

Neste artigo de perguntas da entrevista de teste de software, eu coletei as perguntas mais frequentes dos entrevistadores. Essas perguntas são coletadas após consulta aos principais especialistas da indústria na área de testes manuais e de automação. Se você quiser se atualizar com os fundamentos do teste de software, vale a pena dar uma olhada nesse material.

De forma simples e bem didática através de perguntas você será capaz de assimilar bastante conteúdo em um curto espaço de tempo.

Vamos começar as perguntas de perguntas de nível iniciante primeiro.

Perguntas da entrevista de teste de software de nível iniciante

1. Quais são as fases envolvidas no ciclo de vida do teste de software?

As diferentes fases envolvidas no ciclo de vida do teste de software são:

Análise de Requisitos	Aqui, a equipe de QA entende os requisitos e identifica os requisitos testáveis.
Planejamento de Teste	Nesta fase, a estratégia de teste é definida.
Desenvolvimento de Caso de Teste	Aqui, casos de teste detalhados são definidos e desenvolvidos.
Configuração de ambiente	É uma configuração de software e hardware para as equipes de teste executarem casos de teste.
Execução de Teste	É o processo de execução do código e comparação dos resultados esperados e reais.
Encerramento do ciclo de teste	Envolve a verificação da reunião do membro da equipe de teste e a avaliação dos critérios de conclusão do ciclo com base na cobertura do teste, qualidade, custo, tempo, objetivos críticos de negócios e software.

2. Quais são os diferentes métodos de teste?

Existem três métodos de teste de software e são os seguintes:

- **Teste de caixa preta:** É uma estratégia de teste baseada exclusivamente em requisitos e especificações. Nesta estratégia, não requer nenhum conhecimento de caminhos internos, estruturas ou implementação do software que está sendo testado.
- **Teste de caixa branca:** é uma estratégia de teste baseada em caminhos internos, estruturas de código e implementação do software que está sendo

testado. O teste de caixa branca geralmente requer habilidades de programação detalhadas.

- **Teste de caixa cinza:** É uma estratégia para depuração de software em que o testador tem conhecimento limitado dos detalhes internos do programa.

3. Quais são os diferentes níveis de teste?

Existem principalmente quatro níveis de teste e são:

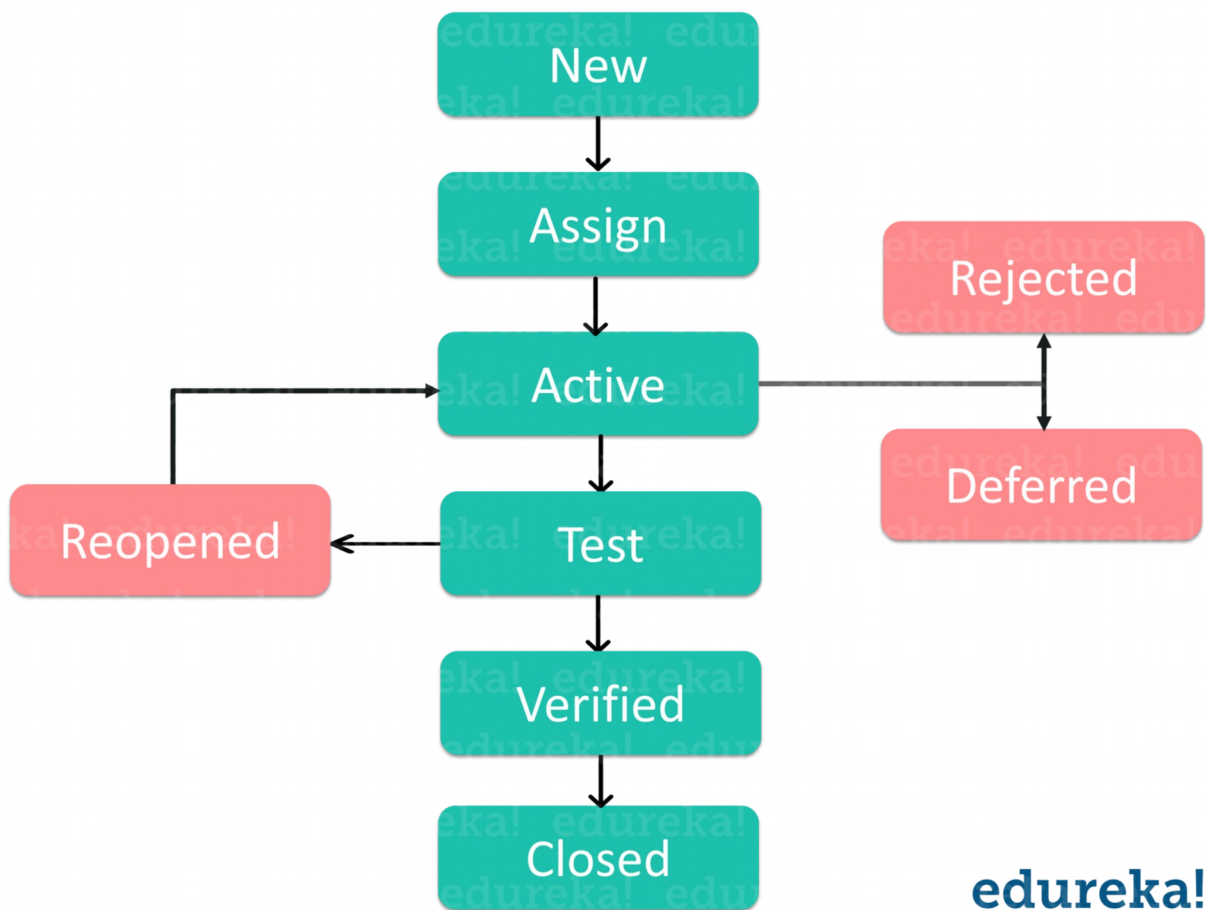
- Teste de Unidade
- Teste de integração
- Teste de Sistema
- Teste de aceitação

