

CONTEÚDO



- Problema e Justificativa
- Objetivo Geral
- Objetivos Específicos
- Introdução Teórica
- Ferramentas
- Estudo de Caso
- Considerações Finais

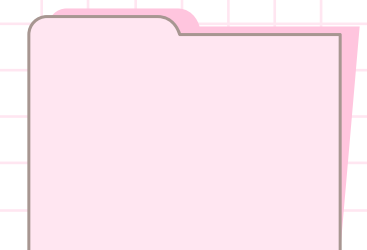
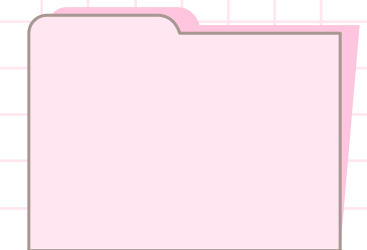
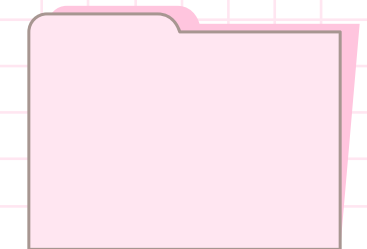
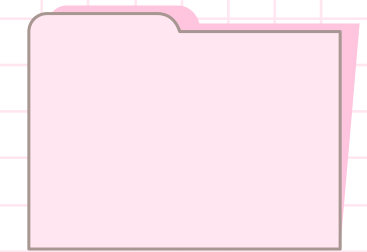


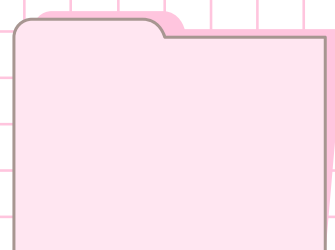
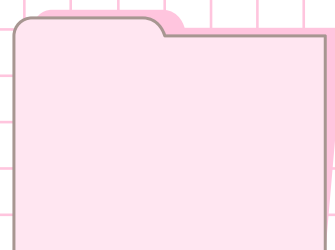
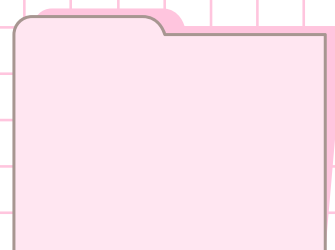
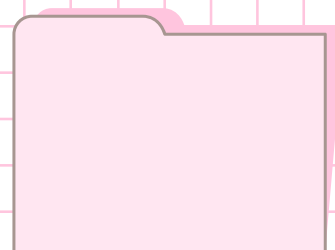
Problema e Justificativa

Testes de ponta-a-ponta representam a oportunidade de escalar atividades e reduzir custos.

O emprego de suítes de teste automatizados aumenta a confiabilidade do projeto. Além de acelerar atividades de teste.

Dessa forma, o desafio inicial de um processo de automatização de testes reside na escolha da ferramenta.





Objetivo Geral

A análise comparativa entre ferramentas contemporâneas voltadas a automação de testes, em um contexto de testes de ponta-a-ponta para WEB.

Objetivos Específicos

- Estudar o paradigma atual do mercado em relação ao emprego de testes de ponta-a-ponta.
- Construir suítes de teste robustas para a demonstração das ferramentas.
- Realizar o estudo de caso comparativo e análise de desempenho das ferramentas selecionadas.

Fundamentação Teórica

No capítulo 3, são abordados tópicos centrais para compreensão do trabalho. Entre eles:

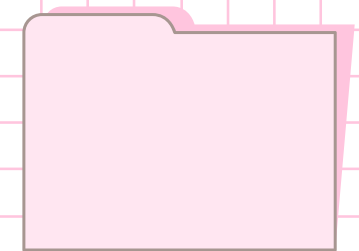
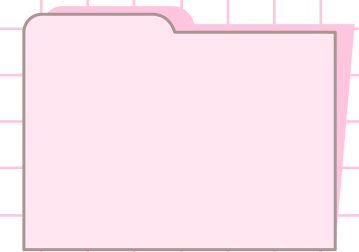
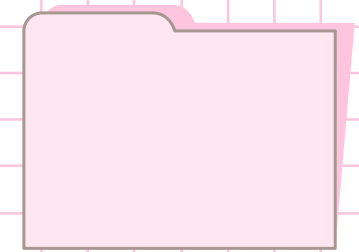
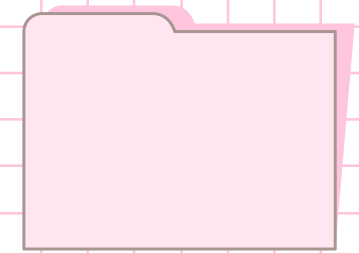
- **3.1 Evolução de ferramentas e estratégias;**
- **3.3 Sobre Qualidade de Software;**
- **3.4 Abordagens de automação para testes E2E em aplicações WEB;**

Evolução de ferramentas e estratégias

A ideia de atividade de teste como algo distinto da programação foi apresentada originalmente em 1972, em Chapel Hill (Carolina do Norte).

Ferramentas de automação aplicadas de maneira independente, ou combinada, auxiliam equipes de controle de qualidade a superar desafios.

Devido ao interesse comercial existente (bem como os benefícios encontrados em sua utilização) diversos pesquisadores mantêm como objetos de trabalho e estudo estes tipos de ferramentas.



Sobre Qualidade de Software

A qualidade de software é comumente definida através de algumas características, normalmente mencionadas como atributos.

