**1. Qual é o principal objetivo da análise de risco no processo de teste?**

O principal objetivo é identificar e priorizar áreas críticas do sistema que apresentam maior probabilidade de falha ou maior impacto no negócio. Isso ajuda a direcionar o esforço de teste para onde ele gera mais valor.

**2. O que é um teste baseado em requisitos? Cite exemplos.**

É o teste planejado a partir dos requisitos documentados do Sistema (critérios de aceitação), garantindo que cada requisito seja validado.  
Exemplo: Requisito: 'O sistema deve permitir login com e-mail e senha.' Teste: Validar login com credenciais válidas e inválidas.

**3. O que é um teste unitário e quem os realiza? Cite exemplos.**

Teste que valida pequenas partes isoladas do código (funções/métodos). Geralmente escrito e mantido por desenvolvedores.  
Exemplo: Função que soma dois números = validar se retorna o resultado correto.

**4. O que é um teste de Integração e quem os realiza? Cite exemplos.**

Valida a comunicação entre diferentes módulos/componentes de um sistema. Pode ser feito tanto por desenvolvedores quanto QAs.  
Exemplo: Testar se o módulo de login consegue acessar corretamente o banco de dados de usuários.

**5. Você precisa realizar testes de Performance em uma aplicação web que calcula o tempo das rotas de entrega para um serviço de entregas por motoboys que realiza 20 milhões de entregas por dia. Quais os conceitos de Testes de Performance você consideraria para esta situação? E quais tipos de Testes de Performance você realizaria nesta situação?**

Conceitos considerados: carga, estresse, picos, escalabilidade, tempo de resposta, throughput, uso de recursos.  
Tipos aplicados: Teste de Carga, Teste de Stress, Teste de Picos e Teste de Endurance.

**6. O que é CI e CD? Cite exemplos.**

CI (Continuous Integration): Integração contínua de código em repositório central, com execução automática de builds e testes.  
CD (Continuous Delivery/Deployment): Automação da entrega em produção após os testes.  
Exemplo: Usar GitLab CI ou Jenkins para rodar testes automatizados a cada merge.

**7. Qual tipo de teste foca no comportamento interno do software?**

Teste “Caixa Branca”.

**8. Qual tipo de teste foca no comportamento externo do software?**

Teste “Caixa Preta”.

**9. Cite a importância dos testes Exploratórios.**

Os testes exploratórios visam utilizar/explorar o sistema como se fossem feitos por um usuário em uma situação real de uso. Eles permitem descobrir falhas que não estavam previstas nos roteiros de teste. São úteis para encontrar bugs rapidamente em áreas críticas.

**10. Quais são os benefícios de termos uma automação de testes funcionais?**

Redução de esforço manual repetitivo, maior cobertura de regressão, execução rápida e contínua, integração em pipelines CI/CD.

**11. Em qual etapa do projeto o QA deve ser envolvido?**

Desde o início do ciclo de desenvolvimento, participando da análise de requisitos, refinamentos e planejamento. Isso evita retrabalho e garante qualidade desde a concepção.

**12. Como manter um trabalho ágil como um todo?**

Comunicação constante entre times, entregas incrementais e frequentes, feedback rápido, testes contínuos (shift-left testing) e transparência nas tarefas.

**13. Na sua visão, a Qualidade do Software é uma responsabilidade apenas do QA?**

A qualidade é responsabilidade de todo o time (desenvolvedores, POs, QAs, DevOps). O QA tem um papel muito importante, mas todos devem contribuir para entregar software com qualidade.

**14. Indique uma ou mais formas de executar testes não funcionais, e explique a importância dos mesmos.**

Performance (JMeter, Gatling), Segurança (OWASP ZAP, BurpSuite), Usabilidade (testes com usuários).  
Importância: validam aspectos que vão além da funcionalidade, garantindo escalabilidade, segurança e experiência do usuário.

**15. Indique maneiras de dar visibilidade das tarefas do QA para o time como um todo.**

Uso de ferramentas de gestão (Jira, Azure DevOps), Dashboards com métricas (testes executados, falhas, cobertura), Daily meetings e relatórios curtos de progresso.

**16. De um modo geral, explique as vantagens dos testes manuais e dos automatizados...**

Manuais: exploratórios, usabilidade, testes ad-hoc. Ex.: avaliar layout responsivo.  
Automatizados: regressão, cenários repetitivos e volumosos.   
Ex.: rodar 500 cenários de login em diferentes navegadores.

**17. Na sua visão, o que a IA pode colaborar para os testes de software?**

A IA pode colaborar facilitando ou agilizando processos que já são realizados manualmenente. Como por exemplo: criação de casos de teste, detecção de padrões de falhas em logs, testes visuais para validar UI, análise de impacto de mudanças (test impact analysis).

**18. Cite exemplos da importância da rastreabilidade nos testes.**

Permite ligar requisitos = casos de teste = defeitos. Isso garante cobertura completa, facilita auditorias e ajuda na priorização de correções.

**19. Na sua visão, quando os testes são o suficiente?**

Quando atingem um equilíbrio entre risco e cobertura. Critérios: Cobertura mínima de requisitos atingida, riscos principais mitigados, nenhum defeito crítico aberto, stakeholders satisfeitos com o nível de confiança.

**20. Vamos imaginar 2 cenários que não são ideais:  
  
1. O QA está atuando sozinho na squad, mas os DEVs entregam as tarefas com certa rapidez (embora seja bastante coisa para apenas uma pessoa).**

**2. A equipe de QA é suficiente, porém os DEVs liberam as tarefas no final da sprint.**

O pior cenário é quando os devs liberam tudo no final da sprint, pois impede testes contínuos e aumenta risco de retrabalho. Um QA atuando sozinho não é ideal, mas ainda permite adaptação com priorização.

**21. Diferencie teste estático e dinâmico.**

Estático: sem execução de código (revisão, inspeção, análise de requisitos).  
Dinâmico: com execução de código (testes funcionais, unitários, E2E).

**22. Diferencie teste alfa e beta.**

Alfa: feito internamente pela equipe de QA/Dev antes da liberação.  
Beta: feito por usuários reais em ambiente próximo ao de produção.

**23. O que é Splunk? Ele pode ser usado por QAs?**

Splunk é uma ferramenta de análise de logs e monitoramento.  
Sim, QAs podem usar para identificar erros em tempo real, validar fluxos de sistema em logs e investigar defeitos reportados.

**24. Cite exemplos de como a AWS pode ser usada no trabalho do QA.**

Ambientes de teste em EC2/Docker, Banco de dados para testes em RDS, Load testing com AWS CloudWatch + AutoScaling, Armazenar evidências (S3).

**25. Você precisa executar um script de Performance, o nome do projeto é “Teste\_TOTVS”. Você precisa executar via linha de comando. Escreva a linha de comando para executar esse script performático.**

jmeter -n -t Teste\_TOTVS.jmx -l resultados.jtl -e -o ./report

**26. Você é um QA especialista em Testes de Performance. Você tem 2 clientes principais: Uma grande indústria de carros e um programa de TV com vários patrocinadores. Nesse cenário, qual dos dois clientes necessita mais de um teste performático e por quê?**

O programa de TV com vários patrocinadores, porque a visibilidade é muito alta, com acessos simultâneos massivos em tempo real. Uma falha afetaria imagem, receita e contratos.

**27. Você é Especialista em Automação de Testes, porém sua empresa tem uma resistência muito grande para aderir à automação (mesmo necessitando). Que estratégia você usaria para convencer seus gestores e a empresa como um todo a usarem a automação de testes?**

Mostrar ROI: redução de retrabalho e tempo de regressão. Apresentar um projeto piloto com métricas reais. Destacar riscos atuais sem automação. Alinhar automação com metas de negócio (qualidade, time-to-market).

**28. O que é teste de confiabilidade?**

Avalia a capacidade do sistema de operar corretamente por um período de tempo sem falhas. Ex.: rodar sistema por 72h seguidas sem crash.

**29. O que é teste de portabilidade?**

Valida se o sistema funciona corretamente em diferentes ambientes (SO, browsers, dispositivos). Ex.: rodar em Windows, Linux e Mac.

**30. O que é teste de segurança?**

Valida a proteção contra acessos indevidos, vazamento de dados e ataques (SQL Injection, XSS, CSRF, etc).