

Task Runners

Pré-processando HTML, JavaScript e CSS



@alessandrosales

- Desenvolvedor Backend na **Mobtelecom**
- Formação em **Análise de Sistemas**
- Ambiente: Linux, php, angular, nodejs, ruby on rails, git, mysql, docker e etc...
- Jogador de **League of Legends** nas horas vagas =D

Tópicos

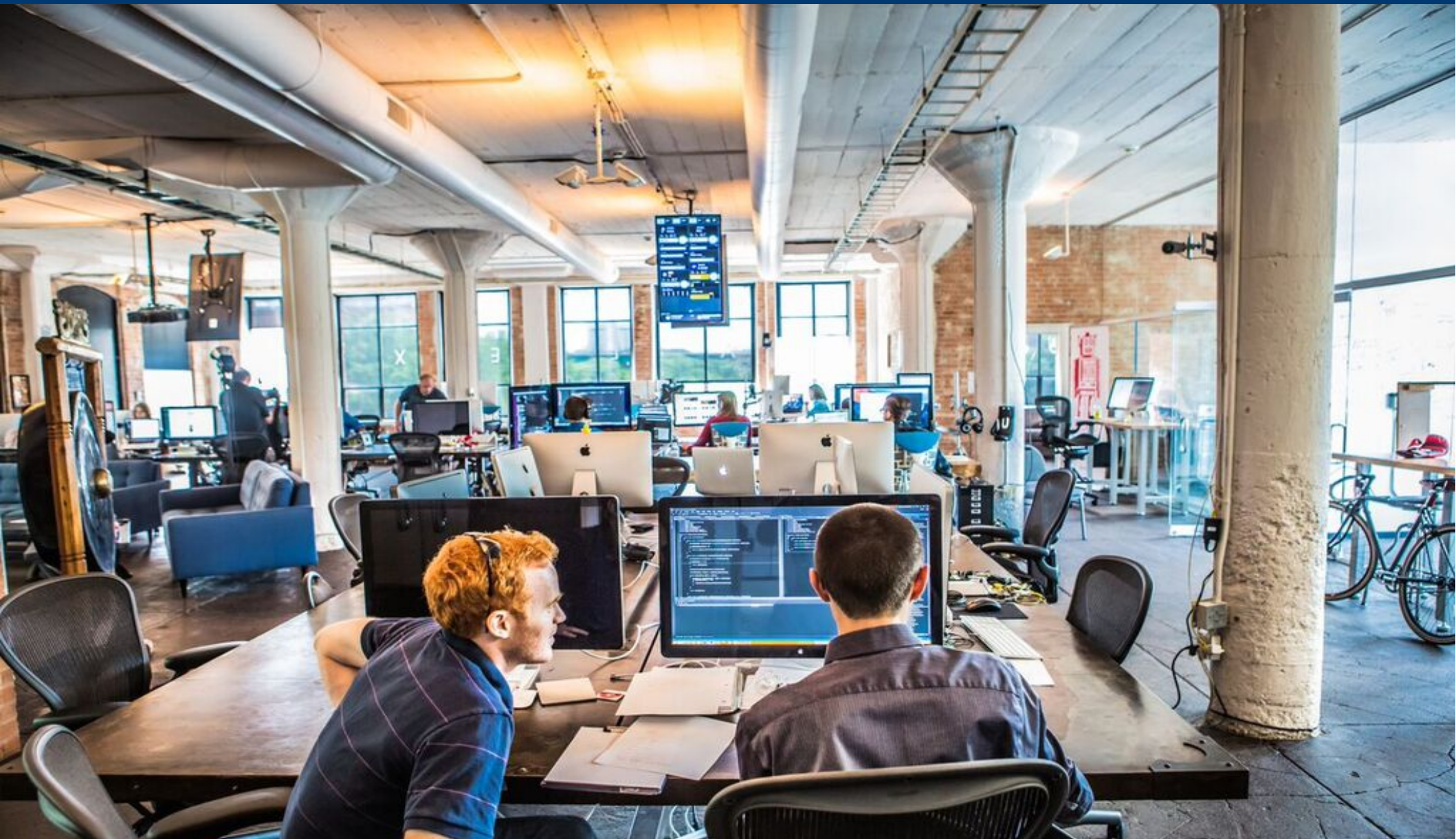
- Por que utilizar Task Runners?
- Conhecendo alguns dos Task Runners mais utilizados e configurando o ambiente
- Testando tasks com Grunt e Gulp

Por que utilizar Task
Runners?

