**Documentação**

**Estudantes**

Alexandre Sebastian **1324553**

Felipe Gesser **1323865**

**Disciplina**

**Qualidade de Software**

**Professor**

**Diego Sauter Possamai**

**Universidade**

**Católica SC – Joinville**

**Data de Entrega:** [25/03/2025]

**2. Introdução**

**2.1 Apresentação do Projeto**

Este projeto consiste no desenvolvimento de uma aplicação web monolítica voltada para análise e tratamento de dados. A ferramenta permitirá a criação, inserção e manipulação de dados, possibilitando sua organização em diferentes formatos, como dashboards, textos e planilhas. O objetivo é fornecer um ambiente eficiente para estudo e análise de informações, garantindo flexibilidade e personalização para os usuários.

**2.2 Justificativa**

Atualmente, muitas pessoas lidam com uma quantidade massiva de informações, o que pode tornar o processo de pesquisa e análise desorganizado e estressante, mesmo com o auxílio de ferramentas tradicionais. Esse software tem como objetivo otimizar a coleta e a análise de dados, proporcionando uma experiência mais eficiente e estruturada. Dessa forma, ele facilita estudos e pesquisas científicas, permitindo que os usuários encontrem, organizem e interpretem informações de maneira mais ágil e precisa.

**2.3 Público-Alvo**

O software será usado por qualquer pessoa que desejar usá-lo, provavelmente tendo um público entre estudantes e pesquisadores.

Forma

**3. Modelagem do Software**

**3.1 Requisitos Funcionais**

**Gerenciamento de usuários**

**• RF001:** O sistema deve permitir a criação de usuários através de e-mail e senha.

**• RF002:** Após a criação do usuário, deve ser possível adicionar informações adicionais, como nome, telefone, entre outros.

**• RF003:** Deve ser possível editar as informações do usuário após sua criação.

**• RF004:** Caso um usuário já tenha sido cadastrado, o acesso deve ser possível através de e-mail ou nome de usuário e senha.

**• RF005:** O sistema deve permitir a recuperação de senha por meio de um código enviado via e-mail ou SMS.

**Manipulação de dados**

**• RF006:** O usuário deve poder criar seus próprios dados em uma planilha dentro da aplicação.

**• RF007:** O sistema deve permitir a importação de dados no formato CSV.

**• RF008:** Os dados inseridos devem poder ser tratados e filtrados conforme as necessidades do usuário.

**• RF009:** Deve ser possível editar os dados inseridos.

**• RF010:** O sistema deve garantir que os dados inseridos sejam salvos e mantidos para uso posterior.

**• RF011:** Deve ser possível salvar as alterações feitas no dashboard

**Visualização e análise de dados**

**• RF012:** Deve ser possível criar dashboards personalizados com os dados inseridos.

**• RF013:** O usuário deve poder visualizar múltiplas telas dentro da aplicação.

**• RF014:** Cada tela deve exibir os dados conforme a configuração escolhida pelo usuário.

**• RF015:** Deve ser possível alternar a visualização dos dados entre formato de texto e dashboard.

**3.2 Requisitos Não Funcionais**

**• RNF001**: O sistema deve ser acessível via navegador web.

**• RNF002:** O sistema deve armazenar os dados em um banco de dados relacional.

**• RNF003:** O tempo de resposta para exibição da lista 2 segundos.

**• RNF004:** O sistema deve implementar a autenticação para proteger as informações dos usuários.

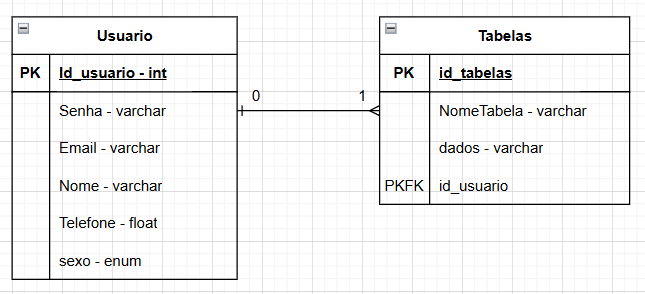
**• RNF005:** O sistema deve utilizar criptografia para armazenar dados sensíveis.

