

Plano de Testes<Notfy>

Versão <1.0>

Histórico de Alterações

Data	Versão	Descrição	Autor
<20/09/21>	<1.0>	<Planejamento de Testes Funcionais e não funcionais>	<Janaina Jamarinor>
<20/09/21>	<1.0>	<Planejamento de Testes de Usabilidade>	<Janaina Jamarinor>

Conteúdo

1	INTRODUÇÃO	3
2	REQUISITOS A TESTAR	3
2.1	ITERAÇÃO 1	3
2.2	ITERAÇÃO 2	3
2.3	ITERAÇÃO 3	3
3	TIPOS DE TESTE	3
3.1	ITERAÇÃO 1	3
3.2	ITERAÇÃO 2	3
4	RECURSOS	3
4.1	AMBIENTE DE TESTE – SOFTWARE & HARDWARE	3
4.2	FERRAMENTAS DE TESTE	3
5	CRONOGRAMA	3
6	REFERÊNCIAS	3

1 Introdução

O fluxo de testes, assim como os demais fluxos, está presente no processo de desenvolvimento de *software* ao longo de todas as suas fases, concentrando-se, no entanto, no planejamento dos testes na iteração inicial e no início de cada nova iteração e, durante as iterações, tendo seu foco no projeto e na execução dos testes, sobretudo nas iterações da fase de Construção.

Este documento descreve os requisitos a testar, os tipos de testes definidos para cada iteração, os recursos de hardware e software a serem empregados e o cronograma dos testes ao longo do projeto. As seções referentes aos requisitos, recursos e cronograma servem para permitir ao gerente do projeto acompanhar a evolução dos testes.

2 o que será testado

Este projeto contempla o planejamento de testes de requisitos funcionais e não funcionais. não obstante a isso, foi decidido tratar os testes de usabilidade e acessibilidade em separado dos outros testes considerados não funcionais.

Isso posto, define-se os seguintes testes à serem realizados:

1. Testes funcionais: Contemplam todos os requisitos funcionais específicos e relacionados às histórias de usuário
2. Testes não-funcionais: Volume, carga, performance, uso multibrowser, responsividade em diferentes aparelhos, segurança (que contemplem a LGPD).
3. Teste de Usabilidade: Contemplam atender às 10 heurísticas de Nielsen

4. Testes de acessibilidade: Contempla os requisitos pré-definidos pelo World Wide Web Consortium (W3C)

** Esta seção contém os requisitos que são objetos dos testes a serem realizados. Esses requisitos são divididos, por iteração, em casos de uso e requisitos não funcionais conforme descrito abaixo.

2.1 Iteração 1

Casos de Uso / Requisitos Funcionais

Identificador do Caso de Uso	Nome do Caso de Uso
<Id UC1>	<Nome UC1>
<Id UC2>	<Nome UC2>
<Id UCN>	<Nome UC3>

Requisitos Não-Funcionais

Identificador do Requisito	Nome do Requisito
<Id Req1>	Funcionamento multibrowser (Chrome, Firefox, Safari, Edge e Opera)
<Id Req2>	Responsividade para Desktop/WEB, Tablet e Dispositivos móveis
<Id Req3>	Carga, Volume e Performance ainda serão definidos após levantamento de todos os requisitos
<Id ReqN>	

Requisitos de Usabilidade

Identificador do Requisito	Nome do Requisito
<Id Req1>	Visibilidade e status do sistema
<Id Req2>	Compatibilidade entre o sistema e o mundo real
<Id Req3>	Controle e liberdade para o usuário
<Id Req4>	Consistência e padronização
<Id Req5>	Prevenção de erros
<Id Req6>	Reconhecimento ao invés de memorização
<Id Req7>	Eficiência e flexibilidade de uso
<Id Req8>	Estética e design minimalista
<Id Req9>	Ajude os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros.
<Id Req10>	Ajuda e documentação
<Id ReqN>	

Requisitos de acessibilidade

Identificador do Requisito	Nome do Requisito
<Id Req1>	Informações perceptíveis e Interface do usuário: 1. Texto alternativo para conteúdo não textual

