

**Emelly Yasmin Andrade Formigário**

**Gabriélly Custódio Ferreira**

**Henrique Siviero  
Marco Antonio Alcindo Gitti**

**SIGEDESP  
Sistema de Gerenciamento de Despesas da Prefeitura de Jales**

**Projeto Integrador**

Disciplinas Envolvidas: Engenharia de Software, Tópicos Avançado de Banco de Dados Relacional e não Relacional, Técnicas Avançadas de Programação, Técnicas Avançadas de Programação Web e Mobile, Gestão Ágil de Projetos de Software.

Jales

2023

Lista de Figuras

Lista de Tabelas

Lista de Quadros

SUMÁRIO

[1 INTRODUÇÃO 3](#_Toc152321107)

[2 LEVANTAMENTO DE REQUISITOS DE SOFTWARE 4](#_Toc152321108)

[2.1 Descrição dos objetivos do sistema 4](#_Toc152321109)

[2.2 Descrição do sistema atual 4](#_Toc152321110)

[2.3 Descrição dos principais problemas 5](#_Toc152321111)

[2.4 Descrição dos requisitos funcionais 5](#_Toc152321112)

[2.5 Descrição dos requisitos não funcionais 9](#_Toc152321113)

[3 VISÃO DE CASO DE USO – UML 11](#_Toc152321114)

[3.1 Diagrama de Classes 11](#_Toc152321115)

[3.2 Dicionário de classes 12](#_Toc152321116)

[3.3 Definição dos Atores 19](#_Toc152321117)

[3.4 Lista de Casos de Uso 21](#_Toc152321118)

[3.4. Diagrama de Casos de Uso 21](#_Toc152321119)

[3.5. Diagrama de Casos de uso individuais 23](#_Toc152321120)

[3.6. Diagrama de Sequência 24](#_Toc152321121)

[3.7. Diagrama de Comunicação 25](#_Toc152321122)

[3.8. Diagrama de Atividade 25](#_Toc152321123)

[4 DEFINIÇÃO DA INTERFACE COM O USUÁRIO (UX) 26](#_Toc152321124)

[4.1 Descrição de cenário 26](#_Toc152321125)

[4.2 Descrição de personas 27](#_Toc152321126)

[4.3 Esboços de tela (wireframes) 29](#_Toc152321127)

[4.4 Protótipos de tela 32](#_Toc152321128)

[5 BANCO DE DADOS 36](#_Toc152321129)

[5.1 Modelo Entidade Relacionamento 36](#_Toc152321130)

[5.2 Script das tabelas 37](#_Toc152321131)

[6 ARQUITETURA DE SOFTWARE 41](#_Toc152321132)

[6.1 Arquitetura de desenvolvimento 41](#_Toc152321133)

[6.2 Telas do sistema 45](#_Toc152321134)

[7 CONCLUSÃO 48](#_Toc152321135)

[8 REFERÊNCIAS 49](#_Toc152321136)

# INTRODUÇÃO

No âmbito da gestão financeira, o sistema de gerenciamento de despesas emerge como uma ferramenta essencial, seja para indivíduos que buscam controlar suas finanças pessoais ou para organizações que necessitam monitorar, registrar, categorizar e gerir suas despesas financeiras de forma eficaz. Essas soluções desempenham um papel fundamental na simplificação do processo de rastreamento de despesas, na melhoria da visibilidade dos gastos e no estabelecimento de um controle orçamentário sólido.

Como destacado pela Oracle, "De forma simples, sem algum tipo de software de contabilidade, uma organização não sobreviveria." (Oracle, 2023). Neste contexto, torna-se evidente que a ausência de software de contabilidade eficiente pode representar um risco significativo para o desempenho financeiro de qualquer organização.

Em um cenário no qual diversas instituições lidam com o desafio de controlar suas despesas, é crucial mencionar as Prefeituras como exemplo paradigmático. Cada município é dotado de sua própria prefeitura, composta por diversos setores, sendo um deles responsável por monitorar as despesas dos demais setores e instituições sob sua jurisdição. No entanto, o volume considerável de despesas torna imperativo que a Prefeitura mantenha uma organização eficaz, uma vez que a falta de controle adequado pode resultar na perda de informações cruciais e, consequentemente, em desafios substanciais.

Com a finalidade de auxiliar a Prefeitura de Jales a enfrentar esse desafio, o presente trabalho se propõe a desenvolver um sistema web dedicado ao gerenciamento de despesas. Este software representará um recurso valioso para a prefeitura, oferecendo funcionalidades que incluem a geração de relatórios detalhados, painéis de controle intuitivos, manutenção de cadastros atualizados, análise orçamentária abrangente, entre outras capacidades.

Com a implementação desse sistema, a Prefeitura de Jales estará melhor preparada para manter um controle preciso das despesas, garantindo a utilização eficiente do orçamento previamente estipulado. A seguir, este trabalho se aprofundará na elaboração, implementação e avaliação desse sistema, destacando sua importância na gestão financeira eficaz do município.

# LEVANTAMENTO DE REQUISITOS DE SOFTWARE

## Descrição dos objetivos do sistema

O principal objetivo do Sistema de Gerenciamento de Despesas (SIGEDESP) é proporcionar à Prefeitura de Jales uma plataforma completa que viabilize o registro, acompanhamento e geração de relatórios detalhados de suas despesas relacionadas às contas de serviços públicos. O foco central deste sistema é abordar e solucionar os desafios identificados no sistema de gestão de despesas atual, com ênfase na promoção da eficiência, transparência e controle eficaz das despesas municipais.

Para alcançar esse objetivo primordial, o SIGEDESP buscará a implementação de um sistema integrado que centralize e unifique todas as informações relacionadas às despesas municipais. Esse processo de integração eliminará as lacunas na comunicação entre as diferentes partes envolvidas no ciclo de despesas, garantindo que todas as partes tenham acesso completo e oportuno aos dados relevantes. Isso resultará em uma gestão financeira mais transparente e eficiente na Prefeitura de Jales.

Além disso, o SIGEDESP estabelecerá um registro de auditoria abrangente, assegurando a integridade dos dados e a rastreabilidade de todas as operações realizadas no sistema. Esta abordagem reforça o compromisso do projeto com a transparência e conformidade com as melhores práticas de governança e gestão municipal.

Em resumo, o SIGEDESP representa um marco significativo na busca pela melhoria do controle financeiro na Prefeitura de Jales, com a aspiração de contribuir para uma administração pública mais transparente, eficiente e adaptada aos desafios modernos. Este projeto visa aprimorar a qualidade da gestão de despesas e, consequentemente, promover o bem-estar dos cidadãos de Jales.

## Descrição do sistema atual

A Prefeitura Municipal de Jales utiliza o Microsoft Excel como ferramenta principal para o registro e gestão de contas a pagar e pendentes. Este sistema baseia-se em uma planilha que abrange diversos tipos de despesas, incluindo: Energia, Água, Telefone, Internet e (Aluguel). Cada departamento da instituição possui seu próprio conjunto de despesas, que são registradas e monitoradas no Excel, com indicação se foram pagas ou estão pendentes.

Atualmente, essa planilha passa por frequentes atualizações diárias, o que torna a tarefa de controle ainda mais desafiadora. Isso cria dificuldades para os responsáveis pelo acompanhamento dessas informações, uma vez que não é fácil determinar o status de pagamento de cada despesa.

## Descrição dos principais problemas

Foram identificados diversos problemas no sistema atual de gerenciamento de despesas. Entre essas questões, destacam-se as dificuldades enfrentadas para adicionar e gerenciar as despesas de forma eficiente. Uma das razões para essas dificuldades está relacionada à presença de outro setor envolvido no processo de despesas, o que resulta em uma divisão de responsabilidades fragmentada em diferentes etapas do processo.

Cada parte envolvida no ciclo das despesas acaba retendo apenas uma parte dos dados, o que gera uma lacuna na informação. Aqueles que criam as despesas têm informações incompletas sobre como elas são posteriormente processadas e pagas, criando um cenário de desconexão. Esse problema resulta em uma grande desorganização no sistema, prejudicando a eficiência e transparência do gerenciamento das despesas.

Portanto, é evidente que a situação atual requer uma revisão profunda do sistema de gerenciamento de despesas. O projeto SIGEDESP se propõe a abordar esses desafios e melhorar a comunicação e o controle das despesas, promovendo uma gestão mais eficiente e transparente das finanças da Prefeitura de Jales.

## Descrição dos requisitos funcionais

Requisitos funcionais são declarações detalhadas que definem as funções, recursos e comportamentos específicos necessários para um sistema ou software atender às expectativas e necessidades dos usuários. Esses requisitos delineiam o que o sistema deve realizar e como deve se comportar em diversas situações. Os requisitos funcionais constituem uma parte vital na especificação de software, servindo como alicerce para o desenvolvimento, teste e validação do sistema.