É possível compreendê-los como “características de um item que afetam sua qualidade”, ou ainda “medidas que definem a performance de um software

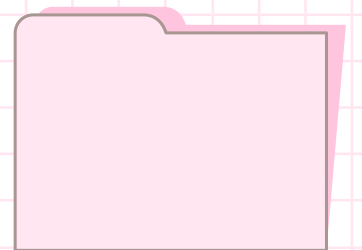
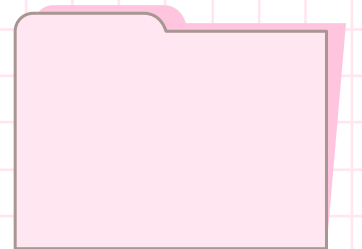
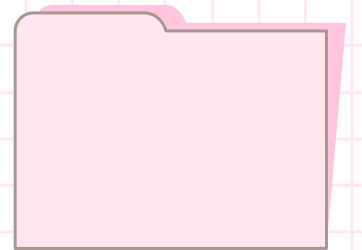
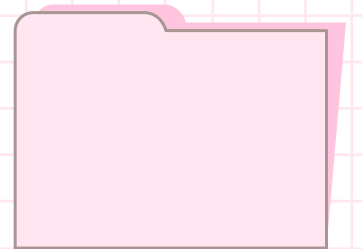
Tanto a parte estrutural do produto como a maneira com que o mesmo se relaciona com o ambiente devem ser levadas em conta. Isso acarreta na criação de atributos internos e externos de qualidade

Atributos Internos de Qualidade

Atributos internos de qualidade são ligados a propriedades estruturais do código, cuja quantificação dependem apenas de sua representação no programa. Podem também ser definidos como propriedades estáticas do código.

Entre eles:

- **Tamanho** → Mensurado a partir do total de linhas de código
- **Complexidade** → Complexidade da implementação
- **Coesão** → Como elementos de um módulo interagem
- **Acoplamento** → Identifica a relação entre diferentes módulos

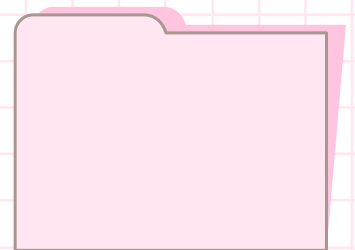
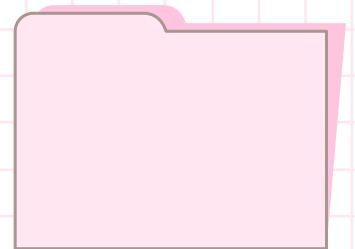
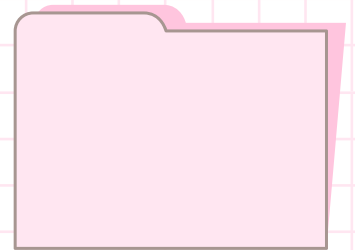
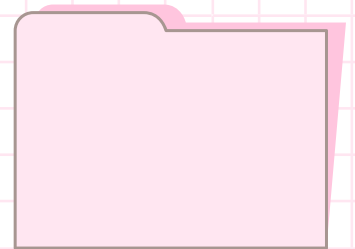


Atributos Externos de Qualidade

Atributos externos de qualidade tornam-se disponíveis após a conclusão da implementação e execução do produto. Refletindo a perspectiva do usuário, pois são afetados diretamente pela máquina, sistema operacional e padrão de utilização.

Entre eles:

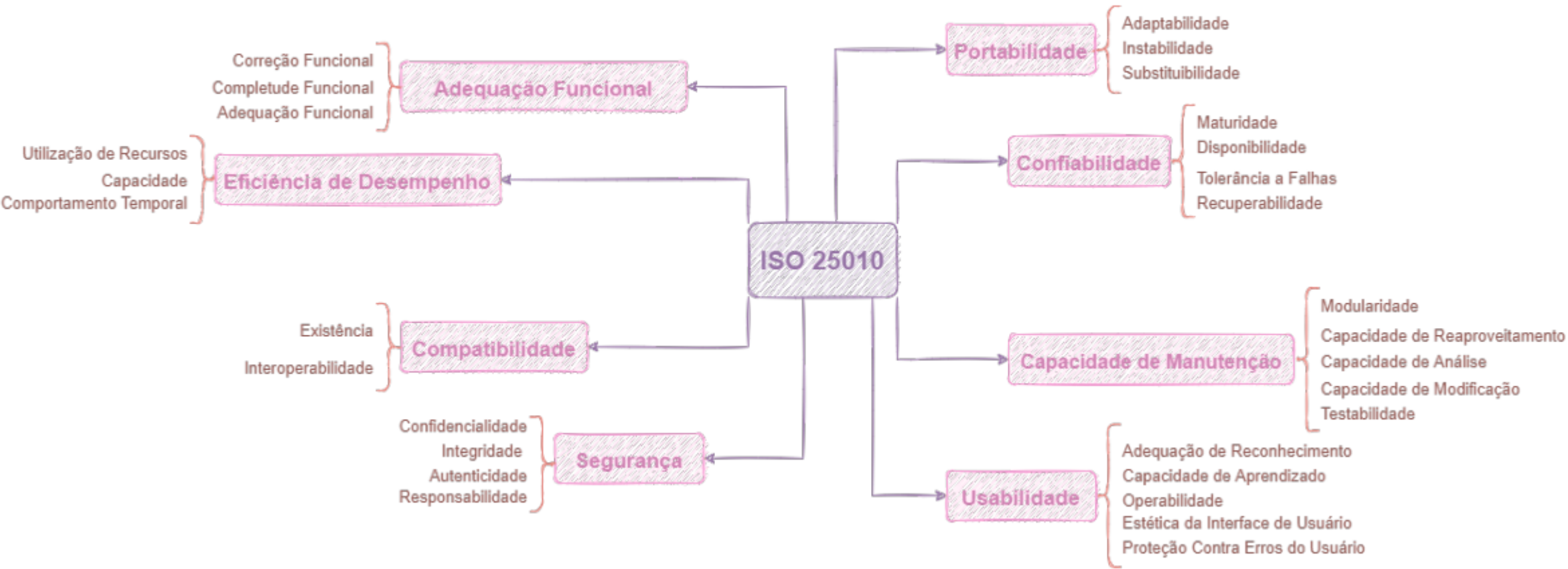
- **Capacidade de suporte** → Capacidade de acomodação das necessidades do usuário
- **Capacidade de Manutenção** → Facilidade de adaptação e modificação
- **Flexibilidade** → Uso efetivo em diferentes aplicações, sistemas ou ambientes
- **Usabilidade** → Facilidade de operação, e execução de atividades pelo usuário
- **Confiabilidade** → Execução das funções definidas com respeito as especificações
- **Portabilidade** → Possibilidade de uso do produto em diferentes ambientes
- **Interoperabilidade** → Capacidade de troca de informação entre partes do sistema
- **Desempenho** → A capacidade de um sistema de realizar suas funções



Modelo de Qualidade ISO 25010

O modelo ISO 25010 considera características de segurança que o tornam o mais relevante que seu antecessor, além de distinguir compatibilidade de portabilidade. Dito isto, similar a ISO 9126-1, não apresenta uma descrição detalhada de medições e também não possui características conectadas as métricas.

