

GTI

SPRINT 2 – MISSÃO 6

PROJETO: “DEPLOYMENT QUALITY ASSURANCE”

ESTUDO DE CASO

Uma certa empresa decidiu estabelecer uma cultura *QUALITY ASSURANCE* em seu modelo de negócio, visando impactar positivamente processos de qualidade em suas áreas de operação e tecnologia.

ESCOPO DO PROJETO

O projeto será composto por 3 Sprints que se complementam, onde os alunos deverão construir ações que validem a empresa a possuir uma cultura orientada a Q.A.

Em **duplas** os alunos desenvolverão projeto 3 em Sprints:

- SPRINT 1: Vale 0,5 ponto na AC-1 e presenças nas aulas
- **SPRINT 2: Vale 1 ponto na AC-2 e presenças nas aulas**
- SPRINT 3: Vale 1 ponto na AC-3 e presenças nas aulas

OBJETIVO

Aprender as nuances e aplicabilidade do *QUALITY ASSURANCE* em uma organização. Construir um projeto de implementação de Gerenciamento de Qualidade Total e realizar atividades que valem nota.

SPRINT 2 (1 ponto)

Início: **18/09** – Término: **09/10**. Vale 1,0 ponto na AC-2 e presenças nas aulas. Composto por 4 missões que se complementam para a entrega total do projeto:

- Missão 5: Automação de Testes I – Vale 25% da AC-2
- **Missão 6: Automação de Testes II – Vale 25% da AC-2**
- Missão 7: Testes de API /QA em Mobile – Vale 25% da AC-2
- Missão 8: Validações e entrega final – Vale 25% da AC-2

MISSÃO 5

VALE 25% DA NOTA AC-2

ETAPA 1 – SELENIUM IDE

FERRAMENTA DE AUTOMAÇÃO DE TESTES – SELENIUM IDE

Selenium IDE (Integrated Development Environment): é uma extensão de navegador usada para gravar, editar e reproduzir interações com páginas web. Ele é muito útil para iniciantes que desejam criar testes simples de maneira rápida, sem precisar escrever código.

Trata-se de uma ferramenta de gravação e reprodução que facilita a criação de testes sem a necessidade de programação. Ideal para iniciantes ou para criação de testes rápidos.

TAREFA 1 – PREPARAÇÃO:

1. Baixe o arquivo esse “**Missão6-Projeto QA - ADS-5.pdf**” disponível no AVA;
2. Abra o GitHub oficial da dupla/trio e em seguida abra o repositório que estão usando para o projeto;
3. Suba no seu repositório o arquivo “**Missão6-Projeto QA - ADS-5.pdf**”;
4. Agora abra o projeto deste repositório e visualize o quadro Kanban que está gerenciando o projeto;
5. Criar e colocar o cartão MISSÃO 6 para a lista EM ANDAMENTO;
6. Abra o Google Forms para criar um formulário que será utilizado como base;
7. Renomeie o formulário para TESTE-1 AUTO;
8. Crie 4 perguntas de pesquisa, todas no formato múltipla escolha, com 4 opções de resposta como se você estivesse realizando uma pesquisa de satisfação do cliente;

TAREFA 2 – CONFIGURAÇÃO SELENIUM IDE

Siga para o processo de instalação do SELENIUM IDE:

9. Instalar o Selenium IDE no Navegador
 - a. Abra o **Chrome**.
 - b. Vá até a Chrome Web Store e pesquise por "Selenium IDE" ou acesse este link.
 - c. Clique em **Adicionar ao Chrome** e depois em **Adicionar Extensão**.
10. Após a instalação, você verá o ícone do **Selenium IDE** na barra de ferramentas do navegador. Clique no ícone para abrir a ferramenta.
11. Na primeira vez que abrir, você verá algumas opções, como:

- a. **Create a new project** (Criar um novo projeto)
- b. **Open an existing project** (Abrir um projeto existente)

12. Clique em "Create a new project";

13. Escolha um nome para o projeto (exemplo: "Teste de Automação") e clique em **OK**.

14. Clique no botão "Record a new test in a new project" (Gravar um novo teste).

15. Escolha um nome para o seu teste (exemplo: "Teste Formulário");

TAREFA 3 – ROBÔ TESTE 1 – SELENIUM IDE

16. Digite a URL do site que você deseja testar (no caso o link do formulário)

17. Clique em **Start recording** (Iniciar gravação). O Selenium IDE começará a gravar todas as interações que você fizer no navegador. Você pode:

- a. Navegar por páginas.
- b. Clicar em botões.
- c. Inserir texto em campos de formulário.
- d. Enviar formulários.
- e. Feche a página do formulário

18. Quando terminar de gravar as interações, volte à janela do Selenium IDE e clique no botão **Stop recording** (Parar gravação).

19. Executar o Teste Gravado: Após parar a gravação, você verá uma lista de comandos gravados no Selenium IDE.

20. Para executar o teste gravado, clique no botão **Run current test** (Executar teste atual). O Selenium IDE abrirá uma nova janela e executará automaticamente os passos que você gravou. (no relógio, diminua a velocidade)

21. Você poderá ver os resultados no painel inferior da interface do Selenium IDE.

22. Selecione Save Project (Salvar Projeto um com um nome exemplo: "Teste Forms") e escolha uma pasta no seu computador para salvar o projeto e os testes.