No contexto do Sistema de Gerenciamento de Despesas (SIGEDESP) destinado à Prefeitura de Jales, destacamos os seguintes requisitos funcionais:

**Quadro ? –** Requisitos Funcionais

|  |  |  |
| --- | --- | --- |
|  | **Requisitos**  **Funcionais** | **Descrição** |
|  |  |  |
| **1** | Autenticação de Usuário | Os usuários devem ser capazes de inserir suas credenciais (nome de usuário e senha) para fazer login. |
| **2** | Controle de Acesso | O sistema deve verificar as credenciais e conceder acesso apenas a usuários registrados e autorizados. |
| **3** | Registro de  Pagamentos | O sistema deve registrar os pagamentos que já foram pagos (incluir data do pagamento, forma que pagou e etc); |
| **4** | Registro de  Log de Acesso | O sistema deve registrar o acesso bem-sucedido e tentativas de acesso malsucedidas para fins de auditoria. |
| **5** | Gerenciamento  de Tipos de Usuário | Os administradores podem definir, modificar ou excluir tipos de usuário, como administrador, público ou atendente. |
| **6** | Registro de Despesas | Os usuários podem registrar informações sobre despesas, como data, valor, descrição e método de pagamento. |
| **7** | Geração de  Relatórios | Os usuários podem gerar relatórios detalhados sobre suas despesas, incluindo gráficos e informações resumidas. |
| **8** | Notificações de  Despesas Pendentes | O sistema envia notificações aos usuários quando despesas estão pendentes de aprovação ou quando atingem limites definidos. |
| **9** | Exportação de  Dados | Os usuários podem exportar dados de despesas para formatos comuns, como CSV ou PDF. |
| **10** | Categorização  de Despesas | O sistema permite aos usuários categorizarem suas despesas em categorias personalizáveis, como alimentação, transporte, etc. |
| **11** | Log de Atividades  do Usuário | Registrar atividades de login, incluindo horário, localização e dispositivo utilizado. |
| **12** | Confirmação de  E-mail | Enviar e-mail de confirmação para verificar a autenticidade do endereço de e-mail fornecido durante o cadastro. |
| **13** | Detalhes de Usuário | Os detalhes do usuário devem incluir nome, sobrenome, endereço de e-mail, senha, tipo de usuário e outras informações. |
| **14** | Permissões e Papéis | Os tipos de usuário (adm, público, atendente) devem ter permissões específicas para ações dentro do sistema. |
| **15** | Pesquisa Avançada de Despesas | Os usuários podem realizar pesquisas avançadas, filtrando despesas por data, categoria, valor, etc. |
| **16** | Acompanhamento de faturas | O sistema deve exibir o status das contas (pago, pendente, vencido); |
| 17 | Backup e Recuperação de Dados | O sistema deve ter mecanismos de backup e recuperação de dados para evitar a perda de informações importantes. |
| 18 | Cadastrar Despesa | Permite aos usuários inserir novas informações sobre despesas, incluindo valor, data e categoria. |
| 19 | Alterar Despesa | Oferece a capacidade de modificar dados de despesas previamente registradas, como ajustar valores ou datas. |
| 20 | Excluir Despesa | Permite a remoção de registros de despesas, contribuindo para a manutenção da precisão da base de dados. |
| 21 | Listar Despesa | Facilita a visualização organizada de todas as despesas registradas para uma análise abrangente. |
| 22 | Entrada de Despesas | Permite registrar e monitorar as despesas incorridas, oferecendo uma visão clara dos valores, datas e categorias associadas às transações financeiras. |
| 23 | Saída de Despesas | Facilita o acompanhamento e controle das despesas já realizadas, proporcionando uma análise detalhada dos gastos e contribuindo para uma gestão financeira eficiente. |
| 24 | Controle de Despesas em Atraso | Permite identificar e gerenciar despesas que estão pendentes, fornecendo informações sobre valores em atraso, datas correspondentes e categorias associadas para facilitar a gestão financeira. |

Fonte: Elaborado pelos autores

## Descrição dos requisitos não funcionais

De acordo com Visure Solutions (2023), os requisitos não funcionais constituem as limitações e características de qualidade impostas a um sistema, delineando atributos como segurança, desempenho e escalabilidade. Esses requisitos, frequentemente descritos por adjetivos, desempenham um papel crucial ao assegurar que o sistema atenda adequadamente às demandas do usuário. Ao contrário dos requisitos funcionais que especificam as funcionalidades diretas do sistema, os focam na qualidade e desempenho, desempenhando um papel essencial na garantia da eficácia global e na satisfação do usuário. Essa categorização e definição clara de requisitos não funcionais são fundamentais para o desenvolvimento e o sucesso de sistemas de software robustos e adaptáveis. Os requisitos não funcionais são classificados em diversas categorias, como: Interface de Usuário, Segurança, Confiabilidade, Performance, Manutenção, entre outros.

Os Requisitos Não Funcionais de um Sistema de Gerenciamento de Despesas para a Prefeitura podem ser diversos e devem ser considerados para garantir a eficiência, segurança e adequação do sistema às necessidades específicas do ambiente municipal. Aqui estão alguns exemplos de RNF que podem ser relevantes.

Quadro ? – Requisitos Não Funcionais

|  |  |
| --- | --- |
| Requisitos Não Funcionais | Descrição |
| Desempenho | Garantir tempos de resposta rápidos para consultas e transações, mantendo alta eficiência. |
| Segurança | Implementar medidas robustas, como controle de acesso e criptografia, para proteger dados sensíveis. |
| Escalabilidade | Capacidade de lidar com um aumento substancial no volume de dados e transações. |
| Disponibilidade | Assegurar que o banco de dados esteja acessível a maior parte do tempo, minimizando períodos de inatividade. |
| Consistência | Manter a integridade e a consistência dos dados, garantindo precisão e atualização constante. |
| Backup e Restauração | Estabelecer procedimentos eficientes para backup regular e recuperação rápida em caso de falhas. |

Fonte: Elaborado pelos autores

# VISÃO DE CASO DE USO – UML

O entendimento profundo de um sistema é crucial para o sucesso de qualquer projeto de desenvolvimento de software. Nesse contexto, o Diagrama de Modelagem Unificada (UML) emerge como uma ferramenta essencial para representar visualmente a estrutura e as interações entre os componentes do sistema. De acordo com Miro (2023), “desenvolvedores criam diagramas UML para entender projetos, arquitetura de código e propostas de implementação de sistemas de software complexos.”

A análise a seguir se dedica à exploração dos principais diagramas da UML aplicados no desenvolvimento do sistema. Estes diagramas desempenham um papel crucial na compreensão da estrutura e na narrativa do processo de desenvolvimento do sistema, oferecendo uma visão aprofundada de sua arquitetura. Ao examinar cada diagrama, buscaremos desvendar não apenas a estrutura técnica, mas também a lógica e a dinâmica subjacentes ao desenvolvimento, proporcionando uma base sólida para a compreensão abrangente do projeto.

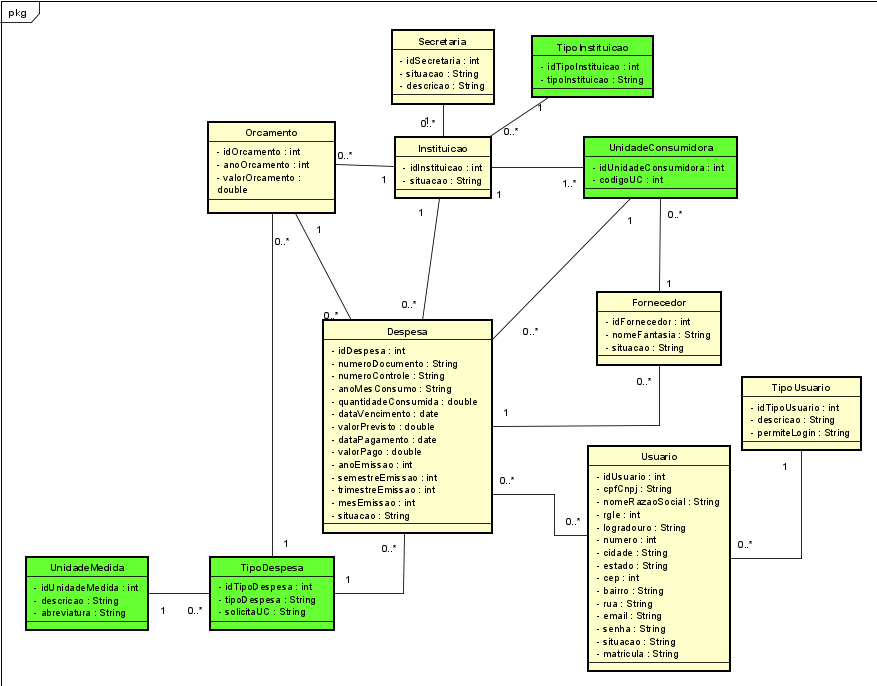
## 3.1 Diagrama de Classes

O diagrama de classe é fundamental na UML, sendo essencial para criar os demais diagramas do conjunto.

