# DEFINIÇÕES INICIAIS

* Não é necessário que você desenvolva tudo que está descrito nos requisitos, você pode focar naquilo que considera mais importante.
* Fique à vontade para entrar em contato e tirar dúvidas e/ou negociar o prazo de desenvolvimento;
* A solução deverá ser apresentada.
* Ao final responda o e-mail com suas considerações, apontando dificuldades e como as contornou.

# CENÁRIO

Garantir a qualidade da página inicial da [www.amazon.com.br,](http://www.amazon.com.br/) um dos sites de e-commerce mais acessados do mundo.

Testar a funcionalidade, usabilidade e desempenho da página inicial, com foco nas seguintes funcionalidades:

# ESPECIFICAÇÕES DO SISTEMA

* Sugestões de Pesquisa: Autocomplete
* Menu de Navegação: Responsividade
  + Desktop
  + Tablet
  + Mobile
* Carregamento e Desempenho: diferentes navegadores e dispositivos.

# TAREFA

1. Criação de Casos de Teste:
   * Elabore uma suíte de casos de teste para as funcionalidades de sugestões de pesquisa (autocomplete), menu de navegação e desempenho de carregamento.
     + Inclua casos de teste, cobrindo principais cenários positivos e negativos, incluindo testes funcionais, de usabilidade e de desempenho.
     + Documente os casos de teste em uma ferramenta como TestRail, Excel ou outro formato estruturado.
     + Identifique quais casos podem ser automatizados e justifique sua escolha.
2. Automação de Testes:
   * Implemente a automação des casos de teste utilizando uma linguagem de **JAVA** e framework de automação de sua escolha (por exemplo, Selenium, Cypress, Cucumber).
     + Certifique-se de que os testes automatizados sejam robustos, reutilizáveis e que gerem relatórios claros.

# APRESENTAÇÃO FINAL

# Apresentação da Suite de Testes

# Foi elaborado uma Suite de Testes, utilizando a ferramenta Excel, onde podemos encotrar:

# Abas de controle de versionamento do documento e alterações realizadas.

# 

# 

# Aba de apresentação gráfica dos índices obtidos.

# 

# Aba de definição dos Cenários a serem cobertos pela Suite de Teste

# 

# Abas com os Roteiros de Teste a serem executados para 03 cliclos de testes.

# 

# Aba para registro de ocorrências

# 

1. Visão geral das funcionalidades testadas.

A Suite de Teste foi elaborada para atender as funcionalidades:

* Autocomplete: campo de pesquisa da página principal da Amazon.com.br
* Responsividade: verificar o comportamento da página quanto a apresentação e funcionalidades quando submetida a diferentes tipos de devices.
* Carregamento e Desempenho: verificar principalmente quanto ao tempo de resposta de quando a página submetida à diferentes navegadores e diferentes tipos de devices.

1. Estratégia de teste adotada.

Testes de aplicações web são essenciais para garantir que um site ou aplicativo web funcione corretamente em diferentes navegadores, dispositivos e condições de uso.

Alguns tipos comuns de testes realizados em aplicações web:

* **Testes Funcionais:** Verificam se as funcionalidades da aplicação estão funcionando conforme o esperado. Isso inclui testes de formulários, links, botões e outras interações do usuário.
* **Testes de Usabilidade:** Avaliam a experiência do usuário, verificando se a interface é intuitiva e fácil de usar.
* **Testes de Compatibilidade:** Garantem que a aplicação funcione corretamente em diferentes navegadores (Chrome, Firefox, Safari, etc.) e dispositivos (desktop, tablet, smartphone).
* **Testes Automatizados:** Utilizam ferramentas e scripts para automatizar a execução de testes, aumentando a eficiência e a cobertura dos testes.

1. Resultados dos testes e principais defeitos encontrados.

Os resultados durante os testes web são fundamentais para o processo de desenvolvimento e manutenção.

* **Relatórios de Teste**: Após a execução dos testes, é comum gerar relatórios que documentam os resultados. Esses relatórios incluem informações sobre quais testes foram realizados, quais passaram, quais falharam e detalhes sobre os defeitos encontrados.
* **Métricas de Qualidade**: Os resultados dos testes podem ser usados para calcular métricas de qualidade, como a taxa de falhas, a cobertura de testes e o tempo médio para resolver defeitos. Essas métricas ajudam a avaliar a saúde do software.
* **Feedback para a Equipe**: Os resultados dos testes fornecem feedback valioso para a equipe de desenvolvimento. Eles ajudam a identificar áreas problemáticas e a priorizar correções.

