

Prova QA

- 1) Quais cenários de teste você especificaria para validar a funcionalidade do aplicativo bancário desenhado abaixo? Liste todas as premissas adotadas.

Banco XYZ

Transferência de valores entre correntistas

Saldo R\$ 200,00

Cliente destino

Valor

Data de efetivação

TransferirCancelar

Podemos começar adotando uma premissa de campo/valor para uma primeira validação.

Em seguida vou detalhar um pouco mais com cenários em Gherkin.

1. Cliente Destino:

- Deve ser um campo obrigatório.
- Deve aceitar somente caracteres válidos (letras e números).
- Deve corresponder a um cliente existente.

2. Valor:

- Deve ser um campo obrigatório.
- Deve aceitar apenas valores numéricos positivos.
- O valor não pode ser maior que o saldo disponível (R\$ 200,00).

3. Data de Efetivação:

- Deve ser um campo obrigatório.
- Deve aceitar somente datas válidas.
- A data não pode ser no passado.

4. Botões:

- O botão "Transferir" deve realizar a transferência se todos os campos estiverem preenchidos corretamente.
- O botão "Cancelar" deve limpar todos os campos e retornar ao estado inicial.

Cenários:

Funcionalidade: Validação de Campos Obrigatórios

Cenário: Cliente destino não preenchido

Dado que o campo "Cliente destino" está vazio

E o campo "Valor" está preenchido com "50"

E o campo "Data de efetivação" está preenchido com "2024-07-20"

Quando o usuário clica no botão "Transferir"

Então o sistema deve exibir uma mensagem de erro "Cliente destino é obrigatório"

Cenário: Valor não preenchido

Dado que o campo "Cliente destino" está preenchido com "Cliente123"

E o campo "Valor" está vazio

E o campo "Data de efetivação" está preenchido com "2024-07-20"

Quando o usuário clica no botão "Transferir"

Então o sistema deve exibir uma mensagem de erro "Valor é obrigatório"

Cenário: Data de efetivação não preenchida

Dado que o campo "Cliente destino" está preenchido com "Cliente123"

E o campo "Valor" está preenchido com "50"

E o campo "Data de efetivação" está vazio

Quando o usuário clica no botão "Transferir"

Então o sistema deve exibir uma mensagem de erro "Data de efetivação é obrigatória"

Funcionalidade: Validação de Dados

Cenário: Cliente destino inválido

Dado que o campo "Cliente destino" está preenchido com "!!@#"

E o campo "Valor" está preenchido com "50"

E o campo "Data de efetivação" está preenchido com "2024-07-20"

Quando o usuário clica no botão "Transferir"

Então o sistema deve exibir uma mensagem de erro "Cliente destino inválido"

Cenário: Valor inválido

Dado que o campo "Cliente destino" está preenchido com "Cliente123"

E o campo "Valor" está preenchido com "-50"

E o campo "Data de efetivação" está preenchido com "2024-07-20"

Quando o usuário clica no botão "Transferir"

Então o sistema deve exibir uma mensagem de erro "Valor inválido"

Cenário: Data de efetivação inválida

Dado que o campo "Cliente destino" está preenchido com "Cliente123"

E o campo "Valor" está preenchido com "50"

E o campo "Data de efetivação" está preenchido com "32/13/2024"

Quando o usuário clica no botão "Transferir"

Então o sistema deve exibir uma mensagem de erro "Data de efetivação inválida"

Cenário: Data de efetivação no passado

Dado que o campo "Cliente destino" está preenchido com "Cliente123"

E o campo "Valor" está preenchido com "50"

E o campo "Data de efetivação" está preenchido com "2023-07-01"

Quando o usuário clica no botão "Transferir"

Então o sistema deve exibir uma mensagem de erro "Data de efetivação não pode ser no passado"

Funcionalidade: Limite de Saldo

Cenário: Valor maior que o saldo disponível

Dado que o campo "Cliente destino" está preenchido com "Cliente123"

E o campo "Valor" está preenchido com "250"

E o campo "Data de efetivação" está preenchido com "2024-07-20"

Quando o usuário clica no botão "Transferir"

Então o sistema deve exibir uma mensagem de erro "Saldo insuficiente"

Funcionalidade: Funcionalidade dos Botões

Cenário: Transferir com todos os campos preenchidos corretamente

Dado que o campo "Cliente destino" está preenchido com "Cliente123"
E o campo "Valor" está preenchido com "50"
E o campo "Data de efetivação" está preenchido com "2024-07-20"
Quando o usuário clica no botão "Transferir"

Então a transferência deve ser realizada com sucesso

Cenário: Cancelar transferência

Dado que o campo "Cliente destino" está preenchido com "Cliente123"
E o campo "Valor" está preenchido com "50"
E o campo "Data de efetivação" está preenchido com "2024-07-20"
Quando o usuário clica no botão "Cancelar"
Então todos os campos devem ser limpos e o sistema deve retornar ao estado inicial