	<ul style="list-style-type: none"> 2. Legendas e outras alternativas paramultimídia 3. O conteúdo pode ser apresentado de formas diferentes 4. O conteúdo é mais fácil de ver e ouvir
<Id Req2>	Interface de usuário operacional e navegação: <ul style="list-style-type: none"> 1. A funcionalidade está disponível em teclado (atalhos) 2. Os usuários tem tempo suficiente para ler todo o conteúdo 3. O conteúdo não causa convulsões e reações físicas (luminosidade, padrão de cores, pop-ups em cores latejantes...) 4. O usuário pode navegar facilmente, encontrar o conteúdo e determinar onde estão. (volume de clicks para realizar uma operação, consigo navegar previamente entre conteúdo, footer ou algo importante que mereça destaque) 5. Os usuários podem usar diferentes modalidades de entrada além do teclado.
<Id Req3>	Informações compreensíveis e Interface do usuário; <ul style="list-style-type: none"> 1. O texto é compreensível e legível 2. O conteúdo aparece e opera de maneiras previsíveis 3. Os usuários são ajudados à corrigir erros
<Id Req4>	Conteúdo robusto e interação confiável <ul style="list-style-type: none"> 1. O conteúdo é compatível com as ferramentas do usuário atuais e futuras
<Id Req5>	Vlibras está sendo utilizado?
<Id ReqN>	

2.2 Iteração 2

<Esta iteração e demais dentro da seção 2 serão utilizadas incrementalmente para testes de novas versões>

...

3 Tipos de Teste

3.1 Iteração 1

Objetivo:	Nesta iteração serão testados todos os requisitos do sistema para primeira entrega do projeto. Em caso de necessidade de retestes ou de incremento de funcionalidades, novas iterações serão criadas
------------------	--

Técnica:	<input checked="" type="checkbox"/> Manual <input checked="" type="checkbox"/> Automática
Estágio do teste: <input type="checkbox"/> Integração <input type="checkbox"/> Sistema <input type="checkbox"/> Unidade <input checked="" type="checkbox"/> Aceitação	Abordagem do teste <input checked="" type="checkbox"/> Caixa branca <input checked="" type="checkbox"/> Caixa preta
Responsável(is):	Janaina Jamarino

...

3.2 Iteração 2

<Esta iteração e demais dentro da seção 2 serão utilizadas incrementalmente para testes de novas versões>

...

4 Recursos

De extrema importância para o bom andamento dos testes, os recursos a serem utilizados durante os testes são descritos nessa seção. Os recursos estão divididos nas subseções que se seguem.

4.1 Ambiente de Teste – Software & Hardware

Criar-se-há 4 branches para o bom gerenciamento do projeto e realização correta dos testes:

- 1 - Desenvolvimento (Onde os desenvolvedores deverão comitar suas soluções e testarem antes de comitar para que o tester realize os teste de qualidade)
- 2 - Testes - Onde o tester realiza pontualmente
- 3 - Homologação - Onde o tester realiza testes de fumaça para garantir que após o deploy nda se quebrou
- 4 - Produção/ main - Onde asolução pronta e estável é colocada para o usuário final.

4.2 Ferramentas de Teste

Serão utilizadas duas abordagens:

1 - Manual :

- 1.1 - <https://www.browserstack.com/> para testar multibrowser
- 1.2 Git para versionamento
- 1.3 Inspeção dos elementos através do atalho F12 para verificar as requisições HTTP.
- 1.4 Análise de logs que podem ser gravados em arquivo .txt ou utilizar alguma ferramenta de gestão como kibana.

2 - Automatizada

- 2.1 Cypress + Javascript ou Robot Framework + Python (ainda avaliaremos qual a melhor opção)
- 2.2 Se possível implementar integração com Kibana para análise de logs)

5 Cronograma

<Ainda não há um cronograma definido.>

6 Referências

<https://www.w3.org/WAI/roles/testers/>

<https://www.nngroup.com/>