

Documentação do Teste_QA_2023 - FORMULÁRIO TRICENTIS

A documentação é composta pelos requisitos solicitados no teste prático, com a descrição dos casos de teste e as evidências dos testes sendo executados no Cypress, ferramenta utilizada para a automação dos mesmos.

Foi criado um projeto no VS Code, contendo toda a documentação, scripts, evidências e descrição dos casos de teste.

Os casos de testes foram criados utilizando a escrita Gherkin.

Given: Dado (condição)

When: Quando (ação)

Then: Então (resultado da ação)

And: e (adição)

2- Acessar o site de seguros:

-Documentar o preenchimento de cada aba do formulário, com capturas de tela.

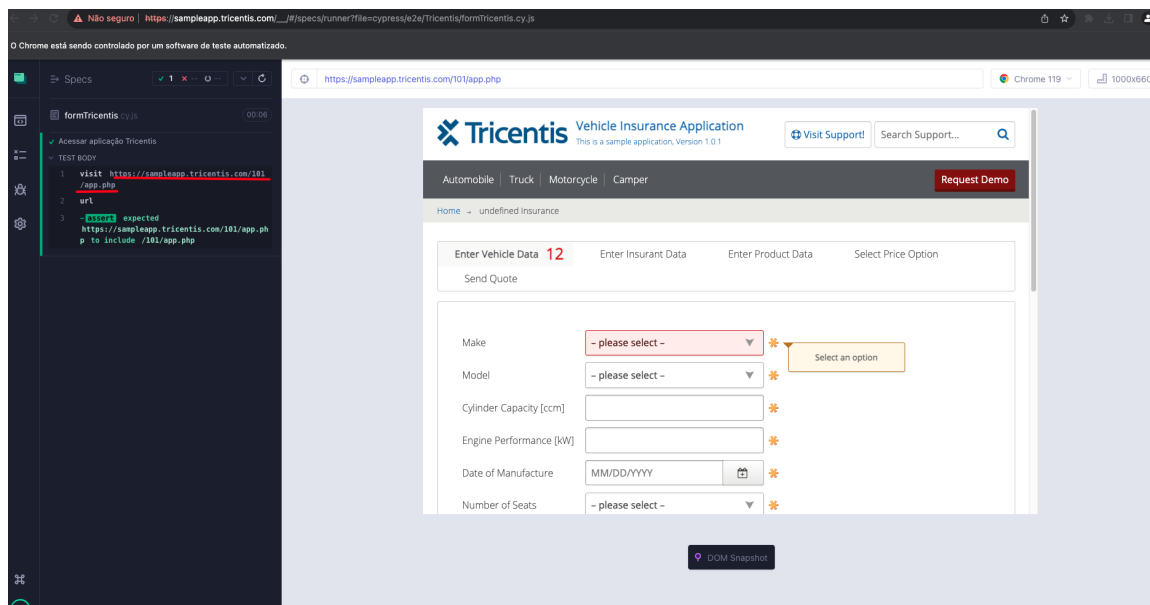
Caso de Teste: Acessar aplicação Tricentis

Dado que possuo o link ('<https://sampleapp.tricentis.com/101/app.php>)

And crio o projeto do teste no VsCode e o mesmo automatizado na ide Cypress

When acesso o navegador com o link

Then posso visualizar a página da Tricentis



Caso de Teste: Preencher todos os campos do formulário e enviar

Given que estou na página da Tricentis

When preencho todos os campos obrigatórios do formulário

Then posso enviar o formulário

Chrome está sendo controlado por um software de teste automatizado.

https://sampleapp.tricentis.com/101/app.php

Tricentis Vehicle Insurance Application
This is a sample application. Version 1.0.1

Automobile | Truck | Motorcycle | Camper Request Demo

Home - undefined Insurance

Enter Vehicle Data 0 Enter Insurant Data Enter Product Data Select Price Option

Send Quote

Make - please select - ✓
Model - please select - ✓
Cylinder Capacity [ccm] 150 ✓
Engine Performance [kW] 100 ✓
Date of Manufacture 10/10/1998 ✓
Number of Seats - please select - ✓

Preenchendo campos na aba EnterVehicleData

9 Pinned before after Highlights X

Chrome está sendo controlado por um software de teste automatizado.