Basicamente, ele começa com a **fase de Teste de Unidade** e termina com o **Teste de Aceitação**.

4. Explique o ciclo de vida do bug ou o ciclo de vida do defeito.

O **ciclo de vida de um defeito** é um processo no qual um defeito passa por várias fases durante toda a sua vida. Começa quando um defeito é encontrado e termina quando um defeito é fechado, após garantir que não seja reproduzido.

O ciclo de vida do bug ou defeito inclui as etapas ilustradas na figura abaixo.



edureka!

Pode variar de organização para organização e também de projeto para projeto com base em vários fatores, como política da organização, modelo de desenvolvimento de software usado (como Agile, Iterativo), cronogramas de projeto, estrutura de equipe, etc.

5. O que é um caso de teste?

Um caso de teste nada mais é do que um conjunto de condições ou variáveis sob as quais um testador determinará se um sistema em teste satisfaz os requisitos ou funciona corretamente.

6. Qual é a diferença entre testes funcionais e não funcionais?

Teste funcional	Teste Não Funcional
Realizado antes do teste não funcional	Realizado após o teste funcional
Com base nos requisitos do cliente	Com base nas expectativas dos clientes
Descreve o que o produto faz	Descreve como o produto funciona

7. O que é Verificação e Validação em Teste de Software?

Verificação: É uma técnica de análise estática. Aqui, o teste é feito sem executar o código. Os exemplos incluem - revisões, inspeção e passo a passo.

Validação: É uma técnica de análise dinâmica onde o teste é feito executando o código. Os exemplos incluem técnicas de teste funcionais e não funcionais.

No modelo V, as atividades de desenvolvimento e QA são feitas simultaneamente. Não existe uma fase discreta chamada Teste, em vez disso, o teste começa logo na fase de requisitos. As atividades de verificação e validação caminham juntas.

8. O que é teste de usabilidade?

É uma metodologia de teste onde o cliente final é solicitado a usar o software para ver se o produto é fácil de usar, para ver a percepção do cliente e o tempo da tarefa. Uma maneira precisa de finalizar o ponto de vista do cliente para usabilidade é usando protótipo ou software mock-up durante os estágios iniciais.

9. Quais são as categorias de defeitos?

Existem três categorias principais de defeitos, conforme mostrado na figura abaixo:



- Errado:** **significa** que os requisitos foram implementados incorretamente.É uma variação da especificação fornecida.
- Ausente:** Esta é uma variação das especificações, uma indicação de que uma especificação não foi implementada ou um requisito do cliente não foi anotado corretamente.
- Extra:** É um requisito incorporado ao produto que não foi fornecido pelo cliente final. É sempre uma variação da especificação, mas pode ser um atributo desejado pelo usuário do produto.

10. Em que base o plano de aceitação é preparado?

Basicamente, o documento de aceitação é preparado usando as seguintes entradas.

