Plano de Teste Completo para o site https://www.google.com.br/

Este plano de teste visa garantir a qualidade, funcionalidade, performance, segurança e acessibilidade do site Google Brasil. O plano abrange diferentes aspectos do processo de teste e inclui diversos artefatos para facilitar a execução e análise dos resultados.

1. Plano de Teste Funcional (PDF)

1.1 Casos de Teste

Cenários:

* **Pesquisa:**
    * Pesquisar por termos simples e complexos.
    * Usar operadores de pesquisa avançada.
    * Filtrar resultados por tipo, data, idioma, região.
    * Validar resultados relevantes e precisos.
    * Testar sugestões de pesquisa e autocompletar.
* **Navegação:**
    * Acessar as diferentes seções do site (Imagens, Vídeos, Notícias, etc.).
    * Validar funcionamento de links e menus.
    * Testar navegação em diferentes dispositivos e resoluções.
* **Login/Logout:**
    * Criar uma nova conta Google.
    * Efetuar login e logout com sucesso.
    * Recuperar senha.
* **Configurações:**
    * Personalizar configurações de pesquisa e privacidade.
    * Alterar idioma e região.
* **Outros:**
    * Testar a funcionalidade do Google Maps.
    * Validar o funcionamento do Google Drive.
    * Testar o Google Translate.

Passos de Execução:

1. **Acessar a página específica do site.**
2. **Realizar a ação desejada (pesquisa, login, etc.).**
3. **Verificar se o resultado esperado é apresentado.**
4. **Documentar os resultados e erros encontrados.**

Dados de Teste:

* Termos de pesquisa diversos (simples, complexos, com operadores avançados).
* Dados de login e senha válidos e inválidos.

* Configurações de pesquisa personalizadas.

Critérios de Aceitação:

* Resultados de pesquisa relevantes e precisos.
* Navegação fluida e intuitiva.
* Login e logout com sucesso.
* Configurações personalizadas aplicadas.
* Funcionalidade das outras ferramentas (Maps, Drive, Translate) conforme esperado.

1.2 Escopo do Teste:

* Todas as páginas principais e subpáginas do site Google Brasil.
* Fluxos de usuário principais (pesquisa, login, configuração).
* Funcionalidades do Google Maps, Google Drive, Google Translate.

1.3 Ambientes de Teste:

* Navegadores: Chrome, Firefox, Safari, Edge.
* Dispositivos Móveis: Android, iOS.
* Resoluções de Tela: 1024x768, 1280x800, 1920x1080.

1.4 Riscos e Dependências:

* Mudanças frequentes no site Google podem afetar os casos de teste.
* Dependência de serviços externos (como Google Maps API) pode impactar a performance dos testes.

1.5 Métricas de Sucesso:

* Taxa de defeitos (número de erros encontrados por número de testes executados).
* Cobertura de teste (porcentagem do código testada).
* Tempo médio de resposta do site.
* Taxa de sucesso dos testes automatizados.

2. Projeto de Automação em Cypress

2.1 Estrutura do Projeto

```
codigo_automacao
% % %  cypress
%    % % %  e2e
%    %    % % %  teste_pesquisa.cy.js
%    %    % % %  teste_login.cy.js
%    %    % % %  teste_config.cy.js
```

```
%   % % %  fixtures
%   %   % % %  pesquisa.json
%   %   % % %  login.json
%   % % %  support
%   %   % % %  commands.js
%   %   % % %  index.js
%   % % %  plugins
%       % % %  index.js
% % %  cypress.config.js
% % %  package.json
```

2.2 Código Cypress

* **teste_pesquisa.cy.js:**
    * Realiza pesquisas com termos diversos.
    * Verifica a relevância dos resultados.
    * Testa sugestões de pesquisa e autocompletar.
* **teste_login.cy.js:**
    * Efetua login com dados válidos e inválidos.
    * Valida mensagens de erro.
    * Testa o fluxo de recuperação de senha.
* **teste_config.cy.js:**
    * Altera configurações de pesquisa e privacidade.
    * Verifica a aplicação das configurações.
    * Testa a mudança de idioma e região.

2.3 Cobertura de Teste:

* Abrange casos de borda, casos de erro, cenários com dados dinâmicos.
* Inclui testes de navegação, pesquisa, login, configuração, outros.

2.4 Integração com Allure

* Configuração do Cypress com Allure para gerar relatórios detalhados.

