

TESTE FRONT-END

PARA COMEÇAR

Este conteúdo aborda conceitos relacionados ao planejamento e à execução de Teste Front-End.

Esperamos que você desenvolva, no decorrer de seus estudos, as seguintes capacidades:

- Reconhecer as especificações técnicas da interface.
- Reconhecer os requisitos da documentação de testes.
- Reconhecer as etapas de planejamento de testes.
- Reconhecer normas e procedimentos de testes.
- Aplicar testes definidos no plano de teste.
- Aplicar testes definidos no plano de teste.
- Reconhecer as técnicas de testes.

Para desenvolver tais capacidades, você deverá estudar os seguintes temas:

- Conceitos, processos fundamentais e planejamento de testes client-side.
- Tipos e técnicas de testes.

O estudo de tais temas será necessário para que você resolva a situação-problema a seguir. Então, avance para conhecê-la.

SITUAÇÃO-PROBLEMA

Você trabalha como programador Front-End e finalizou o desenvolvimento de um site. Agora, precisa atestar que todas as funcionalidades e eventos da interface estão se comportando devidamente, a fim de assegurar a usabilidade do sistema e proporcionar uma experiência positiva ao usuário final, atendendo às expectativas do cliente e garantindo a qualidade da entrega.

Para isso, você deverá pensar em uma estratégia para testar automaticamente os principais fluxos para assegurar que não apresentam ou apresentarão problemas, sobretudo, em relação a alterações ou inclusões de novas funcionalidades.

Para tanto, você deverá realizar os seguintes desafios:

Desafio 1

Fazer o planejamento e o desenho do teste, considerando as especificações técnicas de interface.

Desafio 2

Testar o sistema, aplicando correções de acordo com os resultados obtidos no teste.

DESAFIO 1

Nesta etapa, você deverá resolver o desafio 1:

- Fazer o planejamento e o desenho do teste, considerando as especificações técnicas de interface.

Para isso, você estudará os seguintes conteúdos:

- Conceitos e processo fundamental de testes client-side.
- Planejamento e desenho de testes.
- Tipos e técnicas de testes.

REQUISITOS E AVALIAÇÃO DE QUALIDADE

A qualidade de uma aplicação (sistema, software, site entre outras) pode ser avaliada a partir de diferentes abordagens e aspectos. De acordo com a norma ISO/IEC 25010:2011, uma aplicação deve atender aos requisitos de:

- a) Qualidade interna e externa:** compreende princípios internos ou estruturais da aplicação, como tamanho e complexidade dos códigos, e externos, quando a aplicação é utilizada como parte de um sistema computacional, além de serem resultantes de atributos internos.
- b) Qualidade em uso:** avalia a eficácia, a produtividade, a segurança e a satisfação do usuário.

Em relação às qualidade externas e internas, há seis requisitos que vamos detalhar melhor.

1. **Funcionalidade:** Um site funcional é o que satisfaz a necessidade do usuário, atendendo a tarefas e objetivos específicos (adequação), apresentando resultados corretos (acurácia); interagindo com outros sistemas (interoperabilidade) e protegendo dados e informações (segurança).
2. **Confiabilidade:** Está relacionada à entrada e saída de dados para o usuário, garantindo que a aplicação funcione como esperado em situações normais e anormais. Além disso, deve-se evitar falhas (maturidade), manter um nível de desempenho mínimo e recuperar dados afetados, caso ocorra algum problema (tolerância a falhas e recuperabilidade).
3. **Usabilidade:** Um site usual é aquele que apresenta informações claras e cuja navegação é facilmente compreendida e aprendida pelo usuário (inteligibilidade e apreensibilidade), permitindo a realização de tarefas sem dificuldade (operacionalidade), além de prevenir erros e apresentar interface agradável (atratividade) e acessível.
4. **Eficiência:** Um site é considerado eficiente quando apresenta:
 - Velocidade;
 - Escalabilidade;

Esses conceitos serão aprofundados ao longo deste desafio.

