**Plano de Testes**

***Swag Labs***

**Versão 1.0**

Autores:

A Equipe

Recife, 30 de maio de 2025

**Histórico das Revisões**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor(es)** |
| 26/05/2025 | 1.0 | Elaboração do plano de testes. | A Equipe |

**Sumário**

[**1**](#_5pbabjitcv5d) **INTRODUÇÃO 4**

[1.1](#_6a5kg92t3b93) Objetivo desse documento 4

[1.2](#_3g6uy0wjuyu7) Escopo 4

[1.3](#_qwza1eelj0pv) Identificação de projeto 4

[1.4](#_4h4ygfxym61b) Visão geral do documento 4

[**2**](#_iw3q6ccx6kqo) **REQUISITOS A TESTAR 5**

[2.1](#_nntirz3s45w8) Teste do banco de dados 5

[2.2](#_n18ona39kut3) Teste funcional 5

[2.3](#_g0fh1suzhndv) Teste do ciclo de negócios 5

[2.4](#_4jlso3qsalu4) Teste da interface do usuário 5

[2.5](#_wu0njh41gnah) Perfil da *performance* 5

[2.6](#_xgguvm2avfvp) Teste de carga 5

[2.7](#_n5nxw6diw7ot) Teste de stress 5

[2.8](#_nco0b0gz01ud) Teste de volume 6

[2.9](#_pv74xuykwfuz) Teste de segurança e controle de acesso 6

[2.10](#_usc66fxbeumr) Teste de falha e de recuperação 6

[2.11](#_ej3ev4bmvx3k) Teste de instalação 6

[**3**](#_chjj27fvcj3b) **ESTRATÉGIA DE TESTE 6**

[3.1](#_ai1y2trj0i0k) Tipos de teste 6

[*3.1.1*](#_jb70ujmftike) *Teste de integridade de dados e do banco de dados 6*

[*3.1.2*](#_7aab70luzr2j) *Teste de função 6*

[*3.1.3*](#_takvxsqcarlj) *Teste da interface do usuário 6*

[*3.1.4*](#_ia5egr4xi65v) *Teste de segurança e controle de acesso 7*

[*3.1.5*](#_tfn99vtlhj) *Teste de instalação 7*

[**4**](#_jxz0zl2knwk7) **RECURSOS 7**

[4.1](#_cu3t2i3hfpfh) Trabalhadores 7

[4.2](#_mcg7a8vh7fnl) Ferramentas 8

4.3 Equipamentos 8

[**5**](#_xzyag45v4clk) **CRONOGRAMA 8**

**Plano de Testes**

1. **Introdução**

* 1. **Objetivo desse documento**

Este documento do plano de testes do Swag Labs tem como objetivos:

* Identificar os componentes de *software* que serão testados.
* Listar os requisitos recomendados a serem testados.
* Recomendar e descrever as estratégias a serem empregadas nos testes.
* Identificar os recursos necessários.
* Estimar os esforços de teste.
* Documentar os resultados do projeto de testes.

* 1. **Escopo**

Teste de unidade, de integração e de sistema serão aplicados ao Swag Labs. Os testes unitários e de integração visam à qualidade funcional, das bases de dados, da interface gráfica e do controle de acesso. Os testes de sistema dão conta do desempenho do *software* e da compatibilidade do mesmo com o ambiente de trabalho.

O *software* é para uso individual e não é exigente no tempo de resposta ou do uso de memória, por isso, não serão realizados testes de carga, de *stress*, de *performance* e de volume.

* 1. **Identificação de projeto**

A tabela abaixo mostra a documentação usada para desenvolver o plano de testes com sua respectiva disponibilidade:

|  |  |  |
| --- | --- | --- |
| Documento | Criado ou Disponível | Recebido ou Revisado |
| Especificação de requisitos | ■Sim □ Não | ■Sim □ Não |
| Plano de projeto | ■Sim □ Não | ■Sim □ Não |
| Modelo de análise | □Sim ■ Não | □Sim ■ Não |
| Modelo de projeto | □Sim ■ Não | □Sim ■ Não |
| Documento de arquitetura | □Sim ■ Não | □Sim ■ Não |
| Protótipo | □Sim ■ Não | □Sim ■ Não |
| Manual de usuário | □Sim ■ Não | □Sim ■ Não |
| Lista de riscos | ■Sim □ Não | ■Sim □ Não |

* 1. **Visão geral do documento**

Este documento está dividido da seguinte forma:

* Seção 1: introdução.
* Seção 2: requisitos a testar.
* Seção 3: estratégias de teste.
* Seção 4: recursos.
* Seção 5: cronograma.

1. **Requisitos a testar**

Os itens a seguir foram obtidos através dos casos de uso, requisitos funcionais e não-funcionais. Esta lista representa o que será testado.

* 1. **Teste do banco de dados**
* Verifique que as informações do usuário podem ser cadastradas, alteradas e removidas.
* Verifique que as informações de uma pesquisa podem ser inseridas, alteradas e removidas.
* Verifique que os dados da matriz de repertório podem ser inseridos, alterados e removidos.

* 1. **Teste funcional**
* Verifique que um pesquisador pode acessar sua própria conta através de login e senha.
* Verifique que uma pesquisa pode ser criada, salva, alterada, recuperada e removida pelo pesquisador.
* Verifique que uma pesquisa salva é corretamente recuperada para o entrevistado responder.
* Verifique que uma matriz de repertório poder ser criada, salva, alterada, recuperada e removida pelo entrevistado.
* Verifique que o relatório gerado pelo software faz uma análise por agrupamento hierárquico correta.

