

**INSTITUTO FEDERAL**

Mato Grosso do Sul

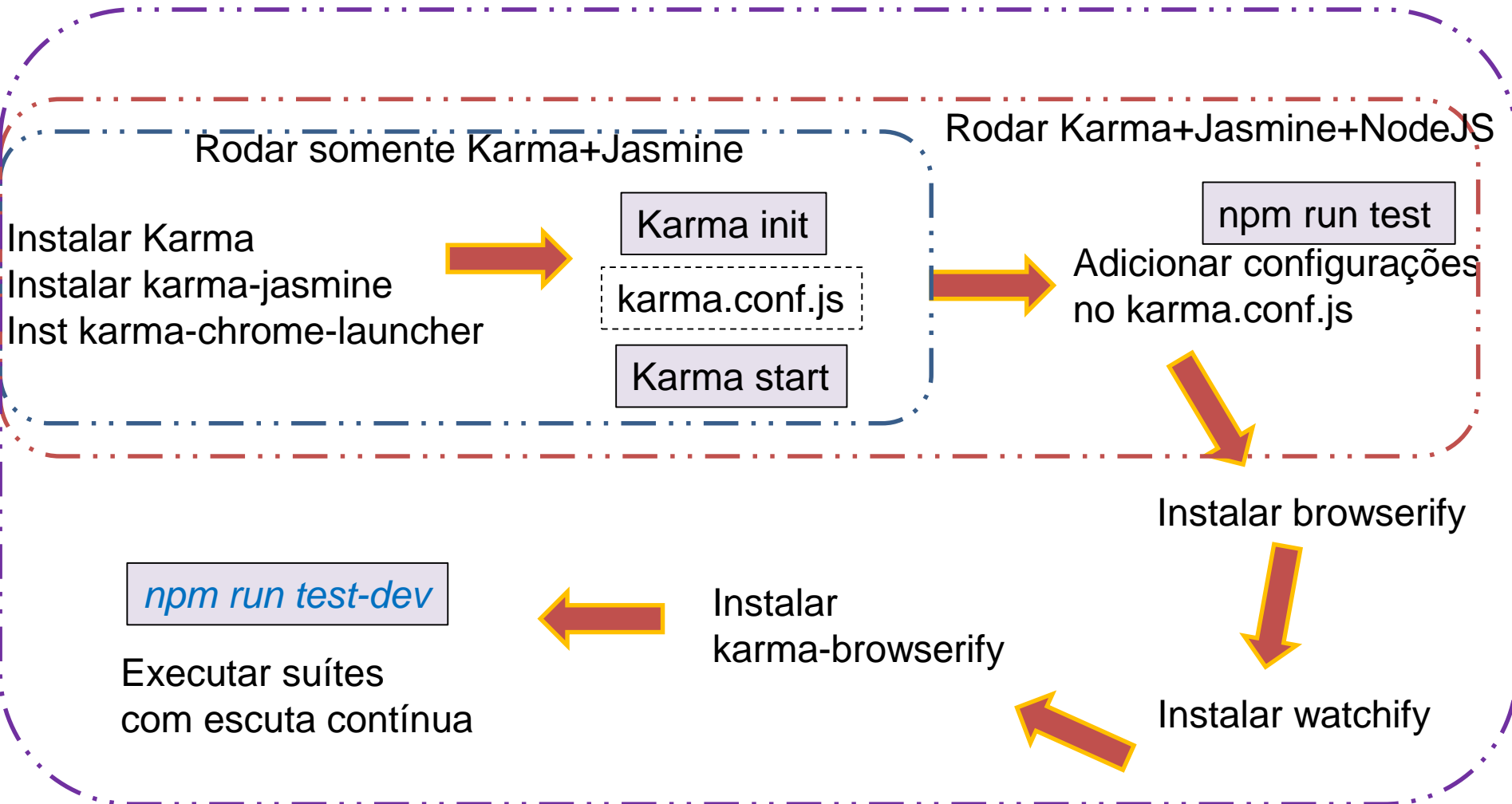
## Karma – Automação de Testes

Engenharia de Software II

# Conteúdo

- O que é o Karma?
- Integrar Karma com o Jasmine para automatizar os testes
- Executar Karma através do NodeJS
- O que é o Browserify?

# Diagrama Geral das Configurações




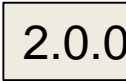
## O que é o Karma?

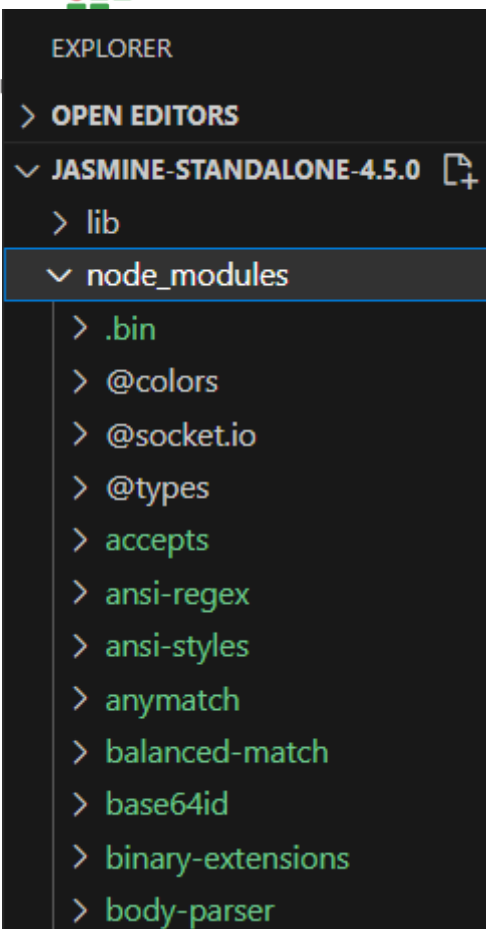
- É uma ferramenta que permite executar os testes diretamente em um ou mais navegadores
- Ele fica monitorando os arquivos de testes, e sempre que uma alteração é detectada ele executa todos os testes
- Ele consegue capturar informações dos navegadores, assim como consegue acessar objetos da página como o objeto 'window'
- Muito útil na automatização do processo de testes, assim como permite fácil integração com servidores de integração contínua

## Instalação

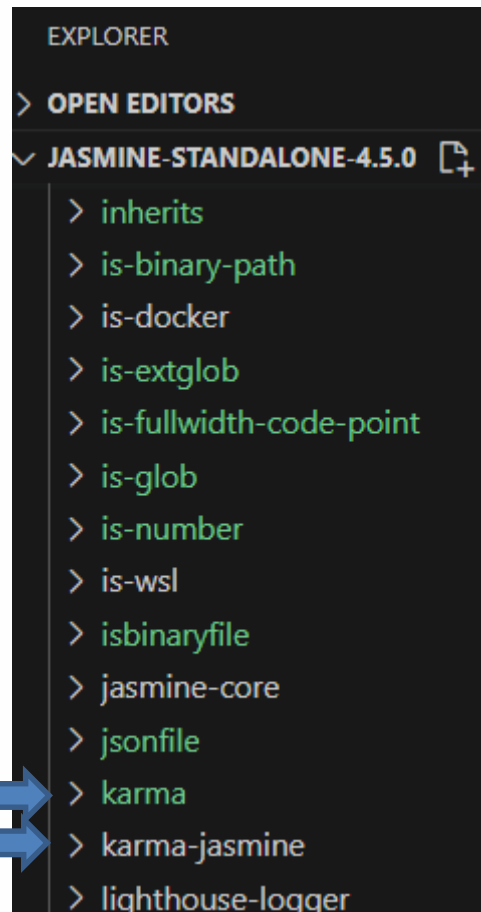
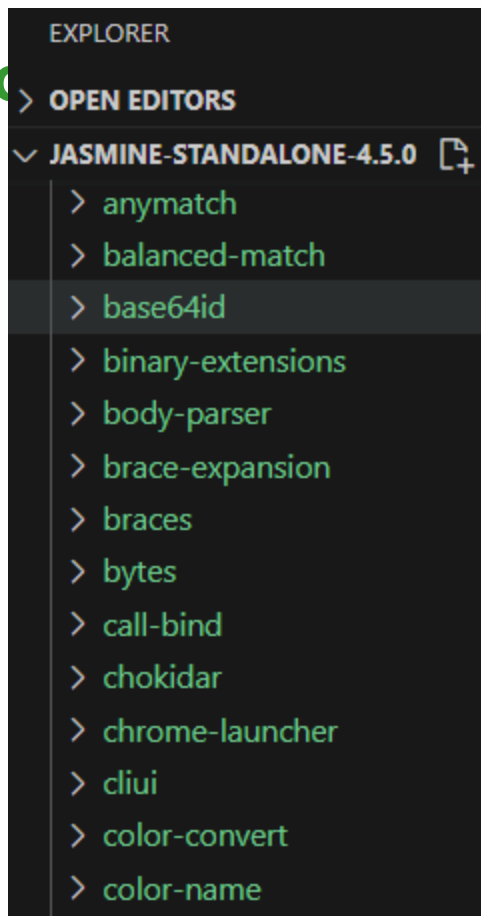
- NPM 'karma-jasmine'
- para a integração, instalar pacote NPM 'karma-jasmine'
- instalar também o pacote: 'karma-chrome-launcher'
- instalar globalmente o pacote NPM 'karma-cli'

## Instalação no terminal

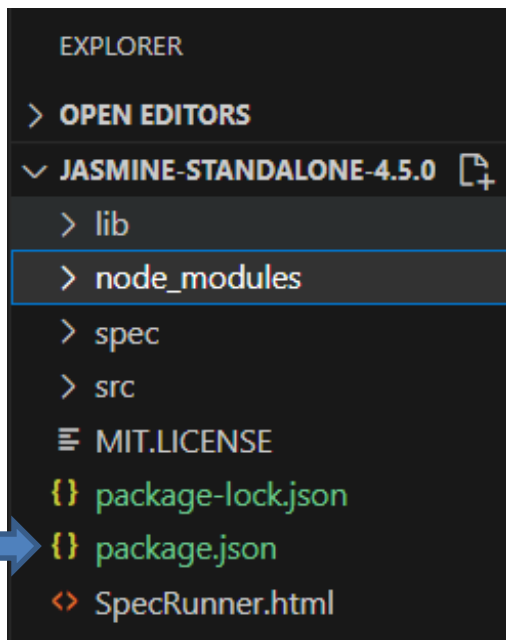
- `npm install --save-dev karma`
- `npm install --save-dev karma-jasmine`
- `npm install --save-dev karma-chrome-launcher`
- `npm install --save-dev -g karma-cli`
- `karma --version` //para testar  
- abrir e visualizar o package.json



ação no



## Configurando o Karma



```
{ } package.json > ...  
1  {  
2    "name": "jasmine-node",  
3    "version": "1.0.0",  
4    "description": "",  
5    "main": "index.js",  
6    "scripts": {  
7      "test": "jasmine"  
8    },  
9    "keywords": [],  
10   "author": "",  
11   "license": "ISC",  
12   "devDependencies": {  
13     "jasmine": "^4.6.0",  
14     "karma": "^6.4.2",  
15     "karma-chrome-launcher": "^3.2.0",  
16     "karma-jasmine": "^5.1.0"  
17   }  
18 }
```



## Configurando o Karma no projeto

- é necessário um arquivo de configuração com os detalhes da aplicação sobre a execução dos testes
- o assistente 'karma init' serve para criar o arquivo de inicialização e recebe como parâmetro o nome do arquivo 'karma.conf.js'
- ficará localizado na raiz do projeto

## Configurando o Karma no projeto

- *karma init karma.conf.js*
  - **responda as questões:**
  - jasmine
  - no
  - Chrome
  - spec/\*\*/\*Spec.js
  - Digite Enter
  - yes
- visualize o arquivo de configuração criado

## Configurando o Karma no projeto

```
PS C:\Users\marci\Desktop\jasmine\jasmine-standalone-4.5.0> karma init karma.conf.js
```

**Which testing framework do you want to use ?**

Press tab to list possible options. Enter to move to the next question.

```
> jasmine
```

**Do you want to use Require.js ?**

This will add Require.js plugin.

Press tab to list possible options. Enter to move to the next question.

```
> no
```

**Do you want to capture any browsers automatically ?**

Press tab to list possible options. Enter empty string to move to the next question.

```
> Chrome
```

# Configurando o Karma no projeto

**What is the location of your source and test files ?**

You can use glob patterns, eg. "js/\*.js" or "test/\*\*/\*.Spec.js".

Enter empty string to move to the next question.

> spec/\*\*/\*.Spec.js

>

**Should any of the files included by the previous patterns be excluded ?**

You can use glob patterns, eg. "\*\*/\*.swp".

Enter empty string to move to the next question.

>

**Do you want Karma to watch all the files and run the tests on change ?**

Press tab to list possible options.