5. **Manutenibilidade:** A manutenibilidade está relacionada à capacidade do site ser facilmente modificado e testado (modificabilidade e testabilidade), apresentando estabilidade e fácil identificação de erros decorrentes dessas modificações (analisabilidade).
6. **Portabilidade:** Capacidade do sistema ser transferido de um ambiente para outro (por exemplo, sistemas operacionais, diferentes versões de banco de dados etc.), apresentando fácil adaptação (adaptabilidade), instalação (capacidade de ser instalado), relação com outros ambientes e softwares (coexistência) ou substituição (capacidade para substituir).

Quanto ao requisito eficiência, conforme já mencionado, o site deve apresentar:

- **Velocidade:** tempo de resposta dentro de especificações. É avaliado o tempo de carregamento de informações, imagens, troca de páginas, eventos, requisições, entre outros.
- **Escalabilidade:** Está relacionada a quantas requisições o sistema suporta. Por exemplo, caso o site seja um e-commerce, quantas requisições feitas ao mesmo tempo o sistema suporta sem apresentar lentidão? Qual é a carga máxima de usuários simultâneos que o sistema comporta?

Para testar a escalabilidade, podemos mesclar outros tipos de teste.

- **Teste de carga:** Capacidade máxima de usuários logados ao mesmo tempo.
- **Teste de estresse:** Capacidade máxima de requisições feitas simultaneamente.
- **Teste de resistência:** Período máximo de tempo que o site suporta em situação adversa.
- **Teste de volume:** Quantidade de dados que podem ser salvos no banco de dados e seu impacto no sistema.

Para testar a escalabilidade de um site de e-commerce, por exemplo, podemos simular um cenário de Black Friday (importante para o setor). No **teste de carga**, verificaremos qual a quantidade máxima de usuários simultâneos logados que o sistema suporta. No **teste de estresse**, determinaremos a quantidade máxima de requisições simultâneas.

No **teste de volume**, testaremos a quantidade de dados que podem ser manipulados simultaneamente no banco de dados, para garantir que o sistema de controle de estoque funcione normalmente.

No teste de resistência, devemos nos certificar que o site estará estável por, no mínimo, 24 hr durante a Black Friday ou verificar quanto tempo é necessário para ele volte a funcionar normalmente depois de um queda.

PENSE NISSO...

Ao analisar todos esses requisitos, além de garantir a qualidade do site, podemos:

- Fazer escolhas mais assertivas, como determinar o melhor perfil de hospedagem para o site;
- Prevenir contratempos que possam fazer o site travar ou cair;
- Evitar falhas, como permitir a compra de um item que já está esgotado no estoque.

NÍVEIS DE TESTES

De acordo com a característica ou o atributo de qualidade que se pretende avaliar na aplicação de um site, que podem estar relacionadas a **usabilidade**, **confiabilidade**, **desempenho**, **manutenção** etc, aplica-se um conjunto de estratégias para encontrar falhas.

Mike Cohn, em seu livro *Succeeding with Agile*, sugere que os testes sejam estruturados em diferentes camadas ou níveis, começando por testes unitários, passando pelos de integração e finalizando com os de interface.

Interface do Usuário

Também chamado de end-to-end, de uma ponta a outra, reproduz o fluxo de interações do usuário com o sistema em forma de scripts de código, a partir de um contexto real. Por exemplo, testar se a agenda mostra corretamente o calendário, as datas e os horários disponíveis, se as mensagens de sucesso ou alerta de sobreposição de eventos aparecem corretamente, se é possível deletar ou mover eventos corretamente, entre outros.

Integração

Testam se as unidades da aplicação desenvolvidas separadamente funcionam juntas conforme o esperado. Por exemplo, testar se a data e o horário da agenda estão no mesmo formato e funcionamento em conjunto. O que aconteceria se a data fosse formatada em 24 horas e o horário fosse formatado em dois períodos de 12 horas? Uma reunião às 15h seria agendada corretamente?

Unitários

Testam um pedaço ou uma unidade de código de uma aplicação isoladamente, sem atingir várias camadas de um sistema, identificando erros com rapidez, precisão e baixo custo. Por exemplo, testar se o campo para entrada de data ou de horário está no formato estabelecido para inserir uma reunião na agenda.