Seu principal enfoque está em permitir a visualização das classes que comporão o sistema com seus respectivos atributos e métodos, bem como em demonstrar como as classes do diagrama se relacionam, complementam e transmitem informações entre si. Esse diagrama apresenta uma visão estática de como as classes estão organizadas, preocupando-se em como definir a estrutura lógica delas. (GUEDES, 2018, p.34)

Neste contexto, apresentaremos o diagrama de classe modelado para o projeto SIGEDEPS.

**Figura ? –** Diagrama de Classe – SIGEDESP



Fonte: Elaborado pelos autores

## 3.2 Dicionário de classes

Para proporcionar uma compreensão mais clara do funcionamento do sistema, vai ser detalhando cada tabela, juntamente com seus atributos.

**UnidadeMedida:** tem como objetivo principal cadastrar e gerenciar as unidades de medida associadas aos diferentes tipos de despesas.

**Quadro ?** – Descrição Classe UnidadeMedida

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idUnidadeMedida | Integer | Código que irá identificar a unidade de medida de cada instituição. |
| descricao | String | Nome da unidade de medida. |
| Abreviatura | String | Abreviatura da unidade de medida. |

Fonte: Elaborado pelos autores

**TipoInstituicao:** tem como objetivo principal cadastrar e classificar os diferentes tipos de instituições no sistema.

**Quadro ?** – Descrição Classe TipoInstituicao

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idTipoInstituicao | Integer | Código que irá identificar o tipo de instituição. |
| tipoInstituicao | String | Classificação do Tipo de instituição. |

Fonte: Elaborado pelos autores

**UnidadeConsumidora:** tem como objetivo principal cadastrar os códigos identificadores de cada instituição em relação a cada despesa que possui.

**Quadro ?** – Descrição Classe UnidadeConsumidora

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idUnidadeConsumidora | Integer | Código que irá identificar a unidade consumidora de cada instituição. |
| codigoUC | Integer | Código da unidade consumidora. |

Fonte: Elaborado pelos autores

**Instituição:** tem como objetivo principal cadastrar os nomes das instituições, sendo utilizada para categorizar a qual instituição uma determinada despesa pertence.

**Quadro ?** – Descrição Classe Instituicao

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idInstituicao | Integer | Código que irá identificar a instituição. |
| Situação | String | Classificação da situação em que a instituição se encontra, como por exemplo, operacional ou inativo. |

Fonte: Elaborado pelos autores

**Secretária:** tem como objetivo principal cadastrar as secretárias que a Prefeitura de Jales possui, sendo utilizada para categorizar a qual secretária uma determinada despesa pertence.

**Quadro ?** – Descrição Classe Secretaria

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idSecretaria | Integer | Código que irá identificar a secretária. |
| Situação | String | Classificação da situação em que a secretária se encontra, como por exemplo, operacional ou inativo. |
| descricao | String | Identificar para qual secretaria a despesa está sendo cadastrada. |

Fonte: Elaborado pelos autores

**TipoDespesa:** tem como objetivo principal cadastrar os diferentes tipos de despesas, sendo utilizada para classificar a qual categoria uma determinada despesa pertence.

**Quadro ?** – Descrição Classe TipoDespesa

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idTipoDespesa | Integer | Código que irá identificar o tipo de despesa. |
| tipoDespesa | String | Classificação do tipo de despesa. |
| solicitaUC | String | Solicita ou Não o Código UC |

Fonte: Elaborado pelos autores

**Fornecedor:** tem como objetivo principal cadastrar as informações sobre os fornecedores, ou seja, as empresas responsáveis por fornecer as despesas. Essa classe é utilizada para associar as despesas aos seus respectivos fornecedores, facilitando a organização e rastreamento.

**Quadro ?** – Descrição Classe Fornecedor

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idFornecedor | Integer | Código que irá identificar o fornecedor. |
| nomeFantasia | String | Nome Comercial da empresa que é fornecedora. |
| Situação | String | Classificação da situação em que o fornecedor se encontra, como por exemplo, operacional ou inativo. |

Fonte: Elaborado pelos autores

**TipoUsuario:** tem como objetivo cadastrar os diferentes tipos de usuários que terão acesso ao sistema. Essa classe é utilizada para determinar as autorizações e restrições de acesso para cada categoria de usuário.

**Quadro ?** – Descrição Classe TipoUsuario

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idTipoUsuario | Integer | Código que irá identificar o tipo de usuário. |
| descricao | String | Qual é o tipo de usuário que está acessando o sistema, como por exemplo, administrador. |
| permiteLogin | String | Permissão para acessar ou não o sistema. |

Fonte: Elaborado pelos autores

**Orçamento:** tem como objetivo cadastrar os orçamentos propostos para a realização dos pagamentos das despesas ao longo do ano. Esses dados são utilizados na geração de relatórios e dashboards, fornecendo uma visão consolidada e analítica das previsões de gastos, e o cálculo do orçamento do próximo ano.

**Quadro ?** – Descrição Classe Orcamento

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idOrcamento | Integer | Código que irá identificar o orçamento. |
| anoOrcamento | Integer | Ano em que o orçamento foi calculado. |
| valorOrcamento | Double | Valor que foi calculado para o orçamento. |

Fonte: Elaborado pelos autores

**Despesa:** é a peça central do sistema, sendo responsável por cadastrar as despesas relacionadas a instituições gerenciadas pelas secretarias da prefeitura. Essa classe possibilita o registro organizado e a gestão eficaz das transações financeiras, sendo fundamental para o monitoramento e controle das despesas no contexto da administração pública.

**Quadro ?** – Descrição Classe Despesa

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idDespesa | Integer | Código que irá identificar o fornecedor. |
| numeroDocumento | String | Número do Documento (Despesa) emitido. |
| numeroControle | String | Número de Controle da Despesa. |
| anoMesConsumo | String | Ano e Mês do Consumo da Despesa. |
| quantidadeConsumida | Double | Quantidade de recursos consumidos. |
| dataVencimento | Date | Data de Vencimento da Despesa. |
| valorPrevisto | Double | Valor total previsto na Despesa. |
| dataPagamento | Date | Data de Pagamento da Despesa. |
| valorPago | Double | Valor que foi pago. |
| anoEmissao | Int | Ano de Emissão da Despesa. |
| semestreEmissao | Int | Semestre de Emissão da Despesa. |
| trimestreEmissao | Int | Trimestre de Emissão da Despesa. |
| mesEmissao | Int | Mês de Emissão da Despesa. |
| Situação | String | Situação que a Despesa se encontra. |

Fonte: Elaborado pelos autores

**Usuário:** tem como objetivo principal armazenar informações fundamentais sobre os usuários que acessam o sistema. Essa classe é essencial para gerenciar e controlar o acesso ao sistema, permitindo a personalização das permissões e proporcionando uma base para a segurança e identificação dos usuários.

**Quadro ?** – Descrição Classe Usuario

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idUsuario | Integer | Código que irá identificar o fornecedor. |
| cpfCnpj | String | CPF/CNPJ do Usuário |
| nomeRazaoSocial | String | Nome do Usuário |
| Rgle | Int | RG/LE do Usuário |
| logradouro | String | Logradouro que o Usuário mora ou trabalha. |
| Numero | Int | Número da habitação ou empresa. |
| Cidade | String | Cidade que o usuário vive ou trabalha. |
| Estado | String | Estado que o usuário vive ou trabalha. |
| Cep | Int | CEP do Usuário. |
| Bairro | String | Bairro do Usuário/empresa que trabalha. |
| Rua | String | Rua do Usuário/empresa que trabalha. |
| Email | String | E-mail do Usuário/empresa que trabalha. |
| Senha | String | Senha do Usuário |
| Situação | String | Situação que o Usuário se encontra em relação ao Sistema. |
| matricula | String | Número da Matrícula do Usuário no sistema. |

Fonte: Elaborado pelos autores

## 3.3 Definição dos Atores

De acordo com Lucidchart (2023), os atores são pessoas, organizações ou sistemas externos que interagem com o sistema. Os atores desempenham papéis específicos e interagem com o sistema para realizar ações, receber informações ou fornecer dados.