Figura 16 – Modelo ISO 25010.



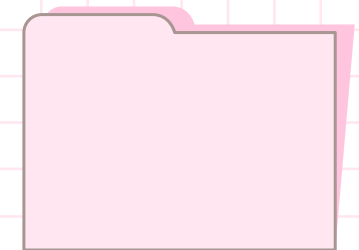
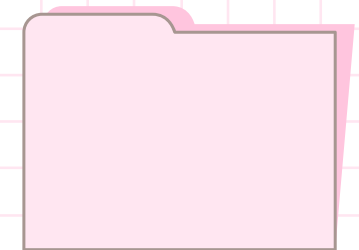
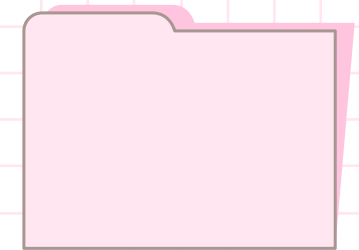
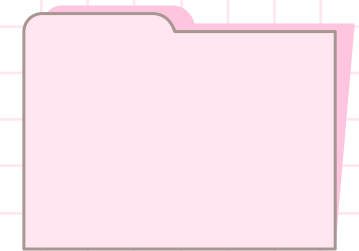
Fonte: Reprodução própria

Abordagens de automação para testes E2E em aplicações WEB

Testes de ponta-a-ponta representam a oportunidade de escalar atividades e reduzir custos.

O emprego de suítes de teste automatizados aumenta a confiabilidade do projeto. Além de acelerar atividades de teste.

Dessa forma, o desafio inicial de um processo de automatização de testes reside na escolha da ferramenta.

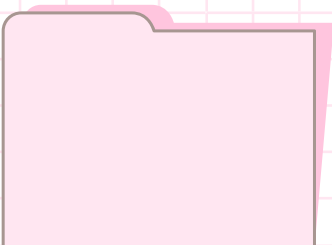
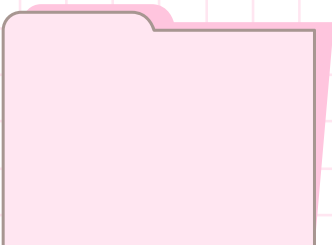
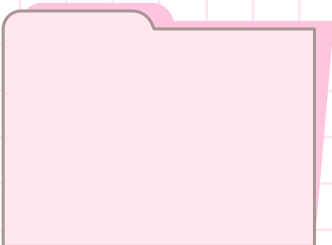
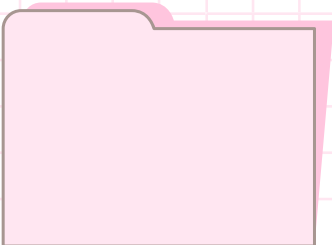


Critérios Ortogonais

- Forma de implementação dos scripts de teste;
- Forma de localização dos elementos em tela;

		Como desenvolver casos de teste			
		Captura-Replay		Programável	
Como localizar elementos web	Geração	3º	Baseado em Visual		
		2º	Baseado em DOM		
		1º	Baseado em Coordenadas		

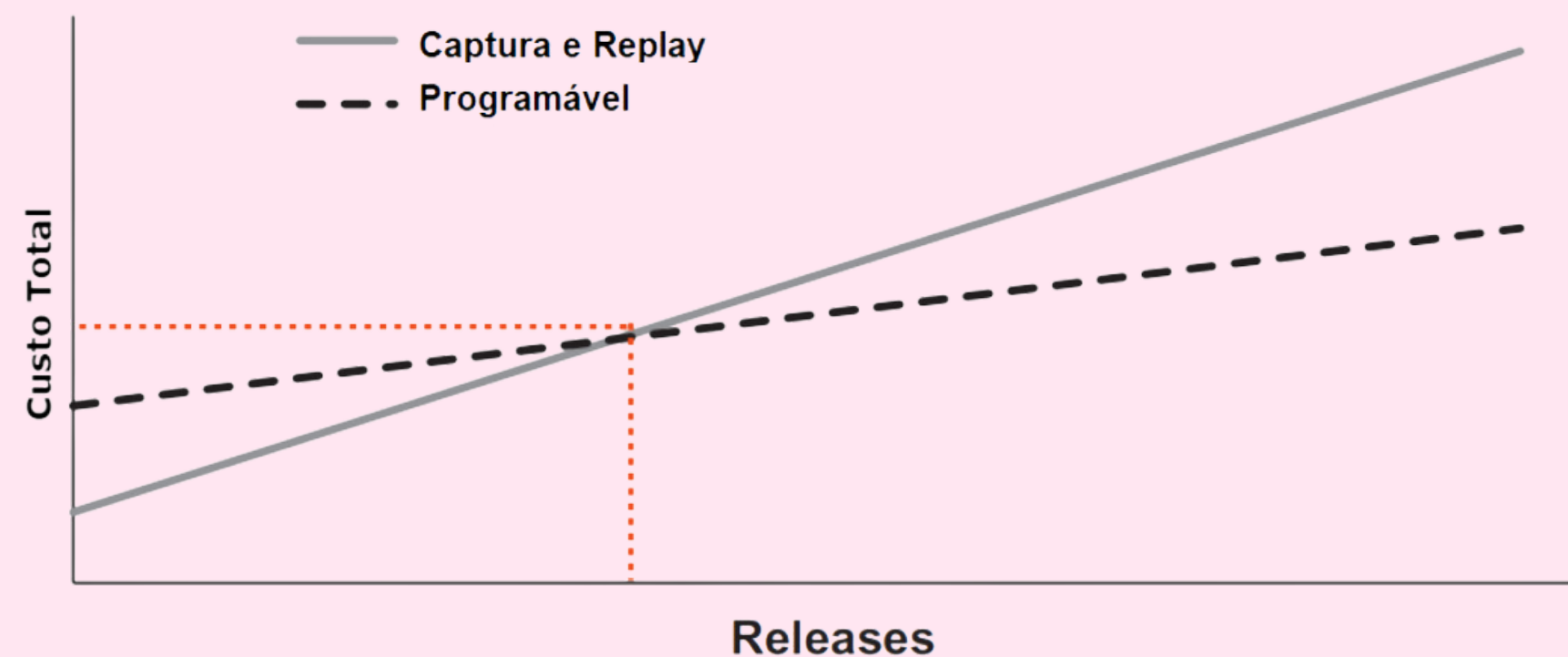
Fonte: (LEOTTA et al., 2016) traduzido pela autora.