- Documento de requisito:** especifica o que exatamente é necessário no projeto da perspectiva do cliente.
- Entrada do cliente:** podem ser discussões, conversas informais, e-mails, etc.
- Plano do projeto:** O plano do projeto preparado pelo gerente do projeto também serve como uma boa entrada para finalizar o teste de aceitação.

11. O que é cobertura e quais são os diferentes tipos de técnicas de cobertura?

O parâmetro usado no teste de software para descrever até que ponto o código-fonte é testado é conhecido como cobertura. Existem três tipos básicos de técnicas de cobertura e são:

- 1.**Cobertura de declaração:** garante que cada linha de código-fonte foi executada e testada.
- 2.**Cobertura de decisão:** garante que todas as decisões (verdadeiro / falso) no código-fonte foram executadas e testadas.
- 3.**Cobertura do caminho:** aqui garantimos que todas as rotas possíveis através de uma determinada parte do código sejam executadas e testadas.

12. Quais são os benefícios dos testes de automação?

Os benefícios dos testes de automação são:

- 1.Suporta a execução de casos de teste repetidos

2. Ajuda no teste de uma grande matriz de teste
3. Ativa a execução paralela
4. Encoraja a execução autônoma
5. Melhora a precisão, reduzindo assim os erros gerados por humanos
6. Economiza tempo e dinheiro

13. Por que o Selenium é a ferramenta preferida para testes de automação?

Selenium é uma ferramenta de código aberto que se destina a automatizar os testes realizados em navegadores web. Como o Selenium é de código aberto, não há custo de licenciamento envolvido, o que é uma grande vantagem sobre outras ferramentas de teste. Outras razões por trás da popularidade cada vez maior do Selenium são:

- Os scripts de teste podem ser escritos em qualquer uma dessas linguagens de programação: **Java, Python, C #, PHP, Ruby, Perl & Internet**
- Os testes podem ser realizados em qualquer um destes sistemas operacionais: **Windows, Mac ou Linux**
- Os testes podem ser realizados em qualquer navegador: **Mozilla Firefox, Internet Explorer, Google Chrome, Safari ou Opera**
- Ele pode ser integrado com ferramentas como **TestNG, e Junit** para gerenciar casos de teste e gerar relatórios
- Ele pode ser integrado com **Maven, Jenkins & Docker** para realizar testes contínuos

14. Quais são os vários componentes do selênio?

Os diferentes componentes do selênio são:

- Selenium Integrated Development Environment (IDE)
- Selenium Remote Control (RC)
- [Selenium WebDriver](#)
- [Grade de selênio](#)

15. Quais são os diferentes tipos de localizadores no Selênio?

O localizador nada mais é que um endereço que identifica um elemento da web exclusivamente dentro da página da web. Assim, para identificar os elementos da web de forma precisa e precisa, temos diferentes tipos de localizadores no Selenium da seguinte forma:

- ID
- Nome da classe
- Nome
- TagName
- Texto do link
- PartialLinkText
- XPath
- Seletor CSS
- DOM

16. O que é XPath?

Xpath também chamado de **XML Path** é uma linguagem para consultar documentos XML. É uma estratégia importante para localizar elementos no selênio. Consiste em uma expressão de caminho junto com algumas condições. Aqui, você pode escrever facilmente um script / consulta XPath para localizar qualquer elemento na página da web. Ele é projetado para permitir a navegação de documentos XML, com o objetivo de selecionar elementos individuais, atributos ou alguma outra parte de um documento XML para processamento específico. Também produz localizadores confiáveis.

17. Qual é a diferença entre Caminho Absoluto e Relativo?

- XPath absoluto

É a maneira direta de localizar o elemento, mas a desvantagem do **XPath absoluto** é que, se houver alguma alteração feita no caminho do elemento, o **Xpath** falhará. Por exemplo: /html/ body / div [1] / section / div [1] / div

•**XPath Relativo**

Para **Relative Xpath**, o caminho começa no meio da estrutura HTML DOM. Ele começa com a barra dupla (//), o que significa que pode pesquisar o elemento em qualquer lugar da página da web. Por exemplo: **// input [@ id = 'ap_email']**

18. Quais são as diferentes exceções no Selenium WebDriver?

As exceções no Selenium são semelhantes às exceções em outras linguagens de programação. As exceções mais comuns no Selênio são:

- TimeoutException**
- NoSuchElementException**
- ElementNotVisibleException**
- StaleElementException**

19. Quando devo usar Selenium Grid?

O Selenium Grid pode ser usado para executar scripts de teste iguais ou diferentes em várias plataformas e navegadores simultaneamente, de modo a obter execução de teste distribuída, testando em ambientes diferentes e economizando tempo de execução notavelmente.

20. Como eu inicio o navegador usando o WebDriver?

A seguinte sintaxe pode ser usada para iniciar o navegador:

WebDriver driver = **new** FirefoxDriver ();

Driver WebDriver = **novo** ChromeDriver ();

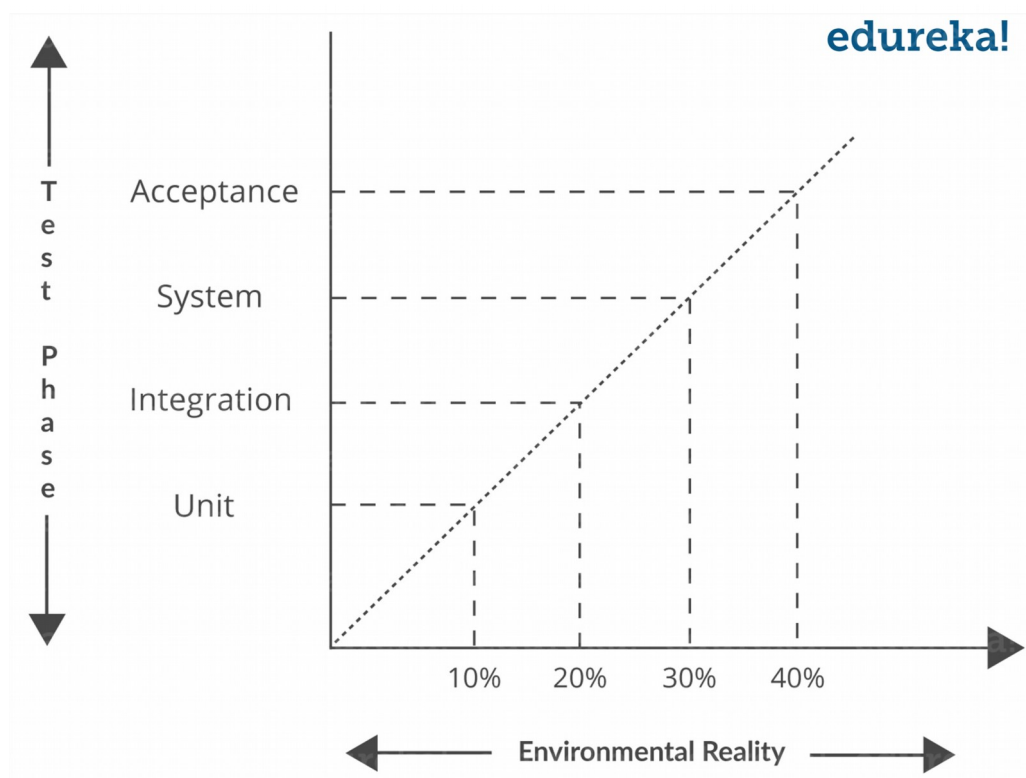
Driver WebDriver = **novo** InternetExplorerDriver ();

21. O teste deve ser feito somente após a conclusão das fases de construção e execução?

O teste é sempre feito após as fases de construção e execução. Quanto mais cedo detectarmos um defeito, mais econômico ele será. Por exemplo, consertar um defeito na manutenção é dez vezes mais caro do que consertá-lo durante a execução.

22. Qual é a relação entre a realidade do ambiente e as fases de teste?

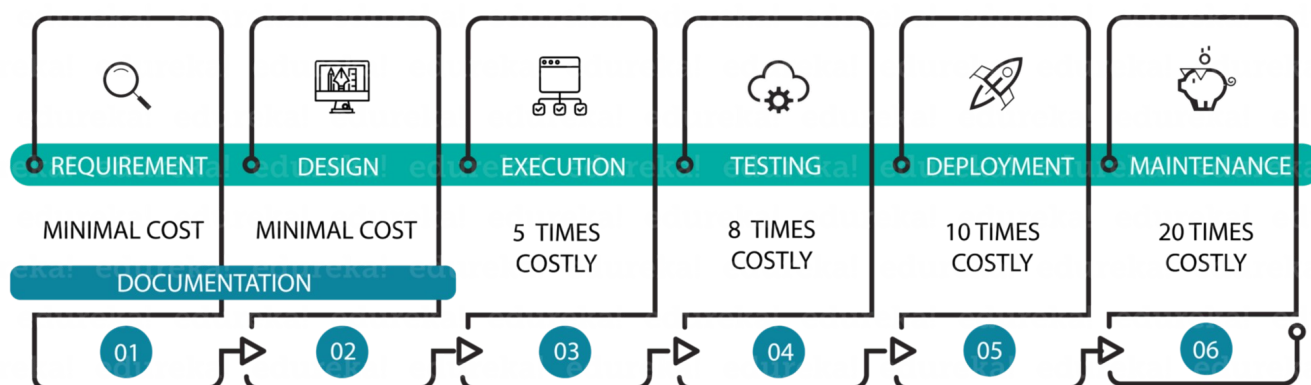
Conforme as fases de teste começam a avançar, a realidade do ambiente se torna mais importante. Por exemplo, durante o teste de unidade, você precisa que o ambiente seja parcialmente real, mas na fase de aceitação você deve ter um ambiente 100% real, ou podemos dizer que deveria ser o ambiente real.