https://sampleapp.tricentis.com/101/app.php

Tricentis Vehicle Insurance Application
This is a sample application. Version 1.0.1

Automobile | Truck | Motorcycle | Camper Request Demo

Home - undefined Insurance

Enter Vehicle Data 0 Enter Insurant Data Enter Product Data Select Price Option

Send Quote

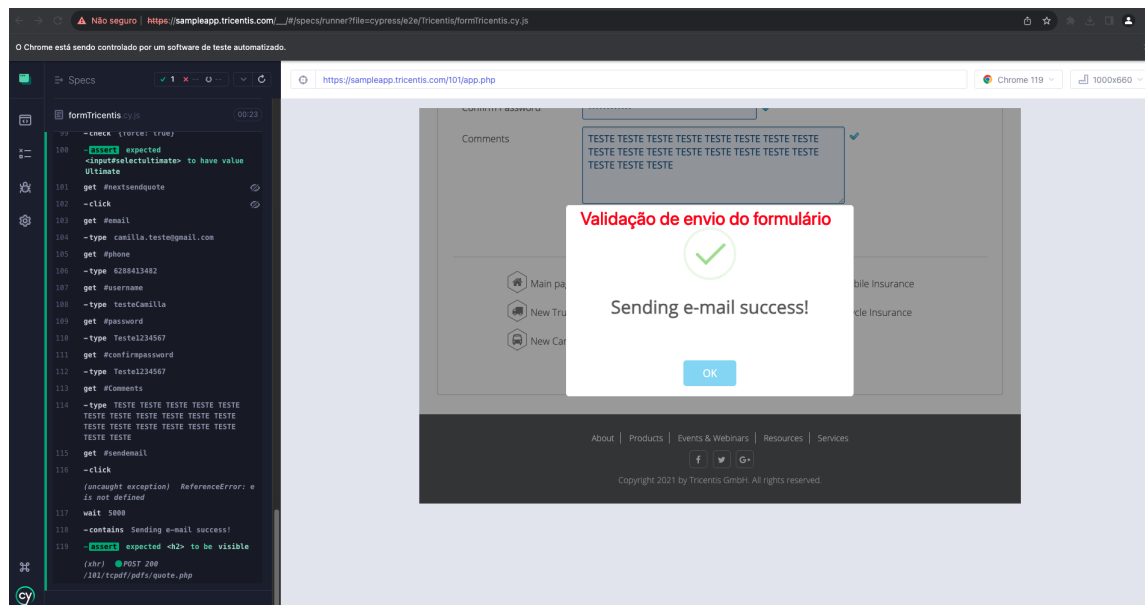
Number of Seats - please select - ✓
Right Hand Drive Yes No ✓
Number of Seats - please select - ✓
Fuel Type - please select - ✓
Payload [kg] 250 ✓
Total Weight [kg] 100 ✓
List Price [\$] 10000 ✓
License Plate Number COO-4017 ✓
Annual Mileage [mi] 30000 ✓