TÉCNICAS DE TESTES

Existem várias técnicas utilizadas para se testar uma aplicação de Front-End. A seguir apresentaremos os testes funcionais (caixa-preta) e estruturais (caixa-branca).

FUNCIONAIS (CAIXA-PRETA)

De acordo com o *international Software Testing Qualifications Board (ISTQB)*¹, o *teste funcional*² envolve a verificação de uma determinada funcionalidade ou de um componente da aplicação, sem que exista a preocupação com a estrutura interna do programa, ou seja, desconsidera-se a estrutura do código-fonte, os detalhes de implementação ou os cenários de execução.

¹ Conselho Internacional de Qualificações de Teste de Software, em português.

² Também denominado teste de caixa-preta, trata-se de um teste comportamental orientado a um dado ou orientado a uma entrada e saída.

Nesse tipo de teste, um código será estruturado a partir da visão do usuário, objetivando testar dados de entradas e de saídas do sistema, cliques, navegabilidade e usabilidade. Além disso, o resultado obtido é comparado a um previamente conhecido, permitindo identificar erros de:

- funções incorretas ou ausentes;
- interface;
- estruturas de dados;
- acesso a bancos de dados externos;
- desempenho;
- inicialização e término.

Essa técnica é aplicável a todas as fases de teste (unitário, integração e interface).

VOCÊ SABIA?

O ISTQB é o órgão internacional responsável pelas certificações para **profissionais da área de TI que trabalham com testes de softwares**. No Brasil, temos o Brazilian Software Testing Qualifications Board (BSTQB) como representante do ISTQB.

ESTRUTURAIS (CAIXA-BRANCA)

Os testes estruturais, também chamados de caixa-branca, avaliam e validam o código-fonte da aplicação, os diferentes algoritmos envolvidos nas decisões lógicas e, por fim, as estruturas de dados. De modo geral, esse tipo de teste simula diferentes valores de entrada, examinando cada um dos possíveis fluxos de execução dentro de funções, classes, módulos, entre outros, e verificando se os valores de saída retornam corretamente.

A equipe responsável pelo teste terá uma visão mais detalhada sobre as condições tanto do Back-End quanto do front-End, das etapas, da segurança, da funcionalidade, da navegabilidade, do fluxo de dados etc. A título de diferenciação, podemos dizer que enquanto o teste de caixa-preta é voltado à interface do sistema, o teste de caixa-branca tem foco na funcionalidade do sistema.

SAIBA MAIS

Além dos teste de caixa-preta e caixa-branca, existem outros testes, como:

- **Caixa-cinza:** aplica técnicas tanto funcionais quanto estruturais para analisar uma aplicação.
- **Regressão:** realizado em novas versões, visando identificar falhas tanto em funcionalidades já implementadas quanto em novas funcionalidades inseridas.
- **Não funcionais:** avaliam casos inválidos ou inesperados de entrada, de adequação a normas e de restrições tecnológicas, bem como os atributos de confiabilidade, eficiência, usabilidade, manutenibilidade e portabilidade.
- **Teste de aceitação e operação:** realizado pelo cliente no ambiente final de publicação, simulando rotinas do sistemas para determinar se a aplicação satisfaz ou não aos critérios de aceitação.
- **Teste alpha** (anterior ao teste de aceitação), **beta** (feito por grupo selecionado de usuários em um ambiente real) e **gama** (teste realizado com os usuários finais).

Pesquise e conheça mais sobre cada um deles.

ETAPAS OU CICLOS DE TESTES

Independentemente da técnica aplicada ou do produto a ser testado, é imprescindível planejar e documentar os testes que serão executados. A norma 829-2008 do Instituto de Engenheiros Eletricistas e Eletrônicos (IEEE) estabelece uma estrutura padrão para o planejamento de testes, do início ao fim, garantindo e controlando a qualidade.

A seguir, você conhecerá detalhadamente as principais etapas orientadas pela norma. São elas:

- Planejamento;
- Elaboração do caso de teste;
- Execução do programa;
- Avaliação de resultados.

IMPORTANTE!

A IEEE é uma instituição estadunidense sem fins lucrativos que tem como finalidade elaborar documentações para diversas áreas, incluindo a de tecnologia.