Funcionalidade: Cenários Adicionais

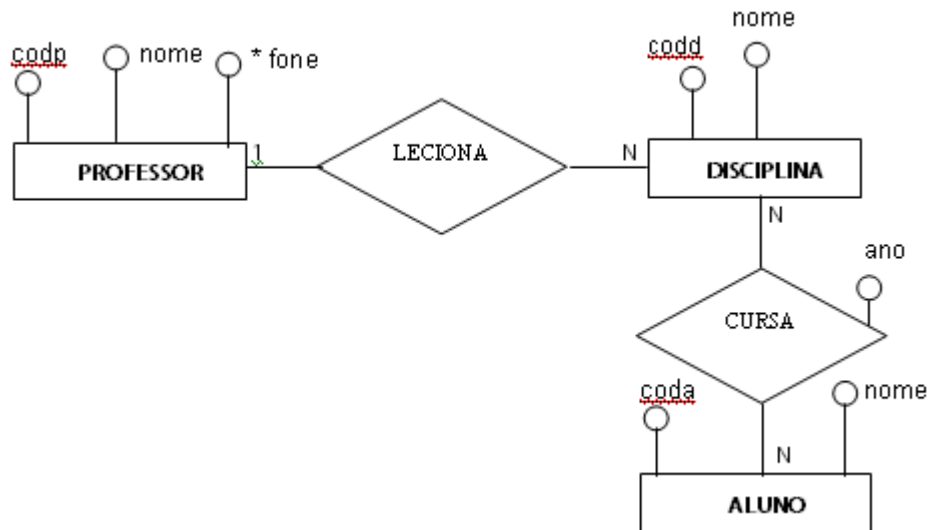
Cenário: Transferência com data de efetivação no mesmo dia

Dado que o campo "Cliente destino" está preenchido com "Cliente123"
E o campo "Valor" está preenchido com "50"
E o campo "Data de efetivação" está preenchido com a data atual
Quando o usuário clica no botão "Transferir"
Então a transferência deve ser realizada com sucesso

Cenário: Transferência com data de efetivação futura

Dado que o campo "Cliente destino" está preenchido com "Cliente123"
E o campo "Valor" está preenchido com "50"
E o campo "Data de efetivação" está preenchido com "2024-12-31"
Quando o usuário clica no botão "Transferir"
Então a transferência deve ser realizada com sucesso

2) Considere o esquema Entidade / Relacionamento abaixo:



- Escreva a consulta SQL para listar o nome de todos os alunos

```
SELECT ALUNO.nome
FROM ALUNO
JOIN CURSA ON ALUNO.coda = CURSA.coda
JOIN DISCIPLINA ON CURSA.codd = DISCIPLINA.codd
JOIN LECIONA ON DISCIPLINA.codd = LECIONA.codd
JOIN PROFESSOR ON LECIONA.codp = PROFESSOR.codp
WHERE DISCIPLINA.nome = 'Cálculo'
AND PROFESSOR.nome = 'João';
```

- matriculados na disciplina de Cálculo do professor João.

```
SELECT ALUNO.nome
FROM ALUNO
JOIN CURSA ON ALUNO.coda = CURSA.coda
JOIN DISCIPLINA ON CURSA.codd = DISCIPLINA.codd
JOIN LECIONA ON DISCIPLINA.codd = LECIONA.codd
JOIN PROFESSOR ON LECIONA.codp = PROFESSOR.codp
WHERE DISCIPLINA.nome = 'Cálculo'
AND PROFESSOR.nome = 'João';
```

- Escreva a consulta SQL para exibir a quantidade de alunos por disciplinas.

```
SELECT DISCIPLINA.nome, COUNT(CURSA.coda) AS quantidade_alunos
FROM DISCIPLINA
JOIN CURSA ON DISCIPLINA.codd = CURSA.codd
GROUP BY DISCIPLINA.nome;
```

- Escreva a consulta SQL para listar as disciplinas que todos os professores lecionam.

```
SELECT DISCIPLINA.nome
FROM DISCIPLINA
JOIN LECIONA ON DISCIPLINA.codd = LECIONA.codd
GROUP BY DISCIPLINA.nome
HAVING COUNT(LECIONA.codp) = (SELECT COUNT(*) FROM PROFESSOR);
```

- Escreva a consulta SQL que exibe o total de professores.

```
SELECT COUNT(*) AS total_professores
FROM PROFESSOR;
```

- Escreva a consulta SQL para listar todos os alunos que cursaram alguma disciplina do ano 2000 até 2020.

```
SELECT DISTINCT ALUNO.nome
FROM ALUNO
JOIN CURSA ON ALUNO.coda = CURSA.coda
WHERE CURSA.ano BETWEEN 2000 AND 2020;
```

3) Para validarmos a sua técnica de automação de testes, vamos utilizar uma api de desenvolvimento referente a plataforma Trello.

Caso não conheça o Trello, acessar: www.trello.com

Link da api para desenvolvimento:

<https://developers.trello.com/docs/api-introduction>

Com o end-point acima, crie testes automatizados para as seguintes ações:

- Cadastrar um board
- Cadastrar um card
- Excluir um card
- Excluir um board

Obs: Utilizar o Cypress como ferramenta de automação.

Para a automação de testes realizada, ilustre através de um mapa mental, a sua estratégia de testes para construção do seu projeto.

Após a conclusão do teste, publique no github, todo material produzido, e não esqueça de nos disponibilizar o link da sua avaliação.

LINK DO GITHUB COM O CÓDIGO DE AUTOMAÇÃO DA API TRELLO:

https://github.com/danilolattera/trello_testepratico

Guia utilizando MAPA MENTAL:

