



# Desafio Vip Commerce

---



# Organização de apresentação



01

Ferramenta utilizada .  
linguagens adjacentes e  
ferramentas de apoio.

02

Estrutura dos projetos.  
Análise dos testes solicitados.

03

Arquitetura de projetos.  
Decisões de testes.

04

Cenário utilizados.  
Steps e ações de comando.



01

Ferramenta escolhida



# Linguagem escolhida

## Cucumber



Por conta de o Cucumber ser uma ferramenta já muito conhecida pela área de teste e também muito intuitiva para se trabalhar. A escolha do cucumber vem de uma preferência particular de uso a priori do desafio.

Além disso o cucumber é uma ferramenta que possibilita com muita facilidade o uso adequado de se usá-lo com metodologias ágeis. E em conjunto com o BDD( desenvolvimento orientado a comportamento) e o Gherkin como sintaxe para apoiar a linguagem para apoiar a construção e lógica dos cenários.

# Linguagens adjacentes e ferramentas utilizadas



Selenium

Capybara





# 02

## Estrutura do projetos

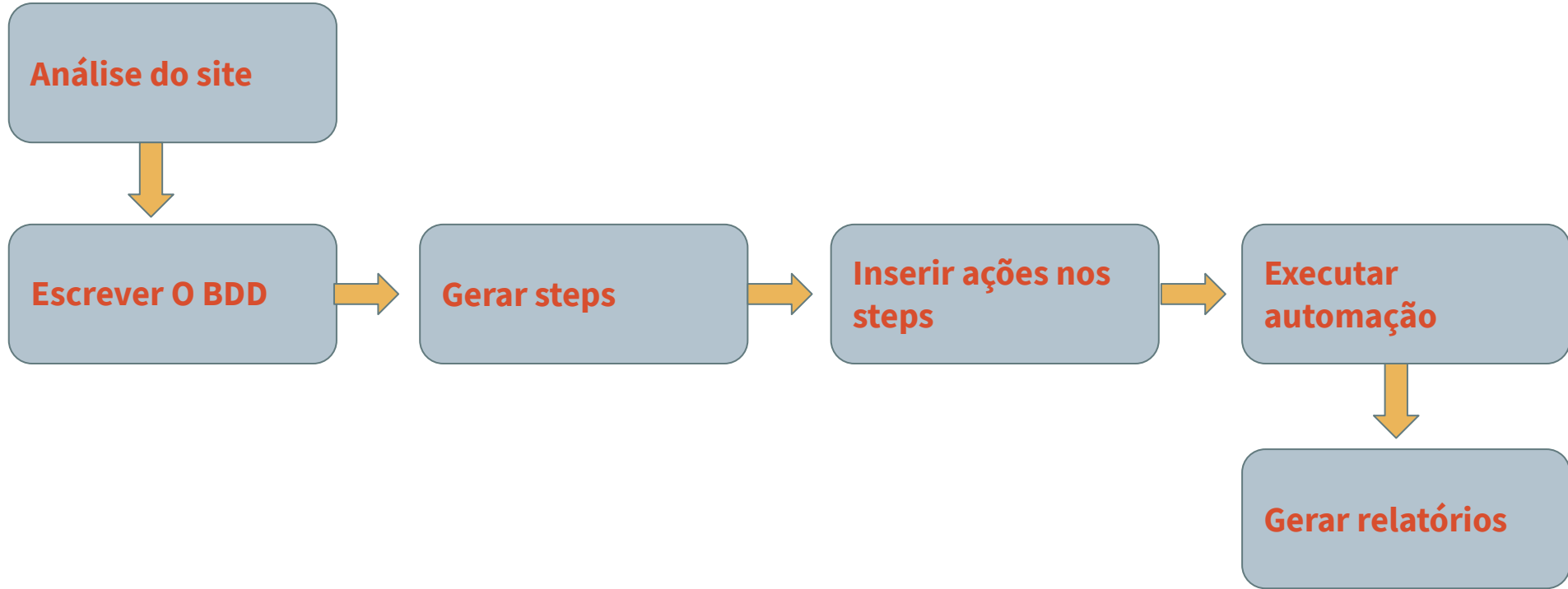




## Análise dos testes propostos.

- 1- Se a página está carregando corretamente a home page;
- 2- Navegar pela página e ir ao sistema de vagas no menú superior;
- 3- Um vez na tela de vagas, nosso sistema de teste automatizado terá que buscar pela empresa de VipCommerce onde terá um ou mais resultados;
- 4- Depois obter o resultado, deveremos abrir a vaga e revisar se o carregamento é realizado com sucesso ou não;
- 5- Por último, precisamos saber se o sistema de canditar está funcionando e para isso é necessário clicar no botão Candidatar-se e revisar se abriu o modal corretamente.