Clicando em Next

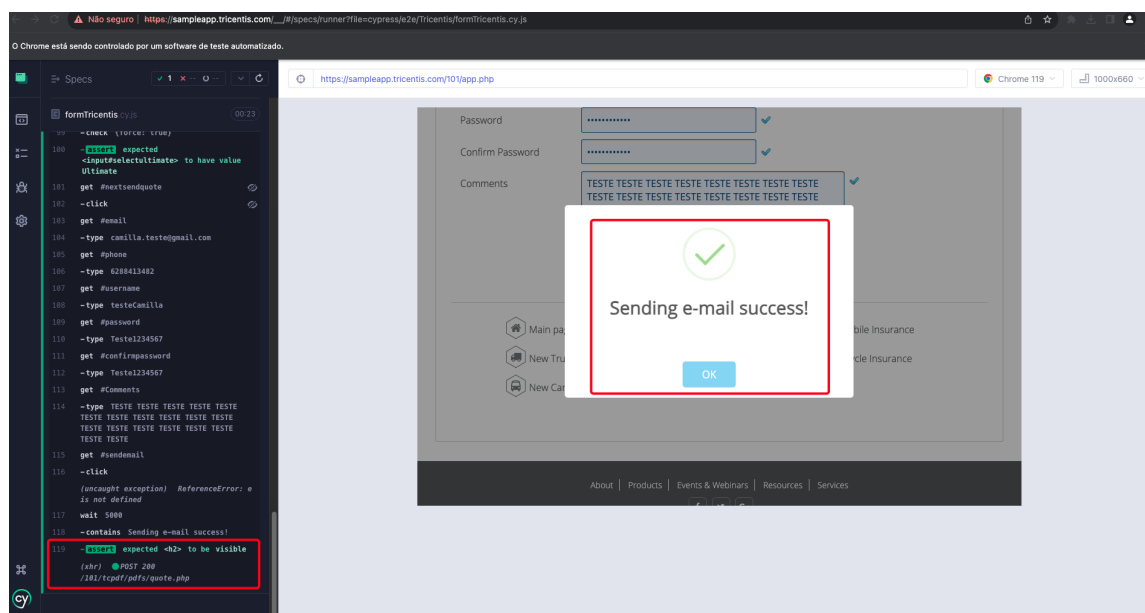
Next

Preenchendo campos na aba EnterVehicleData

9 Pinned before after Highlights X



-Verificação e documentação da mensagem de sucesso “Sending e-mail success!”.



-Observações Adicionais para Documentação:

-Incluir comentários no código fonte do Cypress para explicar a lógica de teste.

Scripts com comentários explicativos adicionados no projeto.

-Documentar as validações realizadas em cada etapa.

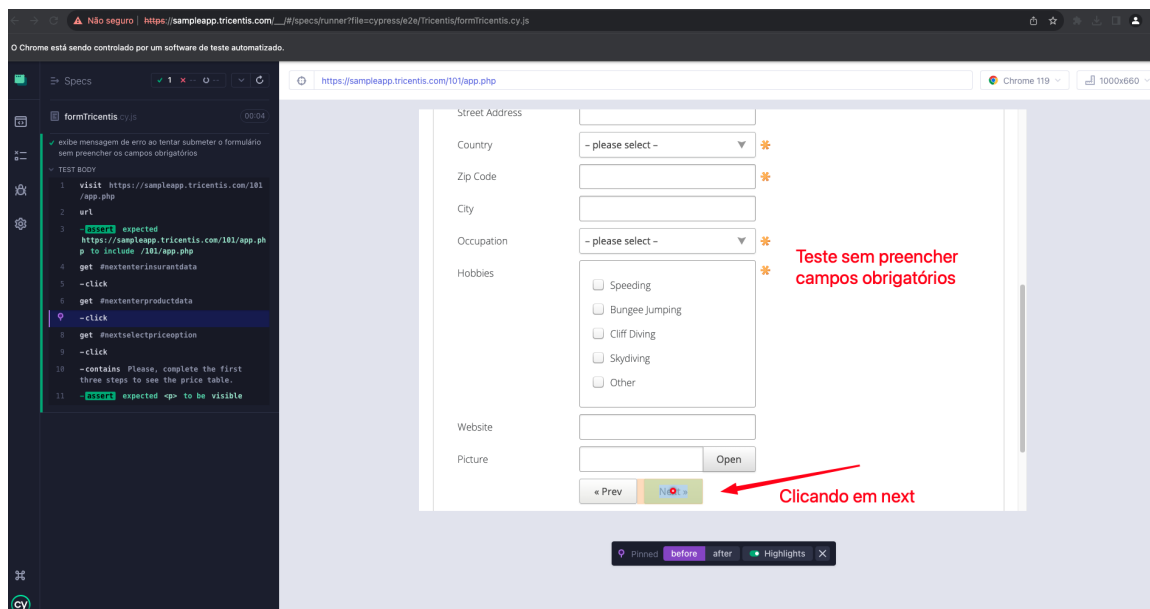
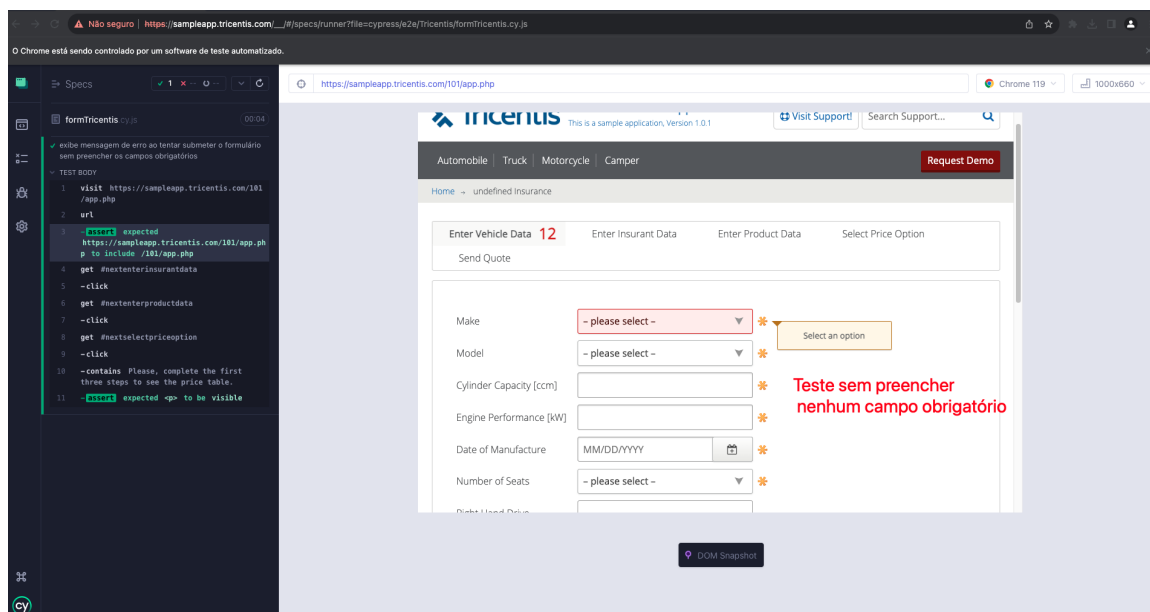
Toda a documentação e validações(evidências) estão registradas no presente documento.

