

1 - O que é teste de software?

✓ Resposta Sugerida:

"Teste de software é o processo de verificar se um sistema ou aplicação funciona conforme o esperado, atendendo aos requisitos definidos. O principal objetivo dos testes é encontrar defeitos (bugs) e garantir a qualidade do produto antes que ele seja entregue ao usuário final. Além disso, os testes ajudam a melhorar a usabilidade, desempenho e segurança do software."

2 - Qual a diferença entre teste funcional e não funcional?

✓ Resposta Sugerida:

"**Testes funcionais** verificam se o software faz o *que deveria fazer*, ou seja, se as funcionalidades estão funcionando conforme os requisitos. Eles focam no comportamento do sistema, como login, cadastro, envio de e-mails, entre outros."

"**Testes não funcionais**, por outro lado, avaliam *como* o sistema se comporta. Eles testam características como desempenho, segurança, usabilidade, escalabilidade, entre outros. Um exemplo seria testar quantos usuários simultâneos o sistema suporta sem travar."

📌 Exemplo para reforçar:

- Teste funcional: Verificar se o botão "Enviar" envia corretamente um formulário.
- Teste não funcional: Verificar se o sistema responde em menos de 2 segundos ao clicar no botão "Enviar".

3 - O que é um caso de teste (test case) ?

✓ Resposta Sugerida:

"Um **caso de teste** (ou *test case*) é um conjunto de condições ou etapas criadas para verificar se uma funcionalidade do sistema está funcionando corretamente. Ele descreve o que deve ser testado, os dados de entrada, os passos a seguir, o resultado esperado e o resultado real obtido. O objetivo é garantir que o software esteja funcionando conforme os requisitos."



Exemplo simples:

Por exemplo, para testar o campo de login:

- **"Título: Verificar login com credenciais válidas"**
- **"Pré-condição: O usuário já está cadastrado no sistema"**
- **"Passos:**
 1. Acessar a página de login
 2. Digitar e-mail e senha válidos
 3. Clicar em "Entrar"
- **"Resultado esperado: Usuário é redirecionado para a página inicial"**
- **"Resultado obtido: (preenchido após o teste)"**

4 - O que é um bug? E como você o reportaria?



Resposta Sugerida:

"Bug é um erro, falha ou comportamento inesperado no software que faz com que ele funcione de forma diferente do que foi especificado nos requisitos. Pode afetar a funcionalidade, desempenho, aparência ou segurança do sistema."

"Para reportar um bug corretamente, é importante ser claro, objetivo e detalhado. Eu costumo seguir uma estrutura com título, descrição, passos para reproduzir o problema, resultado esperado, resultado obtido, evidências (como prints ou vídeos), e a severidade do bug. Normalmente, isso é feito em ferramentas como Jira, Trello ou Azure DevOps."



Exemplo prático de relato de bug:

- **Título:** Erro ao salvar formulário de cadastro
- **Descrição:** O botão "Salvar" não está funcionando ao preencher todos os campos obrigatórios.
- **Passos para reproduzir:**
 1. Acesse a tela de cadastro
 2. Preencha todos os campos obrigatórios
 3. Clique em "Salvar"
- **Resultado esperado:** O usuário é redirecionado para a tela de confirmação.
- **Resultado obtido:** Nada acontece; o botão fica inativo.
- **Evidência:** Print e vídeo do erro
- **Severidade:** Alta (impede a conclusão do fluxo principal)

5 - Qual a diferença entre verificação e validação?

✓ Resposta Sugerida:

"Verificação é o processo de garantir que o software está sendo desenvolvido corretamente, ou seja, que ele atende às especificações e requisitos durante as fases de desenvolvimento. Normalmente envolve revisões, inspeções e testes estáticos (sem executar o código)."

"Validação é garantir que o software faz o que o usuário realmente precisa, ou seja, se atende ao propósito final. Envolve a execução do sistema e testes dinâmicos para garantir que o comportamento esteja correto na prática."

📌 Resumindo com uma frase:

Verificação: "Estamos construindo o software certo?"

Validação: "Estamos construindo o software de forma certa?"

🔧 Exemplo prático:

- **Verificação:** Revisar os documentos de requisitos e o código para ver se estão corretos.
- **Validação:** Executar testes no sistema para ver se ele realmente resolve o problema do usuário.

6 - Você sabe o que é um ciclo de teste ?

✓ Resposta Sugerida:

"Sim, o **ciclo de teste** é o conjunto de etapas que o time de QA segue para planejar, executar e encerrar as atividades de teste dentro de um projeto ou sprint. Ele garante que os testes sejam feitos de forma organizada, controlada e com base em critérios claros."

💬 Etapas comuns de um ciclo de teste:

1. **Planejamento** – Definir escopo, objetivos, recursos, riscos e cronograma dos testes.
2. **Design** – Criar os casos de teste com base nos requisitos ou histórias de usuário.
3. **Preparação do ambiente** – Configurar o ambiente necessário para rodar os testes.
4. **Execução dos testes** – Rodar os testes manuais ou automatizados e registrar os resultados.
5. **Registro de defeitos** – Reportar e acompanhar os bugs encontrados.
6. **Re-teste e regressão** – Verificar se os bugs foram corrigidos e se novas mudanças não afetaram outras partes do sistema.
7. **Encerramento** – Gerar relatórios e lições aprendidas; avaliar a qualidade entregue.

Exemplo:

"Por exemplo, se vou testar uma nova funcionalidade de cadastro, eu sigo esse ciclo desde o planejamento dos testes, criação dos casos, execução e registro dos resultados até encerrar com um relatório ou checklist."

7 - O que são testes manuais e testes automatizados?

Resposta Sugerida:

"Testes manuais são aqueles realizados diretamente por uma pessoa, sem uso de scripts de automação. O testador segue os casos de teste passo a passo e observa os resultados. É ideal para testes exploratórios, testes de interface ou validação rápida de funcionalidades."

"Testes automatizados são executados por scripts ou ferramentas que simulam as ações do usuário. Eles são usados para testar repetidamente funções do sistema, com mais rapidez e precisão. São ótimos para testes de regressão, smoke tests e cenários de alto volume."

Comparação rápida:

Tipo de teste	Vantagens	Desvantagens
Manual	Flexível, bom para testes únicos	Demorado, sujeito a erro humano
Automatizado	Rápido, reutilizável, confiável	Requer tempo para desenvolver scripts

Exemplos:

- **Teste manual:** Verificar se o botão "Comprar" leva corretamente ao carrinho.
- **Teste automatizado:** Rodar todos os testes de login automaticamente a cada nova versão do sistema.

8 - Qual a diferença entre teste de regressão e teste de reteste ?

Resposta Sugerida:

"Teste de reteste é quando verificamos se um bug que foi corrigido está realmente funcionando agora. Ou seja, testamos novamente exatamente aquele cenário onde o problema aconteceu."

"Teste de regressão é quando testamos outras partes do sistema para garantir que a nova correção ou funcionalidade não tenha quebrado nada que já funcionava antes. Ele garante que o sistema continua estável após mudanças."

Exemplo prático:

Imagine que um botão "Salvar" não funcionava e foi corrigido:

- **Reteste:** Você testa novamente aquele botão com os mesmos dados que causavam o erro.
- **Regressão:** Você testa outras áreas que usam a função "Salvar" (ex: salvar perfil, salvar produto) para garantir que nada mais foi afetado pela correção.

Dica para entrevista:

"O reteste é focado no defeito que foi corrigido. A regressão é mais ampla e verifica o impacto da mudança em outras funcionalidades."

9 - Conhece alguma ferramenta de gerenciamento de teste ?

Resposta Sugerida:

"Sim, eu conheço algumas ferramentas de gerenciamento de testes que são bastante usadas no mercado, como:

- "Jira: Uma das ferramentas mais populares, usada para gerenciamento de projetos ágeis. É possível criar, rastrear e organizar casos de teste, bugs e histórias de usuário."
- "TestRail: Uma ferramenta específica para gerenciamento de testes, onde você pode criar planos de teste, gerenciar casos de teste, executar e gerar relatórios detalhados."
- "Zephyr: Outra ferramenta integrada ao Jira, que permite o gerenciamento de casos de teste e execução de testes diretamente no Jira."
- "Xray: Uma ferramenta de gerenciamento de testes que também integra com o Jira, muito utilizada para testes manuais e automatizados."



Dica de Como Responder:

Se você já usou alguma dessas ferramentas, vale a pena citar a sua experiência prática, como:

"No meu último estágio, eu utilizei o **Jira** para criar e acompanhar casos de teste, além de reportar bugs e acompanhar o progresso do time. Também tive experiência com o **TestRail**, onde pude criar casos de teste, executar e gerar relatórios para acompanhamento do ciclo de testes."

Se você nunca usou uma ferramenta específica, você pode mencionar que **está familiarizado** com a ideia de gerenciamento de testes, ou até mesmo demonstrar interesse em aprender novas ferramentas, como:

"Embora eu ainda não tenha trabalhado diretamente com essas ferramentas, estou bastante familiarizado com o conceito de gerenciamento de testes e estou empolgado para aprender e aplicar essas ferramentas no dia a dia."

10 - Você já usou alguma ferramenta de automação ? como Selenium,cypress,Postman)

✓ Resposta Sugerida:

"Sim, eu já tive **experiência básica** com ferramentas de automação de testes, como **Selenium** e **Postman**. Eu usei o **Selenium** para automatizar testes de navegação em navegadores, como preencher formulários e verificar links. No **Postman**, eu criei e executei testes de APIs, garantindo que as respostas da API estivessem corretas, como status codes e o corpo da resposta."

"Ainda estou me aprofundando mais em automação, mas tenho uma boa base e estou sempre buscando melhorar minhas habilidades, principalmente em **Cypress** e outras ferramentas que são muito populares no mercado."

🔧 Exemplos de como você pode detalhar sua experiência:

1. Selenium:

"Durante o meu estágio, usei o Selenium para escrever scripts que automatizavam a validação de funcionalidades em uma aplicação web. Por exemplo, criei scripts para verificar o fluxo de login e logout em diferentes navegadores."

2. Postman:

"No Postman, configurei e executei testes de API para garantir que a resposta da API estivesse conforme o esperado, validando parâmetros como tempo de resposta e dados retornados."

3. Cypress (se tiver alguma experiência):

"Estou aprendendo o Cypress para testar funcionalidades em tempo real, especialmente porque ele oferece um framework muito mais rápido e com uma boa integração para teste de front-end."

💡 Caso você não tenha experiência prática com essas ferramentas, você pode responder da seguinte forma:

"Embora eu ainda não tenha tido a oportunidade de usar essas ferramentas de automação em projetos anteriores, tenho uma boa compreensão do conceito de automação de testes e estou motivado para aprender mais, especialmente com ferramentas como Selenium e Cypress, que são bastante utilizadas no mercado."