Evolução de custos para casos de teste

Em seu trabalho Leotta et al. (2016) verificou empiricamente o trade-off entre as estratégias baseadas em captura e replay, e as abordagens programadas.

Figura 17 – Evolução de custos para casos de teste para estratégia C&R e programável.



Fonte: (LEOTTA et al., 2016) traduzido pela autora.

Ferramentas

```
Describe('Testing Cypress.io', () => {
  before(() => {
    cy.visit('https://cypress.io');
  });

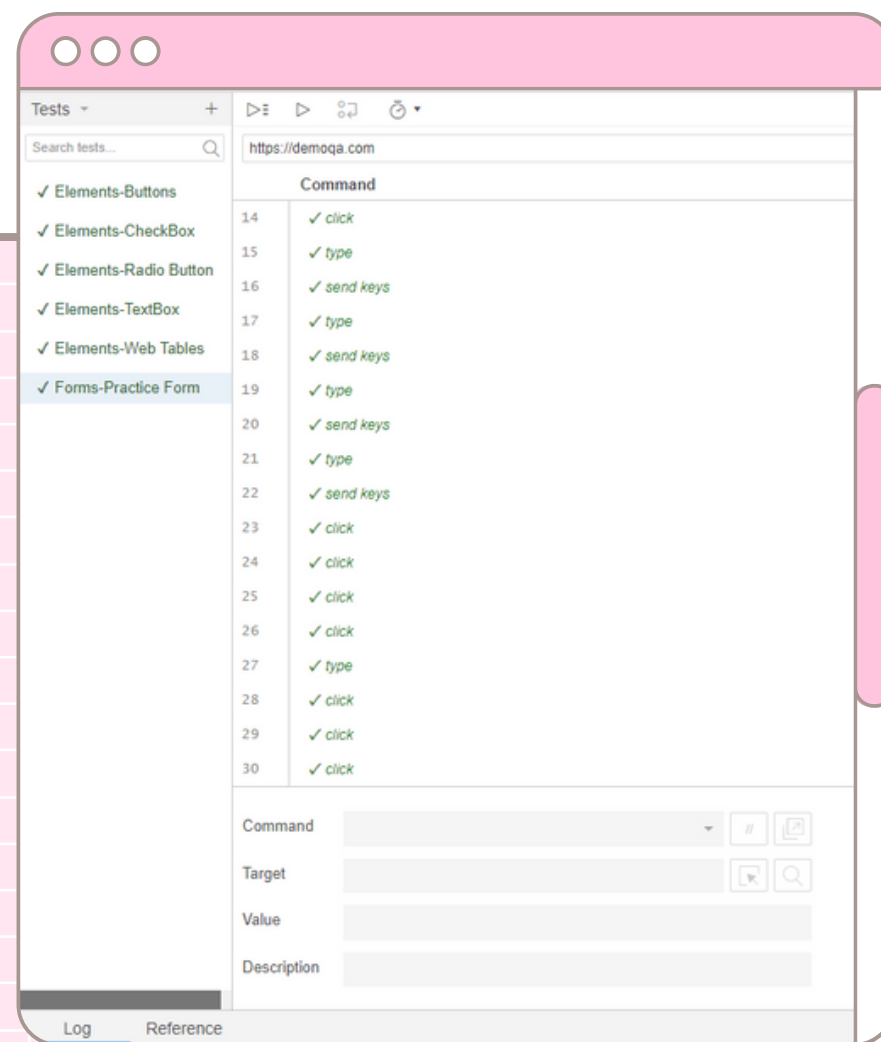
  it('Closing banners should close banners', () => {
    // Testing top banner
    cy.get('.close-top-banner-btn')
      .should('be.visible')
      .click()
      .should('not.exist');

    // Testing cookie consent
    cy.get('.cookieConsent').should('be.visible');
  });
});
```

```
1 Feature: Simple Test
2
3   @test_text
4   Scenario: Test1
5     Given I browse to 'https://pypi.python.org/pypi'
6     #Given I browse to [google_home.url]
7     When I enter 'selenium' to [pypi.home.txt_search]
8     And I click ['#submit']
9     Then I see browser url contains 'action=search&term=selenium'
10    Then I see ['#content > div.section > h1'] text contains 'Index of Packages M
11    #Then I see [pypi.home.txt_search] @value contains 'sele'
12    Then I see [pypi.home.txt_search] @value not contains 'sele'
13    And I see ['#term'] @value is empty
14    And I see ['/*[id="term"]'] @value is empty
15
16   @test_select
17   Scenario: Test select
18     Given I browse to [bootswatch.url.custom]
19     When I select '3' text in [.element.cbx_test]
20     Then I see [...cbx_test] @id contains 'select'
21
22   @test_checkbox
23   Scenario: Test checkbox
24     Given I browse to [bootswatch.url.default]
25     Then I see [bootswatch.element.chk_test] is unchecked
```

