

Bloco Bônus – Mini Desafio






Implementação conceitual de uma folha de testes (CRUD de pedidos):

Busca → Cadastro → Atualização → Exclusão

Proposto o uso de **Cypress + Node.js**:

- Fluxo completo de CRUD;
- Validação de resposta HTTP.

Critérios de Avaliação Atendidos

-  Clareza, organização e estrutura dos artefatos
-  Cumprimento dos requisitos obrigatórios
-  Report de bugs completo e detalhado
-  Proposta de processo de QA alinhada ao contexto da empresa
-  Diferencial: análise crítica e documentação prática de automação E2E

Objetivo

Simular um fluxo completo de **CRUD (Create, Read, Update, Delete)** de pedidos de compra, validando:

- Cadastro de novos pedidos;
- Visualização e consistência dos registros existentes;
- Atualização de informações de pedidos;
- Exclusão de registros e validação da remoção.

O objetivo é demonstrar **habilidades práticas em automação de testes E2E (End-to-End)** em um ambiente simulado, permitindo testes de integração, funcionalidade e consistência de dados.

Cenário de Teste

O ambiente de teste consiste em uma tabela de pedidos contendo informações como ID, produto, fornecedor, quantidade e status.

Ação	Descrição	Tipo de Teste
Create	Inserir novo pedido na tabela	Funcional / E2E
Read	Validar exibição correta dos registros	Validação / Integração
Update	Alterar informações de um pedido existente	Funcional / Regressão
Delete	Remover pedido e verificar exclusão	Funcional / Negativo

Casos de Teste Propostos

- CT01 – Adicionar novo pedido

Objetivo: Validar criação de um novo pedido.

Passos:

1. Acessar a página de pedidos do sistema simulado.
2. Clicar no botão **"Adicionar Pedido"**.
3. Preencher os campos obrigatórios (produto, fornecedor, quantidade, status).
4. Confirmar submissão.

— Resultado Esperado:

O novo pedido é exibido corretamente na tabela com todos os dados informados.

- CT02 – Editar pedido existente

Objetivo: Validar edição de informações de um pedido.

Passos:

1. Selecionar o registro a ser editado.
2. Alterar os campos necessários (ex.: quantidade ou status).
3. Salvar alterações.

— Resultado Esperado:

As alterações são refletidas corretamente na tabela e persistidas no banco simulado.

- CT03 – Excluir pedido

Objetivo: Validar exclusão de um pedido.

Passos:

1. Selecionar o registro a ser removido.
2. Clicar no botão de exclusão.
3. Confirmar a ação, se necessário.

— Resultado Esperado:

O registro é removido da tabela e não aparece mais na listagem.

- CT04 – Buscar pedido inexistente

Objetivo: Validar comportamento da busca para registros inexistentes.

Passos:

1. Inserir um termo que não corresponda a nenhum pedido cadastrado.
2. Observar a resposta da tabela.

— Resultado Esperado:

Mensagem de “Nenhum registro encontrado” é exibida, indicando ausência de resultados.

Automação Proposta (Cypress)

Stack sugerida:

- `Cypress` para testes E2E;
 - `Node.js` + `npm` para execução do ambiente.
-

```
// Exemplo - E2E
// cypress/e2e/crud_pedidos.cy.js
/// <reference types="cypress" />

describe('Mini Desafio - CRUD de Pedidos', () => {
  const pedido = {
    produto: 'Produto Teste',
    fornecedor: 'Fornecedor X',
    quantidade: 10,
    status: 'Pendente'
  }
}
```

```

beforeEach(() => {
  // Simula a página de pedidos
  cy.visit('/pedidos') // No demo, pode usar uma página local HTML ou mock
})

it('CT01 - Criar novo pedido', () => {
  cy.get('#btnAddPedido').click()
  cy.get('#produto').type(pedido.produto)
  cy.get('#fornecedor').type(pedido.fornecedor)
  cy.get('#quantidade').type(pedido.quantidade.toString())
  cy.get('#status').select(pedido.status)
  cy.get('#btnSubmit').click()

  // Valida se o pedido foi adicionado à tabela
  cy.contains('td', pedido.produto).should('exist')
  cy.contains('td', pedido.fornecedor).should('exist')
})

it('CT02 - Editar pedido existente', () => {
  cy.contains('td', pedido.produto)
    .parent('tr')
    .within(() => {
      cy.get('.btnEdit').click()
    })

  cy.get('#quantidade').clear().type('15')
  cy.get('#btnSubmit').click()

  cy.contains('td', '15').should('exist')
})

it('CT03 - Excluir pedido', () => {
  cy.contains('td', pedido.produto)
    .parent('tr')
    .within(() => {
      cy.get('.btnDelete').click()
    })
})

```

```
// Confirma exclusão se houver modal
cy.get('#confirmDelete').click({ force: true })

cy.contains('td', pedido.produto).should('not.exist')
})

it('CT04 - Buscar pedido inexistente', () => {
  cy.get('#searchInput').type('Pedido Inexistente')
  cy.get('#tablePedidos').should('contain.text', 'Nenhum registro encontra
do')
})
})
```

```
// Estrutura Node.js para rodar Cypress
// package.json
{
  "name": "mini-desafio-crud",
  "version": "1.0.0",
  "scripts": {
    "cypress:open": "cypress open",
    "cypress:run": "cypress run"
  },
  "devDependencies": {
    "cypress": "^12.0.0",
    "mochawesome": "^7.0.0"
  }
}
```

- **Create (Adicionar pedido)** → CT01
 - Simula preencher um formulário e adicionar um pedido à tabela.
 - Valida se o registro aparece corretamente.
- **Read (Visualizar pedidos)** → CT01 e CT04
 - CT01 verifica se o pedido criado aparece na tabela.
 - CT04 simula busca por pedidos inexistentes e valida o retorno "Nenhum registro encontrado".

- **Update (Editar pedido)** → CT02
 - Simula clicar em editar, alterar campos (ex.: quantidade) e salvar.
 - Valida se a alteração é refletida na tabela.
 - **Delete (Excluir pedido)** → CT03
 - Simula clicar em excluir e confirma remoção.
 - Valida se o registro não existe mais na tabela.
-

Conclusão

Este mini desafio permitiu aplicar **técnicas de QA, análise de requisitos e automação de testes E2E**, validando fluxos críticos de CRUD de pedidos de forma clara e estruturada.

A abordagem reflete meu perfil como QA: **curiosa, analítica e comprometida com a entrega de valor e qualidade.**

Prioridade do teste: garantir que o produto seja funcional, usável e mantenha a confiança do usuário final.