A figura a seguir ilustra os atores identificados para o sistema SIGEDEPS.

**Figura ? –** Atores do Projeto SIGEDESP

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores

**Administrador:** O ator "Administrador" desempenha o papel central na administração e supervisão do sistema. Suas responsabilidades incluem a configuração e manutenção do sistema, a gestão de usuários e permissões, a definição de parâmetros do sistema, e a geração de relatórios estratégicos. O Administrador possui o controle global sobre as funcionalidades do sistema, garantindo que este atenda aos requisitos da prefeitura e que as operações sejam conduzidas de maneira eficiente e segura.

**Funcionário:** O ator "Funcionário" representa os usuários operacionais que utilizam o sistema para realizar tarefas específicas relacionadas ao gerenciamento e registro de despesas. Suas responsabilidades podem incluir o cadastro de despesas, a associação de despesas a instituições, a consulta de relatórios e a interação direta com as funcionalidades cotidianas do sistema. O Funcionário desempenha um papel crucial na entrada de dados e na execução de operações diárias para garantir a precisão e integridade das informações registradas no sistema de gerenciamento de despesas.

**Público:** O ator "Público" refere-se aos usuários externos, como cidadãos, contribuintes ou qualquer outra parte interessada que tenha acesso limitado ao sistema. O acesso do Público geralmente é direcionado para consultar informações públicas sobre despesas, orçamentos ou relatórios que a prefeitura opta por tornar disponíveis. Essa categoria de usuários pode visualizar dados específicos, geralmente de forma transparente, contribuindo para a prestação de contas e transparência na gestão financeira da prefeitura. Suas interações são frequentemente limitadas a consultas e visualização de informações, sem permissão para realizar ações de entrada de dados ou configuração no sistema.

**Usuário:** O ator "Usuário" representa coletivamente todos os tipos de usuários envolvidos, incluindo Administradores, Funcionários e o Público. Esse ator abrange todas as interações possíveis no sistema, desde administração e manutenção por parte dos Administradores, até as operações diárias dos Funcionários responsáveis pelo registro e acompanhamento das despesas. Além disso, o Público, como Usuário, pode acessar informações específicas de forma mais restrita, promovendo transparência nas atividades financeiras da prefeitura. O ator "Usuário" encapsula as diversas necessidades e permissões associadas aos diferentes papéis dentro do contexto do sistema de gerenciamento de despesas.

**Sistema:** O ator "Sistema" refere-se à própria aplicação ou plataforma. Este ator não representa um usuário humano, mas sim o conjunto de funcionalidades, processos e lógicas incorporadas no software. O Sistema desempenha um papel central na automação de tarefas, processamento de dados, armazenamento de informações e execução de operações específicas relacionadas ao gerenciamento de despesas. Ele é projetado para suportar os diferentes papéis de usuários (Administrador, Funcionário, Público) e facilitar eficientemente as operações e análises relacionadas às atividades financeiras da prefeitura.

## 3.4 Lista de Casos de Uso

A seguir, apresentamos uma lista consolidada dos casos de uso que foram desenvolvidos até o momento no projeto, proporcionando uma visão abrangente das funcionalidades que já foram concebidas e as que ainda serão implementadas.

Quadro 1 – Lista de Casos de Uso

| **Nº** | **Descrição do Caso de Uso** | **Entrada** | **Caso de Uso** | **Resposta** |
| --- | --- | --- | --- | --- |
| 01 | Administrador Cadastra Administrador | Dados administrador | Cadastra Administrador | Msg01 |
| 02 | Administrador Cadastra Funcionário | Dados funcionário | Cadastra Funcionário | Msg02 |
| 03 | Administrador Cadastra TipoDespesa | Dados tipoDespesa | Cadastra TipoDespesa | Msg03 |
| 04 | Administrador Cadastra TipoInstitucao | Dados tipoIntituicao | Cadastra TipoIntituicao | Msg04 |
| 05 | Administrador Cadastra UnidadeMedida | Dados UnidadeMedida | Cadastra UnidadeMedida | Msg05 |
| 06 | Administrador Cadastra UnidadeConsumidora | Dados UnidadeConsumidora | Cadastra UnidadeConsumidora | Msg06 |
| 07 | Administrador Altera Administrador | Dados administrador alterados | Alterar Administrador | Msg07 |
| 08 | Funcionário Altera Funcionário | Dados funcionário alterados | Alterar Funcionário | Msg08 |
| 09 | Funcionário Cadastra Funcionário | Dados funcionario | Cadastra Funcionario | Msg09 |
| 10 | Funcionário Cadastra TipoDespesa | Dados tipoDespesa | Cadastra TipoDespesa | Msg10 |
| 11 | Funcionário Cadastra TipoInstituicao | Dados tipoIntituicao | Cadastra TipoIntituicao | Msg11 |
| 12 | Funcionário Cadastra UnidadeMedida | Dados UnidadeMedida | Cadastra UnidadeMedida | Msg12 |
| 13 | Funcionário Cadastra UnidadeConsumidora | Dados UnidadeConsumidora | Cadastra UnidadeConsumidora | Msg13 |

Fonte: Elaborado pelos autores

## 3.4. Diagrama de Casos de Uso

O Diagrama de Caso de Uso Geral apresenta uma visão abrangente das diversas interações que cada usuário pode realizar no sistema. Essa representação é valiosa para uma compreensão global das funcionalidades disponíveis para cada ator do sistema.

A seguir, serão apresentados os casos de uso gerais, destacando as ações que cada ator pode executar no sistema. Essa abordagem proporciona uma visão consolidada das capacidades do sistema, facilitando a compreensão das funcionalidades disponíveis para cada usuário.

**Figura ? –** Caso de Uso Geral Administrador

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

**Figura ? –** Caso de Uso Geral FuncionárioDiagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

Figura ? – Caso de Uso Geral Público

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

**Figura ? –** Caso de Uso Geral Sistema

Diagrama

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

## 3.5. Diagrama de Casos de uso individuais

*(2 principais: cadastro, consulta, relatório, agendamento, controle, etc..)*

*Inserir o diagrama e abaixo colocar a documentação do mesmo.*

3.5.1 – Caso de uso: Cadastrar Atendimento

**Figura 02** — Diagrama de Contexto – Visão do Paciente

**

Fonte: Silva, 2016.

**Fluxo Normal**

1-AtorAdm solicita exclusão do registro da pessoa

2-Sistema envia mensagem 06: "Deseja excluir o registro?"

3-AtorAdm confirma exclusão do registro

4-Sistema exclui registro

5-Sistema envia msg07: "Registro excluído com sucesso!"

6-Sistema atualiza banco de dados e retorna ao Caso de Uso "Alterar Pessoa"

**Fluxo Exceção**

2-Sistema envia mensagem 06: "Deseja excluir o registro?"

2.1-AtorAdm não confirma exclusão do registro

2.2-Sistema cancela ação

2.3-Sistema retorna ao caso de Uso "Alterar Pessoa"

Fluxo Alternativo

4-Sistema exclui registro

4.1-Sistema identifica erros de conexão com banco de dados ao excluir registro

4.2-Sistema envia msg07:"Erro s de conexão de BD"

4.3-Sistema retorna ao item 1

## 3.6. Diagrama de Sequência

*(2 principais: cadastro, consulta, relatório, agendamento, controle, etc..)*

*Inserir o diagrama e abaixo colocar a documentação do mesmo.*

## 3.7. Diagrama de Comunicação

*(1 diagrama relevante do projeto)*

## 3.8. Diagrama de Atividade

*(1 diagrama de uma ação específica do sistema)*

# DEFINIÇÃO DA INTERFACE COM O USUÁRIO (UX)

## Descrição de cenário

Um cenário, neste contexto, é uma narrativa ou descrição de uma situação hipotética na qual a persona interage com o produto, serviço ou sistema. Os cenários ajudam a visualizar como a persona pode usar, se beneficiar ou enfrentar desafios ao interagir com a solução proposta. Portanto, um "cenário de persona" seria uma descrição ou história que ilustra como uma persona específica pode se envolver ou se beneficiar do produto, serviço ou sistema em diferentes contextos ou situações.