Selenium IDE

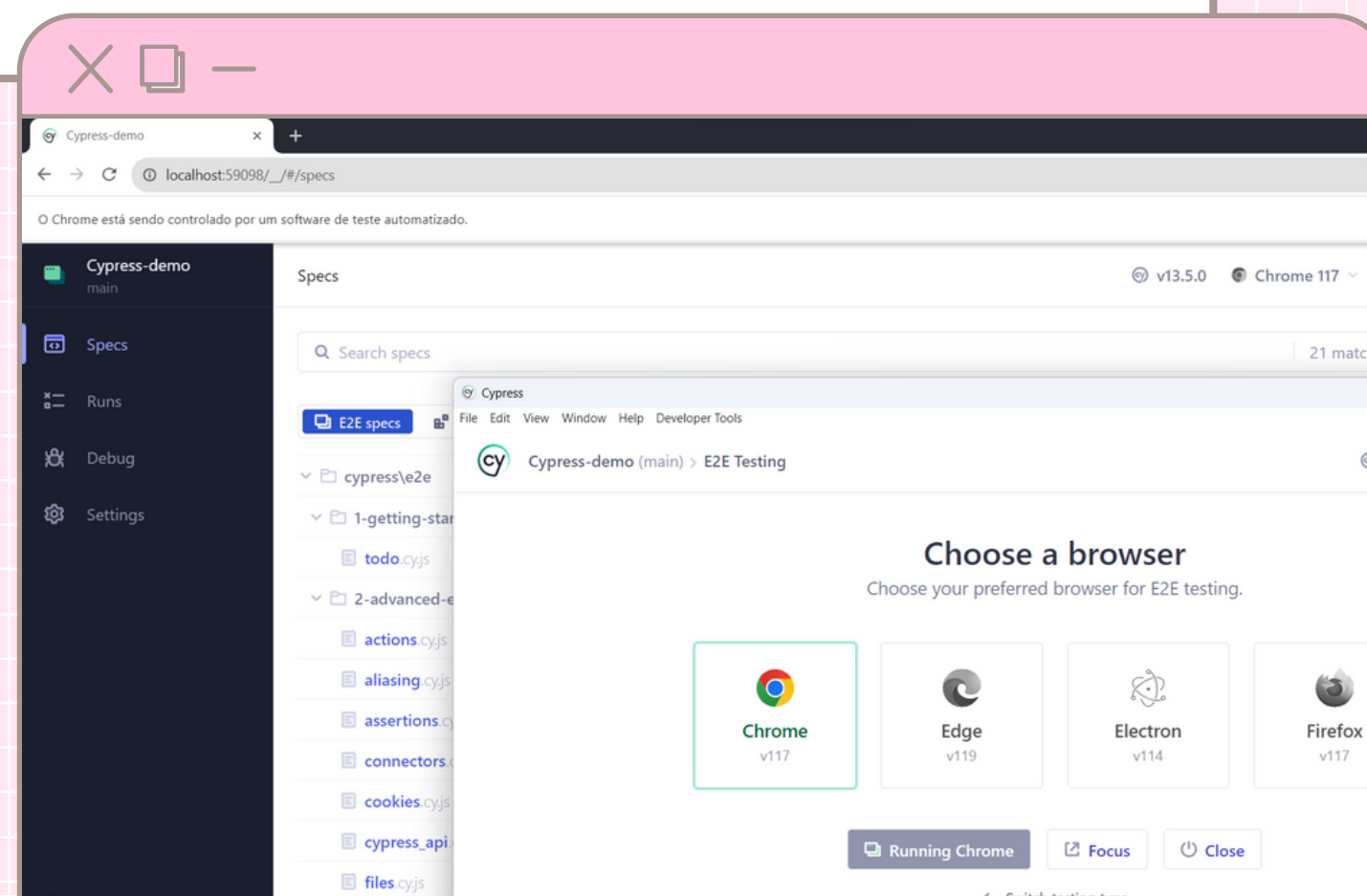
Selenium IDE, onde IDE significa Integrated Development Environment (do inglês "Ambiente de Desenvolvimento Integrado") é uma ferramenta que permite gravar as ações do usuário a partir de comandos Selenium com parâmetros definidos para o contexto de cada elemento disposto em tela. É uma ferramenta que funciona como extensão do navegador web.



Cypress

Cypress é uma ferramenta de teste front end para web, construída pensando nas aplicações mais recentes da dita web 2 e 3.0, onde as páginas são desenvolvidas de maneira mais dinâmica e fluida.