> yes

▼ JASMINE-STANDALONE-4.5.0

> unpipe  
> utils-merge  
> vary  
> void-elements

> which  
> wrap-ansi  
> wrappy  
> ws  
> y18n

> yargs  
> yargs-parser

{ } .package-lock.json

> spec  
> src

🔖 karma.conf.js

☰ MIT.LICENSE

{ } package-lock.json

{ } package.json

🔗 SpecRunner.html

🔖 karma.conf.js U X

🔖 karma.conf.js > ...

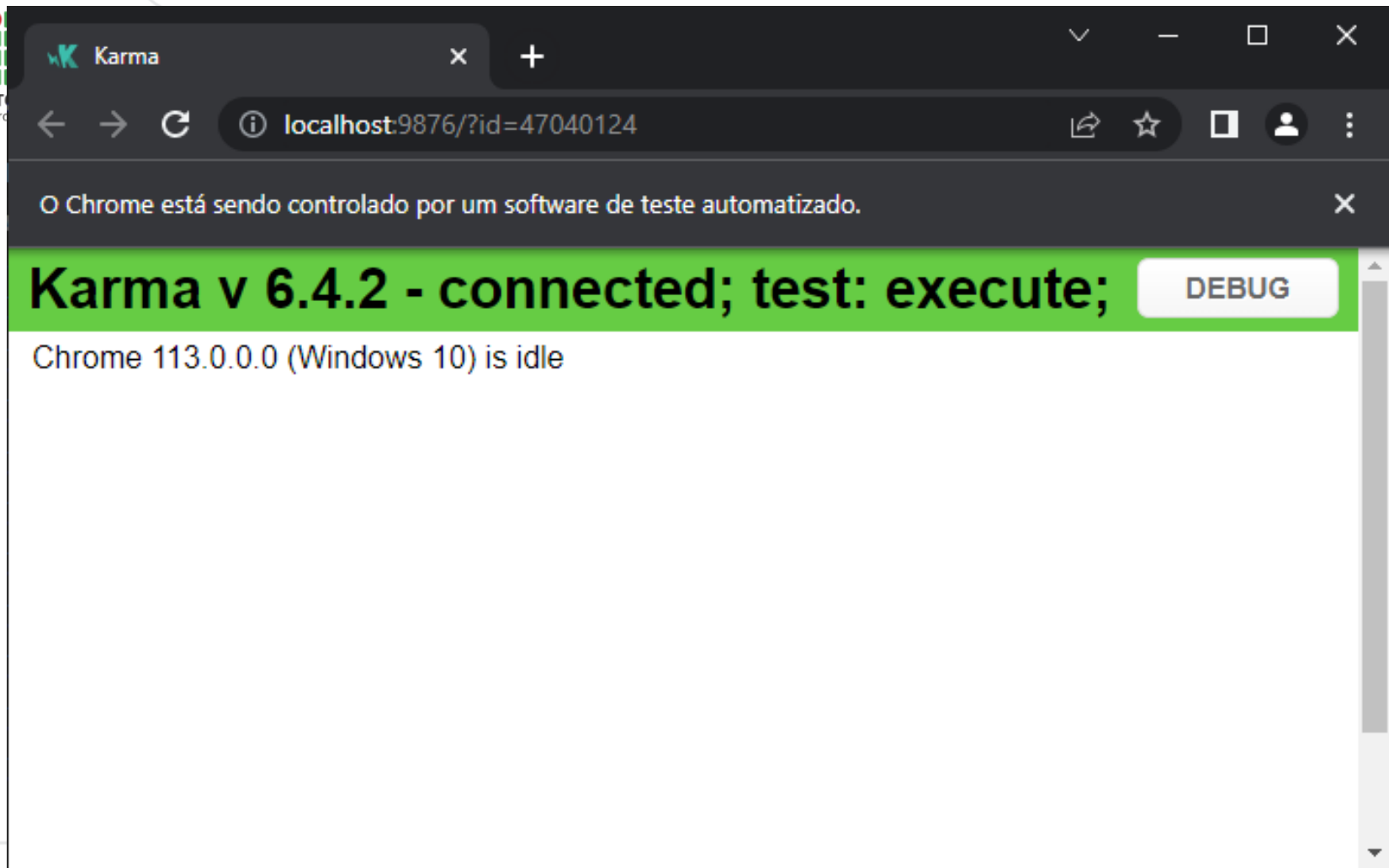
```
1  // Karma configuration
2  // Generated on Thu May 11 2023 22:12:30 GMT-0400 (Horário Padrão do Am
3
4  module.exports = function(config) {
5    config.set({
6
7      // base path that will be used to resolve all patterns (eg. files,
8      basePath: '',
9
10
11     // frameworks to use
12     // available frameworks: https://www.npmjs.com/search?q=keywords:ka
13     frameworks: ['jasmine'],
14
15
16     // list of files / patterns to load in the browser
17     files: [
18       'spec/**/*.Spec.js'
19     ],
20
```

