

ESTRUTURA DE TESTES - [NOME PROJETO]

Feito por [Nome]

versão x.x

Data de criação: **DD.MM.AAAA**

Atualizado em: **DD.MM.AAAA**

Este template está licenciado sob a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

Para mais informações, acesse: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Para mais templates, visite: <https://github.com/seu-usuario/QA-Templates>

Este documento descreve a estrutura de testes utilizada no projeto [Nome do Projeto].

Ele contém detalhes sobre os tipos de testes realizados, os objetivos de cada tipo de teste e as ferramentas utilizadas.

Índice

1. Introdução
2. Tipos de Testes
 - 2.1 Testes Funcionais
 - 2.2 Testes de Regressão
 - 2.1 Testes de Usabilidade
 - 2.1 Testes de Segurança
 - 2.1 Testes de Desempenho
 - 2.1 Testes de Integração
 - 2.1 Testes de Compatibilidade
 - 2.1 Testes de Localização e Internacionalização
3. Plano de Execução dos Testes
4. Resultados Esperados
5. Conclusão

Nota: Também é possível apresentar aqui o programa que será testado.

ESTRUTURA DE TESTES - [NOME PROJETO]

Feito por [Nome]

versão x.x

Data de criação: DD.MM.AAAA

Atualizado em: DD.MM.AAAA

Este template está licenciado sob a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

Para mais informações, acesse: <https://creativecommons.org/licenses/by-nc-sa/4.0/>
Para mais templates, visite: <https://github.com/seu-usuario/QA-Templates>

1. INTRODUÇÃO

- **Objetivo dos Testes:** Garantir que o software funcione conforme os requisitos definidos.
- **Escopo:** Definir as funcionalidades do sistema que serão testadas (ex.: login, cadastro, integração com APIs, etc.).
- **Versão do Software:** [Versão atual do sistema]
- **Ambiente de Testes:** Descrição do ambiente de teste (ex.: servidores de desenvolvimento, QA, produção simulada).

2. TIPOS DE TESTES

2.1 Testes Funcionais

- **Objetivo:** Validar se as funcionalidades do sistema atendem aos requisitos especificados.
- **Descrição:** Testes que verificam o comportamento das funcionalidades principais do sistema.

Técnicas de Teste:

- Teste de Caixa Preta
- Teste de Equivalência de Partições
- Teste de Valor Limite
- Teste de Caminhos

Critérios de Aceitação: Definir o que torna uma funcionalidade "aceita" (ex.: retorno esperado, dados corretos na tela).

Nota: Também é possível apresentar aqui o programa que será testado.

ESTRUTURA DE TESTES - [NOME PROJETO]

Feito por [Nome]

versão x.x

Data de criação: **DD.MM.AAAA**

Atualizado em: **DD.MM.AAAA**

Este template está licenciado sob a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

Para mais informações, acesse: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Para mais templates, visite: <https://github.com/seu-usuario/QA-Templates>

2.2 Testes de Regressão

- **Objetivo:** Verificar se alterações no código não impactaram negativamente o sistema existente.
- **Descrição:** Testes realizados após alterações ou novas funcionalidades para garantir que não houve regressão.
- **Escopo:** Funcionalidades que sofreram modificações no código.
- **Critérios de Aceitação:** Todos os testes devem passar sem falhas.

2.3 Testes de Usabilidade

- **Objetivo:** Avaliar a experiência do usuário e garantir que o sistema seja fácil de usar.
- **Descrição:** Testes focados na interação do usuário, incluindo navegação intuitiva, feedback visual e layout.

Metodologia:

- Testes com usuários reais
- Testes A/B
- Questionários de satisfação (ex: NPS)

Critérios de Aceitação: Alta taxa de aceitação do usuário e poucos erros de navegação.

Nota: Também é possível apresentar aqui o programa que será testado.

ESTRUTURA DE TESTES - [NOME PROJETO]

Feito por [Nome]

versão x.x

Data de criação: **DD.MM.AAAA**

Atualizado em: **DD.MM.AAAA**

Este template está licenciado sob a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

Para mais informações, visite <https://creativecommons.org/licenses/by-nc-sa/4.0/>
Para mais templates, visite <https://github.com/sau-usuario/OA-Templates>

2.4 Testes de Segurança

- **Objetivo:** Garantir que o sistema seja seguro e que os dados dos usuários estejam protegidos.
- **Descrição:** Testes focados em identificar falhas de segurança (ex: SQL Injection, Cross-Site Scripting, vazamento de dados).

Técnicas de Teste:

- Teste de Penetração (Pentesting)
- Análise de Código Fonte
- Testes de Autenticação e Autorização

Critérios de Aceitação: Nenhuma vulnerabilidade crítica deve ser encontrada.

2.5 Testes de Desempenho

- **Objetivo:** Avaliar a performance do sistema sob carga e em diferentes condições.
- **Descrição:** Testes para verificar a velocidade de resposta, a escalabilidade e o comportamento sob carga elevada.

Tipos de testes:

- Teste de Carga (Load Testing)
- Teste de Estresse (Stress Testing)
- Teste de Escalabilidade
- Teste de Estabilidade (Soak Testing)

Critérios de Aceitação: O sistema deve atender aos tempos de resposta estabelecidos, sem deterioração significativa.

