**Projeto de Testes**

***AppSage***

**Versão *0.4***

**Fase de Desenvolvimento: *<Especificação>***

**Iteração: *<0.4>***

**Histórico de Alterações**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 27/05/2021 | 0.2 | Implementação do sistema Kanban | <Gustavo + Matheus + Gabriel> |
| 31/05/2021 | 0.2 | Criação de usuário e banco de dados | <Gabriel + Gustavo + Matheus> |
| 03/06/2021 | 0.3 | Controladores de Usuário, Rotas e Modelos | <Gabriel Santo> |
| 03/06/2021 | 0.3.5 | Funcionalidade atualizar e deletar usuário | <Gabriel Santo> |
| 05/06/2021 | 0.3.5 | Controlador de Kanban | <Alexandre + Gustavo> |
| 05/06/2021 | 0.4 | Criar e entrar em projetos | <Gabriel Santo> |

**Conteúdo**

[**Introdução**](#_30j0zll) **3**

[**Casos de Testes**](#_1fob9te) **3**

[Funcional](#_3znysh7) 3

Usuário 3

Kanban 4

[**Procedimentos de Testes**](#_3dy6vkm) **5**

[Funcional](#_1t3h5sf) 5

[**Referências**](#_4d34og8) **6**

1. Introdução

Este documento descreve os casos e os procedimentos de testes do sistema *AppSage*, com o objetivo de expressar os testes realizados e seus resultados na *0.4* iteraçãoda fase de *Controladores implementados.*

1. Casos de Testes

Esta seção descreve os casos de teste empregados na execução dos testes. Os casos de testes definem todos os testes a serem realizados, com suas pré e pós-condições, passos específicos, critérios de sucesso, estratégia de teste, procedimentos, requisitos e planilha de testes associadas.

* 1. **Funcional**
     1. **Usuário**

|  |  |
| --- | --- |
| **[CT001] Criação\_atualização\_Usuario** | |
| **Descrição: Criar um usuário novo e atualizar suas informações** | |
| **Requisito(s) Associado(s):** [UC001]Login | |
| **Pré-Condições**:   * *Páginas de frontend funcionais* * *Comunicação com banco de dados* | **Pós-Condições:**   * *Usuário logado no sistema* |
| **Critérios de sucesso:**   1. *<Todas as condições de criação de usuário são cumpridas>* 2. *<Um novo usuário é criado no sistema>* 3. *<Informações do usuário são atualizadas ao clicar no botão de atualizar>* | |
| **<Opcional> Cenários:**   1. *Mensagem de erro*    1. *Pré-condições: O usuário não preenche algum campo de informação necessária*    2. *Pós-Condições:Usuário não é criado e a página exibe um erro.*   *2. Operação bem sucedida*  *2.1. Pós-condições:*  *2.1.1. Usuário logado no sistema*  *2.1.2. Informações do usuário são atualizadas* | |
| **Procedimento(s) Associado(s):** *[PT001]Testes\_Usuario* | |

A seguir é apresentado um exemplo de um caso de testes para o caso de uso Realizar doc, que implica numa transferência de fundos entre contas correntes.

**2.1.2 Kanban**

|  |  |
| --- | --- |
| **[CT002] Manipulação\_Tarefas\_Kanban** | |
| **Descrição: Criar uma nova tarefa na aba Kanban e editá-la** | |
| **Requisito(s) Associado(s):** [UC002]Kanban | |
| **Pré-Condições**:   * *Usuário Logado no sistema* * *Páginas frontend funcionais* | **Pós-Condições:**   * *Tarefa Kanban criada* * *Usuário pode editar a tarefa* |
| **Critérios de sucesso:**  *1.Uma nova tarefa é criada no Kanban*  *2.O usuário pode alterar o nome, a descrição e o estado da tarefa* | |
| **<Opcional> Cenários:**   1. *Operação bem sucedida*   *1.1 Pré-condições:*  *1.1.1. O usuário está logado e está em um grupo*  *1.2 Pós-Condições:*  *2.2.1. Uma nova tarefa é criada para o grupo*  *2.2.2 A tarefa pode ser editada e manipulada pelo grupo*  *2. Erro na edição de tarefa*  *2.1. Pré-condições:*  *2.1.1. Tarefa criada*  *2.2. Pós-condições:*  *2.2.1. Algum campo de edição da tarefa não foi preenchido corretamente*  *2.2.2. Uma mensagem de erro aparece informando ao usuário* | |
| **Procedimento(s) Associado(s):** *[PT002]Testes\_Kanban* | |

