**Prova QA**

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

Texto

Descrição gerada automaticamente

Resposta:

Os cenários que eu utilizaria para valida a funcionalidade da tela informada seria:  
- Validação do campo nome do banco

1. Ortografia.

2. Posicionamento na tela.  
3. Responsividade.

- Validação do campo “Saldo Usuário”

1. Valor apresentado condizente com retorno da api.

2. Posicionamento e cor se está de acordo com documentação).

- Validação do campo “Cliente Destino”

1. Validação de valores válidos de acordo com documentação (nome, conta, documento, e-mail).

2. Validar não preenchimento ao tentar fazer transferência.

3. Validar entradas inválidas (aquelas fora do escopo da documentação, como caracteres especiais ou inválidos), informando por alerta no campo sobre a invalidade.

- Validação do campo “Valor”

1. Validar se campo aceita apenas valores numéricos.

2. Validar se exibe uma máscara padronizada para apresentação do valor (ex. R$150,00).

3. Validar se o sistema aceita apenas transferências dentro dos limites do usuário (valores não nulos e negativos até o saldo disponível do usuário).

4. Validar comportamentos não esperado da aplicação como campo vazio.

5. Validar informação de alerta no campo sobre a invalidade de preenchimento do campo.

- Validação do campo “Data Efetivação”

1. Validar se o campo aceita valores apenas no formato data.

2. Validar se exibe uma máscara padronizada para apresentação da data (ex: 01/01/2025).

3. Validar comportamentos não esperado da aplicação como campo vazio.

4. Validar a não permissão de datas passadas e valores futuros longe do permitido pela documentação.

5. Validar informação de alerta no campo sobre a invalidade de preenchimento do campo.

- Validação do botão “Transferir”

1. Validar se ao informar os valores permitidos nos campos da tela, o sistema confirma a transferência após apertar o botão informando ao usuário.

2. Validar se ao informar os valores não permitidos nos campos da tela, o sistema apresenta uma mensagem de erro após apertar o botão, informando o motivo exato do erro (ou dos erros).

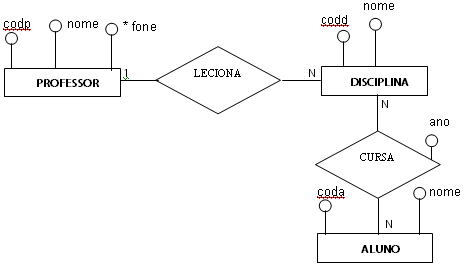
3. Validar a atualização do saldo do usuário após a transação.

4. Validar informação de alerta no campo sobre a invalidade de preenchimento do campo.

5. Validar habilitação de campo após preenchimento correto dos campos para a transferência

- Validação do botão “Cancelar”

1. Validar limpeza dos campos e voltar ao estado inicial da tela.

1. Considere o esquema Entidade / Relacionamento abaixo:

* Escreva a consulta SQL para listar o nome de todos os alunos matriculados na disciplina de Cálculo do professor João.
* Escreva a consulta SQL para exibir a quantidade de alunos por disciplinas.
* Escreva a consulta SQL para listar as disciplinas que todos os professores lecionam.
* Escreva a consulta SQL que exibe o total de professores.
* Escreva a consulta SQL para listar todos os alunos que cursaram alguma disciplina do ano 2000 até 2020.

1.listar o nome de todos os alunos matriculados na disciplina de Cálculo do professor João.  
  
SELECT a.nome

FROM Aluno a

JOIN CURSA c ON a.coda = c.coda

JOIN DISCIPLINA d ON c.codd = d.codd

JOIN LECIONA l ON d.codd = l.codd

JOIN PROFESSOR p ON l.codp = p.codp

WHERE d.nome = 'Calculo' AND p.nome = 'João'

2.exibir a quantidade de alunos por disciplinas.

SELECT d.nome AS disciplina\_nome, COUNT(c.coda) AS quantidade\_alunos

FROM DISCIPLINA d

JOIN CURSA c ON d.codd = c.codd

GROUP BY d.nome;

3.listar as disciplinas que todos os professores lecionam.

SELECT d.nome

FROM DISCIPLINA d

JOIN LECIONA l ON d.codd = l.codd

JOIN PROFESSOR p ON l.codp = p.codp

GROUP BY d.codd, d.nome

HAVING COUNT(DISTINCT p.codp) = (SELECT COUNT(\*) FROM PROFESSOR);

4.exibe o total de professores.

SELECT COUNT(p.codep) AS total\_professores  
FROM PROFESSOR p;

5.listar todos os alunos que cursaram alguma disciplina do ano 2000 até 2020.

SELECT DISTINCT a.nome  
FROM ALUNO a  
JOIN CURSA c ON a.coda = c.coda -- Relaciona ALUNO e CURSA  
WHERE c.ano BETWEEN 2000 AND 2020;

1. **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 mapa mental:  
https://miro.com/app/board/uXjVLs968H0=/?share\_link\_id=55009597042