# Estrutura do projeto







03

Arquitetura do projeto



# Arquitetura de projetos



**Tests** : Pasta raiz de cada projeto.

**Specs** : Onde estão as features de projeto.


**Steps** : Onde estão os testes e comandos de automação.

**Support** : Onde está toda a configuração das gemas necessárias.  
para automação.

**Results** : Onde estão os relatórios gerados a partir da automação.

# Decisões de testes

Com base na análise do site e dos testes propostos os seguintes passos foram seguidos:

- Rodar cada desafio sem estar automatizado no site beta da coodesh.
  - Configurar máquina para a automação
  - Escrever os cenários.
  - Gerar os steps necessários.
  - Inspeção de páginas e elementos necessário para o comando.
  - Implementar ações conforme o solicitado
  - Verificar se há erro com o resultado perante a descrição do desafio.
  - Gerar relatórios.
- 



# 04

## Cenários utilizados



# Cenários utilizados

## Teste 1: Carregar página principal

```
# language: pt
```

```
Funcionalidade: Carregar pagina home
```

```
-Como usuário.
```

```
-Gostaria de entrar na home page do site coodesh.
```

```
Cenário: Carregamento da pagina home com sucesso
```

```
    Dado que o usuário tenha acesso a pagina principal
```

```
    Quando o usuario seleciona a URL da pagina
```

```
    Entao a pagina e carregada com sucesso
```

## Teste 2: Carregar página de vagas

```
# language: pt
```

```
Funcionalidade: Carregar pagina de vagas
```

```
-Como usuário.
```

```
-Gostaria de entrar na pagina de vagas do site coodesh.
```

```
Cenário: Carregamento da pagina de vagas com sucesso
```

```
    Dado que o usuário tenha acesso a pagina principal
```

```
    Quando o usuário requisitar o acesso clicando na pagina de vagas
```

```
    Então o sistema exhibe a página de vagas
```

# Cenários utilizados

## Teste 3: Carregar página vagas e pesquisar Vip commerce

# language: pt

Funcionalidade: Pesquisar sobre Vipcommerce

- Como usuário.
- Gostaria pesquisar sobre a empresa Vipcommerce
- Para melhores informações

Cenário: Listagem de informações sobre empresa com sucesso

Dado que o usuário tenha acesso a pagina de vagas  
Quando o usuário inserir o nome da empresa "Vipcommerce"  
Entao o sistema deve exibir as informações sobre a empresa.

## Teste 4: informações sobre uma vaga

# language: pt

Funcionalidade: Exibir informações de vaga.

- Como usuário.
- Gostaria de selecionar uma vaga na pagina de vagas do site da coodesh.
- Para obter mais informações sobre a vaga.

Cenário: Abrir descrição de vaga selecionada

Dado que o usuário tenha acesso a pagina de vagas  
Quando o usuário seleciona a vaga "Desenvolvedor(a) Fullstack Angular 8 Part-Time"  
Entao a pagina com informações da vaga e carregada com sucesso.

# Cenários utilizados

## Teste 5: Abrir modal de inscrição de vaga

```
# language: pt
```

**Funcionalidade:** Abrir modal de candidatura

- Como usuário.
- Gostaria de acessar o modal de candidatura de uma vaga selecionada.
- Para exibir formulário de inscrição da candidatura

**Cenário:** Exibição do formulário de inscrição

Dado que o usuário tenha acesso ao site de vagas

E tenha selecionado uma vaga

Quando o usuário clicar no botão "Candidatar-se"

Entao o sistema deve exibir um modal de inscrição para a vaga

# Relatórios e conclusões

✓ passado

Resumo de execução

1 cenários

Duração

16,306 s

Implementação

pepino-rubi - 5.2.0

Tempo de execução

rubi - 2.7.2

SO

mingw32 - 10.0.18363

CPU

x64

Algun texto ou @tags

Q

Voê pode usar texto simples para a pesquisa ou [expressões de tag pepino](#) para filtrar a saída.

✓ features / specs / pagina\_home.feature

Funcionalidade : Carregar pagina home

-Como usuário. -Gostaria de entrar na home page do site coodesh.

