

## ## Plano de Teste Completo para o Site do Facebook

Este plano de teste abrangente visa garantir a qualidade, performance, segurança e acessibilidade do site do Facebook, além de cobrir aspectos de teste funcional e de regressão. O plano foi estruturado para facilitar a organização, execução e manutenção dos testes, e inclui artefatos e instruções detalhadas.

### \*\*1. Plano de Teste Funcional (PDF)\*\*

#### \*\*1.1 Casos de Teste\*\*

##### \*\*Funcionalidade | Cenário | Passos | Dados de Teste | Critérios de Aceitação\*\*

---|---|---|---|---

**\*\*Criação de Conta\*\*** | Usuário novo | 1. Acessar página de registro. 2. Preencher campos obrigatórios (nome, email, senha). 3. Confirmar registro. | Nome: "Teste", Email: "teste@email.com", Senha: "senha123" | 1. Mensagem de sucesso após registro. 2. Acesso à conta criada.

**\*\*Login\*\*** | Usuário existente | 1. Acessar página de login. 2. Digitar email e senha. 3. Clicar em "Entrar". | Email: "teste@email.com", Senha: "senha123" | 1. Redirecionamento para a página inicial do usuário. 2. Visualização do nome do usuário no menu.

**\*\*Postagem\*\*** | Criar postagem | 1. Acessar feed de notícias. 2. Clicar em "Criar Postagem". 3. Digitar texto e/ou adicionar imagens. 4. Publicar postagem. | Texto: "Teste de postagem" | 1. Postagem publicada no feed de notícias. 2. Possibilidade de editar e excluir postagem.

**\*\*Curtidas e Compartilhamentos\*\*** | Interagir com postagens | 1. Acessar postagem. 2. Clicar em "Curtir" ou "Compartilhar". | N/A | 1. Contagem de curtidas e compartilhamentos atualizada. 2. Opção de descurtir ou descompartilhar.

**\*\*Mensagens\*\*** | Enviar mensagens | 1. Acessar inbox. 2. Selecionar contato. 3. Escrever mensagem e enviar. | Mensagem: "Olá, tudo bem?" | 1. Mensagem enviada para o contato selecionado. 2. Possibilidade de responder à mensagem.

**\*\*Grupos\*\*** | Criar e gerenciar grupos | 1. Acessar página de grupos. 2. Clicar em "Criar Grupo". 3. Definir nome, descrição e membros. | Nome: "Grupo de Teste", Descrição: "Grupo para testes" | 1. Grupo criado e visível na lista de grupos. 2. Possibilidade de adicionar e remover membros.

**\*\*Eventos\*\*** | Criar e gerenciar eventos | 1. Acessar página de eventos. 2. Clicar em "Criar Evento". 3. Definir nome, data, horário e local. | Nome: "Evento de Teste", Data: "2024-03-01", Horário: "10:00" | 1. Evento criado e visível na lista de eventos. 2. Possibilidade de editar e cancelar eventos.

**\*\*Busca\*\*** | Encontrar pessoas, grupos e páginas | 1. Acessar barra de busca. 2. Digitar termo de busca. | Termo: "Teste" | 1. Resultados relevantes exibidos na página de busca. 2. Possibilidade de filtrar resultados.

**\*\*Configurações\*\*** | Alterar configurações de perfil | 1. Acessar página de configurações. 2. Alterar informações pessoais, privacidade, notificações etc. | N/A | 1. Configurações atualizadas de acordo com as alterações efetuadas. 2. Alterações salvas com sucesso.

**\*\*Segurança\*\*** | Alterar senha e ativar autenticação de dois fatores | 1. Acessar página de segurança. 2. Alterar senha ou ativar autenticação de dois fatores. | N/A | 1. Senha alterada com sucesso. 2. Autenticação de dois fatores configurada corretamente.

**\*\*Ajuda\*\*** | Acessar FAQ e suporte ao cliente | 1. Acessar página de ajuda. 2. Consultar FAQs ou entrar em contato com o suporte. | N/A | 1. Informações relevantes disponíveis na página de ajuda. 2. Resposta do suporte dentro do tempo previsto.