Automatização de atividades repetitivas



Ganho de produtividade



Possibilita a utilização de
pré-processadores para HTML,
JavaScript, CSS e etc...



Otimização do fluxo de trabalho



Deploy automatizado



Conhecendo alguns dos Task
Runners mais utilizados e
configurando o ambiente

Alguns dos Task Runners mais utilizados





É uma plataforma para desenvolvimento de aplicações server-side baseadas em rede utilizando JavaScript e o V8 JavaScript Engine, ou seja, com Node.js podemos criar uma variedade de aplicações Web utilizando apenas código em JavaScript.

Fonte: <https://tableless.com.br/o-que-nodejs-primeiros-passos-com-node-js>

Instalação os módulos globais de gulp e grunt

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


Inicialização do projeto

```
$ npm init -y
```

Instalação do gulp e do grunt local (no projeto)

```
$ npm install gulp grunt --save-dev
```

Instalação de plugins do gulp

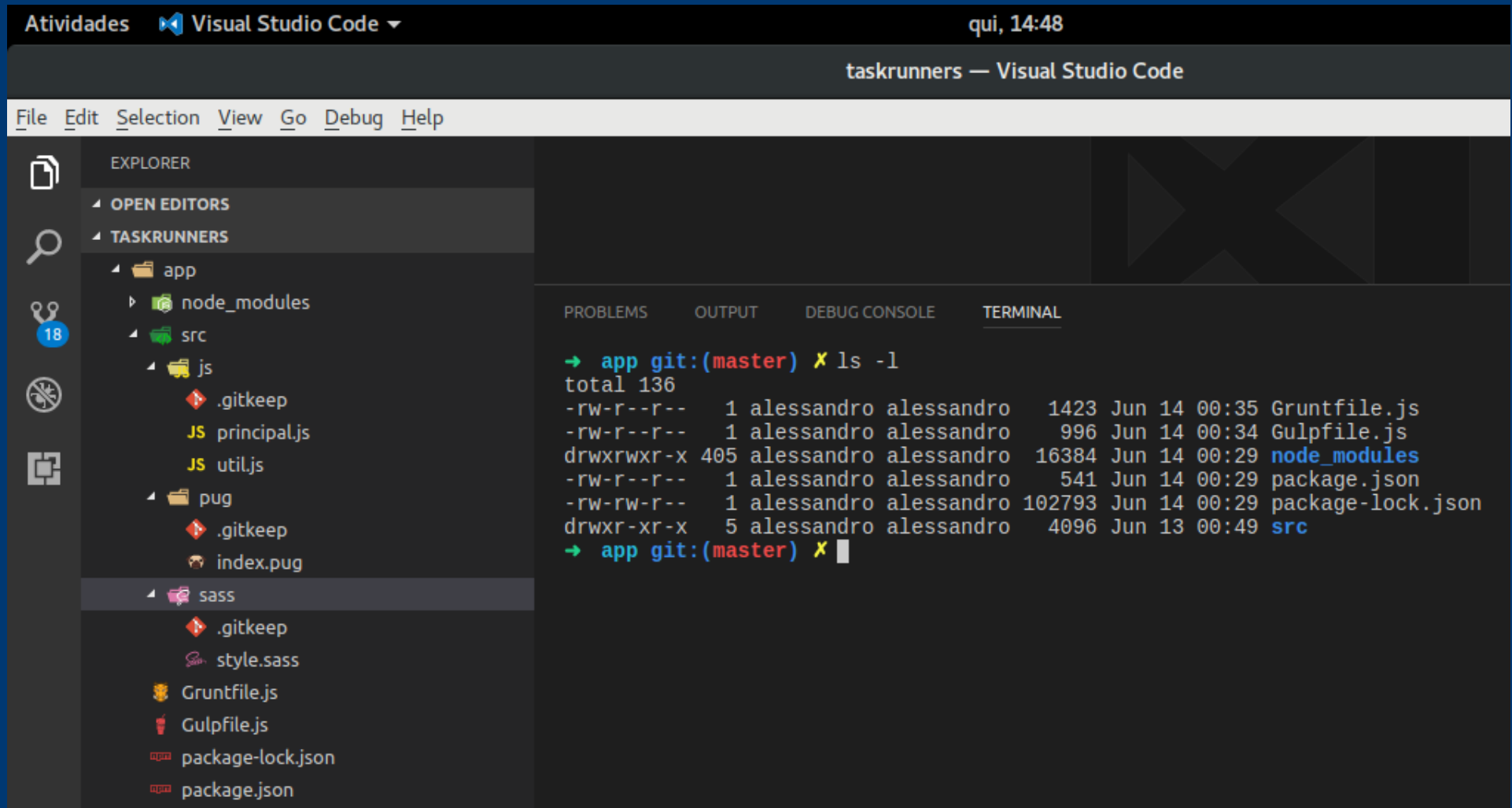
```
$ npm install gulp-pug gulp-sass gulp-uglify  
gulp-concat --save-dev
```

