

Plano de Teste Completo para o Instagram

Este plano de teste abrangente foi desenvolvido para garantir a qualidade e funcionalidade do site do Instagram, considerando diversos aspectos, desde funcionalidade básica até testes de segurança e acessibilidade.

1. Plano de Teste Funcional (PDF).

1.1. Casos de Teste:

| **Caso de Teste** | **Cenário** | **Passos de Execução** | **Dados de Teste** |
Critérios de Aceitação |

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

| **CT-01 - Cadastro de Usuário** | Criar uma nova conta no Instagram. | 1. Acessar a página de cadastro. 2. Preencher os campos obrigatórios (nome de usuário, email, senha). 3. Confirmar a senha. 4. Clicar em "Criar conta". | Nome de usuário válido, email válido, senha segura. | O usuário é direcionado para a página inicial, com a conta criada e login automático. |

| **CT-02 - Login de Usuário** | Acessar a conta do usuário. | 1. Acessar a página de login. 2. Inserir o nome de usuário ou email. 3. Inserir a senha. 4. Clicar em "Entrar". | Nome de usuário/email válido, senha correta. | O usuário é direcionado para a página inicial, com a conta logada. |

| **CT-03 - Publicar Post** | Criar um novo post com texto, imagem e vídeo. | 1. Acessar a página de publicação. 2. Adicionar texto, imagem ou vídeo. 3. Escolher filtros e adicionar localização. 4. Clicar em "Compartilhar". | Texto, imagem ou vídeo válidos. | O post é publicado com sucesso na página do usuário. |

| **CT-04 - Curtir e Comentar Post** | Interagir com posts de outros usuários. | 1. Acessar um post de outro usuário. 2. Clicar no botão de "curtir". 3. Clicar no campo de comentário e escrever um comentário. 4. Clicar em "Enviar". | Comentário válido. | O post é curtido e o comentário é adicionado com sucesso. |

| **CT-05 - Seguir Usuário** | Seguir outros usuários. | 1. Acessar o perfil de um usuário. 2. Clicar em "Seguir". | Usuário válido. | O usuário é seguido com sucesso. |

| **CT-06 - Explorar** | Navegar pela página "Explorar". | 1. Acessar a página "Explorar". 2. Buscar por hashtags ou temas. 3. Navegar pelos posts. | Hashtags ou temas válidos. | A página "Explorar" exibe os posts relevantes à pesquisa. |

| **CT-07 - Mensagens Diretas** | Enviar e receber mensagens diretas. | 1. Acessar a caixa de entrada de mensagens diretas. 2. Selecionar um usuário. 3. Escrever uma mensagem. 4. Clicar em "Enviar". | Mensagem válida, usuário válido. | A mensagem é enviada com sucesso para o usuário. |

| **CT-08 - Histórias** | Criar, visualizar e interagir com histórias. | 1. Acessar a câmera para criar uma história. 2. Gravar um vídeo ou tirar uma foto. 3. Adicionar texto, filtros e figurinhas. 4. Clicar em "Compartilhar". 5. Visualizar histórias de outros usuários. 6. Interagir com as histórias (enviar mensagens, emojis, etc.). | Vídeo ou foto válidos. | A história é criada e publicada com sucesso, e as interações são registradas. |

| ****CT-09 - Reels**** | Criar, visualizar e interagir com Reels. | 1. Acessar a página de criação de Reels. 2. Gravar um vídeo curto. 3. Adicionar música, filtros e efeitos. 4. Clicar em "Compartilhar". 5. Visualizar Reels de outros usuários. 6. Interagir com os Reels (curtir, comentar, salvar). | Vídeo válido, música válida. | O Reel é criado e publicado com sucesso, e as interações são registradas. |

| ****CT-10 - Lives**** | Criar e visualizar lives. | 1. Acessar a câmera para iniciar uma live. 2. Escolher o título da live. 3. Clicar em "Iniciar live". 4. Visualizar lives de outros usuários. | Título válido. | A live é iniciada e exibida com sucesso. |

| ****CT-11 - Configurações de Perfil**** | Gerenciar as configurações de perfil. | 1. Acessar o menu de configurações do perfil. 2. Alterar foto de perfil, nome de usuário, biografia, etc. 3. Alterar as configurações de privacidade. 4. Gerenciar as notificações. | Dados válidos, configurações válidas. | As alterações são salvas com sucesso e as configurações são atualizadas. |

| ****CT-12 - Pesquisa**** | Pesquisar por usuários, hashtags e posts. | 1. Acessar a barra de pesquisa. 2. Inserir um termo de pesquisa (usuário, hashtag, post). | Termo de pesquisa válido. | Os resultados da pesquisa são relevantes e exibidos com sucesso. |

****1.2. Critérios de Aceitação:****

- * Todos os casos de teste devem ser executados com sucesso sem erros.
- * Os tempos de resposta da aplicação devem ser dentro dos limites aceitáveis.
- * As funcionalidades devem ser implementadas de acordo com os requisitos de design.
- * A experiência do usuário deve ser intuitiva e amigável.