O gráfico acima mostra durante o teste de aceitação que deve ser 100% real.

23. Um defeito que poderia ter sido removido durante o estágio inicial é removido em um estágio posterior. Como isso afeta o custo?

Se no estágio inicial um defeito for identificado, ele deve ser removido durante esse estágio / fase em si, em vez de em algum estágio posterior. É um fato que se um defeito for atrasado para fases posteriores, ele se torna mais caro. A figura a seguir mostra como um defeito é caro à medida que as fases avançam.



Se um defeito for identificado e removido durante a fase de projeto, é o mais econômico, mas quando removido durante a manutenção torna-se vinte vezes mais caro.

24. O que você quer dizer com teste de regressão e confirmação?

Teste de regressão: é definido como um tipo de teste de software para confirmar que uma alteração recente no código não afetou adversamente os recursos existentes.

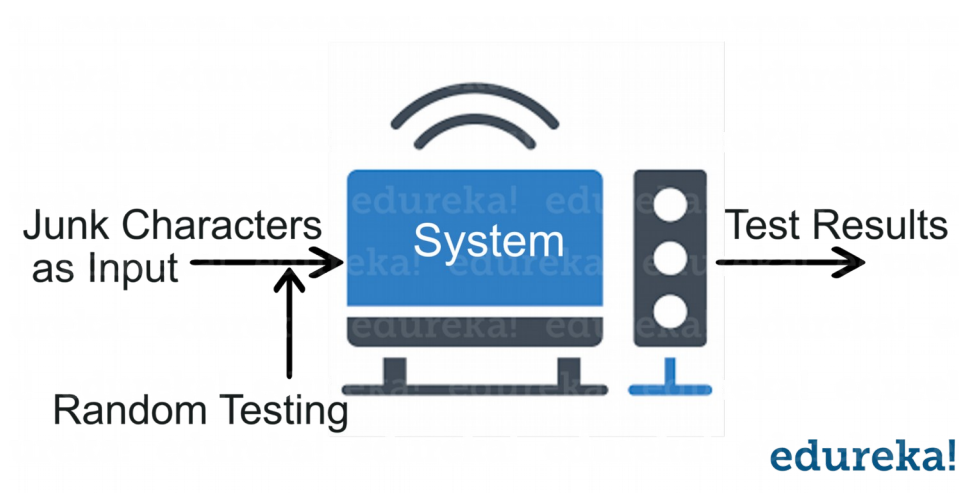
Teste de confirmação: Quando um teste falha devido ao defeito, o defeito é relatado. Em seguida, uma nova versão do software é enviada e o defeito foi corrigido. Isso é chamado de **teste de confirmação** ou novo **teste**.

25. O que você quer dizer com análise de valor limite?

A **análise de valor de limite (BVA)** é uma técnica de design de **teste de caixa preta** que é aplicada para ver se há algum **bug** no limite do domínio de entrada.

26. O que é teste aleatório?

Normalmente, em testes aleatórios, os dados são gerados aleatoriamente, muitas vezes usando uma ferramenta. Por exemplo, a figura a seguir mostra como os dados gerados aleatoriamente são enviados ao sistema.



Esses dados são gerados usando uma ferramenta ou algum mecanismo automatizado. Com essa entrada gerada aleatoriamente, o sistema é então testado e os resultados são observados de acordo.