Instalação de plugins do grunt

```
$ npm install grunt-contrib-pug grunt-sass  
grunt-contrib-uglify --save-dev
```

Testando tasks com Gulp e Grunt

Testando tasks com Gulp e Grunt



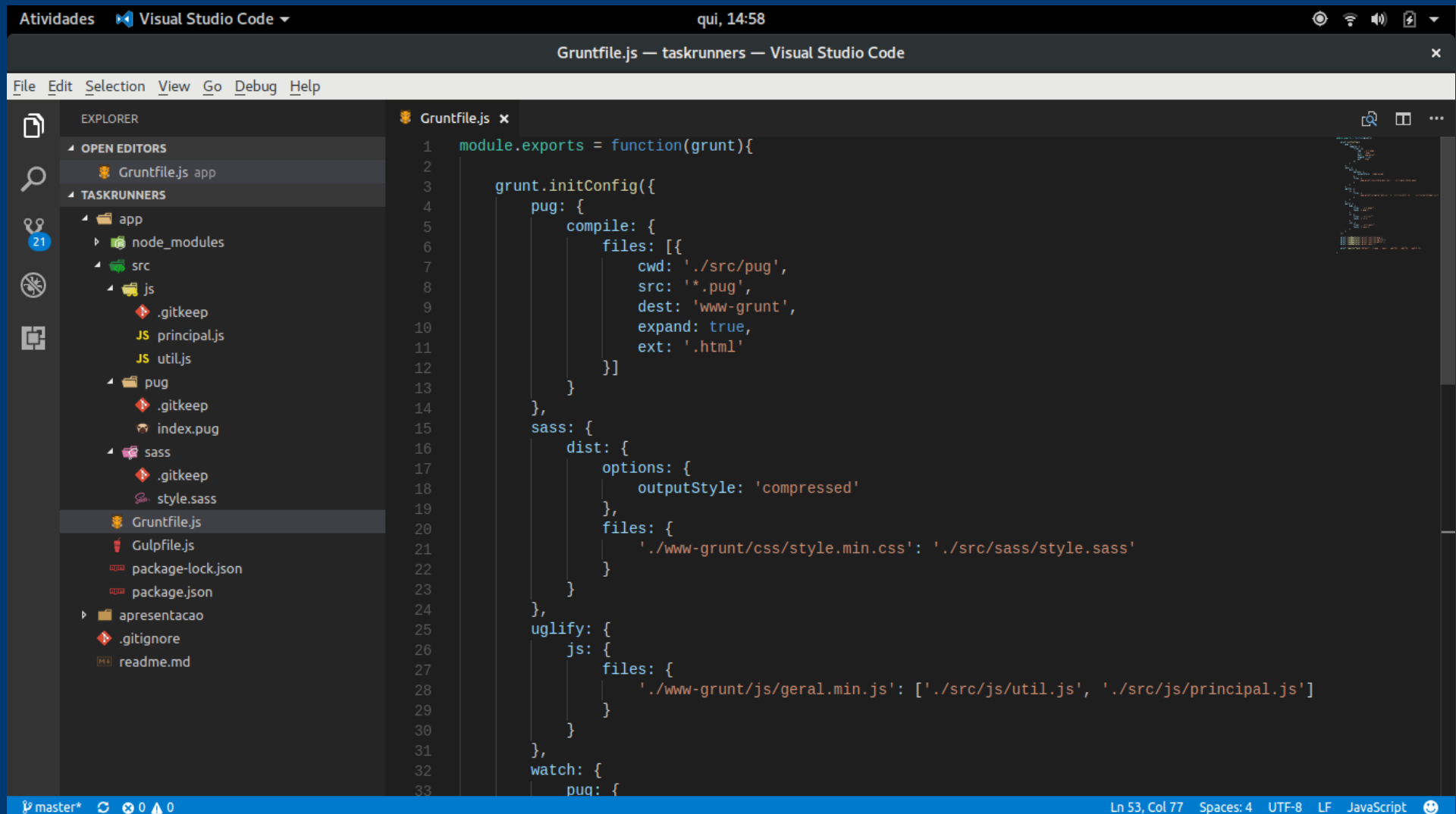
Gulpfile.js

```
1  var gulp = require('gulp');
2  var pug = require('gulp-pug');
3  var sass = require('gulp-sass');
4  var uglify = require('gulp-uglify');
5  var concat = require('gulp-concat');
6
7  gulp.task('html', function(){
8
9      return gulp.src('./src/pug/*.pug')
10         .pipe(pug())
11         .pipe(gulp.dest('./www-gulp/'));
12  });
13
14  gulp.task('css', function(){
15
16      return gulp.src('./src/sass/style.sass')
17         .pipe(sass())
18         .pipe(concat('style.min.css'))
19         .pipe(gulp.dest('./www-gulp/css/'));
20  });
21
22  gulp.task('js', function(){
23
24      return gulp.src(['./src/js/util.js', './src/js/principal.js'])
25         .pipe(uglify({mangle: false}))
26         .pipe(concat('geral.min.js'))
27         .pipe(gulp.dest('./www-gulp/js/'));
28  });
29
30  gulp.task('watch', function(){
```