**• RNF005:** O sistema deve permitir o uso de dashboards diversos como gráfico pizza.

**3.3 Casos de Uso**

* **Caso de Uso: Inserir dados**
  + **Ator:** Usuário Autenticado
  + **Fluxo Principal:** O usuário faz o upload de um arquivo (CSV) ou insere manualmente os dados em um formulário. O sistema valida as informações, armazena no banco de dados e exibe uma confirmação.
* **Caso de Uso: Filtrar e organizar dados**
  + **Ator:** Usuário Autenticado
  + **Fluxo Principal:** O usuário seleciona critérios de filtragem (datas, categorias, palavras-chave). O sistema processa a solicitação, exibe os dados organizados e permite ajustes.
* **Caso de Uso: Gerar Dashboard**
  + **Ator:** Usuário Autenticado
  + **Fluxo Principal:** O usuário seleciona os dados e escolhe o tipo de visualização (gráficos, tabelas, indicadores). O sistema gera automaticamente um dashboard interativo.

**3.4 Modelagem de Dados (Diagrama Entidade-Relacionamento - ER)**



**3.5 Arquitetura do Sistema**

A arquitetura será um arquivo único, que ficara responsável por todas as ações referentes ao banco de dados, essa arquitetura é possível por conta da supabase, que facilita a hospedagem e manipulação de nosso banco de dados.

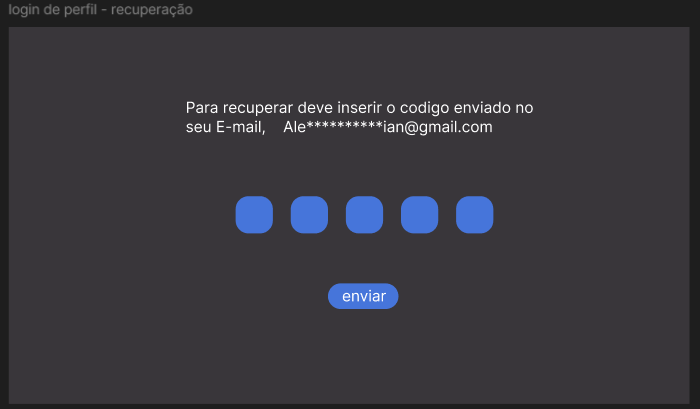
**3.6 Planejamento da Infraestrutura**

* Banco de Dados: Supabase
* Back-end: Python
* Front-end: React.js + Bootstrap
* Hospedagem: InfinityFree

**3.7 Diagramas**

**-> Diagramas UML, Wireframes de Interface, Diagrama de Casos de Uso**

Wireframe de interface:  
Descrição: Login e register são as duas telas principais, onde a de login será a primeira tela vista pelo usuário assim que entregar no programa, nele caso o usuário não possua um login, poderá ser redirecionado para o register, a segunda tela onde poderá criar seu usuário.

Recuperação de senha:  
Descrição: esta é a tela que o usuário será redirecionado caso opte por alterar ou recuperar a sua senha, um código de recuperação será enviado para o e-mail cadastrado, e após a inserção, o mesmo poderá alterar sua senha.

Inserção e visualização de dados:

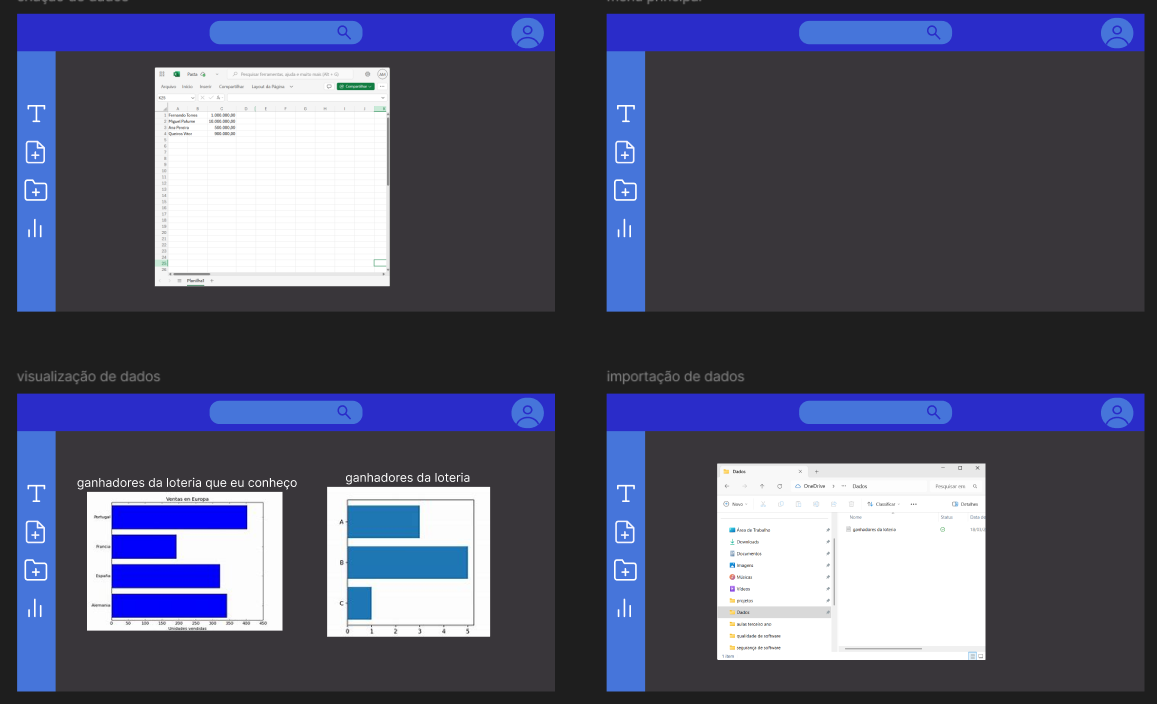
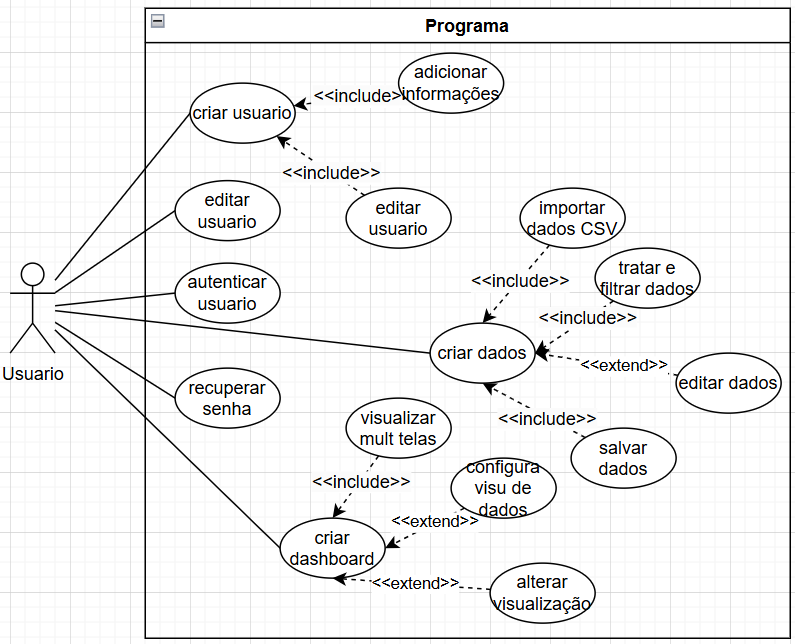
Descrição: esta será a tela principal do programa, após efetuar o login o usuário se deparará com ela, ele terá 4 principais ferramentas para tratamento de dados, a primeira representada pelo símbolo de um ”T”, é a opção de adicionar texto em qualquer região do quadro, a seguinte é a criação de dados, ao selecioná-la será aberto uma planilha do Exel onde poderá ser adicionado quais quer dados que queira para visualização, em seguida temos a opção de importação de dados, onde você ponderar selecionar dados do seu computador que estiverem em um formato aceitável pelo programa, a última feramente serve para definir quais tipos de dashboards os dados serão organizados.