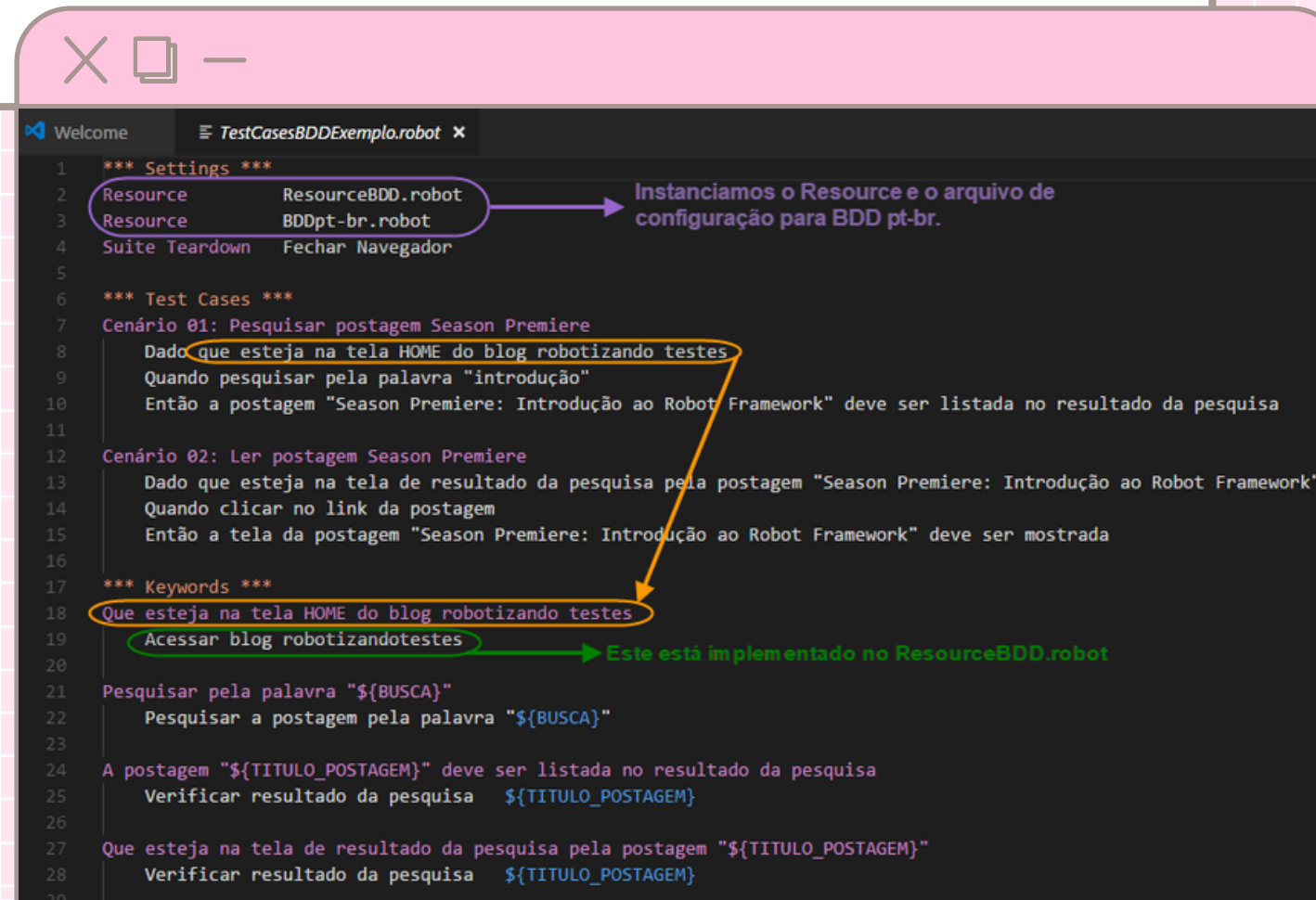
A ferramenta consiste no Cypress Cloud (para gravar as execuções dos cenários de teste), e uma aplicação gratuita, open source, instalada localmente. Após a construção da suite de testes, há a possibilidade de utilizar a Cypress Cloud para um esforço de Integração Contínua (CI, do inglês "Continuous Integration")



Robot

Robot é um framework de automação open source genérico, sendo utilizado tanto para automação de testes quanto para processos de automação robótica (RPA, do inglês "Robotic Process Automation"). É de uso gratuito, sem custos de licenciamento.

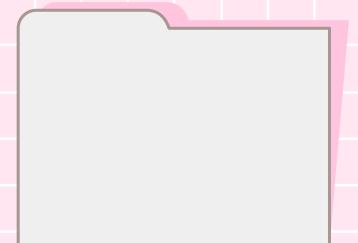
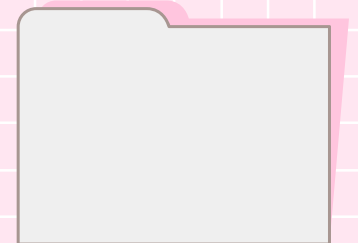
O framework possui um rico ecossistema, composto por bibliotecas e ferramentas que são desenvolvidas como projetos separados.



```
1 *** Settings ***
2 Resource      ResourceBDD.robot
3 Resource      BDDpt-br.robot
4 Suite Teardown Fechar Navegador
5
6 *** Test Cases ***
7 Cenário 01: Pesquisar postagem Season Premiere
8   Dado que esteja na tela HOME do blog robotizando testes
9   Quando pesquisar pela palavra "introdução"
10  Então a postagem "Season Premiere: Introdução ao Robot Framework" deve ser listada no resultado da pesquisa
11
12 Cenário 02: Ler postagem Season Premiere
13   Dado que esteja na tela de resultado da pesquisa pela postagem "Season Premiere: Introdução ao Robot Framework"
14   Quando clicar no link da postagem
15   Então a tela da postagem "Season Premiere: Introdução ao Robot Framework" deve ser mostrada
16
17 *** Keywords ***
18 Que esteja na tela HOME do blog robotizando testes
19   Acessar blog robotizandotestes
20
21 Pesquisar pela palavra "${BUSCA}"
22   Pesquisar a postagem pela palavra "${BUSCA}"
23
24 A postagem "${TITULO_POSTAGEM}" deve ser listada no resultado da pesquisa
25   Verificar resultado da pesquisa  ${TITULO_POSTAGEM}
26
27 Que esteja na tela de resultado da pesquisa pela postagem "${TITULO_POSTAGEM}"
28   Verificar resultado da pesquisa  ${TITULO_POSTAGEM}
```