Gulpfile.js

```
14
15 gulp.task('css', function(){
16
17     return gulp.src('./src/sass/style.sass')
18         .pipe(sass())
19         .pipe(concat('style.min.css'))
20         .pipe(gulp.dest('./www-gulp/css/'));
21 });
22
23
24 gulp.task('js', function(){
25
26     return gulp.src(['./src/js/util.js', './src/js/principal.js'])
27         .pipe(uglify({mangle: false}))
28         .pipe(concat('geral.min.js'))
29         .pipe(gulp.dest('./www-gulp/js/'));
30 });
31
32
33 gulp.task('watch', function(){
34
35     gulp.watch(['./src/pug/*'], ['html']);
36     gulp.watch(['./src/sass/*'], ['css']);
37     gulp.watch(['./src/js/*'], ['js']);
38 });
39
40
41 gulp.task('default', ['html', 'css', 'js', 'watch']);
```

Gruntfile.js

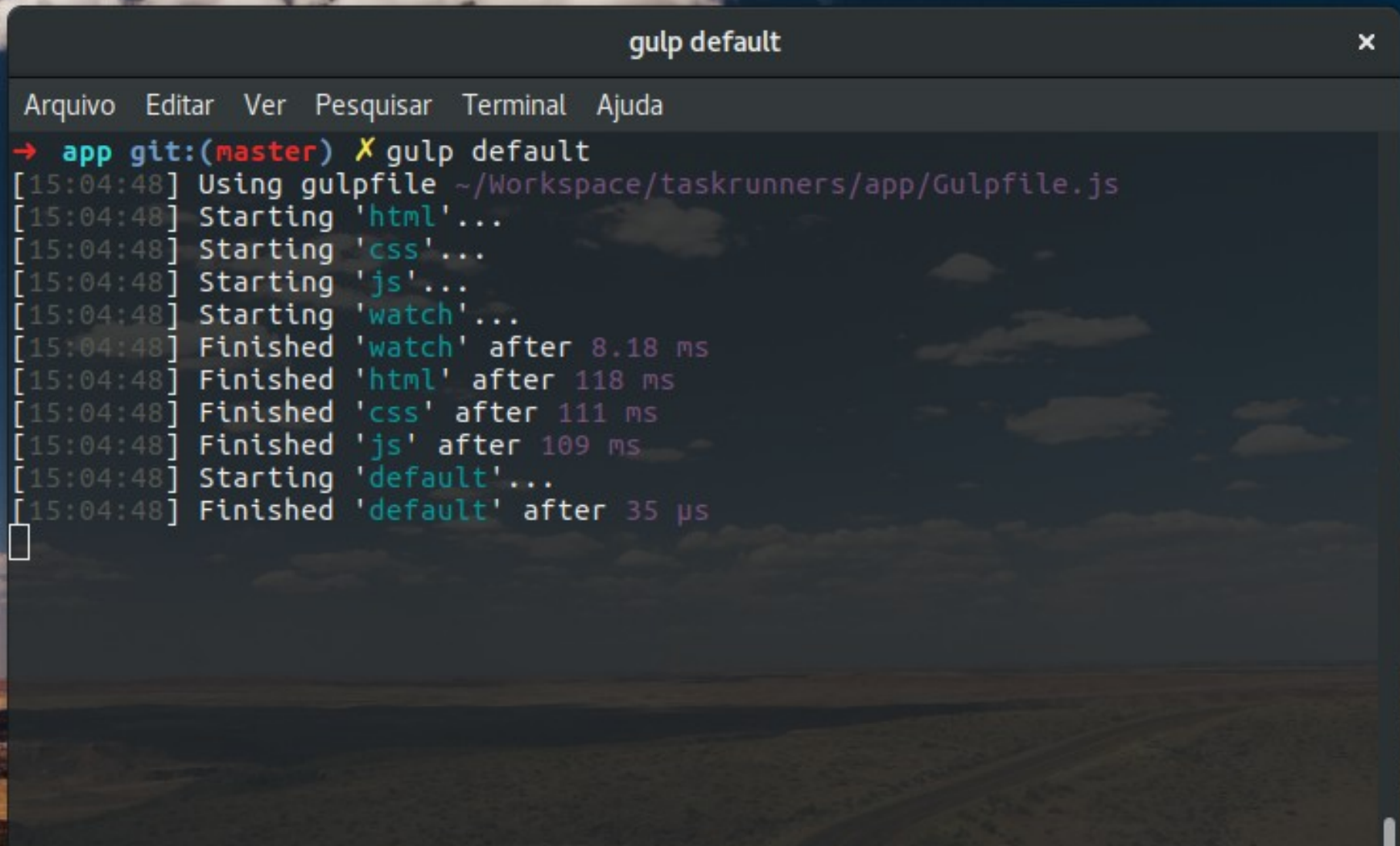


```
1 module.exports = function(grunt){
2
3   grunt.initConfig({
4     pug: {
5       compile: {
6         files: [{
7           cwd: './src/pug',
8           src: '*.pug',
9           dest: 'www-grunt',
10          expand: true,
11          ext: '.html'
12        }]
13      }
14    },
15    sass: {
16      dist: {
17        options: {
18          outputStyle: 'compressed'
19        },
20        files: {
21          './www-grunt/css/style.min.css': './src/sass/style.sass'
22        }
23      }
24    },
25    uglify: {
26      js: {
27        files: {
28          './www-grunt/js/geral.min.js': ['./src/js/util.js', './src/js/principal.js']
29        }
30      }
31    },
32    watch: {
33      pug: {
```