27. Com base em que você pode chegar a uma estimativa para o seu projeto?

Para estimar seu projeto, você deve considerar os seguintes pontos:

- Divida todo o projeto nas menores tarefas
- Atribuir cada tarefa aos membros da equipe
- Faça uma estimativa do esforço necessário para completar cada tarefa
- Valide a estimativa

28. Quais casos de teste são escritos primeiro: caixas brancas ou caixas pretas?

Normalmente, os casos de teste de caixa preta são escritos primeiro e os casos de teste de caixa branca depois. Para escrever casos de teste de caixa preta, precisamos do documento de requisitos e, design ou plano de projeto. Esses documentos estão

facilmente disponíveis no início do projeto. Os casos de teste de caixa branca não podem ser iniciados na fase inicial do projeto porque eles precisam de mais clareza de arquitetura que não está disponível no início do projeto. Portanto, normalmente os casos de teste de caixa branca são escritos após os casos de teste de caixa preta serem escritos.

29. Mencione os componentes básicos do formato do relatório de defeitos.

Os componentes básicos do formato do relatório de defeito incluem:

- Nome do Projeto
- Nome do módulo
- Defeito detectado em
- Defeito detectado por
- ID e nome do defeito
- Instantâneo do defeito
- Status de prioridade e gravidade
- Defeito resolvido por
- Defeito resolvido em

30. Os testes de automação na metodologia ágil são úteis?

O teste de automação é muito útil na metodologia ágil e ajuda a atingir a cobertura máxima de teste em um tempo menor do sprint.

31. Quais casos de teste podem ser automatizados?

- Casos de teste de fumaça
- Casos de teste de regressão
- Casos de teste de cálculo complexo
- Casos de teste baseados em dados
- Casos de teste não funcionais

32. Com que base você pode mapear o sucesso dos testes de automação?

Seguindo os critérios, o sucesso do teste de automação pode ser mapeado:

- Taxa de detecção de defeitos
- Tempo de execução da automação e economia de tempo para lançar o produto
- Redução de mão de obra e outros custos

33. Explique o teste de carga em sites?

Para acessar um site, um usuário envia uma “solicitação” ao servidor desse site, e o servidor envia de volta uma resposta na forma do site que você deseja acessar. Para testar a carga de um site, engenheiros de garantia de qualidade e engenheiros de automação precisam apenas multiplicar o número de respostas enviadas para simular diferentes cargas de tráfego. A resposta do servidor da web ao influxo de usuários virtuais pode então ser medida. Isso é usado para determinar problemas de desempenho e capacidade do servidor.

34. Qual é a diferença entre Selênio e Sikuli?

Selênio	Sikuli
Não pode automatizar objetos flash como reprodutor de vídeo, reprodutor de áudio, etc.	Ele fornece amplo suporte para automatizar objetos flash
Tem API complicada	Tem uma API simples
Ele pode automatizar apenas aplicativos da web	Ele pode automatizar a web e também um aplicativo do Windows.

35. Como clicar em um hiperlink usando linkText ()?

```
1driver.findElement(By.linkText("Google")).click();
```

Este comando encontra o elemento usando o texto do link e, a seguir, clique nesse elemento. Assim, o usuário seria redirecionado para a página correspondente.

36. O que é TestNG?

É uma estrutura avançada projetada de forma a aproveitar os benefícios dos desenvolvedores e testadores. Ele também possui um mecanismo de tratamento de exceções embutido que permite que o programa seja executado sem encerrar inesperadamente.

37. Como definir a prioridade do caso de teste no TestNG?

O código abaixo ajuda você a entender como definir a prioridade do caso de teste no TestNG.

```
1package TestNG;
2import org.testng.annotations.*;
3public class SettingPriority {
4@Test(priority=0)
5public void method1() {
6}
7@Test(priority=1)
8public void method2() {
9}
10@Test(priority=2)
11public void method3() {
12}
13}
```

Sequência de execução de teste:

```
1Method1
2Method2
3Method3
```

38. Qual é a diferença entre Selênio e QTP?

Selênio	Profissional de teste rápido
Selenium suporta quase todos os navegadores populares como Firefox, Chrome, Safari, Internet Explorer, Opera etc	QTP suporta Internet Explorer, Firefox e Chrome. QTP suporta apenas o sistema operacional Windows
Selenium é distribuído como uma ferramenta de código aberto e está disponível gratuitamente	QTP é distribuído como uma ferramenta licenciada e é comercializado
Selenium oferece suporte a testes apenas de aplicativos baseados na web	QTP suporta testes de aplicativos baseados na web e aplicativos baseados em Windows

39. O que é repositório de objetos? Como podemos criar um repositório de objetos no Selenium?

O repositório de objetos refere-se à coleção de elementos da web pertencentes ao Application Under Test (AUT) junto com seus valores de localizador. Com relação ao Selenium, os objetos podem ser armazenados em uma planilha do Excel que pode ser preenchida dentro do script sempre que necessário.