TAREFA 4 – ROBÔ TESTE 2 – SELENIUM IDE

23. Agora você irá usar o site de simulação de eCommerce SWAG LABS. Acesse pelo Chrome o endereço abaixo:

[Swag Labs \(saucedemo.com\)](http://saucedemo.com)

24. Conheça um pouco a DEMO antes de iniciar o próximo teste. Você irá precisar fazer um teste completo de usabilidade do sistema, realizando o login, escolhendo quatro produtos, indo para o carrinho confirmar a compra, preencher dados e forma de pagamento
25. Realize um novo processo de gravação com o Selenium IDE. Inicie um novo teste com uma nova gravação
26. Insira o endereço citado acima;
27. Faça o Login e Senha: Login: **standard_user** – Pswd: **secret_sauce**.
28. Adicione 4 itens no carrinho;
29. Vá para o carrinho e remova um dos 4 itens;
30. Faça checkout;
31. Insira nome e sobrenome e caixa postal (CEP) fictício e clique em Continue;
32. Confira os valores e carrinho e clique em finish;
33. Ao completar o Checkout, clique em Back Home;
34. Na home, clique na lista de item e faça logout;
35. Feche a janela do browser;
36. Quando terminar de gravar as interações, volte à janela do Selenium IDE e clique no botão **Stop recording** (Parar gravação).
37. Executar o Teste Gravado: Após parar a gravação, você verá uma lista de comandos gravados no Selenium IDE.
38. Para executar o teste gravado, clique no botão **Run current test** (Executar teste atual). O Selenium IDE abrirá uma nova janela e executará automaticamente os passos que você gravou.

TAREFA 5 – FINALIZAÇÃO ETAPA 1

39. Após gravar, clique no menu de três pontos no canto superior direito da interface do Selenium IDE.

40. Selecione **Save Project** (Salvar Projeto) e escolha uma pasta no seu computador para salvar o projeto e os testes.
41. Chame o professor para validar todos os dois testes (representa 50% do valor da nota do dia)
42. Salve os códigos no MISSÃO 6;

ETAPA 2 – CYPRESS

Instalação e Configuração do Cypress Usando VS Code

TAREFA 6 – INSTALAÇÃO E TESTE DO CYPRESS

43. Instalar o Node.js (O Cypress depende do Node.js para funcionar)
44. Acesse <https://nodejs.org/> e baixe a versão LTS.
45. Instale o Node.js com as configurações padrão.
46. Após a instalação, abra o VS Code e pressione `Ctrl + `` para abrir o terminal integrado.
47. Verifique se o Node.js e o npm foram instalados corretamente com os comandos abaixo (Ambos devem retornar as versões do Node.js e npm, confirmando a instalação).

node -v

npm -v

48. Criar uma Pasta de Projeto no VS Code. Crie uma nova pasta no local desejado (por exemplo, CypressProject).
49. No VS Code, clique em File > Open Folder e selecione a pasta recém-criada.
50. Inicializar o Projeto no VS Code: No terminal integrado do VS Code (abra com `Ctrl + ``), inicialize o projeto Node.js executando:

npm init -y

//Isso criará automaticamente um arquivo package.json, onde as dependências e scripts do projeto serão gerenciados.

51. Instalar o Cypress no Projeto: No terminal do VS Code, execute o comando para instalar o Cypress como dependência de desenvolvimento:

npm install cypress --save-dev

52. Após a instalação, você pode verificar no package.json que o Cypress foi adicionado como uma dependência.

53. Abrir o Cypress Usando o VS Code: No terminal, execute o seguinte comando para abrir o Cypress:

npx cypress open

54. O Cypress abrirá uma interface gráfica e automaticamente criará uma estrutura de pastas no seu projeto, incluindo a pasta cypress/integration, onde você poderá criar seus testes.

55. Escrever um Teste Automatizado Básico no VS Code: agora que o Cypress está configurado, vamos criar um teste básico:

56. Dentro da pasta cypress/integration, crie um arquivo de teste chamado test_spec.js no VS Code.

57. No arquivo test_spec.js, insira o seguinte código:

```
describe('Página de exemplo', () => {  
  it('Deve carregar a página inicial com sucesso', () => {  
    cy.visit('https://example.cypress.io') // Abre a página de exemplo do Cypress  
    cy.contains('type').click() // Clica em um link com o texto 'type'  
    cy.url().should('include', '/commands/actions') // Verifica se a URL mudou  
    cy.get('.action-email').type('teste@exemplo.com') // Digita em um campo de e-mail  
    cy.get('.action-email').should('have.value', 'teste@exemplo.com') // Verifica se o  
    e-mail foi inserido corretamente  
  })  
})
```

58. Executar o Teste no VS Code: Para rodar o teste, abra o Cypress novamente com o comando no terminal do VS Code:

npx cypress open

59. Na interface gráfica, você verá o arquivo de teste test_spec.js. Clique nele para executar o teste. O Cypress abrirá um navegador, executará o teste e exibirá os resultados em tempo real.

60. Configurar Scripts para Facilitar a Execução dos Testes

61. Para facilitar a execução dos testes no futuro, adicione um script no arquivo package.json:

62. Abra o package.json no VS Code. Dentro da seção "scripts", adicione:

```
"scripts": {  
  "test": "cypress open"  
}
```

63. Agora, você pode abrir o Cypress apenas rodando o seguinte comando no terminal do VS Code:

```
npm run test
```

TAREFA 5 – FINALIZAÇÃO ETAPA 2

64. Coloque no fim o nome e RA dos alunos presentes na atividade no cartão de hoje;

65. Coloque o cartão na lista EM VALIDAÇÃO.

66. Mande email para o professor com a URL do projeto no GITHUB:

flavio.santarelli@pro.fecaf.com.br

SUCESSO A TODOS!