

Avaliação – Área de Qualidade e Testes

Nome: Maria Angélica Silva Lima

**Questão 1** - Com base na sua experiência profissional, por que o teste de software é necessário numa instituição Financeira?

**Resposta:** Um instituição financeira lida com dados sensíveis, como, informações pessoais e transações monetárias. O teste de software é importante para proteger o negócio, o cliente e a reputação da instituição, assim assegurando segurança, confiabilidade, estabilidade e qualidade em cada entrega.

**Questão 2** - Quais defeitos podem ser identificados na tela abaixo?

Funcionalidade: Manter Ator

Busca

Chave

CPF

Valor

22.815.761/0001-34

Nome: ID: 0-121

Atributos Localização Campanhas Métricas Cupons Fidelização Pendências Histórico

Numero do Cartão

5449 0939 7907 8170

CPF\*

Sexo\*

Masculino

Name: \*

Pablo Henrique de Oliveira Neto

Data de Nascimento\*

11072001

senha\*

Amerlinda1056\*\*

RG

2589500

Logradouro

SWCSS 1500

numero

Bairro:

Brasília

Versão - 3.21.0 | admin | 08/2018 18:12:54PM

Server: ON

**Resposta:**

ID	DESCRIÇÃO	ESPERADO
01	O campo “Número do Cartão” está exibindo o número completo do cartão (ex: 5449 0939 7907 8170), o que viola práticas de segurança e LGPD.	maskar parte do número (ex: **** * 8170).
02	O campo “Senha” está visível em texto plano (“Amerlinda1056**”), o que é uma falha grave de segurança.	campo do tipo password, com ocultação do valor.

03	O campo RG sem máscara (2589500)	maskarar parte do número (ex: *****00)
04	Data de nascimento aparece como 11072001 sem formatação.	11/07/2001
05	Campo “número” não segue boas práticas de rotulagem	deveria iniciar com letra maiúscula e acento “Número”
06	No filtro de busca à esquerda, o campo “Valor” está preenchido com 22.815.761/0001-34, que é formato de CNPJ, mas o filtro selecionado é CPF.	o sistema deveria validar o tipo de dado conforme o filtro selecionado.
07	O campo “Name” está em inglês, enquanto o restante da interface está em português	padronizar o idioma (“Nome”).
08	O cabeçalho mostra o rótulo “PRODUCAO”, indicando que a tela pode estar sendo exibida em ambiente produtivo, expondo dados reais.	Grave falha de controle de acesso e ambiente.
09	Campos obrigatórios (*) não têm validação visual clara além do asterisco.	
10	Ausência de feedback visual ou botões de ação claros para “Salvar”, “Editar” ou “Excluir”.	
11	O layout parece desorganizado (campos desalinhados e espaçamento irregular).	

Cadastro de Pessoa

Dados da pessoa

Financeiro

Histórico de compras

Id:

6

Cadastrado em:

01/10/2019 10:10:05

Alterado em:

05/10/2019 09:30:10

Criado por:

Matheus Agnes Dias

Data de ativação:

02/10/2019

Data de desativação:

Dados gerais

Endereços

Contatos

Dados gerais da pessoa

Endereço principal

Nome:

Matheus

Email:

matheus@adianti.com.br

Observação:

Documento:

111.111.111-11

Telefone:

(51) 9 9999-9999

Estado:

Rio Grande do Sul

Cidade:

Lajeado

Bairro:

Rua:

Número:

Compl.:

CEP:

Título do caso de teste	Pré-condições	Etapas para reprodução	Resultado esperado	Tags
Criar pessoa com dados mínimos válidos	Usuário autenticado com permissão de cadastro	1. Abrir 'Cadastro de Pessoa'. 2. Preencher Nome. 3. Salvar.	Registro criado, ID gerado e mensagem de sucesso.	Funcional
Campos obrigatórios sinalizados	Tela de novo cadastro aberta	1. Não preencher Nome. 2. Salvar.	Erro informando campo obrigatório.	Negativa, Validação
Validação de formato de CPF	-	1. Informar CPF inválido. 2. Salvar.	Erro de formato ou CPF inválido.	Validação, Negativa
Validação de e-mail	-	1. Preencher e-mail inválido. 2. Salvar.	Erro de e-mail inválido.	Validação, Negativa
Validação de telefone com máscara	-	1. Preencher telefone sem máscara. 2. Salvar.	Erro ou autocorreção de formato.	Validação