## Iniciar o Karma

- *karma start*

Ou *karma start nome\_do\_arquivo.js*

- Será executado em modo de escuta (watch)
- Para sair do modo de escuta CTRL + C



ReferenceError: Player is not defined

at UserContext.<anonymous> (spec/PlayerSpec.js:6:5)

at <Jasmine>

Chrome 113.0.0.0 (Windows 10) Operação de Adição deve garantir que 1+1=2 FAILED

ReferenceError: somar is not defined

at UserContext.<anonymous> (spec/somarSpec.js:4:9)

at <Jasmine>

Chrome 113.0.0.0 (Windows 10) Suíte de testes do beforeEach deve incrementar o contador para 2 FAILED

Expected 1 to equal 2.

at <Jasmine>

at UserContext.<anonymous> (spec/beforeEachSpec.js:10:26)

at <Jasmine>

Chrome 113.0.0.0 (Windows 10) Suíte de testes do beforeEach deve incrementar o contador para 1 FAILED

Expected 2 to equal 1.

at <Jasmine>

at UserContext.<anonymous> (spec/beforeEachSpec.js:7:26)

at <Jasmine>

Chrome 113.0.0.0 (Windows 10): Executed 36 of 36 (11 FAILED) (0.041 secs / 0.007 secs)

TOTAL: 11 FAILED, 25 SUCCESS



## Iniciar o Karma

- Embora o Karma foi executado em modo de escuta, os erros aconteceram por que tem uma configuração que o browser não entende, que é o uso do pacote **require** pelo código de teste do NodeJS, utilizado no lado servidor para importar arquivos.
- Daí produz o erro de referência: **“Require is not defined!”**
- Para corrigir vamos usar o Browserify que fará a conversão do **require** para navegadores
- CTRL + C para encerrar o modo de escuta do Karma

# Executando o Karma com NodeJS

- Acessar arquivo de configuração do projeto package.json
- renomear o script do Jasmine de 'test' para 'test-jasmine'
- adicionar os novos scripts  
*"test-dev": "karma start",*  
*"test": "karma start --single-run"*

```
{ } package.json > ...
1  {
2    "name": "jasmine-node",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test-jasmine": "jasmine",
8      "test-dev": "karma start",
9      "test": "karma start --single-run"
10   },
11   "keywords": [],
12   "author": "",
13   "license": "ISC",
14   "devDependencies": {
15     "jasmine": "^4.6.0",
16     "karma": "^6.4.2",
17     "karma-chrome-launcher": "^3.2.0",
18     "karma-jasmine": "^5.1.0"
19   }
20 }
```

# Executando NodeJS

- Para testar
  - *npm run test*
- O erro continuará, porém agora foi executado com o NodeJS
- Como o *test* contém o *single-run*, executará um só vez

```
PS C:\Users\marci\Desktop\jasmine\jasmine-node-karma> npm run test
⊗ npm WARN config global `--global`, `--local` are deprecated. Use `--location` instead.
Debugger attached.

> jasmine-node@1.0.0 test
> karma start --single-run

Debugger attached.
12 05 2023 22:04:37.232:INFO [karma-server]: Karma v6.4.2 server started at host:9876/
12 05 2023 22:04:37.235:INFO [launcher]: Launching browsers Chrome with no capabilities, unlimited
12 05 2023 22:04:37.243:INFO [launcher]: Starting browser Chrome
12 05 2023 22:04:37.857:INFO [Chrome 113.0.0.0 (Windows 10)]: Connected to browser at a-PN_tjzFKLPAAAB with id 80503009
Chrome 113.0.0.0 (Windows 10) ERROR
  An error was thrown in afterAll
  Uncaught ReferenceError: require is not defined
  ReferenceError: require is not defined
    at spec/jasmine_examples/PlayerSpec.js:1:16
Chrome 113.0.0.0 (Windows 10): Executed 1 of 1 ERROR (0.012 secs / 0.012 secs)
Waiting for the debugger to disconnect...
Waiting for the debugger to disconnect...
⊗ PS C:\Users\marci\Desktop\jasmine\jasmine-node-karma>
```

# Executando o Karma com NodeJS

- Testar com a escuta contínua
  - *npm run test-dev*
- É equivalente ao karma start

Como resolver o erro do *require*?

