

Entrega: Enviar para o email [ledsouza@senacrs.com.br](mailto:ledsouza@senacrs.com.br) com seu nome e o assunto Revisão – Questões.

#### Pergunta 1: Conceitos Básicos de Teste de Software

- a) O que é teste de unidade e qual é o seu propósito no ciclo de vida do desenvolvimento de software?

Os testes de unidade são testes realizados em partes específicas do código individualmente, eles têm o propósito de avaliar o desempenho de cada parte, a fim de que elas funcionem adequadamente de acordo com os requisitos funcionais estabelecidos.

#### Pergunta 2: Estratégias de Teste

- a) Descreva a diferença entre teste de regressão e teste de integração.

A principal diferença entre eles é que o teste de regressão busca testar se novas funcionalidades incluídas no software afetaram de modo negativo as funcionalidades antigas, para assim evitar possíveis regressões futuras no código. Já na integração, vai ocorrer uma junção de algumas funcionalidades individuais, as unidades, e será testado como elas se interagem entre si, e assim monitorar e rastrear prováveis erros que possam vir a ocorrer.

- b) Quando você escolheria realizar um teste de desempenho em um aplicativo?

Possivelmente nas etapas finais, quando o sistema já estivesse pronto para o deploy. Já que assim seria possível testar como o sistema reagiria a uma série de fatores, como múltiplos acesso de usuários, alto volume de transferência de dados, etc. Desse modo, simulando como o aplicativo seria usado na vida real e com diversos acessos simultâneos.

- c) O que é teste de aceitação do usuário (UAT) e qual é seu papel no processo de teste?

Ela é a última etapa no processo de teste de software, por isso ela desempenha um papel importante, pois é nela que o público alvo e os usuários irão testar a aplicação e ver se ela atende aos requisitos estabelecidos anteriormente. Caso os usuários aprovem a aplicação, então ela ficará totalmente pronta para o deploy, caso contrário, se eles não aprovarem, será necessário rever algumas funcionalidades, ou ainda eventuais erros que possam surgir durante os testes finais.

#### Pergunta 3: Defeitos, erros e falhas

- a) O que é um defeito de software? É quando ocorre algum erro inesperado e indesejado no código. Também pode ser definido como bug.
- b) O que é erro? É quando o desenvolvedor comete um equívoco e realiza alguma funcionalidade errada no código.

c) O que é falha? **É quando o sistema não funciona do jeito esperado, resultando em um comportamento indesejado e fora dos requisitos esperados.**

#### **Pergunta 4: Casos de Teste**

a) O que são casos de teste e como eles são usados no processo de teste de software?

**Os casos de teste são documentos técnicos com informações sobre como deverá ser realizado o teste em determinada funcionalidade específica do código. Nele contém os objetivos e resultados esperados daquela funcionalidade específica.**

b) Explique a diferença entre plano de testes e caso de testes.

**O plano de teste é um documento que apresenta uma visão mais abrangente de como funcionará os testes. Ele leva em conta outros fatores além do teste em si. Ainda assim, ele cita determinados testes que deverão ser realizados, porém ele abrange outros pontos como quais dispositivos serão usados para o teste, número de testadores, cronograma de teste, entre outros. Já o caso de teste foca especificamente em como o teste vai se proceder.**

**Dentro do plano de teste contém todos os casos de teste que deverão ser realizados. E os casos de teste, são os testes propriamente ditos. Cada caso de teste visa testar uma funcionalidade específica.**

#### **Pergunta 5: Teste de Desempenho**

a) Explique o que é um teste de carga e como ele difere de um teste de estresse.

**O teste de carga irá verificar o desempenho do sistema quando submetido a um número esperado e definido de carga ou acessos. Já o teste de estresse vai ser com um número grande e maior de acessos e condições extremas, procurando levar o sistema ao limite.**

#### **Pergunta 6: Teste de Regressão**

a) O que é teste de regressão e por que é importante?

**Ele é um tipo de teste que visa testar se novas funcionalidades afetam as funcionalidades já existentes e como elas interagem entre si. Ele tem o objetivo de evitar eventuais regressões que possam vir a ocorrer no sistema por conta de funcionalidades novas. Ele visa garantir que as funcionalidades antigas permanecem intactas mesmo com novas funcionalidades surgindo no sistema.**

b) Descreva uma estratégia eficaz para realizar testes de regressão em um ambiente de desenvolvimento ágil.

### Pergunta 7: Teste Exploratório

a) Explique o conceito de teste exploratório e em que situações ele é mais apropriado.

Ele é um conceito aonde os testadores não seguem um script definido de testes. Eles possuem total liberdade para testarem o sistema sem necessidade de seguir uma sequência definida de testes. Isso permite que eles aprendam e se adaptem com o sistema em questão, permitindo uma abordagem mais profunda no sistema como um todo. Ele pode ser realizado em diferentes etapas e situações dentro do processo de teste. Geralmente quando o sistema já passou por testes mais específicos e já apresenta funcionalidades mais compactas e estáveis.

b) Quais são os benefícios e desafios associados ao teste exploratório?