Para obter uma compreensão mais aprofundada do funcionamento do projeto, foram desenvolvidos dois cenários de uso. Esses cenários oferecem narrativas que ilustram como os usuários poderão interagir com o sistema

**Cenário de Uso 1 -** Registro de Despesas por um Funcionário da Prefeitura

Em um tranquilo escritório da prefeitura, um funcionário autenticado no sistema de gerenciamento de despesas se encontra diante da tarefa de registrar uma nova despesa mensal. Na atmosfera organizada e focada do ambiente de trabalho, o funcionário se sente determinado a preencher os detalhes precisos da despesa, inserindo cuidadosamente informações como data, categoria, valor e descrição. Ao clicar em "Salvar", uma sensação de realização permeia o escritório, pois o sistema valida os dados, confirmando o registro e redirecionando o funcionário de volta à sua rotina com a despesa devidamente documentada.

**Cenário de Uso 2 -** Observação de Despesas e Geração de Relatórios pelo Funcionário da Secretaria da Fazenda

No ambiente operacional da Secretaria da Fazenda, um funcionário, devidamente autenticado no sistema de gerenciamento de despesas, concentra-se na tarefa estratégica de observar as despesas cadastradas e gerar relatórios diretamente na plataforma. Em meio a uma atmosfera técnica e focada, o funcionário explora as funcionalidades do sistema para analisar detalhadamente dados cruciais, incluindo datas, categorias e valores das despesas registradas. Ao utilizar as ferramentas integradas, o funcionário elabora relatórios abrangentes que proporcionam uma visão clara das tendências financeiras municipais. A sensação de eficiência e controle permeia o ambiente quando o funcionário, por meio do sistema, contribui para uma gestão financeira mais informada e eficaz. Este cenário destaca a importância do papel desempenhado pelos membros da Secretaria da Fazenda na análise detalhada e na produção de relatórios estratégicos diretamente na plataforma de gerenciamento de despesas.

## Descrição de personas

Para que um serviço ou produto seja bem-sucedido, é essencial direcioná-lo para um público específico. Imagine criar algo incrível, mas sem saber para quem. É como navegar sem uma bússola, tornando a eficácia e a satisfação do público incertas. Aqui entra o conceito de 'persona'.

Persona é a representação fictícia do seu cliente ideal. Ela é baseada em dados reais sobre comportamento e características demográficas dos seus clientes. Apresenta, também, uma criação de suas histórias pessoais, motivações, objetivos, desafios e preocupações (RESULTADOS DIGITAIS, 2022)

Seguidamente, mostraremos em detalhes as personas desenvolvidas para o projeto SIGEDEPS. Estas representações foram meticulosamente elaboradas, levando em consideração os distintos perfis de usuários que se envolvem diariamente com o software. Esta análise visa não apenas revelar suas características individuais, mas também destacar as necessidades e responsabilidades específicas que cada persona desempenha dentro do sistema. Essa compreensão aprofundada proporcionará uma visão abrangente das demandas do usuário, informando as estratégias e decisões fundamentais ao longo deste trabalho.

**Figura ?** – Persona 1

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

**Figura ?** – Persona 2

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Elaborado pelos autores.

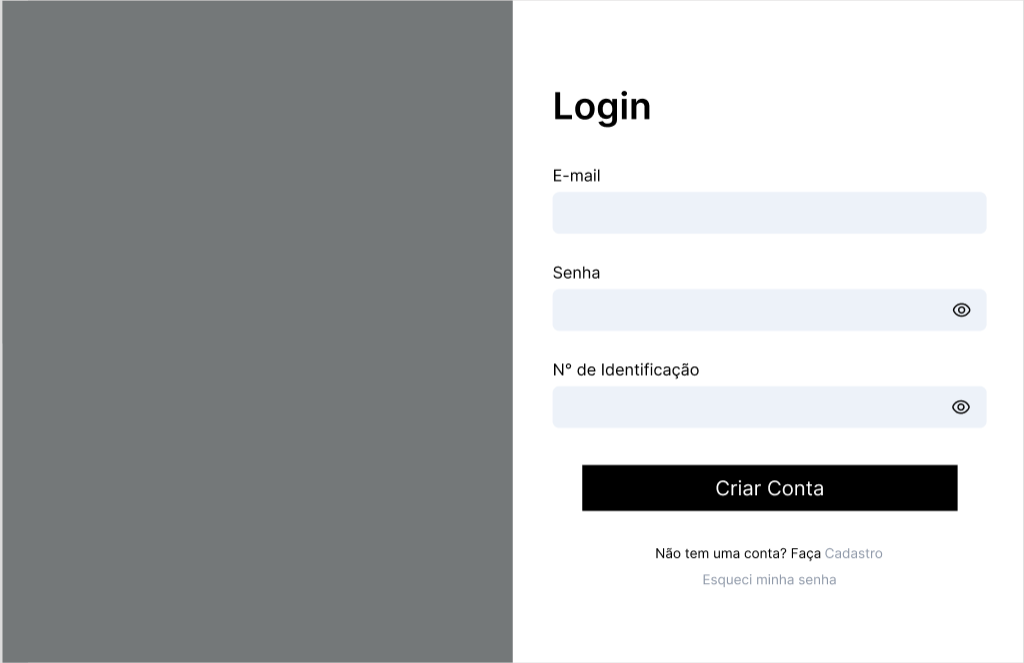
## Esboços de tela (wireframes)

Antes de um site, aplicativo ou produto digital se tornar visualmente atrativo com cores vivas e elementos interativos, existe uma fase fundamental de desenvolvimento que constitui o cerne do processo de design. Essa etapa, frequentemente subestimada por sua aparente simplicidade, desempenha um papel essencial na criação de experiências do usuário coesas e intuitivas. Estamos nos referindo aos wireframes.

De acordo com Kriger (2021), os wireframes são como rascunhos, as primeiras ideias de como será um projeto visual, antes de definir o design e o conteúdo. O wireframe é o que vai mostrar de forma direta a estrutura, a disposição dos elementos e as partes de uma página.

A seguir, serão apresentados os wireframes que foram desenvolvidos para, posteriormente, serem implementados no design do projeto SIGEDESP.

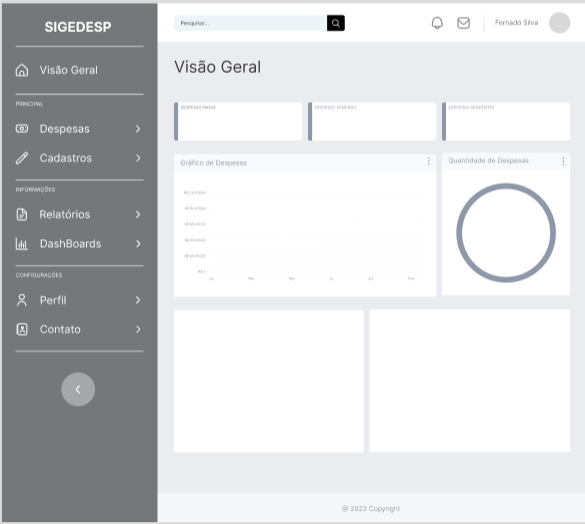
**Figura ?** – Tela de Login



Fonte: Elaborado pelos autores.

Tela de Login: Após o cadastro do usuário no sistema, esta tela permite o acesso mediante a inserção das credenciais, como email, senha e um número de identificação que especifica o tipo de usuário. Essas informações possibilitam o login no sistema.

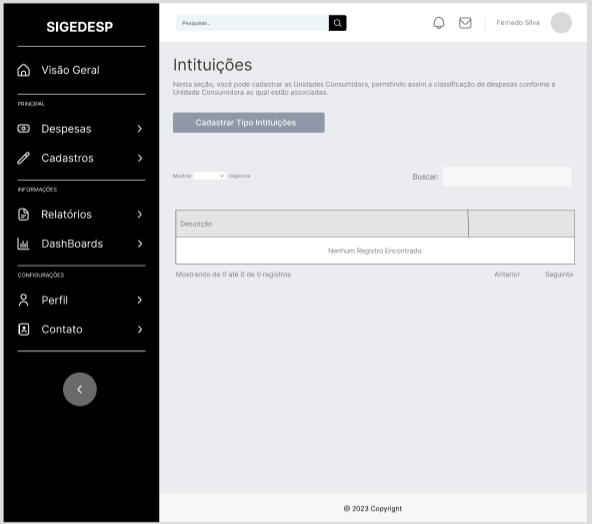
**Figura ?** – Tela Principal Administrador



Fonte: Elaborado pelos autores.

Tela Principal do Administrador: Após o registro de informações no sistema, a tela principal exibirá gráficos detalhados sobre a movimentação de despesas, proporcionando ao administrador uma visão abrangente e atualizada das operações do sistema.