Cenário : Carregamento da pagina home com sucesso

✓ Dado que o usuário tenha acesso a pagina principal

✓ Quando o usuário seleciona um URL da pagina

✓ Entao a pagina e carregada com sucesso



# Relatórios e conclusões

<div><div>✓ passado</div><div>1 cenários</div></div>	<div><div>Duração</div><div>25.895 s</div></div> <div><div>Implementação</div><div>pepino-rubi - 5.2.0</div></div> <div><div>Tempo de execução</div><div>rubi - 2.7.2</div></div> <div><div>SO</div><div>mingw32 - 10.0.18363</div></div> <div><div>CPU</div><div>x64</div></div>	<div><div>Algun texto ou @tags</div><div>Q</div></div> <div><div>ⓘ</div><div>^Você pode usar texto simples para a pesquisa ou <a href="#">expressões de tag pepino</a> para filtrar a saída.</div></div>
--	---	--

✓ recursos / especificações / vagas.feature

## Funcionalidade : Carregar pagina de vagas

-Como usuário. -Gostaria de entrar na página de vagas do site coodesh.

### Cenário : Carregamento da pagina de vagas com sucesso

- ✓ **Dado** que o usuário tenha acesso a pagina principal
- ✓ **Quando** o usuário requisitar o acesso clicando na página de vagas
- ✓ **Então** o sistema exibe uma página de vagas

# Relatórios e conclusões

passed

Execution summary

1 scenarios

Duration

30.277s

Implementation

cucumber-ruby - 5.2.0

Runtime

ruby - 2.7.2

OS

mingw32 - 10.0.18363

CPU

x64

Some text or @tags

q

You can use either plain text for the search or [cucumber tag expressions](#) to filter the output.

✓ features/specs/pesquisavpc.feature

Funcionalidade: Pesquisar sobre Vipcommerce

-Como usuário. -Gostaria pesquisar sobre a empresa Vipcommerce -Para melhores informações

Cenário: Listagem de informações sobre empresa com sucesso

✓ Dado que o usuário tenha acesso a pagina de vagas

✓ Quando o usuário inserir o nome da empresa "Vipcommerce"

✓ Entao o sistema deve exibir as informações sobre a empresa.

# Relatórios e conclusões

✓ passado

Resumo de execução

1 cenários

Duração

23.542 s

Implementação

pepino-rubi - 5.2.0

Tempo de execução

rubi - 2.7.2

SO

mingw32 - 10.0.18363

CPU

x64

Algun texto ou @tags

Q

?

Você pode usar texto simples para a pesquisa ou [expressões de tag pepino](#) para filtrar a saída.

✓ recursos / especificações / vagas.feature

**Funcionalidade :** Exibir informações de vaga.

-Como usuário. -Gostaria de selecionar uma vaga na pagina de vagas do site da coodesh. -Para obter mais informações sobre a vaga.

**Cenário :** Abrir descrição de vaga restituída

✓ **Dado** que o usuário tenha acesso a página de vagas

✓ **Quando** o usuário seleciona a vaga *"Desenvolvedor (a) Fullstack Angular 8 Part-Time"*

✓ **Entao** a pagina com informações da vaga e carregada com sucesso.



# Relatórios e conclusões

✓ passed

**Execution summary**  
1 scenarios

**Duration** 25.698s  
**Implementation** cucumber-ruby - 5.2.0  
**Runtime** ruby - 2.7.2  
**OS** mingw32 - 10.0.18363  
**CPU** x64

Q

• You can use either plain text for the search or [cucumber tag expressions](#) to filter the output.

✓ features/specs/candidatar.feature

**Funcionalidade:** Abrir modal de candidatura  
-Como usuário, -Gostaria de acessar o modal de candidatura de uma vaga selecionada. -Para exibir formulário de inscrição da candidatura  
**Cenário:** Exibição do formulário de inscrição

- ✓ **Dado** que o usuário tenha acesso ao site de vagas
- ✓ **E** tenha selecionado uma vaga
- ✓ **Quando** o usuário clicar no botão "Candidatar-se"
- ✓ **Entao** o sistema deve exibir um modal de inscrição para a vaga

# Relatórios e conclusões

**Teste 1 (Carregar pagina home) :** O teste rodou sem empecilhos, e o resultado foi o esperado.

**Teste 2( Carregar página de vagas):** O teste rodou sem empecilhos, e o resultado foi o esperado.

**Teste 3 (Pesquisar Vipcommerce):** O teste rodou sem empecilhos, mas na pesquisa não houve resultados na página. O resultado não foi o esperado.

**Teste 4 (Exibir informações de vaga):** O teste rodou sem empecilhos, mas como não houve resultados na pesquisa do teste anterior, foi decidido a escolha de uma outra vaga para teste.E o resultado foi como o esperado

**Teste 5 ( Abrir modal de candidatura):** O teste rodou sem empecilhos, e o resultado foi como o esperado.

# Obrigado

Dúvidas?

joysimoes98@gmail.com

<https://www.linkedin.com/in/joycesim%C3%B5es-41b6a4131/>



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**