#### **\*\*1.2 Critérios de Aceitação\*\***

- **\*\*Funcionalidade\*\***: Todas as funcionalidades devem funcionar como esperado, sem erros ou comportamentos inesperados.
- **\*\*Usabilidade\*\***: A interface do site deve ser intuitiva e fácil de usar.
- **\*\*Performance\*\***: O site deve carregar rapidamente e responder com agilidade às interações do usuário.
- **\*\*Segurança\*\***: O site deve estar protegido contra ataques maliciosos e garantir a privacidade dos dados do usuário.
- **\*\*Acessibilidade\*\***: O site deve ser acessível para todos os usuários, incluindo pessoas com deficiência.

#### **\*\*1.3 Escopo do Teste\*\***

O escopo dos testes abrange todas as funcionalidades principais do site do Facebook, incluindo:

- Criação de conta e login
- Navegação no feed de notícias
- Postagem de conteúdo
- Interação com postagens (curtidas, compartilhamentos, comentários)
- Mensagens privadas
- Grupos
- Eventos
- Busca
- Configurações de perfil
- Segurança
- Ajuda

#### **\*\*1.4 Ambientes de Teste\*\***

Os testes serão realizados em diversos ambientes, incluindo:

- **\*\*Navegadores\*\***: Google Chrome, Mozilla Firefox, Safari, Microsoft Edge
- **\*\*Dispositivos móveis\*\***: Smartphones e tablets com diferentes sistemas operacionais (Android, iOS)
- **\*\*Resoluções de tela\*\***: Diferentes tamanhos e resoluções de tela
- **\*\*Plataformas\*\***: Windows, macOS, Linux

## **\*\*1.5 Riscos e Dependências\*\***

- **\*\*Disponibilidade do ambiente de teste\*\***: A indisponibilidade do ambiente de teste pode atrasar a execução dos testes.
- **\*\*Dependências de terceiros\*\***: Falhas em APIs ou serviços de terceiros podem impactar a funcionalidade do site.
- **\*\*Mudanças frequentes no site\*\***: A frequência de mudanças no site pode exigir adaptações nos casos de teste e scripts de automação.

## **\*\*1.6 Métricas de Sucesso\*\***

- **\*\*Taxa de defeitos\*\***: O número de erros encontrados durante os testes.
- **\*\*Cobertura de teste\*\***: A porcentagem de código ou funcionalidades cobertas pelos testes.
- **\*\*Tempo de resposta\*\***: O tempo que o site leva para responder às solicitações do usuário.
- **\*\*Taxa de sucesso\*\***: A porcentagem de testes que foram executados com sucesso.

## **\*\*2. Projeto de Automação em Cypress\*\***

### **\*\*2.1 Estrutura do Projeto\*\***

```
...
codigo_automacao/
% % % cypress/
% % % integration/
% % % login.spec.js
% % % postagem.spec.js
% % % grupos.spec.js
% % % etc...
% % % support/
% % % commands.js
% % % index.js
% % % plugins/
% % % index.js
% % % fixtures/
% % % usuarios.json
% % % postagens.json
% % % .env
...
```

### **\*\*2.2 Código Cypress\*\***

**\*\*login.spec.js\*\***

```
````javascript
```

```
describe('Login', () => {
  it('Deve logar no Facebook com sucesso', () => {
    cy.visit('/');
    cy.get('#email').type('teste@email.com');
    cy.get('#pass').type('senha123');
    cy.get('#loginbutton').click();
    cy.url().should('include', '/home');
    cy.get('.user-name').should('contain', 'Teste');
  });
});
```

```

**\*\*postagem.spec.js\*\***

```
```javascript
describe('Postagem', () => {
  it('Deve criar uma postagem com sucesso', () => {
    cy.visit('/home');
    cy.get('#create-post').click();
    cy.get('#post-text').type('Teste de postagem');
    cy.get('#post-submit').click();
    cy.get('.post-text').should('contain', 'Teste de postagem');
  });
});
```

```

## **\*\*2.3 Cobertura de Teste\*\***

Os testes Cypress abrangem casos de borda, casos de erro, e cenários com dados dinâmicos, garantindo uma cobertura abrangente das funcionalidades do site.