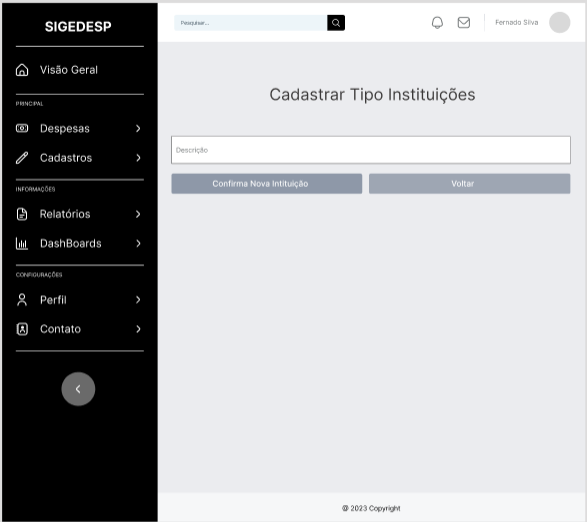
**Figura ?** – Tela Listar



Fonte: Elaborado pelos autores.

Tela de Listagem: Antes do cadastro de um novo registro por parte do administrador ou funcionário, esta tela permite a pesquisa para verificar a existência prévia do registro desejado. Caso não seja encontrado, possibilita o cadastro do mesmo.

**Figura ?** – Tela Cadastro



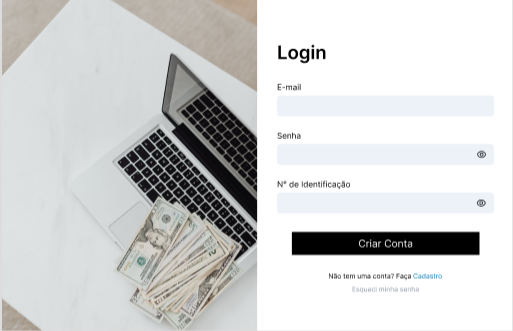
Fonte: Elaborado pelos autores.

Tela de Cadastro: Utilizada como interface padrão para alguns cadastros, como tipo de instituição e tipo de despesa. Esta tela simplifica o processo de cadastro ao requerer apenas uma informação específica.

## Protótipos de tela

A seguir, serão apresentados os protótipos das telas desenvolvidos, os quais serão posteriormente aplicados no design do sistema, como parte integrante deste trabalho de Projeto Integrador (PI).

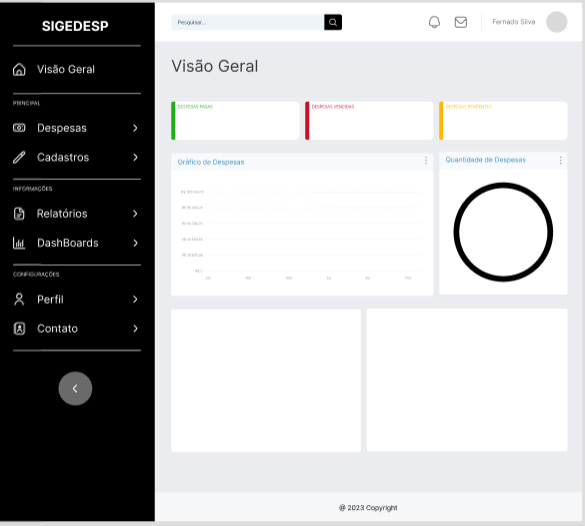
**Figura ?** – Tela de Login



Fonte: Elaborado pelos autores.

Tela de Acesso: Após o usuário ter se cadastrado no sistema, esta tela possibilita a entrada ao solicitar as credenciais, incluindo e-mail, senha e um número de identificação que define o tipo de usuário. Esses dados são essenciais para efetuar o login no sistema.

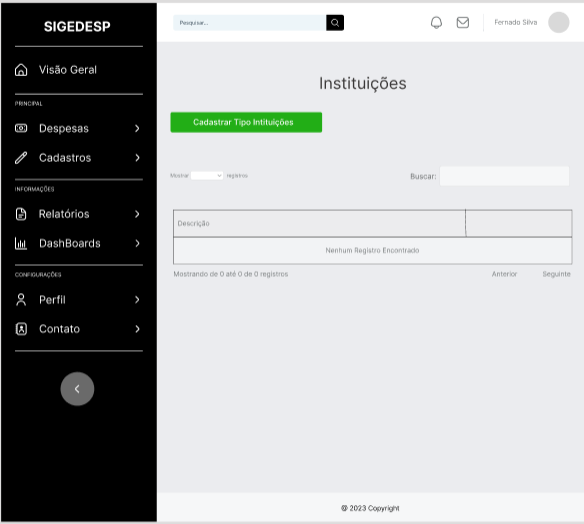
**Figura ?** – Tela Principal Administrador



Fonte: Elaborado pelos autores.

Página Principal do Administrador: Após o registro de informações no sistema, a página principal exibirá gráficos detalhados sobre a movimentação de despesas, proporcionando ao administrador uma visão completa e atualizada das operações do sistema.

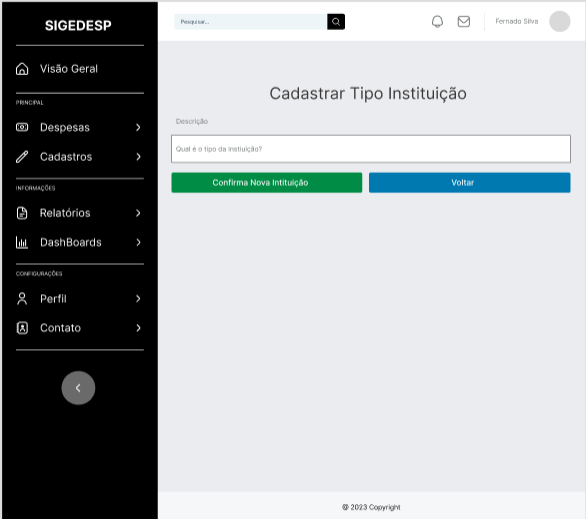
**Figura ?** – Tela Listar



Fonte: Elaborado pelos autores.

Tela de Listagem: Antes que um novo registro seja cadastrado pelo administrador ou funcionário, esta tela permite a pesquisa para verificar a existência prévia do registro desejado. Caso não seja encontrado, possibilita o cadastro do mesmo.

**Figura ?** – Tela Cadastro



Fonte: Elaborado pelos autores.

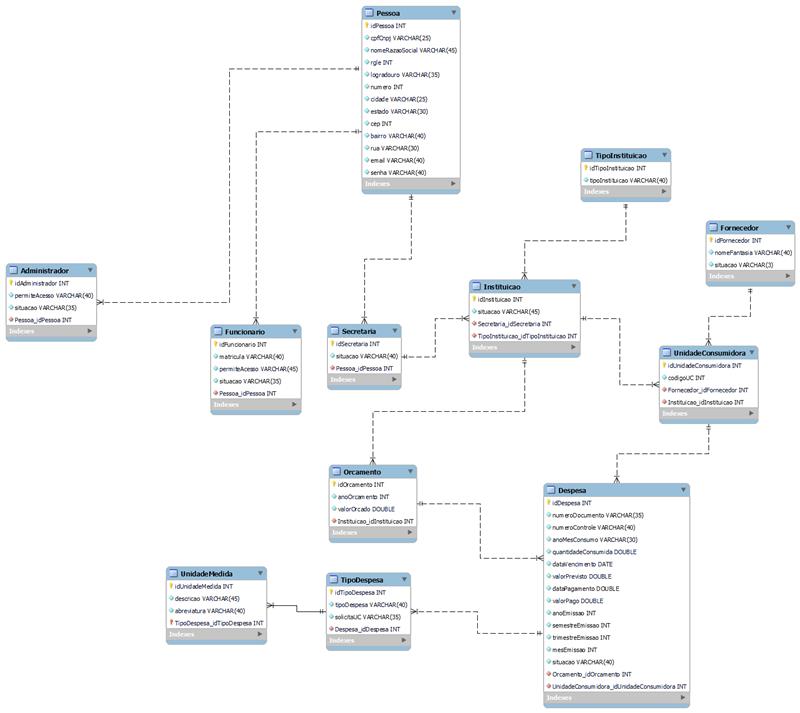
Tela de Cadastro: Utilizada como interface padrão para alguns cadastros, como tipo de instituição e tipo de despesa. Esta tela simplifica o processo de cadastro ao solicitar apenas uma informação específica.

# BANCO DE DADOS

## Modelo Entidade Relacionamento

O Modelo Entidade-Relacionamento (MER) é uma técnica de modelagem de dados que representa as relações entre entidades em um sistema de informação. Ele utiliza entidades para representar objetos ou conceitos do mundo real e relacionamentos para descrever as associações entre essas entidades. Cada entidade é definida por seus atributos, e os relacionamentos especificam como as entidades interagem. O modelo fornece uma visualização abstrata e intuitiva da estrutura do banco de dados, facilitando o design e a compreensão dos relacionamentos complexos dentro de um sistema. O MER é amplamente utilizado no desenvolvimento de sistemas de banco de dados, servindo como uma base para a criação de esquemas de banco de dados relacionais eficientes e coerentes.