Data de desativação anterior à ativação	-	1. Inserir data de ativação > data de desativação. 2. Salvar.	Erro de inconsistência temporal.	Negativa, Regra de negócio
Edição preserva histórico	Pessoa já cadastrada	1. Alterar telefone. 2. Salvar.	Campos de auditoria preservados, 'Alterado em' atualizado.	Funcional
Estado/Cidade dependentes	Listas disponíveis	1. Selecionar Estado. 2. Verificar Cidade.	Cidade lista apenas cidades do estado selecionado.	Funcional, Usabilidade
Validação de CEP	-	1. Informar CEP inválido. 2. Salvar.	Erro de formato de CEP.	Validação
Limite de tamanho no campo Nome	-	1. Digitar nome > 255 caracteres. 2. Salvar.	Erro ou bloqueio de tamanho.	Limite, Negativa

**Questão 4** - Utilizando as seguintes tabelas, escreva as cláusulas SQL solicitadas:

#### Tabela Livro

	codigo	titulo	lançamento	assunto	editora	preco
1	2	Programando na Linguagem	1997-01-10 00:00:00	P	1	35.00
2	1	Banco de Dados	1999-10-01 00:00:00	B	1	31.20
3	5	Banco de Dados SQL Server	NULL	D	1	60.00
4	6	Banco de Dados SQL Server	NULL	B	1	85.00
5	7	Linguagem Procedural	2004-10-09 00:00:00	D	1	60.00

#### Tabela Aluno

	matricula	nome	cpf	endereco	data_nascimento	nacionalidade
1	10000	Jose Bonifacio	10001	Rua Jose	2010-01-09 00:00:00	Brasileira
2	10001	Mane Gamincha	10002	Rua Lovan	2011-01-09 00:00:00	Italiana
3	10002	Antonio	10003	Estrada de Itapeperica	1998-02-01 00:00:00	Portuguesa

#### Tabela ControleAlunoLivro

	codigo_livro	matricula
1	2	10000
2	3	10002
3	4	10001
4	1	10000
5	5	10001
6	6	10002
7	7	10003

a) Descreva qual o resultado para a consulta realizada abaixo:

```
SELECT titulo, lancamento FROM livro v Where assunto = 1 and editora = 'P' ;
```

**Resposta:**

Nenhuma linha retornada.

na tabela **livro** da imagem **assunto** contém letras ('P', 'B', 'D') e **editora** contém números (1). A query está comparando **assunto = 1** nunca verdadeiro e **editora = 'P'** também nunca verdadeiro, portanto nenhuma linha atende aos dois critérios.

b) A cláusula WHERE é usada para adicionar condições de filtragem a uma consulta SQL, permitindo que você especifique critérios para selecionar apenas os registros desejados. Descreva a cláusula SQL que retorne a Consulta que retorne o título e o preço de todos os livros pertencentes ao aluno Jose Bonifácio:

**Resposta:**

```
SELECT l.titulo, l.preco  
FROM aluno a  
JOIN controlealunolivro c ON a.matricula = c.matricula  
JOIN livro l ON l.codigo = c.codigo_livro  
WHERE a.nome = 'Jose Bonifacio';
```

JOIN une as tabelas pelas chaves (matricula e codigo\_livro), WHERE a.nome = 'Jose Bonifacio' filtra apenas o aluno desejado, SELECT l.titulo, l.preco exibe as colunas solicitadas.

c) Descreva a cláusula sql que retorne todos os registros da tabela aluno.

**Resposta:**

```
SELET * FROM aluno;
```

SELECT \* seleciona todas as colunas da tabela, FROM aluno indica que os dados devem ser buscados da tabela chamada aluno.

**Questão 5** - Os aplicativos móveis são uma tendência atual e têm se tornado cada vez mais populares nos últimos anos. Com o avanço da tecnologia móvel e o aumento do uso de smartphones e tablets, quais tipos de teste podem ser

realizados para garantir a qualidade e segurança da entrega de um projeto de aplicativo móvel?