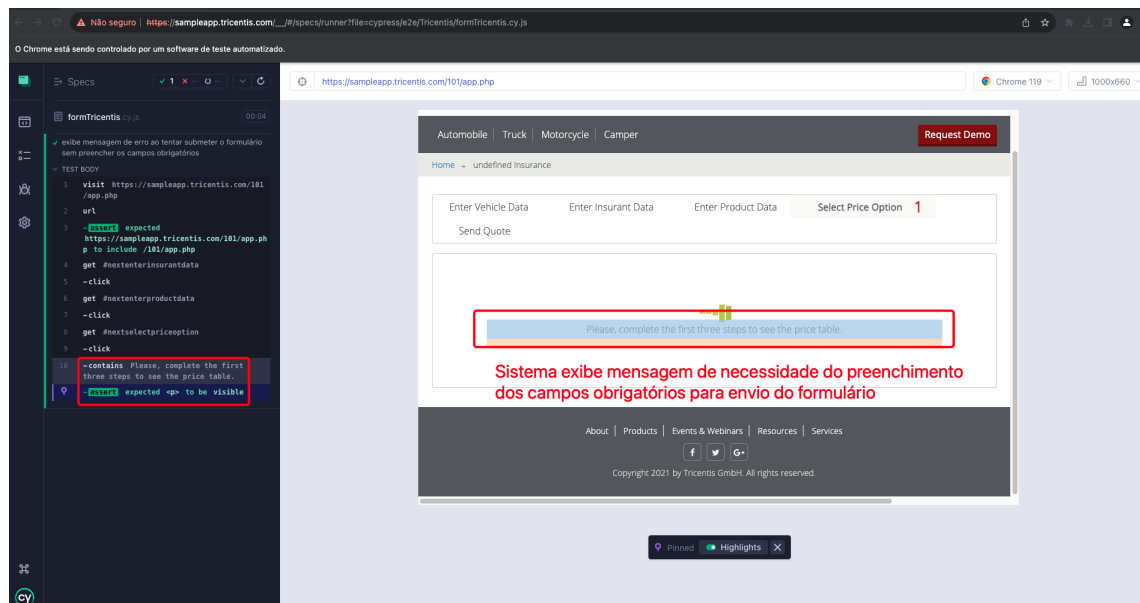
-Registrar quaisquer problemas ou peculiaridades encontradas durante os testes.

As peculiaridades e problemas encontrados se trata da criação dos casos de testes negativos da aplicação e a validação de como ela deve se comportar.

Caso de Teste: Exibe mensagem de erro ao tentar submeter o formulário sem preencher os campos obrigatórios

Given que estou na página da Tricentis
When não preencho todos os campos obrigatórios do formulário
Then sistema não permite envio do formulário e informa necessidade de preenchimento dos campos obrigatórios



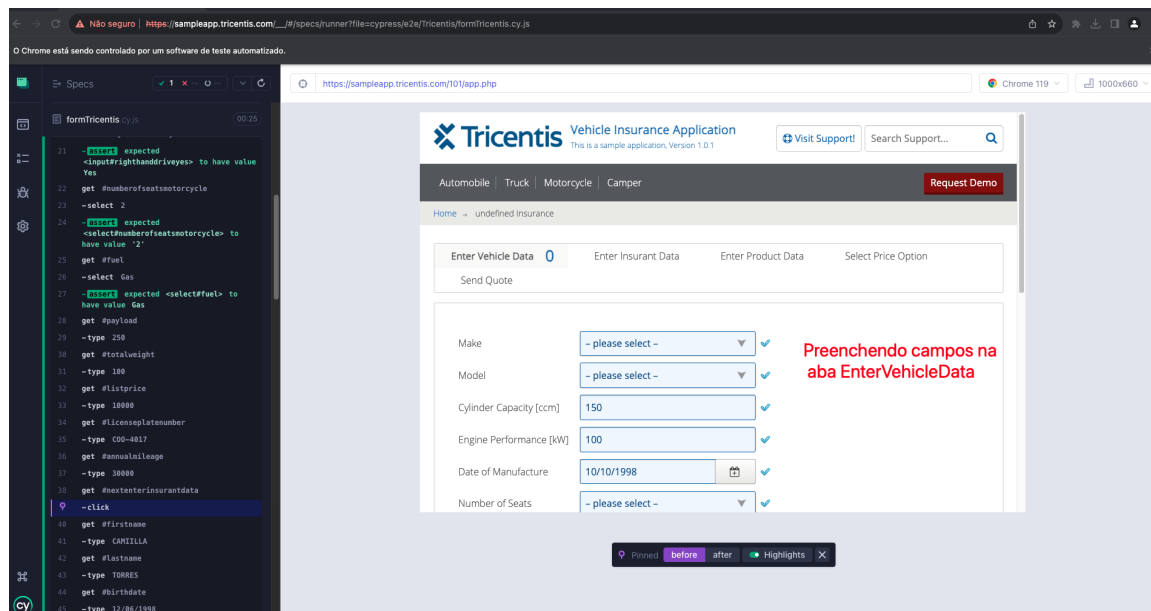


Caso de Teste: Exibe mensagem de erro ao submeter o formulário com um site com formatação inválida

Given que estou na página da Tricentis

When não preencho o campo Website com uma formatação válida

Then sistema informa que deve ser um URL válido e campo fica sinalizado em vermelho



Chrome DevTools interface showing a Cypress test run on the left and a web application on the right.

Left Panel (Cypress Test Runner):

- Specs: formTricentis.cy.js
- Test Steps:
 - 27: `expect(select#fuel to have value Gas)`
 - 28: `get #payload`
 - 29: `-type 250`
 - 30: `get #totalweight`
 - 31: `-type 100`
 - 32: `get #listprice`
 - 33: `-type 10000`
 - 34: `get #licenseplatenumber`
 - 35: `-type COO-4017`
 - 36: `get #annualmileage`
 - 37: `-type 30000`
 - 38: `get #nextenterinsurantdata`
 - 39: `-click`
 - 40: `get #firstname`
 - 41: `-type CAMILLA`
 - 42: `get #lastname`
 - 43: `-type TORRES`
 - 44: `get #birthdate`
 - 45: `-type 12/06/1998`
 - 46: `get input[type='radio'] [value='Female']`
 - 47: `-check (force: true)`
 - 48: `expect(input#genderfemale to have value Female)`
 - 49: `get #streetaddress`
 - 50: `-type RUA 5, N.193, SETOR CENTRAL`
 - 51: `get #country`
 - 52: `-select Brazil`
 - 53: `expect(select#country to have value Brazil)`

Right Panel (Web Application):

Vehicle Insurance Application - Enter Vehicle Data

Fields filled in the form:

- Number of Seats: - please select -
- Right Hand Drive: Yes
- Number of Seats: - please select -
- Fuel Type: - please select -
- Payload [kg]: 250
- Total Weight [kg]: 100
- List Price [\$]: 10000
- License Plate Number: COO-4017
- Annual Mileage [mi]: 30000

Buttons: Next, Previous, Cancel

Annotations:

- Red arrow pointing to the "Next" button: **Clicando em Next**
- Red text: **Preenchendo campos na aba EnterVehicleData**

Chrome DevTools interface showing a Cypress test run on the left and a web application on the right.

Left Panel (Cypress Test Runner):

- Specs: formTricentis.cy.js
- Test Steps:
 - 56: `get #city`
 - 57: `-type GOIANIA`
 - 58: `get #occupation`
 - 59: `-select Selfemployed`
 - 60: `expect(select#occupation to have value Selfemployed)`
 - 61: `get input[type='checkbox'] [value='BungeeJumping']`
 - 62: `-check (force: true)`
 - 63: `expect(input#bungeejumping to have value BungeeJumping)`
 - 64: `get input[type='checkbox'] [value='Skydiving']`
 - 65: `-check (force: true)`
 - 66: `expect(input#skydiving to have value Skydiving)`
 - 67: `get #website`
 - 68: `-type www.teste.com.br`
 - 69: `get [type='file']`
 - 70: `expect(input#picturecontainer.ideal-file not to have value undefined)`
 - 71: `-selectFile img.jpg, (force: true)`
 - 72: `get #nextenterproductdata`
 - 73: `-click`
 - 74: `get #startdate`
 - 75: `-type 01/03/2024`
 - 76: `get #insurancesum`
 - 77: `-select 10000000`
 - 78: `expect(select#insurancesum to have value 10000000)`

Right Panel (Web Application):

Vehicle Insurance Application - Enter Insurant Data

Fields filled in the form:

- First Name: CAMILLA
- Last Name: TORRES
- Date of Birth: 12/06/1998
- Gender: Female

Buttons: Previous, Next, Cancel

Annotations:

- Red text: **Preenchendo campos da aba EnterInsurantData**

Chrome DevTools interface showing a Cypress test run on the left and a web application on the right.

Left Panel (Cypress Test Runner):

- Specs: formTricentis.cy.js
- Test Steps:
 - 48: `get #streetaddress`
 - 49: `-type RUA 5, N.193, SETOR CENTRAL`
 - 50: `get #country`
 - 51: `-select Brazil`
 - 52: `expect(select#country to have value Brazil)`
 - 53: `get #zipcode`
 - 54: `-type 75250004`
 - 55: `get #city`
 - 56: `-type GOIANIA`
 - 57: `get #occupation`
 - 58: `-select Selfemployed`
 - 59: `expect(select#occupation to have value Selfemployed)`
 - 60: `get input[type='checkbox'] [value='BungeeJumping']`
 - 61: `-check (force: true)`
 - 62: `expect(input#bungeejumping to have value BungeeJumping)`
 - 63: `get input[type='checkbox'] [value='Skydiving']`
 - 64: `-check (force: true)`
 - 65: `expect(input#skydiving to have value Skydiving)`
 - 66: `get #website`
 - 67: `-type teste@teste`
 - 68: `-contains Must be a valid URL`
 - 69: `expect(span.error to be visible)`

Right Panel (Web Application):

Vehicle Insurance Application - Enter Insurant Data

Fields filled in the form:

- Country: Brazil
- Zip Code: 75250004
- City: GOIANIA
- Occupation: Selfemployed
- Hobbies: Bungee Jumping, Cliff Diving, Skydiving
- Website: teste@teste

Buttons: Previous, Next, Cancel

Annotations:

- Red text: **Teste preenchendo o campo Website com uma formatação inválida, sistema sinaliza**
- Red arrow pointing to the error message: **Must be a valid URL**

3- Produto do Teste com Documentação:

-Scripts Cypress no GitHub, com comentários explicativos.

Scripts com comentários explicativos adicionados no projeto.

-Documentação detalhada no formato de markdown ou PDF junto ao repositório.

Documentação do presente documento adicionada no repositório.

-Vídeo do teste no diretório “video” no repositório, com anotações se necessário.

Evidência em vídeo do teste Formulário Tricentis:

