



escola  
britânica de  
artes criativas  
& tecnologia

# Profissão: Engenheiro Front-End



# Automação de tarefas com Grunt



# GUIA DA AULA 3



# Use o Grunt para compilar LESS

- Plugins
- Exemplo de configuração de plugin



Acompanhe aqui os temas que serão tratados na videoaula



## Plugins

As automações no Grunt funcionam com o uso de *plugins*. Possuímos *plugins* para compilar SASS e LESS, para pré-processador código de estilos, minificar arquivos, limpar pastas, entre outros.

Os plugins também são tarefas, porém não utilizamos o **grunt.registerTask** para eles, mas sim a função **grunt.loadNpmTasks("nomeDoPacote")**, e dentro da função **initConfig**, após “**pkg**”, configuramos o *plugin*.



A configuração de um *plugin* pode ser dividida entre os ambientes.

Por exemplo, podemos compilar um código LESS para produção aplicando a minificação e compilar o mesmo arquivo, mas não minificando para o ambiente de desenvolvimento.

As configurações de *plugins* serão inseridas logo após a propriedade “**pkg**” dentro da função **grunt.initConfig**.



# Exemplo de configuração de plugin

```
module.exports = function(grunt) {
  grunt.initConfig({
    pkg: grunt.file.readJSON('package.json'),
    less: {
      desenvolvimento: {
        files: {
          'final.css': 'origem.less'
        }
      }
    }
  });

  grunt.loadNpmTasks('grunt-contrib-less');
}
```



Na configuração, onde temos a palavra “**desenvolvimento**” podemos separar as configurações no que o Grunt chama de *targets* (alvos), então podemos ter um alvo para desenvolvimento e outro para produção, enviando o arquivo final para outra pasta e comprimindo ele.

...

```
less: {  
    desenvolvimento: { ... }  
    producao: {  
        options: { compress: true }  
        files: { 'dist/final.css': 'origem.less' }  
    }  
}
```