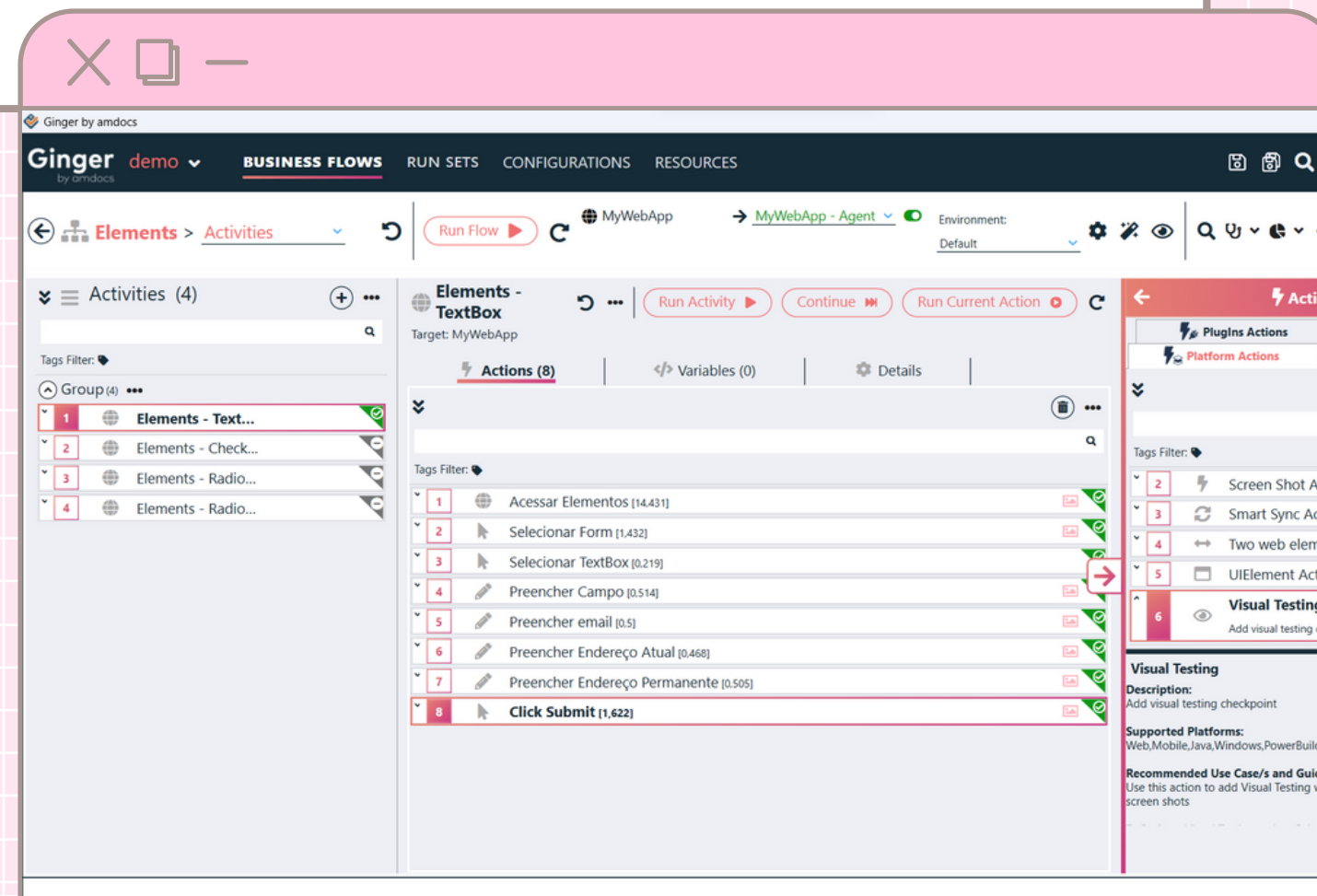
Annotations in the image:

- Line 2: ResourceBDD.robot → Instanciamos o Resource e o arquivo de configuração para BDD pt-br.
- Line 8: Dado que esteja na tela HOME do blog robotizando testes → (points to line 18)
- Line 18: Que esteja na tela HOME do blog robotizando testes → (points to line 19)
- Line 19: Acessar blog robotizandotestes → Este está implementado no ResourceBDD.robot



Ginger by Amdocs

Ginger é um framework de automação de código aberto que maximiza os benefícios do DevOps, metodologias de desenvolvimento ágil e migração para nuvem. Permite testes contínuos com automação orientada por IA, orquestração de scripts (criados por ferramentas de qualquer linguagem de programação), criação de dados de teste e automação de gerenciamento de ambiente de teste.



Testes E2E e o mercado

Em uma aplicação *frontend* testes E2E automatizam a execução de um fluxo através de um navegador web para verificar o comportamento do sistema.

Os testes E2E exemplificam o papel do teste de software em um mercado em que a utilização de estratégias de DevOps prevalece de maneira marcante.

Isso ocorre por levarem em conta a conexão inerente entre como software é construído e utilizado. Ao garantir aos desenvolvedores uma visão de como alterações impactam a jornada do usuário final, os mesmos tornam-se melhores equipados para entregar valor a esses usuários

Testes E2E e o mercado

É possível dividir as histórias comerciais de fracasso de suítes E2E em duas grandes categorias:

- Suítes de teste com **crescimento desordenado**;
- Suítes de teste com **manutenção defasada**;

Dentro da indústria ainda se compreende a estratégia de testes de ponta-a-ponta como viável e válida.

Entretanto alguns pontos de dor reduzem o valor entregue por esse tipo de produto, o que não permite que essa seja uma experiência universal:

- **Tempo de espera entre execuções**;
- **Falta de confiança devido a instabilidade**;
- **Custo de manutenção**;
- **Dubug nebuloso**;
- **Entrega lenta**;
- **Ineficiência**;
- **Ineficácia**;

Considerações Finais

O desenvolvimento do presente trabalho possibilitou a análise de um conjunto de ferramentas de automação, em um contexto de testes de ponta-a-ponta para web. Além disso, também permitiu a investigação do paradigma atual da indústria em relação a essa categoria de testes.

Da mesma maneira, os testes de ponta-a-ponta se provam instrumentais para a verificação da integração entre os diferentes componentes destes complexos sistemas de software modernos. Auxiliando no trabalho de manter uma experiência de usuário consistente, além de identificar os desafios e fragilidades de um projeto de software.

A partir da análise das ferramentas realizada utilizando o modelo ISO 25010, foi possível constatar que (em um contexto de automação de testes E2E para web) a ferramenta Ginger mostrou-se mais versátil. Ademais, demonstrou-se a mais amigável e de fácil instalação em relação aos recursos disponíveis.

Em relação a posição da indústria no que toca aos testes de ponta-a-ponta, é possível argumentar que - apesar das vantagens citadas anteriormente que derivam dessa estratégia de teste -, os testes E2E costumam ser implementados de maneira errônea. Muitas vezes tendo um número exacerbado de cenários a cobrir, com uma manutenção mínima das suítes de automação. O que resulta no aumento do número de falhas (além de falsos negativos e positivos), e acarreta no abandono das mesmas.

Referências

ACKERSON, Z.; SAJJAD, D. **Why End-to-End (E2E) Testing Is Often Good Enough**. 2022. Disponível em: <<https://semaphoreci.com/blog/e2e-testing>>.

AMARA, D.; RABAI, L. B. A. **Software quality measurement: State of the art**. In: . [S.l.:s.n.], 2019. p. 150–181. ISBN 9781522557944.

AMDOCS. **Ginger by Amdocs Help**. [S.l.], 2023. Disponível em: <https://ginger-automation.github.io/Ginger-Web-Help/assets/Ginger_By_Amdocs/Getting_Started/What_is_Ginger.htm>.

ANGMO, R.; SHARMA, M. **Performance evaluation of web based automation testing tools**. 2014. p. 731-735.

ATESOGULLARI, D.; MISHRA, A. Automation testing tools: A comparative view. **International Journal of Information and Computer Security**, v. 12, p. 63–76, 12 2020. BADAL, L.; GRUNLER, D. Automated end-to-end testing: Useful practice or frustrating time sink? v. 1, p. 01–07, 04 2021.

BUXTON, J.; RANDELL, B. **Software engineering techniques** - homepages.cs.ncl.ac.uk. NATO Science Committee, 1970. Disponível em: <<http://homepages.cs.ncl.ac.uk/brian.randell/NATO/nato1969.PDF>>.

CERQUOZZI, R.; DECOUTERE, W.; DUSSA-ZIEGER, K.; RIVERIN, J.-F.; HRYSZKO, A.; KLONK, M.; PILAETEN, M.; POSTHUMA, M.; REID, S.; COSQUER, E. Riou du; ROMAN, A.; STAPP, L.; ULRICH, S.; ZAKARIA, E. [s.n.], 2023. Disponível em: <https://istqb-main-web-prod.s3.amazonaws.com/media/documents/ISTQB_CTFL_Syllabus-v4.0.pdf>.

CHELUVARAJU, B.; NAGAL, K.; PASALA, A. Mining software revision history using advanced social network analysis. **2012 19th Asia-Pacific Software Engineering Conference**, p.717–720, 2012.

Referências

CYPRESS. **Cypress Documentation**. [S.l.], 2023. Disponível em: <<https://docs.cypress.io/guides/overview/why-cypress>>.

DAM, K. **The future of testing: Digging in the past of software testing and unearthing the future**. In: . [S.l.: s.n.], 2020. p. 197–205. ISBN 978-3-030-29508-0.

ELM, D. **Dominic Elm - Cypress: The future of E2E testing**. 2019. Disponível em:<<https://www.youtube.com/watch?v=pXyBligMMr0>>.

EMMANUEL, O. **The Problem with End-to-End Tests**. 2019. Disponível em:<<https://levelup.gitconnected.com/the-problem-with-end-to-end-tests-65509df4bc7a>>.

FOGG, E. **Why E2E Test Suites Fail—And How to Avoid Breakdowns**. 2020. Disponível em: <<https://prodperfect.com/blog/end-to-end-testing/why-test-suites-fail-and-how-to-avoid-breakdowns/>>.

FREIRE, A. **Why we killed our end-to-end test suite**. Nubank, 2022. Disponível em: <<https://building.nubank.com.br/why-we-killed-our-end-to-end-test-suite/>>.

GEORGIAN, S. **What is end-to-end testing and when should you use it?** freeCodeCamp.org, 2021. Disponível em: <<https://www.freecodecamp.org/news/end-to-end-testing-tutorial/>>.

HUGHES, B. **In Defense of Comprehensive End-to-End Testing**. 2023. Disponível em: <<https://www.mabl.com/blog/in-defense-of-comprehensive-end-to-end-testing>>.

JOHNSTON, B. **Tech blog**. 2016. Disponível em: <<https://tech.justeattakeaway.com/2016/10/04/why-fewer-end-to-end-tests/>>.

Referências

LEOTTA, M.; CLERISSI, D.; RICCA, F.; TONELLA, P. **Chapter five - approaches and tools for automated end-to-end web testing.** In: MEMON, A. (Ed.). Elsevier, 2016, (Advances in Computers, v. 101). p. 193–237. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0065245815000686>>.

PINHEIRO, P. **Tipos de frameworks de automação de testes.** LinkedIn, Feb 2023. Disponível em: <<https://www.linkedin.com/pulse/tipos-de-frameworks-automa%C3%A7%C3%A3o-testes-paulo-pinheiro>>.

PRESSMAN, R.; MAXIM, B. **Engenharia de Software - 8ª Edição.** [s.n.], 2016. ISBN 9788580555349. Disponível em: <<https://books.google.com.br/books?id=wexzCwAAQBAJ>>.

RAVIART, D. **Amdocs Simplifies the Test Case-to-Test Script Process with Ginger.** NelsonHall, 2019. Disponível em: <https://research.nelson-hall.com/blogs/?avpage-views=blog&type=post&post_id=976>.

ROBOT. **Robot framework user guide.** [S.l.], 2023. Disponível em: <<https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html>>.

SELENIUM. **Selenium - Getting Started.** [S.l.], 2019. Disponível em: <<https://www.selenium.dev/selenium-ide/docs/en/introduction/getting-started>>.

SMART, J.; MOLAK, J. BDD in Action, Second Edition: **Behavior-Driven Development for the Whole Software Lifecycle.** Manning, 2023. ISBN 9781617297533. Disponível em: <<https://books.google.com.br/books?id=hIK3EAAAQBAJ>>.

SOMMERVILLE, I. **Engenharia de software.** Pearson Prentice Hall, 2011. ISBN 9788579361081. Disponível em: <<https://books.google.com.br/books?id=H4u5ygAACAAJ>>.

TECHGEEKNEXT. **RobotFramework tutorial - robotframework architecture [updated] 2023.** TechGeekNext, 2023. Disponível em: <<https://www.techgeeknext.com/robot-framework-architecture>>.

Muito obrigada pela atenção!