A norma 829-2008 define o padrão básico para a documentação de teste e especifica o formato dos documentos, mas não orienta se todos devem ser produzidos, nem determina os critérios próprios em relação ao conteúdo.

Portanto, cada organização pode estabelecer critérios próprios em relação ao conteúdo da documentação, além de utilizar documentos complementares, de acordo com o projeto.

Pesquise como criar uma documentação de testes baseada em um plano de teste (estágio 1).

PLANEJAMENTO

O planejamento, ou plano de teste, é um documento que especifica, principalmente:

- o objetivo do teste;
- o escopo do sistema e os requisitos que deverão ser avaliados;
- os elementos que serão testados e quais dados serão monitorados durante a execução do teste;
- o cronograma de atividades;
- a técnica que será utilizada (funcional, estrutural etc.);
- os recursos que serão utilizados (tipos de software);
- o responsável pelo monitoramento dos dados de erros, pelo recebimento dos dados coletados e pela correção de erros;
- As referências às normas aplicadas ao teste e a outros documentos associados ao projeto.

É recomendado que o planejamento inicie já na fase de levantamento de requisitos, uma vez que, após serem determinados os requisitos que o projeto deverá complementar, deverão haver meios de testá-los e avaliá-los para confirmar se, de fato, atenderam ao que foi solicitado.

Como resultado dessa etapa, elabora-se o documento denominado **plano de teste**.

IMPORTANTE!

É de extrema importância que as informações do documento estejam claras e objetivas, pois são destinadas à equipe que realizará o teste.

É possível usar um modelo de plano de teste predefinido ou criar um modelo novo.

Acesse o modelo de referência de plano de teste sugerido pela IBM:

<https://ibm.com/3kvq35n>

Você também pode pesquisar por “template de plano de teste” e conhecer outras opções.

ELABORAÇÃO DO CASO DE TESTE

Trata-se do documento que define as unidades que serão testadas, incluindo dados de entrada, resultados, ações e condições gerais para a execução do teste, ou seja, responde à pergunta: o que será testado? A resposta poderia ser caso de teste 1: efetuar login.

Um caso de teste pode estar agrupado com outros casos para formar uma suíte de testes. Por exemplo:

- caso de teste 1: efetuar login;
- caso de teste 2: incluir novos produtos;
- caso de teste 3: finalizar compra;
- caso de teste 4: efetuar logout;

Os casos de teste são derivados dos casos de uso, portanto, devem apresentar o fluxo básico de uma ação do usuário e, também, os fluxos alternativos, conforme mostra a imagem a seguir:



Após elaborar os casos de teste, a próxima etapa é criar o desenho do teste, ou seja, a codificação para testar a aplicação.

SAIBA MAIS

Casos de uso referem-se à documentação visual do projeto, na qual são criados diagramas para apresentar as funcionalidades e para possibilitar o entendimento do sistema do sistema e de sua navegabilidade antes de iniciar a programação.

Veja o material complementar e leia um exemplo de documento de especificação de casos de teste.

Arquivo: *Exemplo - Documento de Caso de Teste.*

EXECUÇÃO DO PROGRAMA

Nessa etapa, executa-se o programa ou o código utilizando os atributos do HTML, como *class*, *id*, *title*, *text*, de acordo com o plano de teste e com os casos de teste elaborados, objetivando identificar os elementos da página e simular as interações do usuário. É por meio dessa interação que é possível obter os resultados e a coleta de erros.

MONITORAÇÃO E CONTROLE

Durante a execução do teste, dados são monitorados e coletados visando avaliar fatores que influenciam o erro. Como resultado dessa etapa, sugere-se a elaboração de um documento chamado **relatório de incidente de teste**, que deve descrever todos os eventos ocorridos, viabilizando, posteriormente, a análise pela equipe de desenvolvimento, responsável por realizar a correção dessas falhas.

SAIBA MAIS

Pesquise por “template de relatório de incidente de teste” e conheça exemplos de como esse documento pode ser estruturado.