Gruntfile.js

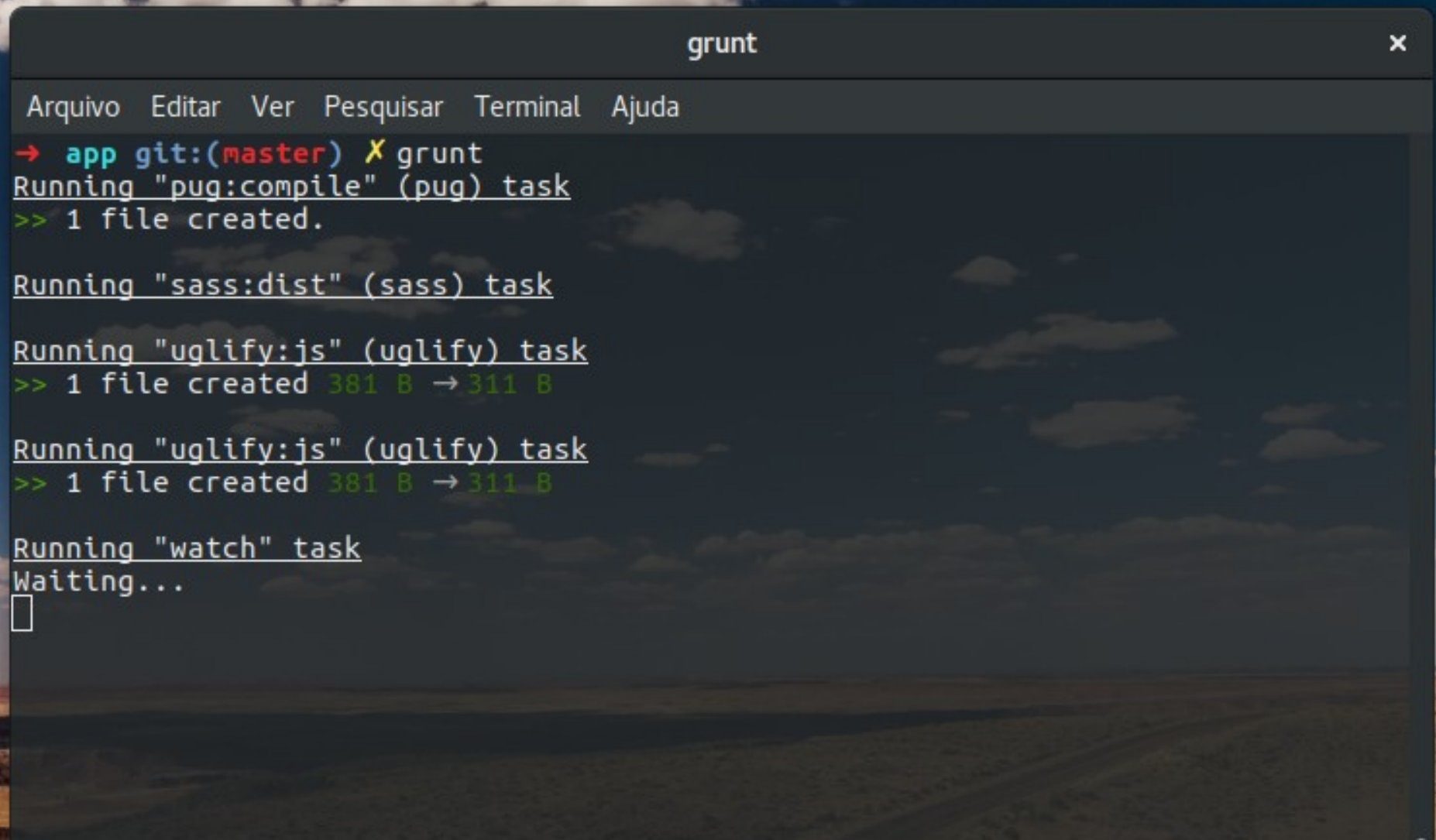
```
25  uglify: {
26    js: {
27      files: {
28        './www-grunt/js/geral.min.js': ['./src/js/util.js', './src/js/principal.js']
29      }
30    },
31  },
32  watch: {
33    pug: {
34      files: './src/pug/*',
35      tasks: ['pug']
36    },
37    js: {
38      files: './src/js/*',
39      tasks: ['js']
40    },
41    css: {
42      files: './src/sass/*',
43      tasks: ['sass']
44    }
45  }
46  });
47
48  grunt.loadNpmTasks('grunt-contrib-pug');
49  grunt.loadNpmTasks('grunt-contrib-uglify');
50  grunt.loadNpmTasks('grunt-contrib-watch');
51  grunt.loadNpmTasks('grunt-sass');
52
53  grunt.registerTask('default', ['pug', 'sass', 'uglify', 'uglify', 'watch']);
54
55 }
```


Task default do Gulp



```
gulp default
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
→ app git:(master) X gulp default
[15:04:48] Using gulpfile ~/Workspace/taskrunners/app/Gulpfile.js
[15:04:48] Starting 'html'...
[15:04:48] Starting 'css'...
[15:04:48] Starting 'js'...
[15:04:48] Starting 'watch'...
[15:04:48] Finished 'watch' after 8.18 ms
[15:04:48] Finished 'html' after 118 ms
[15:04:48] Finished 'css' after 111 ms
[15:04:48] Finished 'js' after 109 ms
[15:04:48] Starting 'default'...
[15:04:48] Finished 'default' after 35 μs
```