Figura ? – Mapeamento do Objeto Relacional



Fonte: Elaborado pelos autores.

## Script das tabelas

O "Script das Tabelas" é uma instrução em linguagem SQL (Linguagem de Consulta Estruturada), que descreve a estrutura de uma tabela em um banco de dados. Discutido no livro "Banco de Dados: Projeto e Implementação", esse script engloba comandos como `CREATE TABLE`, especificando os nomes, tipos de dados e restrições das colunas. Tem como principal objetivo: Criar, Alterar ou Excluir tabelas em um banco de dados.

A seguir, serão expostos os códigos referentes a cada classe no banco de dados do sistema SIGEDESP.

Quadro 1 – Script SQL – Tabela Pessoa

|  |
| --- |
| CREATE TABLE Pessoa (    idPessoa INT NOT NULL PRIMARY KEY,    cpfCnpj VARCHAR(25) NOT NULL,    nome\_razao\_social VARCHAR(45) NOT NULL,    orgao\_expedidor VARCHAR(25),    logradouro VARCHAR(35) NOT NULL,    numero INT NOT NULL,    cidade VARCHAR(25) NOT NULL,    estado VARCHAR(30) NOT NULL,    cep INT NOT NULL,    bairro VARCHAR(40),    rua VARCHAR(30),    email VARCHAR(40)  ); |

Fonte: Elaborado pelos autores.

Quadro 2 – Script SQL – Tabela Administrador

|  |
| --- |
| CREATE TABLE administrador (    idAdministrador INT NOT NULL,    permiteAcesso VARCHAR(40) NOT NULL,    situacao VARCHAR(35) NOT NULL,    Pessoa\_idPessoa INT NOT NULL,    PRIMARY KEY (idAdministrador),    FOREIGN KEY (Pessoa\_idPessoa) REFERENCES pessoa (idPessoa)  ); |

Fonte: Elaborado pelos autores.

Quadro 3 – Script SQL – Tabela Funcionario

|  |
| --- |
| CREATE TABLE funcionario (    idFuncionario INT NOT NULL,    matricula VARCHAR(40) NOT NULL,    permiteAcesso VARCHAR(45) NOT NULL,    situacao VARCHAR(35) NOT NULL,    Pessoa\_idPessoa INT NOT NULL,    PRIMARY KEY (idFuncionario),    FOREIGN KEY (Pessoa\_idPessoa) REFERENCES pessoa (idPessoa)  ); |

Fonte: Elaborado pelos autores.

Quadro 4 – Script SQL – Tabela Secretaria

|  |
| --- |
| CREATE TABLE secretaria (    idSecretaria INT NOT NULL,    situacao VARCHAR(40) NOT NULL,    Pessoa\_idPessoa INT NOT NULL,    PRIMARY KEY (idSecretaria),    FOREIGN KEY (Pessoa\_idPessoa) REFERENCES pessoa (idPessoa)  ); |

Fonte: Elaborado pelos autores.

Quadro 5 – Script SQL – Tabela Orcamento

|  |
| --- |
| CREATE TABLE orcamento (    idOrcamento INT NOT NULL,    anoOrcamento INT NOT NULL,    valorOrcado DOUBLE NOT NULL,    Instituicao\_idInstituicao INT NOT NULL,    PRIMARY KEY (idOrcamento),    FOREIGN KEY (Instituicao\_idInstituicao) REFERENCES instituicao (idInstituicao)  ); |

Fonte: Elaborado pelos autores.

Quadro 6 – Script SQL – Tabela UnidadeMedida

|  |
| --- |
| CREATE TABLE UnidadeMedida (    idUnidade\_Medida INT NOT NULL,    descricao VARCHAR(45) NOT NULL,    abreviatura VARCHAR(40) NOT NULL,    TipoDespesa\_idTipoDespesa INT NOT NULL,    PRIMARY KEY (idUnidade\_Medida),    FOREIGN KEY (TipoDespesa\_idTipoDespesa) REFERENCES tipo\_despesa (idTipoDespesa)  ); |

Fonte: Elaborado pelos autores.

Quadro 7 – Script SQL – Tabela TipoDespesa

|  |
| --- |
| CREATE TABLE TipoDespesa (    idTipoDespesa INT NOT NULL,    tipoDespesa VARCHAR(40) NOT NULL,    solicitaUC VARCHAR(35) NOT NULL,    PRIMARY KEY (idTipoDespesa)  ); |

Fonte: Elaborado pelos autores.

Quadro 8 – Script SQL – Tabela Instuicao

|  |
| --- |
| CREATE TABLE instituicao (    idInstituicao INT NOT NULL,    situacao VARCHAR(40) NOT NULL,    PRIMARY KEY (idInstituicao)  ); |

Fonte: Elaborado pelos autores.

Quadro 9 – Script SQL – Tabela TipoInstituicao

|  |
| --- |
| CREATE TABLE tipoinstituicao (    idTipoInstituicao INTEGER NOT NULL,    tipo\_instituicao VARCHAR(40) NOT NULL,    CONSTRAINT pk\_tipo\_instituicao PRIMARY KEY (id\_tipo\_instituicao)  ); |

Fonte: Elaborado pelos autores.

Quadro 10 – Script SQL – Tabela Fornecedor

|  |
| --- |
| CREATE TABLE fornecedor (    idFornecedor INTEGER NOT NULL,    nome\_fantasia VARCHAR(40) NOT NULL,    situacao VARCHAR(3) NOT NULL,    CONSTRAINT pk\_fornecedor PRIMARY KEY (idFornecedor)  ); |

Fonte: Elaborado pelos autores.

Quadro 11 – Script SQL – Tabela UnidadeConsumidora

|  |
| --- |
| CREATE TABLE UnidadeConsumidora (    idUnidadeConsumidora INTEGER NOT NULL,    codigo\_uc INTEGER NOT NULL,    fornecedor\_id\_fornecedor INTEGER NOT NULL,    instituicao\_id\_instituicao INTEGER NOT NULL,    CONSTRAINT pk\_unidade\_consumidora PRIMARY KEY (idUnidadeConsumidora)  ); |

Fonte: Elaborado pelos autores.

Quadro 12 – Script SQL – Tabela Despesa

|  |
| --- |
| CREATE TABLE despesa (    id\_despesa INTEGER NOT NULL,    numero\_documento VARCHAR(35) NOT NULL,    numero\_controle VARCHAR(40) NOT NULL,    ano\_mes\_consumo VARCHAR(30) NOT NULL,    quantidade\_consumida DOUBLE NOT NULL,    data\_vencimento DATE NOT NULL,    valor\_previsto DOUBLE NOT NULL,    data\_pagamento DOUBLE,    valor\_pago DOUBLE,    ano\_emissao INTEGER NOT NULL,    semestre\_emissao INTEGER NOT NULL,    trimestre\_emissao INTEGER NOT NULL,    mes\_emissao INTEGER NOT NULL,    situacao VARCHAR(40) NOT NULL,    orcamento\_id\_orcamento INTEGER NOT NULL,    unidade\_consumidora\_id\_unidade\_consumidora INTEGER NOT NULL,    CONSTRAINT pk\_despesa PRIMARY KEY (id\_despesa),    CONSTRAINT fk\_despesa\_orcamento FOREIGN KEY (orcamento\_id\_orcamento) REFERENCES orcamento (id\_orcamento),    CONSTRAINT fk\_despesa\_unidade\_consumidora FOREIGN KEY (unidade\_consumidora\_id\_unidade\_consumidora) REFERENCES unidade\_consumidora (id\_unidade\_consumidora)  ); |

Fonte: Elaborado pelos autores.

# ARQUITETURA DE SOFTWARE

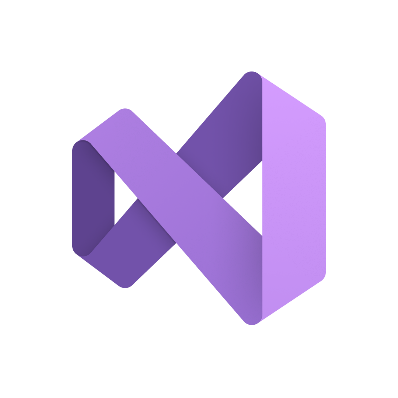
## 6.1 Arquitetura de desenvolvimento

A arquitetura de software é uma disciplina crucial no desenvolvimento de sistemas, visando estruturar o design de forma eficaz para atender requisitos funcionais e não funcionais. Essa abordagem abstrata oferece uma visão conceitual dos componentes e padrões, facilitando a compreensão da equipe e estabelecendo bases sólidas para a evolução do software ao longo do tempo. Além de orientar a construção técnica, a arquitetura abrange aspectos como segurança e desempenho, influenciando diretamente a qualidade e adaptabilidade do sistema resultante. Em resumo, é uma peça fundamental no ciclo de vida do desenvolvimento, moldando soluções alinhadas com as expectativas do usuário e do negócio.