1. Procedimentos de Testes

Esta seção descreve os procedimentos de teste empregados na execução dos testes. Os procedimentos de testes indicam como o ambiente deve estar configurado e as ferramentas de apoio aos testes que devem ser empregadas para execução dos testes. Além disso, os procedimentos de testes definem técnicas (passos) a serem observadas e executadas durante a execução dos casos de testes que se baseiam nesses procedimentos.

* 1. **Funcional**

|  |
| --- |
| **[PT001] Testes\_Usuario** |
| **Descrição:** *Criação e atualização de um usuário* |
| **Configuração inicial do ambiente:**  Configuração da(s) máquina(s):   * *1 cliente <Windows 10, AMD eight-core, 8GB, internet ruim>* * *1 servidor <Mac, I9, 16GB, >*   Periféricos necessários:*<teclado>*  Processos executando em paralelo: *<browser exibindo o app, PyCharm>*  Rede: Cliente conectado a internet através de 4g do celular conectado ao computador |
| **Ferramentas/Componentes de apoio:**  Ferramentas: *<Flask, SQLAlchemy, Flask-migrate, Flask\_cors, MySQL, yarn, pycharm>*  Componentes: *<Indique os componentes de apoio que serão utilizados. Ex.: Stubs>*  Scripts a serem executados: *yarn start, flask run* |
| **Passos:**   * *Iniciar Yarn* * *Iniciar Flask pelo comando python3 -m flask run* * *Acessar o endereço da sua network iniciada pelo projeto* * *no Browser, clicar no botão Sign Up* * *Preencher os campos name, email e password* * *Clicar no botão Create User* * *Caso haja mensagem de erro, repetir os dois processos anteriores seguindo a instrução* * *No index, clicar no seu espaço de usuário* * *fazer as atualizações desejadas e preencher a senha* * *clicar no botão update User* * *clicar no botão de X e verificar se o perfil foi atualizado* |

*Segundo caso de testes:*

|  |
| --- |
| **[PT002] Testes\_Kanban** |
| **Descrição:** *Procedimento para realização de testes funcionais para operações que envolvem transferência de fundos on line.* |
| **Configuração inicial do ambiente:**  Configuração da(s) máquina(s):   * *1 cliente <Windows 10, AMD eight-core, 8GB, internet ruim>* * *1 servidor <Mac, I9, 16GB, >* * *Celular android utilizado como provedor de internet*   Periféricos necessários:*mouse, teclado*  Rede: Cliente conectado a internet através de 4g do celular conectado ao computador |
| **Ferramentas/Componentes de apoio:**  Ferramentas: *<Flask, SQLAlchemy, Flask-migrate, Flask\_cors, MySQL, Yarn, Pycharm>*  Scripts a serem executados: *yarn start, flask run* |
| **Passos:**   1. *Iniciar Yarn pelo comando “yarn start”* 2. *Iniciar Flask pelo comando “python3 -m flask run”* 3. *Acessar o endereço da sua network iniciada pelo projeto* 4. *Realizar o login*   *5. Acessar a aba Kanban clicando nela (opção no topo)*  *6. Clicar no botão de adicionar tarefa*  *7. Clicar na nova tarefa nomeada “New item”*  *8. Editar a tarefa digitando as informações desejadas e selecionando seu estado*  9. Clicar no botão de Save  10. Verificar se o processo foi bem sucedido |

1. Referências

[***https://github.com/gabssanto/engenharia-de-software***](https://github.com/gabssanto/engenharia-de-software)

[***https://github.com/facebook/create-react-app***](https://github.com/facebook/create-react-app)

[***https://flask-sqlalchemy.palletsprojects.com/en/2.x/***](https://flask-sqlalchemy.palletsprojects.com/en/2.x/)