Diagrama de Caso de Uso:  


**4. Planejamento da Qualidade do Software**

**4.1 Normas e Padrões Utilizados**

Para garantir a qualidade do software, serão seguidas normas e padrões reconhecidos internacionalmente:

* ISO/IEC 25010: Define critérios de qualidade para produtos de software, garantindo adequação funcional, confiabilidade e eficiência.
* ISO 9001: Padroniza a gestão da qualidade no desenvolvimento de software.
* ISO/IEC 12207: Especifica os processos do ciclo de vida do software.
* IEEE 829: Normatiza a documentação de testes de software.

**4.2 Planejamento de Testes**

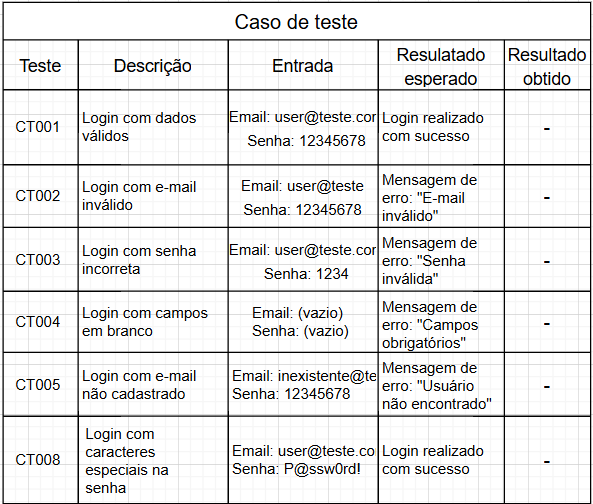
O planejamento de testes será realizado para garantir que todas as funcionalidades do sistema estejam implementadas corretamente e sigam os padrões de qualidade definidos. Os seguintes tipos de testes serão aplicados:

**4.2.1 Testes Unitários**

Objetivo: Garantir que cada componente do sistema funcione corretamente de forma isolada.  
 Como será feito: teste manualmente com frequência.  
 Medição: Cobertura mínima de 80% do código-fonte.  
 Validação: Nem uma função deve apresentar falhas.

**4.2.2 Testes de Integração**

Objetivo: Avaliar a comunicação entre os diferentes módulos do sistema.  
 Como será feito: Será efeituada diversas inserções com diferentes formatos.  
 Medição: Testes devem validar todas as rotas de API e possíveis interações entre serviços.  
 Validação: Todas as inserções devem ser efetivadas de maneira correta incluído suas tipagens.



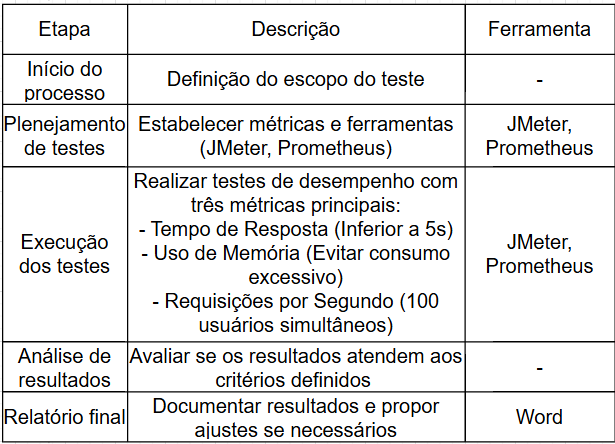
**4.3 Métricas de Qualidade e Validação**

Para garantir a qualidade do software ao longo do desenvolvimento, serão monitoradas e analisadas as seguintes métricas:

**4.3.1 Métricas de Desempenho**

* Tempo de Resposta: Deve ser inferior a 5 segundos por requisição, medido com JMeter.
* Uso de Memória: Monitorado via Prometheus para evitar consumo excessivo.
* Número de Requisições por Segundo: Deve suportar 100 usuários simultâneos sem degradação.

**4.4 Diagrama**  
  
**Diagrama de Processo de Testes**



**CONCLUSÃO**

Nesta primeira etapa do planejamento, foram apresentadas as ideias para o desenvolvimento da aplicação web, incluindo as ferramentas que serão utilizadas, a divisão de tarefas e as normas padrão para o processo de desenvolvimento. O próximo passo será iniciar o desenvolvimento, utilizando essa documentação como referência