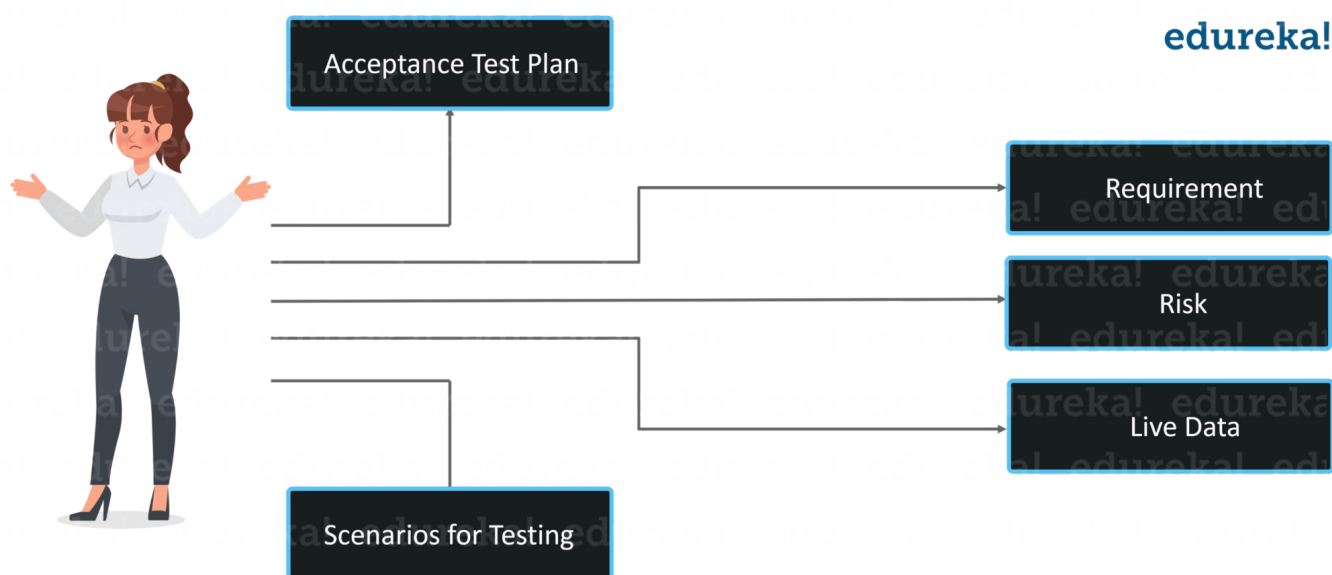
40. Como inserir texto na caixa de texto usando Selenium WebDriver?

Usando o método **sendKeys()**, podemos inserir o texto na caixa de texto usando o Selenium WebDriver.

Perguntas da entrevista de teste de software de nível avançado

41. Que tipo de informação precisamos do usuário final para começar o teste adequado?

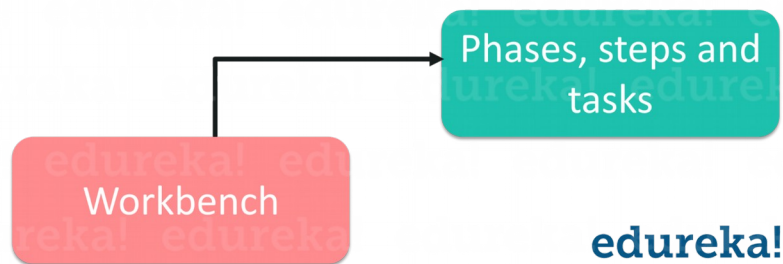
Um usuário final é a pessoa mais importante porque é ele quem deve usar o produto e tem grande interesse de qualquer outra pessoa no projeto.



A figura acima ilustra a entrada que é essencial do usuário final.

42. O que significa conceito de bancada de trabalho?

Um workbench em seu núcleo é uma forma de documentar como uma atividade específica deve ser executada. Geralmente é referido como fases, etapas e tarefas, conforme mostrado na figura a seguir.