## **\*\*2.4 Integração com Allure\*\***

A configuração para integrar o Cypress com Allure é incluída no arquivo `cypress.config.js`.

```
```javascript
module.exports = defineConfig({
  e2e: {
    setupNodeEvents(on, config) {
      // implement node event listeners here
      require('@cypress/allure-plugin/support')(on);
    },
    baseUrl: 'https://www.facebook.com/',
  },
});
```

```

...

## **\*\*2.5 Configurações de Variáveis de Ambiente\*\***

Variáveis de ambiente são configuradas no arquivo `.env` para suportar múltiplos ambientes (desenvolvimento, homologação, produção).

...

```
environment=development
baseUrl=https://www.facebook.com/
...
```

## **\*\*3. Instruções de Execução (README.md)\*\***

### **\*\*3.1 Configuração do Ambiente\*\***

1. Instalar Node.js: <https://nodejs.org/>
2. Instalar Cypress: ``npm install cypress``
3. Configurar variáveis de ambiente: ``cp .env.example .env`` e modificar as variáveis de acordo com o ambiente desejado.

### **\*\*3.2 Execução dos Testes\*\***

- Executar todos os testes: ``npx cypress run``
- Executar testes específicos: ``npx cypress run --spec cypress/integration/login.spec.js``
- Executar testes em um ambiente específico: ``npx cypress run --env environment=staging``

### **\*\*3.3 Interpretação dos Resultados\*\***

- Resultados no terminal: Verificar mensagens de sucesso ou falha, e erros específicos.
- Dashboard do Cypress: Visualizar resultados detalhados de cada teste, incluindo screenshots e logs.
- Relatórios Allure: Gerar relatórios avançados com informações detalhadas sobre os testes, incluindo gráficos e tabelas.

### **\*\*3.4 Configuração de CI/CD\*\***

- Integrar os testes Cypress em uma pipeline CI/CD usando GitHub Actions ou Jenkins.
- Configurar o pipeline para executar os testes a cada build e gerar relatórios Allure.

## **\*\*4. Plano de Testes de Performance (PDF)\*\***

### **\*\*4.1 Testes de Carga\*\***

- Usar ferramentas como JMeter ou k6 para simular tráfego pesado no site.

- Definir cenários de carga com diferentes números de usuários e taxas de requisição.
- Monitorar indicadores de performance como tempo de resposta, throughput e taxa de erros.

#### **\*\*4.2 Testes de Estresse\*\***

- Submeter o site a um alto nível de carga para identificar o ponto de falha.
- Aumentar gradualmente a carga até que o site comece a apresentar problemas de performance.
- Analisar os resultados para identificar os pontos fracos do sistema.

#### **\*\*4.3 Testes de Capacidade\*\***

- Determinar a capacidade máxima do site em termos de número de usuários simultâneos.
- Executar testes de carga com diferentes números de usuários até que a performance do site comece a degradar significativamente.
- Ajustar os recursos do servidor para otimizar a capacidade do site.

#### **\*\*4.4 Métricas de Performance\*\***

- **\*\*Tempo de resposta\*\***: O tempo que o site leva para responder às solicitações do usuário.
- **\*\*Throughput\*\***: O número de requisições por segundo que o site consegue processar.
- **\*\*Taxa de erros\*\***: A porcentagem de requisições que resultam em erros.

### **\*\*5. Plano de Testes de Segurança (PDF)\*\***

#### **\*\*5.1 Testes de Vulnerabilidade\*\***

- Identificar vulnerabilidades comuns como injeção de SQL, XSS e CSRF.
- Usar ferramentas como Burp Suite ou ZAP para realizar testes de vulnerabilidade.
- Verificar a existência de falhas de segurança e corrigir as vulnerabilidades encontradas.

#### **\*\*5.2 Teste de Autenticação/Autorização\*\***

- Verificar se as políticas de autenticação e autorização estão implementadas corretamente.
- Simular diferentes tipos de ataques como brute force e phishing para testar a robustez do sistema de autenticação.
- Assegurar que apenas usuários autorizados tenham acesso às áreas restritas do site.