**Resposta:**

<b>Tipo de Teste</b>	<b>Objetivo</b>	<b>Exemplos</b>	<b>Categoria</b>
<b>Funcionais</b>	Verifica se as funcionalidades principais funcionam conforme esperado	Ex: login, cadastro, busca, checkout, push notifications	Funcional
<b>Compatibilidade/Fragmentação</b>	Assegura funcionamento em diferentes dispositivos, SOs e telas	Ex: Android/iOS, tamanhos de tela, versões de sistema	Compatibilidade
<b>Usabilidade e Acessibilidade</b>	Avalia a experiência do usuário e suporte a acessibilidade	Ex: TalkBack, VoiceOver, contraste, gestos	Usabilidade
<b>Performance</b>	Avalia tempos de resposta e uso de recursos	Ex: tempo de inicialização, FPS, consumo de bateria, memória	Performance
<b>Rede e Resiliência</b>	Testa comportamento em condições adversas de rede	Ex: modo offline, latência, perda de pacotes, reconexão	Rede
<b>Segurança</b>	Garante proteção de dados e resistência a ataques	Ex: criptografia, autenticação	Segurança
<b>Integração e Contratos</b>	Valida integração com APIs e serviços externos	Ex: versionamento de APIs e pagamentos	Integração
<b>Regressão e Automatizados</b>	Assegura que novas mudanças não quebrem funções existentes	Ex: smoke tests, regressão em	Manual e Automação
<b>Testes Visuais</b>	Verifica consistência e layout entre dispositivos	Ex: temas claro/escuro, rotação, snapshots visuais	UI/UX
<b>Loja e Compliance</b>	Confirma conformidade com políticas da App Store/Play Store	Ex: permissões, assinaturas, privacidade	Compliance
<b>Operação e Observabilidade</b>	Verifica monitoramento e logs	Ex: crash reporting, métricas de performance, rollout	Monitoramento

**Questão 6** - Ao realizar testes de API, é importante observar vários critérios de validação tais como a Cobertura, Entrada de Dados, Formato da Resposta, entre outros. A seguir responda as seguintes perguntas:

a) Encontre as possíveis divergências entre o resultado esperado constante no Swagger e a resposta da API.

Code

Description

202

Response Hello

Media type

application/json

Controls Accept header.

Example Value | Schema

ListaConsultaParticipanteResposta

paginacao

participantes

▼ string

Identificador único pra realizar a consulta dos dados sob demanda.

▼ [ConsultaParticipanteResposta

▼ integer(\$int32)

pattern: ^\d{10}\$

Identificador único do participante

▼ string

Nome do participante

▼ string

Nome de exibição do participante. Pode ser alterado e apelidado.

▼ string

Situação do participante:

• ATIVO - Participante está ativo para o PIX

• INATIVO - Participante está inativo para o PIX

Enum:

▼ [ ATIVO, INATIVO ]

▼ string

Tipo do participante:

• DIRETO - Participante com acesso direto ao SPI

• INDIRETO - Participante sem acesso direto ao SPI e utiliza um Participante Direto para a comunicação no PIX

▼ string(\$date)

Data da criação do participante

▼ [EventoParticipante > {...}]

GET

http://{{spi-participantes-host}}/participantes

Send

ParamsAuthHeaders (8)BodyScriptsSettingsCookies

Query Params

	Key	Value	Description	...	Bulk Edit
	Key	Value	Description		

Body

200 OK • 316 ms • 276.95 KB • Save Response

{}

JSON

PreviewVisualize

```
1 {
2   "paginacao": "1489",
3   "participantes": [
4     {
5       "_id": 16,
6       "ispb": "00000000",
7       "bolCooperativa": false,
8       "nome": "BANCO DO BRASIL S.A.",
9       "nomeExibicao": "BCO DO BRASIL S.A.",
10      "tipo": "DIRETO",
11      "situacao": "ATIVO",
12      "dataCriacao": "2020-11-03T08:55:40.954Z",
13      "historico": []
14    },
15  ]
16 }
```

Resposta:

**b)** Um endpoint de uma API REST retorna 404 quando você tenta buscar um recurso que sabe que existe. Explique **duas possíveis causas para isso**, considerando o uso do Postman.

**Resposta:**

**c)** Você está testando uma API de pagamentos. O fluxo é:

1. Criar pagamento (POST)
  2. Consultar pagamento (GET)
  3. Cancelar pagamento (DELETE)
- O ID da transação é gerado na etapa 1.

Como você estruturaria essa sequência no Postman de forma automatizada e reutilizável?

**Resposta:**

**d)** Crie uma collection para testar 5 endpoints da API, abaixo.

<https://petstore.swagger.io/>

Obs.: envie o arquivo da collection junto com as respostas deste questionário.