Existem cinco tarefas para cada bancada e são as seguintes:

- 1.Entrada
- 2.Executar
- 3.Verificar
- 4.Saída de Produção
- 5.Retrabalho

43. O que significa cascata de defeitos?

O cascadeamento de defeitos é um defeito causado por outro defeito. Um defeito aciona o outro defeito. Quando um defeito estiver presente em qualquer etapa, mas não for identificado, oculo para as outras fases sem ser notado. Isso resultará em um aumento no número de defeitos.

Vamos entender isso por um exemplo.

Você está projetando o Módulo de Login de uma página da web:

Na fase 1- Você está projetando o Módulo de Registro de Usuário para Login e o número do celular é obrigatório, mas você pode deixá-lo em branco devido a um bug que passa despercebido.

Na Fase 2 - Você vai desenhar o formulário de login com nome de usuário e senha. A senha é OTP que será enviada ao número de celular cadastrado do Usuário.

Agora, como o módulo Register tem um bug que o número do celular pode ser deixado em branco, então isso pode levar a uma falha de login ou talvez algum erro do sistema ou travamento se um número de celular nulo não for tratado. Isso é conhecido como cascata de defeitos.

44. Quais são as diferentes estratégias de distribuição para os usuários finais?

As estratégias a serem seguidas para o lançamento são as seguintes:

- Piloto
- Implementação Gradual
- Implementação em fases
- Implementação Paralela

45. Explique como você pode encontrar links quebrados em uma página usando o Selenium WebDriver?

Esta é uma pergunta complicada que o entrevistador pode apresentar a você. Ele pode fornecer uma situação em que existem 20 links em uma página da web, e temos que verificar quais desses 20 links estão funcionando e quantos não estão funcionando (quebrados).

Como você precisa verificar o funcionamento de cada link, a solução alternativa é enviar solicitações HTTP a todos os links na página da web e analisar a resposta. Sempre que você usar o método `driver.get ()` para navegar para uma URL, ele responderá com um status de **200 - OK**. Isso indica que o link está funcionando e foi obtido. Considerando que qualquer outro status indica que o link está quebrado.

Agora vamos entender como fazer isso.

Primeiro, temos que usar as marcas âncora <a> para determinar os diferentes hiperlinks na página da web. Para cada tag <a>, podemos usar o valor do atributo 'href' para obter os hiperlinks e então analisar a resposta recebida quando usado no método **driver.get ()**.

46. Qual técnica deve ser considerada no script " se não houver id do frame nem nome do frame"?

Se o nome do quadro e o id do quadro não estiverem disponíveis, podemos usar **quadro por índice**. Por exemplo, existem 3 frames em uma página da web e se nenhum deles tiver um nome de frame e id de frame, então ainda podemos selecionar esses frames usando o atributo de índice de frame (baseado em zero). Todo o quadro terá um número de índice como o primeiro quadro estaria no índice "0", o segundo no índice "1" e o terceiro no índice "2".

```
1driver.switchTo().frame(int arg0);
```

47. Como tirar screenshots no Selenium WebDriver?

Usando a função **TakeScreenshot**, você pode fazer uma captura de tela. Com a ajuda do método **getScreenshotAs ()**, você pode simplesmente salvar a captura de tela.

Exemplo: Arquivo scrFile = ((TakeScreenshot) driver) .getScreenshotAs
(outputType.FILE);

48. Explique como você logará em qualquer site se ele está mostrando qualquer pop-up de autenticação para nome de usuário e senha?

Se houver um pop-up para fazer login, precisamos usar o comando explícito e verificar se o alerta está realmente presente. O código a seguir ajuda a entender o uso do comando de espera explícito.

```
1WebDriverWait wait = new WebDriverWait(driver, 10);  
2Alert alert = wait.until(ExpectedConditions.alertIsPresent());
```

```
3 alert.authenticateUsing(new UserAndPassword(**username**,  
  **password**));
```

49. Como pular um método ou bloco de código no TestNG?

Para pular um determinado método de teste ou código, você pode definir o parâmetro 'enabled' na anotação de teste como false.

@Test (ativado = falso)

50. Explique resumidamente o que o trecho de código abaixo indica?

```
1 WebElement sample = driver.findElement(By.xpath("//*[contains(text(),  
  'data')]"));
```

Ele define uma *amostra* de variável do tipo WebElement e usa uma pesquisa XPath para inicializá-la com uma referência a um elemento que contém o valor de texto "dados".

Isso nos leva ao final deste artigo sobre as 50 principais perguntas da entrevista de teste de software. Espero que tenha ajudado a aumentar o seu conhecimento. Desejo a você tudo de bom para sua entrevista. Bom aprendizado.