**Plano de Teste**

****

**Aplicação Financeira BugBank**

**Histórico da Alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 14/06/2024 | 1.0 | Testes Manuais e Automatizado | Chaiene |
|  |  |  |  |

**Conteúdo**

1. Introdução 4

1.1. Visão Geral do Projeto 4

1.2. Objetivos 4

2. Escopo dos Testes 5

2.1. Funcionalidades incluídas: 5

2.2. Funcionalidades não incluídas: 5

2.3. Técnicas de Testes: 5

2.4. Revisão e Aprovação de Testes: 5

3. Ambiente e Ferramentas 6

4. Classificação de Bugs 7

5. Cronograma de Testes: 8

6. Definição de Pronto 9

# Introdução

# Visão Geral do Projeto

O projeto em questão é uma aplicação web que simula um site bancário, focado principalmente em operações de transferência de fundos entre contas. Este projeto foi desenvolvido com o objetivo de fornecer uma plataforma prática para o treinamento e a aplicação de testes automatizados e manuais em um ambiente seguro e controlado. A interface e as funcionalidades buscam reproduzir as operações típicas de um banco digital, como gerenciamento de contas, verificação de saldo e execução de transferências, permitindo que os testadores avaliem o comportamento da aplicação em diferentes cenários e identifiquem possíveis falhas e melhorias. Através desse projeto, pretende-se consolidar boas práticas de testes e garantir a qualidade e a segurança das transações simuladas.

# Objetivos

Verificar as funcionalidades, com o foco na usabilidade, juntamente com o a performance, acessibilidade e a responsividade da aplicação.

# Escopo dos Testes



# Funcionalidades incluídas:

* Página de Login;
* Página de Cadastro;
* Funcionalidade de Transferência;
* Funcionalidade de Extrato;

# Funcionalidades não incluídas:

# Técnicas de Testes:

|  |  |  |
| --- | --- | --- |
| **Técnicas** | **Descrição** | **Responsável** |
| Teste Funcional | Terá uma cobertura de 100% das funcionalidades | QA |
| Testes de Limite |  | QA |
| Teste de Estado |  | QA |
| Teste de Performance | Verificar a performance do site, gerando um relatório com sugestões de melhorias. | QA |
| Teste de Usabilidade | Verificar se o site é intuitivo e fácil de usar para os usuários. | QA |
| Teste de Acessibilidade | Avaliar a acessibilidade do site utilizando ferramentas como leitores de tela. | QA |
| Testes Automatizados | Realizar testes automatizados nas funcionalidades implementadas para garantir qualidade. | QA |
| Teste Exploratório |  | QA |

# Revisão e Aprovação de Testes:

Ao final de cada cenário de teste, será elaborado um relatório contendo um resumo detalhado do teste realizado. Este relatório incluirá evidências relevantes e o Behaviour-Driven Development (BDD) realizado, seguindo o template do Documento de Execução de Testes (DET). Além disso, será identificado o responsável pela execução do teste, facilitando a rastreabilidade e responsabilização.

# Ambiente e Ferramentas

Os testes serão realizados em ambiente de produção, no site https://bugbank.netlify.app/, que contém as configurações finais. Utilizaremos uma massa de dados geradas pelo time de qualidade, com o propósito para a realização dos testes.

Abaixo, planilha com as ferramentas que serão utilizadas para a realizações dos testes:

|  |  |  |
| --- | --- | --- |
| **Ferramenta** | **Time** | **Descrição** |
| Selenium | Qualidade | Testes End-to-End |
| Cypress | Qualidade | Testes End-to-End |
| PageSpeed Insights | Qualidade | Testes de Performance |
| NVDA | Qualidade | Teste de Acessibilidade |

# Classificação de Bugs

A seguir, uma lista classificando os níveis de bugs, com descrições de seus impactos e seus respectivos níveis de bloqueio, conforme a tabela abaixo:

|  |  |  |
| --- | --- | --- |
| **ID** | **Nível de Severidade** | **Descrição** |
| 1 | Blocker | Bug que bloqueia o teste de uma função ou feature causa crash na aplicação.  Botão não funciona impedindo o uso completo da funcionalidade.  Bloqueia a entrega. |
| 2 | Grave | Funcionalidade não funciona como o esperado  Input incomum causa efeitos irreversíveis |
| 3 | Moderada | Funcionalidade não atinge certos critérios de aceitação, mas sua funcionalidade em geral não é afetada  Mensagem de erro ou sucesso não é exibida |
| 4 | Pequena | Quase nenhum impacto na funcionalidade, porém atrapalha a experiência  Erro ortográfico  Pequenos erros de UI |

# Cronograma de Testes:

Abaixo está o cronograma que demonstra as atividades a serem realizadas, seguindo o tempo das iterações, podendo variar até 3 dias úteis:

|  |  |
| --- | --- |
| **Iteração** | **Atividades realizadas** |
| 1 | Preparação do ambiente e das ferramentas, criação da planilha com os suítes de testes |
| 2 | Teste exploratório das funcionalidades já implementadas no ambiente de Produção |
| 3 | Criação dos cenários e casos de teste, execução dos testes manuais |
| 4 | Continuação da execução dos testes manuais |
| 5 | Execução dos testes automatizados e análise dos resultados obtidos |

# Definição de Pronto

Os critérios de aceitação são essenciais para atender aos requisitos estabelecidos pelo cliente e concluir com sucesso a fase de homologação, permitindo assim a liberação do produto para produção. Estes critérios incluem:

* Todos os casos de testes, devem ter sua execução bem sucedida e sem erros, aplicado através;
* As funcionalidades deverão seguir conforme as regras de negócios;