#### **\*\*5.3 Teste de Penetração\*\***

- Contratar especialistas em segurança para realizar testes de penetração.

- Simular ataques reais para avaliar a resistência do site a ataques maliciosos.
- Identificar vulnerabilidades que podem ser exploradas por hackers.

## **\*\*6. Plano de Testes de Acessibilidade (PDF)\*\***

### **\*\*6.1 Conformidade com WCAG\*\***

- Testar o site contra as diretrizes WCAG (Web Content Accessibility Guidelines) para garantir a acessibilidade para todos os usuários.
- Usar ferramentas como Axe, Lighthouse e NVDA para realizar testes de acessibilidade.

### **\*\*6.2 Ferramentas de Acessibilidade\*\***

- **\*\*Axe\*\***: Extensão de navegador que identifica problemas de acessibilidade no código HTML.
- **\*\*Lighthouse\*\***: Ferramenta que analisa a performance, acessibilidade, SEO e segurança do site.
- **\*\*NVDA\*\***: Leitor de tela que permite que pessoas cegas ou com baixa visão naveguem na web.

### **\*\*6.3 Casos de Teste Acessíveis\*\***

- **\*\*Navegação com teclado\*\***: Verificar se todas as funcionalidades do site podem ser acessadas utilizando apenas o teclado.
- **\*\*Uso de leitores de tela\*\***: Testar a compatibilidade do site com leitores de tela como NVDA.
- **\*\*Acessibilidade para pessoas com deficiências visuais\*\***: Verificar se o site possui contraste de cores adequado, legendas e textos alternativos para imagens.
- **\*\*Acessibilidade para pessoas com deficiências motoras\*\***: Testar a usabilidade do site para pessoas que têm dificuldades de usar o mouse, como a possibilidade de usar o teclado para clicar em links e botões.

## **\*\*7. Estratégia de Teste de Regressão\*\***

### **\*\*7.1 Seleção de Casos de Teste\*\***

- Priorizar casos de teste que cobrem funcionalidades críticas do site.
- Incluir casos de teste que foram afetados por mudanças recentes no código.

### **\*\*7.2 Automação de Regressão\*\***

- Criar scripts de regressão automatizados usando Cypress ou outras ferramentas de automação de testes.
- Executar os scripts de regressão a cada build para garantir que as novas alterações não quebrem funcionalidades existentes.

### **\*\*7.3 Planejamento de Execução\*\***

- Executar testes de regressão diariamente ou a cada pull request.
- Ajustar a frequência de execução dos testes de acordo com a frequência de mudanças no código do site.

### **\*\*8. Documentação de Integração de APIs\*\***

#### **\*\*8.1 Testes de API Automatizados\*\***

- Criar casos de teste para validar as respostas de APIs usando ferramentas como Postman ou Newman.
- Verificar se as APIs retornam os dados esperados e em formato válido.
- Validar os códigos de status das respostas da API.

#### **\*\*8.2 Testes de Contrato\*\***

- Definir como garantir que as APIs respeitam os contratos esperados usando Pact.
- Criar contratos para cada API que define os dados de entrada e saída esperados.
- Executar testes de contrato para verificar se as APIs estão em conformidade com os contratos.

#### **\*\*8.3 Testes de Carga em APIs\*\***

- Realizar testes de carga especificamente para as APIs usando ferramentas como k6.
- Simular tráfego pesado nas APIs para verificar se elas podem suportar o volume de requisições esperado.
- Monitorar indicadores de performance como tempo de resposta e taxa de erros.

### **\*\*Organização e Armazenamento\*\***

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

### **\*\*Conclusão\*\***

Este plano de teste completo e abrangente visa garantir a qualidade, performance, segurança e acessibilidade do site do Facebook. Ao seguir este plano, os testes serão realizados de forma eficiente e abrangente, resultando em um site de alta qualidade e confiabilidade.

**\*\*Observação\*\*:** Este plano de teste é apenas um exemplo e pode ser adaptado de acordo com as necessidades específicas do site do Facebook. É importante realizar uma análise detalhada do site e de suas funcionalidades para garantir que o plano de



teste atenda aos requisitos específicos do projeto.