AVALIAÇÃO DE RESULTADOS

Nessa etapa, a equipe de desenvolvimento avaliará o erro, buscando compreender o que o ocasionou, por exemplo erro de sistema, de código, de internet, de hardware, para, assim, corrigi-lo.

É normal que aplicações apresentem erros de modo não consecutivo. Alguns fatores podem motivar a ocorrência de erros, como rede, requisições ou caso o sistema esteja consumindo um API que está fora do ar por razões de atualização, de implantação do sistema ou do serviço do site em que o sistema esteja hospedado ou, ainda, em razão de uma atualização ou correção de erro no site que está no ambiente de produção.

VOCÊ SABIA?

O termo ambiente de produção é utilizado para designar o site que está no ar, aquele que é acessado pelo usuário final. Ambiente de homologação é o termo utilizado para designar um ambiente de testes, que tem suas funcionalidades testadas e corrigidas antes de ser disponibilizado para o usuário final.

Ao final dessa etapa, sugere-se a elaboração de um relatório de resumo de teste que descreva:

- a equipe que realizou o teste;
- quantos casos de teste foram executados e se foi necessário criar outros durante o teste;
- quantos casos retornaram respostas adequadas e quantos casos apresentaram erros.

SAIBA MAIS

Pesquise por “template de relatório de resumo de teste” e conheça exemplos de como esse documento pode ser estruturado.

NESTE DESAFIO...

SITUAÇÃO-PROBLEMA | DESAFIO 1

Neste desafio, você estudou que o teste é uma etapa imprescindível para verificar e validar se as funcionalidades e os requisitos documentados foram implementados ou se possuem algum tipo de erro, que possa prejudicar a experiência do usuário, a fim garantir a qualidade de uma aplicação.

Você estudou também que a qualidade de qualquer aplicativo deve atender aos requisitos internos e externos, como funcionalidade, confiabilidade, usabilidade, eficiência, manutenção e portabilidade. E, para avaliação desses requisitos, aplica-se tipos e técnicas de teste específicos, que podem ser um teste unitário, de integração, de interface, de caixa preta ou branca.

Além disso, você estudou que é imprescindível planejar e documentar os testes que serão executados.

NO PRÓXIMO DESAFIO...

Você estudará como criar testes automatizados, por meio da ferramenta Selenium.

DESAFIO 2

Após fazer o planejamento e o desenho do teste, considerando as especificações técnicas de interface, nesta etapa, você deverá resolver o desafio 2:

- Testar o sistema e aplicar correções, quando necessário, a partir dos resultados obtidos no teste.

Para este desafio, você estudará o seguinte conteúdo:

- Automação de testes

AUTOMAÇÃO DE TESTES

Conforme você estudou, testes automatizados simulam a interação que um usuário faria, quantas vezes forem necessárias, como cliques, inserção de dados, preenchimento de formulários, abertura de documentos, seleção de itens, pesquisas, entre outras ações, a fim de encontrar possíveis erros.

Existem ferramentas que realizam esse tipo de teste. Neste desafio, trabalharemos com um conjunto de ferramentas chamado **Selenium**.

Selenium é um framework utilizado para testar aplicações web que permite a automatização do browser. Essa ferramenta possibilita escrever testes em várias linguagens de programação (C#, PHP, Python, Java, entre outras). Utilizaremos, a título de exemplo, a linguagem Java para a escrita do teste.

O Selenium possui um recurso chamado **WebDriver**, que permite a realização de testes automatizados dentro do seu navegador (Chrome, Firefox, Opera etc), traduzindo o código do servidor para o código de cliente (site). É uma API compacta orientada a objetos que conduz o browser de maneira eficaz.

CONFIGURAÇÃO DE AMBIENTE

Para realização do teste, é necessário ter algumas ferramentas instaladas e/ou configuradas em sua máquina para implementar o ambiente de desenvolvimento integrado.

Eclipse

Ambiente de desenvolvimento para executar a automação. Acesse seu material complementar e conheça o passo a passo para realizar o download dessa ferramenta em PDF...

JDK-8

Trata-se de um conjunto de utilitários para a plataforma Java, cujo objetivo é facilitar a criação de sistemas a partir de alguns recursos, como Java Virtual Machine (JVM), bibliotecas e compiladores.