2.5 Configuração de Variáveis de Ambiente

* Variáveis de ambiente para suportar múltiplos ambientes (desenvolvimento, homologação, produção).

3. Instruções de Execução (README.md)

3.1 Configuração do Ambiente:

1. Instalar Node.js.

2. Instalar Cypress: `npm install cypress --save-dev`
3. Instalar Allure: `npm install allure-commandline --save-dev`

3.2 Execução dos Testes:

* Localmente: `npx cypress run`
* Em pipeline CI/CD: `npx cypress run --env environment=staging`

3.3 Interpretação dos Resultados:

* Resultados no terminal: informações sobre testes executados, erros e falhas.
* Dashboard do Cypress: visão geral dos testes, gráficos de performance, falhas.
* Relatórios Allure: relatórios detalhados com screenshots, logs e informações de cada teste.

3.4 Configuração de CI/CD:

* Integração do Cypress em pipeline CI/CD (GitHub Actions, Jenkins) para execução automatizada dos testes.

4. Plano de Testes de Performance (PDF)

4.1 Testes de Carga:

* Usar ferramentas como JMeter, k6.
* Simular tráfego de usuários para testar a performance do site sob carga pesada.
* Monitorar tempo de resposta, throughput, taxa de erros.

4.2 Testes de Estresse:

* Aumentar gradualmente a carga para identificar o ponto de falha do site.
* Avaliar a capacidade de resposta do site em situações de alta demanda.

4.3 Testes de Capacidade:

* Determinar a capacidade máxima do site sem degradação significativa da performance.
* Avaliar o tempo de resposta, throughput e utilização de recursos em diferentes níveis de carga.

4.4 Métricas de Performance:

* Tempo de resposta: tempo que o site leva para responder a uma solicitação.
* Throughput: número de solicitações atendidas por segundo.
* Taxa de erros: porcentagem de solicitações que falham.

5. Plano de Testes de Segurança (PDF)

5.1 Testes de Vulnerabilidade:

* Identificar vulnerabilidades comuns: injeção de SQL, XSS, CSRF, etc.
* Usar ferramentas de análise de código e scanners de vulnerabilidades.

5.2 Teste de Autenticação/Autorização:

* Validar se as políticas de autenticação e autorização estão implementadas corretamente.
* Testar o acesso a recursos restritos com credenciais válidas e inválidas.

5.3 Teste de Penetração:

* Simular ataques maliciosos para avaliar a resistência do site.
* Contratar especialistas em segurança para realizar testes de penetração.

6. Plano de Testes de Acessibilidade (PDF)

6.1 Conformidade com WCAG:

* Testar o site contra as diretrizes WCAG para garantir a acessibilidade para todos os usuários.

6.2 Ferramentas de Acessibilidade:

* Usar ferramentas como Axe, Lighthouse, NVDA para realizar testes de acessibilidade.

6.3 Casos de Teste Acessíveis:

* Validar a navegação com teclado, uso de leitores de tela, acessibilidade para pessoas com deficiências visuais e motoras.

7. Estratégia de Teste de Regressão

7.1 Seleção de Casos de Teste:

* Identificar casos de teste chave que representam as funcionalidades principais do site.
* Priorizar testes de alta criticidade e que abranjam os fluxos de usuário mais importantes.

7.2 Automação de Regressão:

* Criar scripts automatizados para executar os testes de regressão.

* Utilizar Cypress para automatizar os testes de regressão.

7.3 Planejamento de Execução:

* Executar testes de regressão em cada build do site.
* Definir a frequência de execução (diariamente, em cada pull request).

8. Documentação de Integração de APIs

* Se o site usa APIs, incluir documentação de teste para essas APIs.

8.1 Testes de API Automatizados:

* Criar casos de teste para validar as respostas de APIs, usando ferramentas como Postman ou Newman.

8.2 Testes de Contrato:

* Garantir que as APIs respeitam os contratos esperados (por exemplo, usando Pact).

8.3 Testes de Carga em APIs:

* Realizar testes de carga especificamente para as APIs, usando ferramentas como k6.

Organização dos Artefatos:

* Todos os arquivos e pastas serão organizados de forma clara e acessível.
* Relatórios e artefatos serão armazenados em locais facilmente acessíveis para revisão posterior.

Observação: Este plano de teste é um exemplo e pode ser adaptado às necessidades específicas do projeto. É fundamental realizar uma análise detalhada do site e de seus requisitos para garantir uma cobertura completa dos testes.