****1.3. Escopo do Teste:****

* ****Funcionalidades:**** Cadastro de usuário, login de usuário, publicação de posts, curtir e comentar posts, seguir usuário, explorar, mensagens diretas, histórias, reels, lives, configurações de perfil, pesquisa.

* ****Páginas:**** Página de login, página de cadastro, página inicial, perfil do usuário, página de publicação, página "Explorar", caixa de entrada de mensagens diretas, página de criação de histórias, página de criação de reels, página de lives, menu de configurações de perfil, barra de pesquisa.

* ****Fluxos de Usuário:**** Cadastro e login, publicação de posts e interação, navegação pela página "Explorar", comunicação por mensagens diretas, criação e visualização de histórias e reels, gerenciamento de configurações de perfil.

****1.4. Ambientes de Teste:****

* ****Navegadores:**** Chrome, Firefox, Safari, Edge

* ****Dispositivos Móveis:**** Android, iOS

* ****Resoluções de Tela:**** 1024x768, 1280x1024, 1920x1080, 2560x1440

****1.5. Riscos e Dependências:****

* ****Riscos:**** Dependência de APIs de terceiros, integração com outros serviços,

mudanças frequentes nos requisitos.

* **Dependências:** Equipe de desenvolvimento, equipe de design, infraestrutura de testes.

1.6. Métricas de Sucesso:

* Taxa de defeitos (número de defeitos encontrados dividido pelo número total de casos de teste).

* Cobertura de teste (porcentagem de funcionalidades e páginas cobertas pelos testes).

* Tempo de resposta da aplicação (tempo médio para carregar a página inicial e outras funcionalidades).

* Taxa de sucesso dos testes automatizados (porcentagem de testes automatizados que passam).

2. Projeto de Automação em Cypress:

2.1. Estrutura do Projeto:

...

```
codigo_automacao/  
% % % cypress/  
%   % % % integration/  
%   %   % % % cadastro.cy.js  
%   %   % % % login.cy.js  
%   %   % % % publicar-post.cy.js  
%   %   % % % ...  
%   % % % support/  
%   %   % % % commands.js  
%   % % % plugins/  
%   %   % % % index.js  
%   % % % fixtures/  
%   %   % % % usuarios.json  
%   % % % e2e/  
%     % % % ...  
% % % .eslintrc.js  
% % % cypress.config.js  
...
```

2.2. Código Cypress:

```
``javascript  
// cadastro.cy.js  
describe('Cadastro de Usuário', () => {  
  it('deve permitir cadastrar um novo usuário', () => {  
    cy.visit('/accounts/emailsignup/');  
  });  
});
```

```

    cy.get('#email').type('usuario@email.com');
    cy.get('#username').type('username');
    cy.get('#password').type('senha123');
    cy.get('#password_confirm').type('senha123');
    cy.get('#signup').click();
    cy.url().should('include', '/accounts/login/');
    // ... validações adicionais
  });
});

// login.cy.js
describe('Login de Usuário', () => {
  it('deve permitir logar um usuário existente', () => {
    cy.visit('/accounts/login/');
    cy.get('#username').type('username');
    cy.get('#password').type('senha123');
    cy.get('#login').click();
    cy.url().should('include', '/');
    // ... validações adicionais
  });
});

// ... outros arquivos de teste
...

```

****2.3. Cobertura de Teste:****

- * Casos de borda, como campos vazios, formatos inválidos de dados, senhas fracas, etc.
- * Casos de erro, como erros de login, erros de publicação de posts, erros de conexão, etc.
- * Cenários com dados dinâmicos, como pesquisa por hashtags, interações com stories, etc.

****2.4. Integração com Allure:****

- * Instalar o plugin `cypress-allure-plugin`: `npm install cypress-allure-plugin`
- * Configurar o plugin no arquivo `cypress.config.js`.

****2.5. Configuração de Variáveis de Ambiente:****

- * Definir variáveis de ambiente para cada ambiente (desenvolvimento, homologação, produção).
- * Acessar as variáveis de ambiente no código Cypress usando `Cypress.env().variavel`.

****3. Instruções de Execução (README.md):****

****3.1. Configuração do Ambiente:****

- * Instalar Node.js: [Link para instruções de instalação](https://nodejs.org/)
- * Instalar Cypress: `npm install cypress`
- * Instalar dependências: `npm install`

****3.2. Execução dos Testes:****

- * Executar os testes localmente: `npx cypress run`
- * Executar os testes em um pipeline CI/CD: `npx cypress run --env environment=staging`

****3.3. Interpretação dos Resultados:****

- * Visualizar os resultados no terminal.
- * Visualizar os relatórios do Cypress no dashboard do Cypress.
- * Visualizar os relatórios do Allure no dashboard do Allure.