Além disso, seria interessante acrescentar um exemplo de "tempo de resposta estabelecido" para tornar os critérios mais objetivos.

Nota: Também é possível apresentar aqui o programa que será testado.

ESTRUTURA DE TESTES - [NOME PROJETO]

Feito por [Nome]

versão x.x

Data de criação: **DD.MM.AAAA**

Atualizado em: **DD.MM.AAAA**

Este template está licenciado sob a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

Para mais informações, acesse: <https://creativecommons.org/licenses/by-nc-sa/4.0/>
#Para mais templates, veja: <https://github.com/seu-usuario/QA-Templates>

2.6 Testes de Integração

- **Objetivo:** Validar a integração entre os diferentes módulos do sistema e com sistemas externos.
- **Descrição:** Testes que verificam a comunicação entre os componentes do sistema (bancos de dados, APIs, etc.).

Critérios de Aceitação: Todas as integrações devem funcionar corretamente.

2.7 Testes de Compatibilidade

- **Objetivo:** Garantir que o sistema funcione corretamente em diferentes dispositivos, navegadores e sistemas operacionais.
- **Descrição:** Testes realizados para verificar a compatibilidade do software com diversas versões de navegadores, dispositivos e sistemas operacionais.

Critérios de Aceitação: O sistema deve funcionar como esperado em todas as configurações suportadas.

Nota: Também é possível apresentar aqui o programa que será testado.

ESTRUTURA DE TESTES - [NOME PROJETO]

Feito por [Nome]

versão x.x

Data de criação: DD.MM.AAAA

Atualizado em: DD.MM.AAAA

Este template está licenciado sob a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

Para mais informações, consulte <https://creativecommons.org/licenses/by-nc-sa/4.0/>

2.8 Testes de Localização e Internacionalização

- **Objetivo:** Validar a adequação do sistema para diferentes regiões e idiomas.
- **Descrição:** Testes para garantir que o sistema funcione corretamente em múltiplos idiomas e fusos horários.

CrITÉRIOS de Aceitação: Todos os textos devem ser traduzidos corretamente, sem falhas de exibição ou formatação.

3. PLANO DE EXECUÇÃO DOS TESTES

- **Cronograma de Testes:** Definir o período de execução dos testes e entregas programadas. É importante revisar o cronograma periodicamente para garantir que os testes sejam ajustados conforme necessário, especialmente para testar mudanças de última hora.
- **Responsáveis pelos Testes:** Definir quem é responsável por cada tipo de teste (ex: equipe de QA, desenvolvedores, etc.).

Ferramentas Utilizadas:

- **Automação de Testes:** Selenium, Cypress
- **Segurança:** Burp Suite, OWASP ZAP
- **Desempenho:** JMeter, LoadRunner
- **Gerenciamento de Testes:** TestRail, Jira

Nota: Também é possível apresentar aqui o programa que será testado.

ESTRUTURA DE TESTES - [NOME PROJETO]

Feito por [Nome]

versão x.x

Data de criação: **DD.MM.AAAA**

Atualizado em: **DD.MM.AAAA**

Este template está licenciado sob a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

Para mais informações, acesse: <https://creativecommons.org/licenses/by-nc-sa/4.0/>
#20230928 | github.com/seu-usuario/QA-Templates

4. RESULTADOS ESPERADOS

- **Relatórios de Testes:** Cada teste será documentado com o status (Pass/Fail), logs e evidências (prints de tela, arquivos de log, etc.).
- **Defeitos Encontrados:** Relatórios detalhados de defeitos encontrados, incluindo severidade, impacto no projeto e status de correção.
- **Métricas de Qualidade:** Taxa de cobertura de testes, tempo de resposta, número de testes automatizados, entre outros.

5. CONCLUSÃO

- **Resumo dos Resultados:** Um resumo dos testes realizados, incluindo as descobertas mais importantes e o status do sistema.
- **Recomendações:** Se houver falhas ou áreas de melhoria, recomendações para correção ou melhorias contínuas.

Nota: Também é possível apresentar aqui o programa que será testado.

ESTRUTURA DE TESTES - [NOME PROJETO]

Feito por [Nome]

versão x.x

Data de criação: **DD.MM.AAAA**

Atualizado em: **DD.MM.AAAA**

Este template está licenciado sob a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0).

Para mais informações, acesse: <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Para mais templates, visite: <https://github.com/seu-usuario/QA-Templates>

7. FERRAMENTAS UTILIZADAS

- [Ferramenta 1] para [tipo de teste].
- [Ferramenta 2] para [tipo de teste].

8. AMBIENTES

- Ambiente de desenvolvimento (DEV): [link]
- Ambiente de homologação (HML): [link]
- Ambiente de produção (PRD): [link]

9. DOCUMENTAÇÃO RELACIONADA

Consulte também:

- Requisitos do Sistema: [Pequena descrição e link dos requisitos]
- Cronograma do Projeto: [Pequena descrição e link do cronograma.]
- Guia de Ferramentas: Pequena descrição e link do guia de ferramentas.]

Nota: Também é possível apresentar aqui o programa que será testado.