* 1. **Teste do ciclo de negócios**

Nenhum.

* 1. **Teste da interface do usuário**
* Navegue através de todos os casos de uso, verificando se cada tela de interface gráfica é compreensível e intuitiva.
* Verifique se a língua usada é o português.
* Verifique se as palavras nos textos estão seguindo a norma culta da língua portuguesa.

* 1. **Perfil da *performance***

Nenhum.

* 1. **Teste de carga**

Nenhum.

* 1. **Teste de stress**

Nenhum.

* 1. **Teste de volume**

Nenhum.

* 1. **Teste de segurança e controle de acesso**
* Verifique que apenas o usuário pode adicionar ou remover itens no seu pedido.

* 1. **Teste de falha e de recuperação**

Nenhum.

* 1. **Teste de instalação**

Nenhum

1. **Estratégia de teste**

* 1. **Tipos de teste**

* + 1. **Teste de integridade de dados e do banco de dados**

|  |  |
| --- | --- |
| Objetivo do teste: | Verificar se os métodos e processos de acesso ao banco de dados funcionam corretamente, sem perdas e sem corrupção de dados. |
| Técnica: | Invocar cada método e processo de acesso ao banco de dados, alimentando cada um com dados ou requerimentos válidos e inválidos. |
| Critério de finalização: | Todos os métodos e processos de acesso à base de dados funcionam como projetados. Nenhum dado é perdido ou corrompido. |
| Considerações especiais: | Nenhuma. |

* + 1. **Teste de função**

|  |  |
| --- | --- |
| Objetivo do teste: | Verificar se o alvo de teste está funcionando de acordo com as especificações dadas nos requisitos, inclusive as mensagens de erro. |
| Técnica: | Executar cada caso de uso, seguindo todos os fluxos. Utilizar dados válidos e inválidos. |
| Critério de finalização: | Execução de todos os testes.  Os defeitos foram identificados e resolvidos. |
| Considerações especiais: | Nenhuma. |

* + 1. **Teste da interface do usuário**

|  |  |
| --- | --- |
| Objetivo do teste: | Verificar se a navegação através dos alvos reflete as funções e os requisitos. |
| Técnica: | Executar testes de navegação passando de janela para janela. |
| Critério de finalização: | Execução de todos os testes.  Os defeitos foram identificados e resolvidos. |
| Considerações especiais: | Nenhuma. |

* + 1. **Teste de segurança e controle de acesso**

|  |  |
| --- | --- |
| Objetivo do teste: | Verificar se um usuário pode acessar apenas as funções destinadas a ele. |
| Técnica: | Como entrevistado, tentar editar uma pesquisa criada por um pesquisador. |
| Critério de finalização: | Execução de todos os testes para cada tipo de acesso. |
| Considerações especiais: | Nenhuma. |

* + 1. **Teste de instalação**

|  |  |
| --- | --- |
| Objetivo do teste: | Verificar se o *software* executa corretamente todas as suas funcionalidades no ambiente mínimo exigido nos requisitos. |
| Técnica: | Executar todos os testes anteriores em uma máquina com configuração mínima. |
| Critério de finalização: | Execução de todos os testes.  Os defeitos foram identificados e resolvidos. |
| Considerações especiais: | Nenhuma. |

1. **Recursos**

Essa seção detalha os recursos necessários para a realização dos testes.

* 1. **Trabalhadores**

A tabela a seguir mostra o recrutamento para o projeto:

|  |  |  |
| --- | --- | --- |
| Função | Nome | Atribuições |
| Gerente de teste, gerente do projeto de teste, implementador | Charles | Supervisionar a aplicação de testes.  Fornecer relatórios de gerenciamento dos testes. |
| *Designer* de teste, implementador | Antonio | Identifica, prioriza e implementa casos de teste.  Cria plano de teste.  Cria o modelo de teste. |
| Administrador do sistema de teste | Daniel | Administra o sistema de gerenciamento de testes. |
| Gerente do banco de dados, administrador do banco de dados | Antonio | Garante que o ambiente e bens de teste de dados sejam gerenciados e mantidos. |
| *Designer* | Eric | Identifica e define as classes de teste. |
| Implementador | Todos | Implementa e faz os testes unitários das classes.  Cria as classes de teste implementados no modelo de teste. |

* 1. **Ferramentas**

|  |  |  |
| --- | --- | --- |
|  | Ferramenta | Vendedor |
| Gerenciamento de projeto | Jira | Atlassian |
| Desenvolvimento | Visual Studio Code, Python 3.11 | Microsoft |
| Automação | Selenium, WebDriver | Open Source |
| Navegador | Google Chrome | Google |

* 1. **Equipamentos**

|  |  |  |
| --- | --- | --- |
|  | Ferramenta | Vendedor |
| MacBook Air, M1 2020  8GB Memória, 256GB Armazenamento | macOS Sequoia 15.5 | Apple |
| Notebook Acer Aspire 5  8GB Memória, 256GB Armazenamento | Microsoft Windows 10 | Microsoft |

1. **Cronograma**

|  |  |  |
| --- | --- | --- |
| *Milestone* | Data de início | Data de término |
| Planejar teste | 26/05/2025 | 27/05/2025 |
| Projetar teste | 27/05/2025 | 27/05/2025 |
| Implementar teste | 27/05/2025 | 28/05/2025 |
| Executar teste | 28/05/2025 | 29/05/2025 |
| Revisar teste | 29/05/2025 | 29/05/2025 |
| Apresentar teste | 30/05/2025 | 30/05/2025 |