## Browserify

- é um framework que converte as chamadas do require utilizadas no NodeJS para um formato compreensível pelo navegador
- ele analisa o código NodeJS e converte todas as chamadas 'require' para a tag HTML '<script>' de modo que o navegador consiga importar todas as dependências do projeto

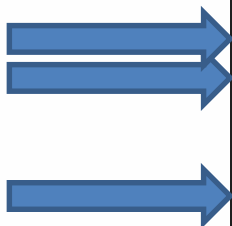
## Como ele soluciona o problema com o Karma?

- O código de exemplo do Jasmine faz uso da função require para importar arquivos, recursos que não é suportado pelos navegadores
- com o Browserify configurado como pré-processador do Karma, antes do código Javascript ser enviado para o navegador, ele será convertido para um formato compreensível e executará sem erros

## Configurando o Karma com o Browserify

- Instale o pacote NPM browserify
- *npm install --save-dev karma*
- Instale o pacote NPM watchify, que é uma dependência para o pacote de integração do Karma com o Browserify
- *npm install --save-dev watchify*
- Instale o pacote de integração do Karma com o Browserify
- *npm install --save-dev karma-browserify*

# Configurando o Karma



```
{} package.json X  {} karma.conf.js

{} package.json > ...
1  {
2    "name": "jasmine-node",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test-jasmine": "jasmine",
8      "test-dev": "karma start",
9      "test": "karma start --single-run"
10   },
11   "keywords": [],
12   "author": "",
13   "license": "ISC",
14   "devDependencies": {
15     "jasmine": "^4.6.0",
16     "karma": "^6.4.2",
17     "karma-browserify": "^8.1.0",
18     "karma-chrome-launcher": "^3.2.0",
19     "karma-jasmine": "^5.1.0",
20     "watchify": "^4.0.0"
21   }
22 }
```



## Edite karma.conf.js

- Em frameworks adicione *'browserify'*
- adicione o diretório helpers em files: “
- adicione em 'preprocessors' *'spec/\*\*/\*.Spec.js':['browserify']*

karma.conf.js X

```
karma.conf.js > <unknown> > exports > preprocess
11 // frameworks to use
12 // available frameworks: https://www.npmjs.com/search?q=frameworks
13 frameworks: ['jasmine', 'browserify'],
14
15
16 // list of files / patterns to load in
17 files: [
18   'spec/**/*.Spec.js',
19   'spec/helpers/**/*.SpecHelper.js'
20 ],
21
22
23 // list of files / patterns to exclude
24 exclude: [
25 ],
26
27
28 // preprocess matching files before serving
29 // available preprocessors: https://www.npmjs.com/search?q=preprocessors
30 preprocessors: {
31   'spec/**/*.Spec.js': ['browserify']
32 },
```

## Finalmente

- execute os testes
  - *npm run test-dev*
- edite o arquivo de teste:
  - 'spec/exemplo/exemploSpec.js' e salve-o para que os testes sejam executados novamente de modo automático
- Termine os testes: CTRL + C

PS C:\Users\marci\Desktop\jasmine\jasmine-node-karma> npm run test-dev

npm WARN config global `--global`, `--local` are deprecated. Use `--location=global` instead.

> jasmine-node@1.0.0 test-dev

> karma start

12 05 2023 22:45:45.247:INFO [framework.browserify]: registering rebuild (autoWatch=true)

12 05 2023 22:45:46.034:INFO [framework.browserify]: 3249 bytes written (0.03 seconds)

12 05 2023 22:45:46.035:INFO [framework.browserify]: bundle built

12 05 2023 22:45:46.037:WARN [karma]: No captured browser, open http://localhost:9876/

12 05 2023 22:45:46.040:INFO [karma-server]: Karma v6.4.2 server started at http://localhost:9876/

12 05 2023 22:45:46.040:INFO [launcher]: Launching browsers Chrome with concurrency unlimited

12 05 2023 22:45:46.045:INFO [launcher]: Starting browser Chrome

12 05 2023 22:45:46.671:INFO [Chrome 113.0.0.0 (Windows 10)]: Connected on socket lt5VZ5a0VMw4s0xiAAAB with id 70149910

Chrome 113.0.0.0 (Windows 10): Executed 6 of 6 SUCCESS (0.017 secs / 0.004 secs)

TOTAL: 6 SUCCESS