Os benefícios dessa abordagem é que permite o testador ter mais independência e autonomia para realizar os testes, e isso leva a um teste mais profundo no sistema como um todo. O testador pode se adaptar e ajustar possíveis estratégias na hora do teste e testar diversas funcionalidades ao mesmo tempo. O desafio que isso acarreta é que por ser um teste mais flexível e abrangente, o testador pode ter problemas de adaptação com o sistema ou ainda seguir por caminhos mais generalizados e difíceis de encontrar eventuais problemas que poderiam ser descobertos com uma metodologia clara e definida de testes.

### Pergunta 8: Teste de Integração

a) Explique o propósito do teste de integração no contexto do desenvolvimento de software.

No contexto de desenvolvimento de software, o teste de integração seria uma junção de partes desenvolvidas tanto no Back-end, quanto no Front-end. Na parte de Back-end poderia ser a integração de uma API com um banco de dados, ou ainda uma resposta do servidor a uma requisição vinda do Front-end, um formulário vindo do Front-end e que será salvo em um banco de dados, etc. Seria essa integração entre ambas as partes no processo de desenvolvimento de software, tanto no Client-side, quanto no Server-side.

### Pergunta 9: Níveis de Teste

a) Quais são os níveis de teste? Explique brevemente sobre o que ocorre em cada fase.

1. **Teste de unidade:** Nessa fase são executados testes em unidades específicas do sistema individualmente. As funcionalidades são testadas individualmente e separadas das outras.
2. **Teste de Integração:** Nessa etapa ocorre a integração das funcionalidades testadas anteriormente. Elas são testadas em conjuntos umas com as outras, visando verificar se elas funcionam bem em grupo e não apresentam falhas.

3. **Teste de sistema:** Nessa etapa o sistema é testado como um todo. Todas as unidades são agrupadas e o sistema final é colocado em teste antes da entrega.
4. **Teste de Aceitação do Usuário:** É a etapa mais importante, pois é o usuário final que irá testar e aprovar o sistema completo, validando e verificando se ele atende aos requisitos solicitados no início do projeto.

### **Elaboração de Plano de Testes para Calculadora (10 pontos)**

Imagine que você foi designado para criar um plano de testes para uma aplicação que é uma calculadora simples. O objetivo é garantir que a calculadora execute suas funções básicas de forma precisa e que seja uma experiência de usuário livre de problemas. Elabore o plano de testes seguindo o modelo passado em aula.

#### **Plano de Teste para uma calculadora mobile que execute as quatro operações básicas**

##### **Objetivo:**

Verificar a usabilidade de uma calculadora básica que execute as quatro operações básicas: adição, subtração, multiplicação e divisão. Identificar e corrigir eventuais defeitos no sistema de cálculo e usabilidade.

##### **Escopo:**

O sistema deverá testar as quatro operações básicas de soma, adição, subtração e divisão, de modo que o resultado obtido em cada operação esteja correto e atenda as expectativas.

##### **Estratégia de Teste:**

Realização de testes unitários em cada uma das funcionalidades mencionadas, testes de integração entre ambas e testes de sistema.

##### **Cenário de testes:**

Função somar: Verificar se a função somar retorna o número esperado após a entrada de dois números selecionados na tela.

Função subtrair: Verificar se a função subtrair retorna o número esperado após a entrada de dois números selecionados na tela.

Função dividir: Verificar se a função dividir retorna o número esperado após a entrada de dois números selecionados na tela.

Função multiplicar: Verificar se a função multiplicar retorna o número esperado após a entrada de dois números selecionados na tela.

Exibição das operações e números na tela: Verificar se após a entrada do usuário na calculadora os números e as operações são exibidos na tela.

Seleção de números e operações: Testar se é possível a seleção dos números e a operação. Após a seleção de um número, o mesmo deve ser exibido na tela. Assim deve se repetir com o símbolo da operação e o segundo número.

Exibição do resultado na tela: Verificar se após selecionado os dois números e a operação, o resultado é exibido na tela corretamente.

**Recursos:**

Equipe composta por dois testadores. Ambiente de teste com acesso a calculadora em tempo real.

**Ambiente de Teste:**

Sistemas operacionais: Windows, macOS e Linux.

Navegadores: Google Chrome, Mozilla Firefox, Microsoft Edge, Opera e Brave.

Dispositivos: Computadores desktop com acesso ao código fonte e celulares para teste final.

**Cronograma de teste:**

Início dos testes: 05.12.2023

Término dos testes: 15.12.2023

**Critérios de aceitação:**

98% dos casos de teste aprovados. É necessário que todas as funções de cálculos estejam funcionando adequadamente. Interface gráfica deve ser de fácil navegação e com boa usabilidade. Problemas específicos na interface, desde que não comprometam a funcionalidade principal de cálculo serão permitidos.

**Riscos e Mitigações:**

Risco: Falhas em alguma funcionalidade crítica que atrase o desenvolvimento e a entrega final do aplicativo.

Mitigações: Comunicação regular entre a equipe de teste e a equipe de desenvolvimento. Comunicar qualquer eventual falha e procurar resolver o mais breve possível. Comunicação deve ser clara e concisa.

**Critérios de saída:**

Todos os testes de caso foram executados e documentados.

- Relatório final informando os resultados obtidos e qualquer eventual problema ocorrido no teste do sistema.