****3.4. Configuração de CI/CD:****

- * Configurar o Cypress em um pipeline CI/CD (por exemplo, usando GitHub Actions ou Jenkins).

****4. Plano de Testes de Performance (PDF):****

****4.1. Testes de Carga:****

- * ****Ferramenta:**** JMeter, k6
- * ****Cenário:**** Simular um grande número de usuários acessando o site simultaneamente.
- * ****Métricas:**** Tempo de resposta, throughput, taxa de erros.

****4.2. Testes de Estresse:****

- * ****Cenário:**** Simular uma carga ainda maior do que nos testes de carga, até atingir o ponto de falha do site.
- * ****Métricas:**** Tempo de resposta, throughput, taxa de erros, tempo de recuperação.

****4.3. Testes de Capacidade:****

- * ****Cenário:**** Determinar a capacidade máxima do site em termos de número de usuários simultâneos.
- * ****Métricas:**** Tempo de resposta, throughput, taxa de erros.

****4.4. Métricas de Performance:****

- * Tempo de resposta: tempo que a aplicação leva para responder a uma solicitação.
- * Throughput: número de solicitações que a aplicação pode atender por segundo.
- * Taxa de erros: porcentagem de solicitações que falham.

****5. Plano de Testes de Segurança (PDF):****

****5.1. Testes de Vulnerabilidade:****

- * ****Injeção de SQL:**** Inserir código SQL malicioso para explorar vulnerabilidades no banco de dados.
- * ****XSS (Cross-Site Scripting):**** Injetar código JavaScript malicioso para roubar informações do usuário.
- * ****CSRF (Cross-Site Request Forgery):**** Forçar um usuário a executar ações não autorizadas.
- * ****Outras vulnerabilidades:**** Falhas de autenticação, gerenciamento de sessões, etc.

****5.2. Testes de Autenticação/Autorização:****

- * Validar se as credenciais do usuário são verificadas corretamente.
- * Validar se as permissões do usuário são verificadas corretamente.

****5.3. Teste de Penetração:****

- * Simular ataques maliciosos para identificar vulnerabilidades e falhas de segurança.
- * ****Ferramentas:**** Burp Suite, Metasploit

****6. Plano de Testes de Acessibilidade (PDF):****

****6.1. Conformidade com WCAG:****

- * ****Ferramentas:**** Axe, Lighthouse, NVDA
- * ****Diretrizes:**** WCAG 2.1 (níveis A, AA, AAA)
- * Validar se o site atende aos critérios de acessibilidade para todos os usuários.

****6.2. Casos de Teste Acessíveis:****

- * Navegação com teclado: navegar pelo site usando apenas o teclado.
- * Uso de leitores de tela: garantir que o conteúdo seja legível por leitores de tela.
- * Acessibilidade para pessoas com deficiências visuais: verificar se as cores e o contraste são adequados.
- * Acessibilidade para pessoas com deficiências motoras: verificar se as interações são fáceis de usar.

****7. Estratégia de Teste de Regressão:****

****7.1. Seleção de Casos de Teste:****

- * Casos de teste críticos, que são essenciais para o funcionamento do site.
- * Casos de teste que foram afetados por alterações recentes.
- * Casos de teste que cobrem funcionalidades frequentemente usadas.

****7.2. Automação de Regressão:****

- * Criar scripts de teste automatizados para executar os casos de teste de regressão.
- * ****Ferramentas:**** Cypress, Selenium, Appium

****7.3. Planejamento de Execução:****

- * Executar os testes de regressão diariamente ou a cada build.
- * Integrar os testes de regressão com o pipeline CI/CD.

****8. Documentação de Integração de APIs:****

****8.1. Testes de API Automatizados:****

- * ****Ferramentas:**** Postman, Newman, RestAssured
- * Criar casos de teste para validar as respostas das APIs.
- * Validar os códigos de status, os cabeçalhos, os dados e a formatação das respostas.

****8.2. Testes de Contrato:****

- * ****Ferramentas:**** Pact, Spring Cloud Contract
- * Garantir que as APIs respeitem os contratos esperados.

****8.3. Testes de Carga em APIs:****

- * ****Ferramentas:**** k6, Gatling
- * Simular um grande número de solicitações para as APIs.
- * Validar a performance das APIs sob carga.

****Organização:****

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

Este plano de teste fornece uma base sólida para garantir a qualidade e funcionalidade do site do Instagram. É importante lembrar que o plano deve ser atualizado e adaptado conforme os requisitos do projeto evoluem.