Acesse seu material complementar e conheça o passo a passo para realizar o download dessa ferramenta, em PDF.

Apache Maven

Trata-se de um gerenciador de dependências utilizado para incrementar o projeto, configurando o arquivo pom.xml.

Acesse seu material complementar e conheça o passo a passo para realizar o download dessa ferramenta, em PDF.

Para usar o JDK e o Maven, é necessário configurar as variáveis de ambiente do windows.

Acesse seu material complementar e acesse o tutorial que ensina como realizar essa configuração, em PDF.

WebDriver

Para utilizar os recursos do WebDriver, é necessário baixar o servidor autônomo ChromeDriver, que implementa o padrão W3C WebDriver.

Acesse seu material complementar e conheça o passo a passo para realizar o download ChromeDriver, em PDF.

IDENTIFICANDO ELEMENTOS

Para identificar os elementos do site, você precisa inspecionar os elementos da página, a fim de obter os dados (**class**, **id**, **value**, **name**, entre outros) que iremos utilizar.

Para compreender melhor, imagine que sua automação precisa acessar o site do Google, mas como ela fará isso? A resposta é: por meio de algoritmos, que realizarão o “clique” na linha passada na URL do site.

PENSE NISSO...

O que o sistema precisará fazer para acessar a página que foi repassada como um comando?

Acesse o input para fazer uma pesquisa.

Portanto, como estamos trabalhando com uma máquina, precisamos deixar claras essas informações pela *id* ou pela *class*, informado, no caso de nosso exemplo: acesse o site do Google, clique no input com o id x e pesquise por “SENAI”.

INTERAGINDO COM A PÁGINA

Agora que o sistema já acessa o site do SENAI, vamos fazer com que ele interaja com a página e realize o teste de navegabilidade. Para que isso ocorra, é necessário que ele acesse mais elementos e teste os links.

Existem duas maneiras possíveis para passarmos um valor, ou seja, com o que (entrada de dados - login e senha) e como o sistema irá interagir (clikando em um elemento - botão, input, imagem, entre outros):

- Deixando visível no código;ou
- Passando como uma variável dentro de Eclipse e chamando-a no código.

COMUNICAÇÃO COM A EQUIPE DE AUTOMAÇÃO DE TESTES

A equipe de automação de testes é responsável por desenvolver a codificação que testará o aplicativo, o serviço ou o sistema online.No planejamento de testes, cada equipe se responsabiliza por uma atividade, por exemplo, criar a automação, monitorar o teste, fazer a coleta dos erros etc.

Se um trecho do código estiver errado, toda a codificação será afetada posteriormente. É o que chamamos de estrutura de filas. Devido a esse motivo, é necessário ter uma equipe exclusiva de monitoramento, que após a conclusão da codificação de teste, executa-o diversas vezes, pois pode ser que ocorra alguma alteração no código-raiz, como o acréscimo de um modal ou a troca de um *class* ou de um *id*.

Conforme você estudou no desafio 1, a última etapa a ser realizada após a aplicação dos testes é documentar os erros, por meio de relatórios, para que a equipe de desenvolvimento possa realizar correções ou melhorias dos erros identificados.

NESTE DESAFIO...

SITUAÇÃO-PROBLEMA | DESAFIO 2

Neste desafio, você estudou como criar um teste automatizado, utilizando o framework Selenium; Configurar o ambiente para a realização do teste, utilizando as ferramentas eclipse, JDK, Apache Maven e WebDriver; e aplicar a linguagem de programação Java para o sistema identificar os elementos do site, interagir com a página, simulando a interação de usuários.

PARA CONCLUIR...

PARABÉNS, VOCÊ CONCLUIU A ETAPA DE TESTE FRONT-END!

Neste conteúdo, você reconheceu a importância dos testes Front-End para um projeto. Estudou conceitos essenciais, como os níveis, as técnicas e as etapas de testes.

Em automação de teste, aprendeu a instalar ferramentas e a configurar ambientes para executar o teste automatizado (feito no site do SENAI).

Continue estudando e se aprimorando. Até breve e sucesso!