Defeitos Encontrados podem ser classificados como:

* **Críticos**: Defeitos que impedem o uso da aplicação ou causam falhas graves.
* **Altos**: Defeitos que afetam funcionalidades importantes, mas que não impedem o uso.
* **Médios**: Defeitos que têm impacto moderado na experiência do usuário.
* **Baixos**: Defeitos menores que não afetam significativamente a funcionalidade.

**Registro de Defeitos**: É importante registrar cada defeito encontrado em um sistema de rastreamento de defeitos (bug tracker). O registro deve incluir detalhes como descrição do defeito, passos para reproduzi-lo, ambiente de teste, gravidade e status (aberto, em progresso, resolvido, etc.).

**Ciclo de Vida do Defeito**: Os defeitos passam por um ciclo de vida que inclui identificação, registro, triagem, resolução e verificação. Após a correção, é importante realizar testes de regressão para garantir que a correção não introduziu novos problemas.

**Análise de Causa Raiz**: Após a identificação de defeitos, pode ser útil realizar uma análise de causa raiz para entender por que o defeito ocorreu e como evitar que problemas semelhantes aconteçam no futuro.

Os resultados dos testes e os defeitos encontrados são essenciais para melhorar a qualidade do software e garantir que ele atenda às expectativas dos usuários.

Um processo bem estruturado de testes e gerenciamento de defeitos pode levar a um produto final mais robusto e confiável.

1. Recomendações para melhorar a qualidade da página inicial da Amazon.

Melhorar a qualidade de uma página web envolve uma combinação de boas práticas de desenvolvimento, design, testes e otimização.

Algumas recomendações para garantir que a página web possua alta qualidade:

### Design Responsivo

**Compatibilidade com Dispositivos**: Certifique-se de que sua página funcione bem em diferentes dispositivos (desktop, tablet, smartphone) e tamanhos de tela.

**Frameworks de CSS**: Utilize frameworks como Bootstrap ou Foundation para facilitar o design responsivo.

### Otimização de Performance

**Minificação de Recursos**: Minifique arquivos CSS, JavaScript e HTML para reduzir o tamanho e melhorar o tempo de carregamento.

**Imagens Otimizadas**: Use formatos de imagem adequados (como WebP) e comprima imagens para reduzir o tempo de carregamento.

1. **Carregamento Assíncrono**: Carregue scripts JavaScript de forma assíncrona para não bloquear o carregamento da página.

### Acessibilidade

**Padrões de Acessibilidade**: Siga as diretrizes WCAG (Web Content Accessibility Guidelines) para garantir que sua página seja acessível a todos os usuários, incluindo aqueles com deficiências.

**Uso de ARIA**: Utilize atributos ARIA para melhorar a acessibilidade de componentes interativos.

### SEO (Otimização para Motores de Busca)

**Meta Tags**: Use meta tags apropriadas, como título e descrição, para melhorar a indexação nos motores de busca.

**URLs Amigáveis**: Crie URLs que sejam descritivas e fáceis de entender.

**Conteúdo de Qualidade**: Produza conteúdo relevante e de alta qualidade que atenda às necessidades dos usuários.

### Testes Rigorosos

**Testes Funcionais**: Realize testes para garantir que todas as funcionalidades da página estejam funcionando corretamente.

**Testes de Usabilidade**: Avalie a experiência do usuário e faça ajustes com base no feedback.

**Testes de Performance**: Utilize ferramentas como Google PageSpeed Insights ou GTmetrix para identificar áreas de melhoria.

### Segurança

**HTTPS**: Utilize HTTPS para garantir que a comunicação entre o usuário e o servidor seja segura.

**Validação de Entrada**: Implemente validação de entrada para proteger contra injeções de SQL e outros ataques.

**Atualizações Regulares**: Mantenha todos os componentes da sua página (CMS, plugins, bibliotecas) atualizados para evitar vulnerabilidades.

### Monitoramento e Análise

**Ferramentas de Análise**: Utilize ferramentas como Google Analytics para monitorar o tráfego e o comportamento dos usuários.

**Feedback do Usuário**: Colete feedback dos usuários para identificar áreas de melhoria.

### Documentação e Manutenção

**Documentação Clara**: Mantenha uma documentação clara e atualizada sobre o código e as funcionalidades da página.

**Manutenção Regular**: Realize manutenções regulares para corrigir bugs, atualizar conteúdo e melhorar a performance.

Seguir essas recomendações pode ajudar a melhorar significativamente a qualidade da página web, proporcionando uma melhor experiência para os usuários e aumentando a eficácia do site.

Muito obrigado,

Juliano José Vilar.