Task default do Grunt



```
grunt
Arquivo  Editar  Ver  Pesquisar  Terminal  Ajuda
→ app git:(master) X grunt
Running "pug:compile" (pug) task
>> 1 file created.

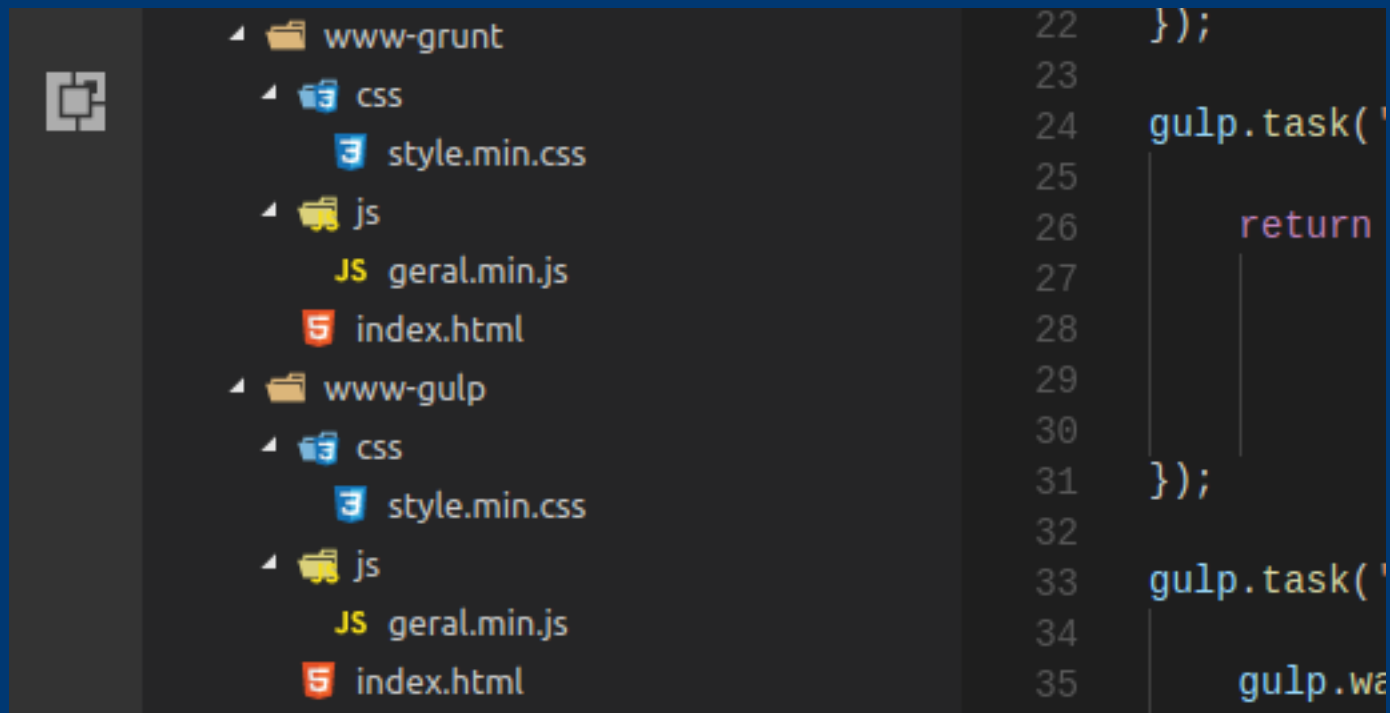
Running "sass:dist" (sass) task

Running "uglify:js" (uglify) task
>> 1 file created 381 B → 311 B

Running "uglify:js" (uglify) task
>> 1 file created 381 B → 311 B

Running "watch" task
Waiting...
█
```

Resultado



The image shows a screenshot of a code editor with a dark theme. On the left, a file explorer displays a project structure with two main folders: 'www-grunt' and 'www-gulp'. Each folder contains subfolders for 'css' and 'js', and a file named 'index.html'. The 'css' folders contain 'style.min.css', and the 'js' folders contain 'geral.min.js'. On the right, a code editor shows a JavaScript snippet with line numbers 22 through 35. The code defines a Gulp task that returns a series of tasks.

```
22  });  
23  
24  gulp.task('default', function() {  
25    return gulp.series(  
26      'build',  
27    );  
28  });  
29  
30  
31  });  
32  
33  gulp.task('build', function() {  
34    return gulp.series(  
35      'clean',  
      'copy',  
      'css',  
      'js',  
      'html',  
    );  
  });  
});
```

Repositório

```
git clone git@github.com:alessandrosales/taskrunners.git
```

Obrigado! =D

Onde me encontrar

 alessandroaug

 alessandrosales

 contato@alessandrosales.com.br