Como destaca Sommerville, a arquitetura do software deve ser definida em um estágio anterior à implementação, pois isso permite uma conexão limpa entre os requisitos do sistema e a elaboração das partes que o compõem.

A seguir, serão mencionadas algumas das ferramentas essenciais na área de arquitetura de software. Estas ferramentas desempenham um papel vital no desenvolvimento e na gestão de sistemas, contribuindo para a criação de estruturas robustas e eficientes. Elas abrangem desde plataformas de modelagem arquitetônica até ambientes de simulação, proporcionando aos arquitetos de software recursos poderosos para visualizar, analisar e otimizar o design de sistemas complexos. Ao integrar essas ferramentas ao processo de desenvolvimento, as equipes podem tomar decisões informadas, melhorar a colaboração e assegurar a conformidade com os requisitos, resultando em soluções de software mais sólidas e alinhadas com as metas do projeto.

**Figura ? -** Visual Studio 2022



Fonte: Microsoft Store

O Visual Studio é um ambiente de desenvolvimento integrado (IDE) da Microsoft, amplamente utilizado para criar aplicativos para Windows, web, dispositivos móveis e serviços em nuvem. Oferece uma ampla gama de recursos, como edição de código, depuração, testes e integração com ferramentas de controle de versão. Com suporte a diversas linguagens de programação, é uma escolha popular entre desenvolvedores devido à sua robustez e integração eficiente com outras tecnologias Microsoft.

**Figura ? -** CSharp (C#)



Fonte: Seeklogo

C# é uma linguagem de programação desenvolvida pela Microsoft, projetada para criar aplicativos Windows e integrar-se à plataforma .NET. Oferece recursos modernos e orientados a objetos, tornando-se uma escolha preferencial para o desenvolvimento de software robusto e escalável. C# é especialmente utilizado no desenvolvimento de aplicativos empresariais, jogos e soluções para a plataforma Microsoft.

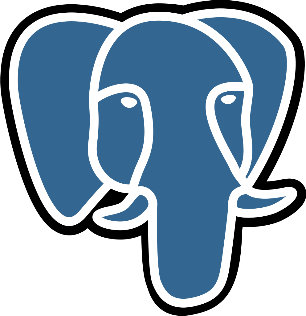
**Figura ? –** Bootstrap



Fonte: GitHub

Bootstrap é um framework front-end de código aberto que simplifica o processo de design e desenvolvimento de interfaces web responsivas. Baseado em HTML, CSS e JavaScript, o Bootstrap oferece uma variedade de componentes e estilos pré-definidos, facilitando a criação de páginas web atraentes e compatíveis com dispositivos de diferentes tamanhos.

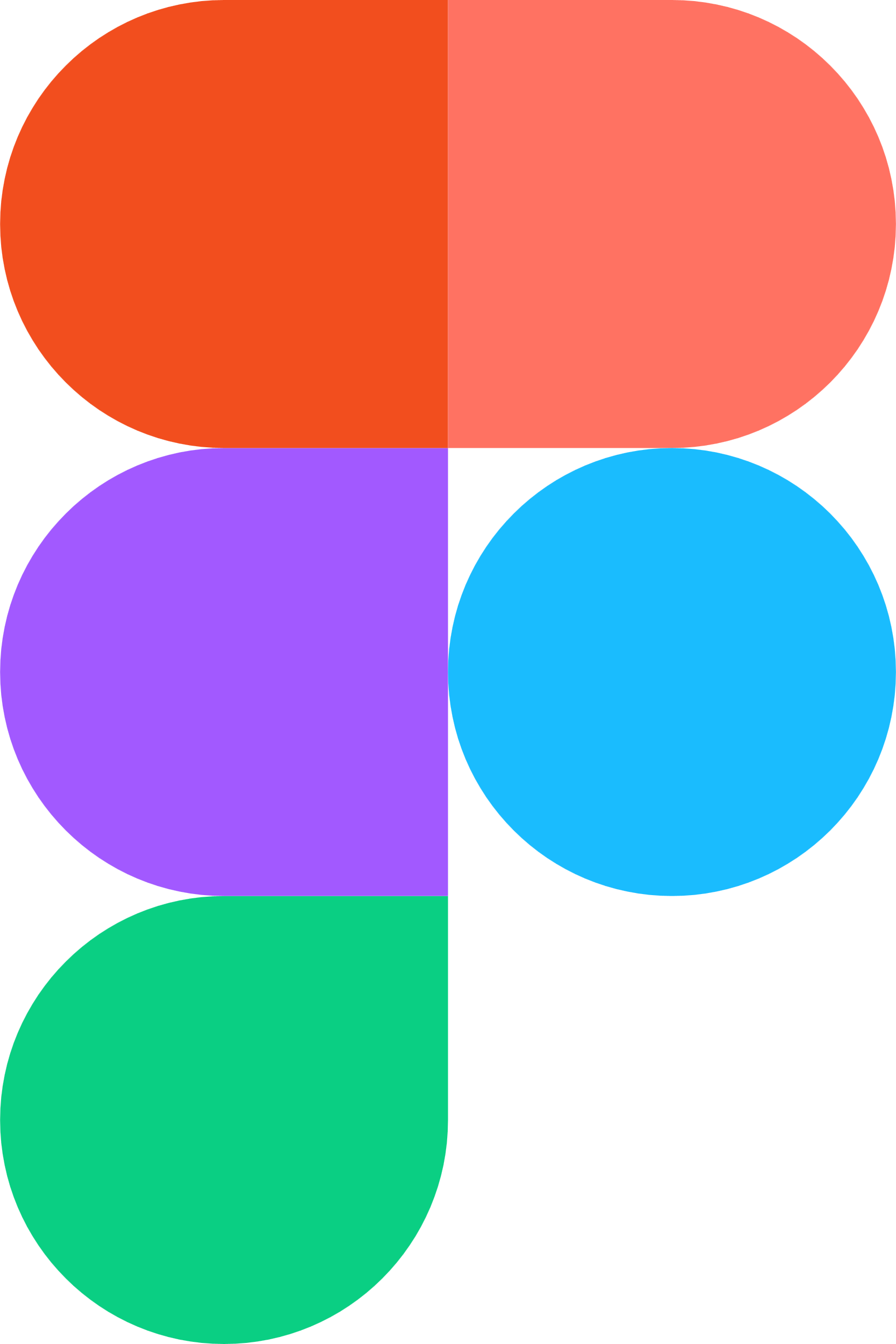
**Figura ? –** PostgreSQL



Fonte: Wikipédia

PostgreSQL é um sistema de gerenciamento de banco de dados relacional de código aberto conhecido por sua confiabilidade, desempenho e recursos avançados. Amplamente utilizado em aplicações web e empresariais, o PostgreSQL suporta consultas complexas, transações ACID, e extensões, tornando-se uma escolha robusta para ambientes de dados críticos.

**Figura ? –** Figma



Fonte: Wikipédia

Figma é uma ferramenta de design colaborativo baseada na nuvem, adequada para criar interfaces de usuário, protótipos e designs interativos. Sua abordagem colaborativa permite que equipes trabalhem simultaneamente em projetos, facilitando a comunicação e a iteração rápida no processo de design.

**Figura ? –** Asta



Fonte: Astah

Astah é um ambiente de modelagem e análise arquitetônica, auxiliando arquitetos de software na visualização e organização de sistemas complexos. Essa ferramenta permite criar representações visuais da arquitetura de software, facilitando a compreensão e comunicação entre membros da equipe e partes interessadas.

**Figura ? –** Azure DevOps

Logotipo

Descrição gerada automaticamente

Fonte: Visure Solutions

Azure DevOps é uma plataforma abrangente da Microsoft que oferece um conjunto integrado de ferramentas para planejamento, colaboração, entrega contínua e monitoramento de aplicações. Inclui recursos como repositórios de código, pipelines de integração contínua, gerenciamento de projetos e ferramentas de rastreamento de problemas. Essa solução facilita a implementação de práticas DevOps, promovendo a colaboração eficiente entre desenvolvedores, operações e equipes de teste.

**Figura ? –** GitHub



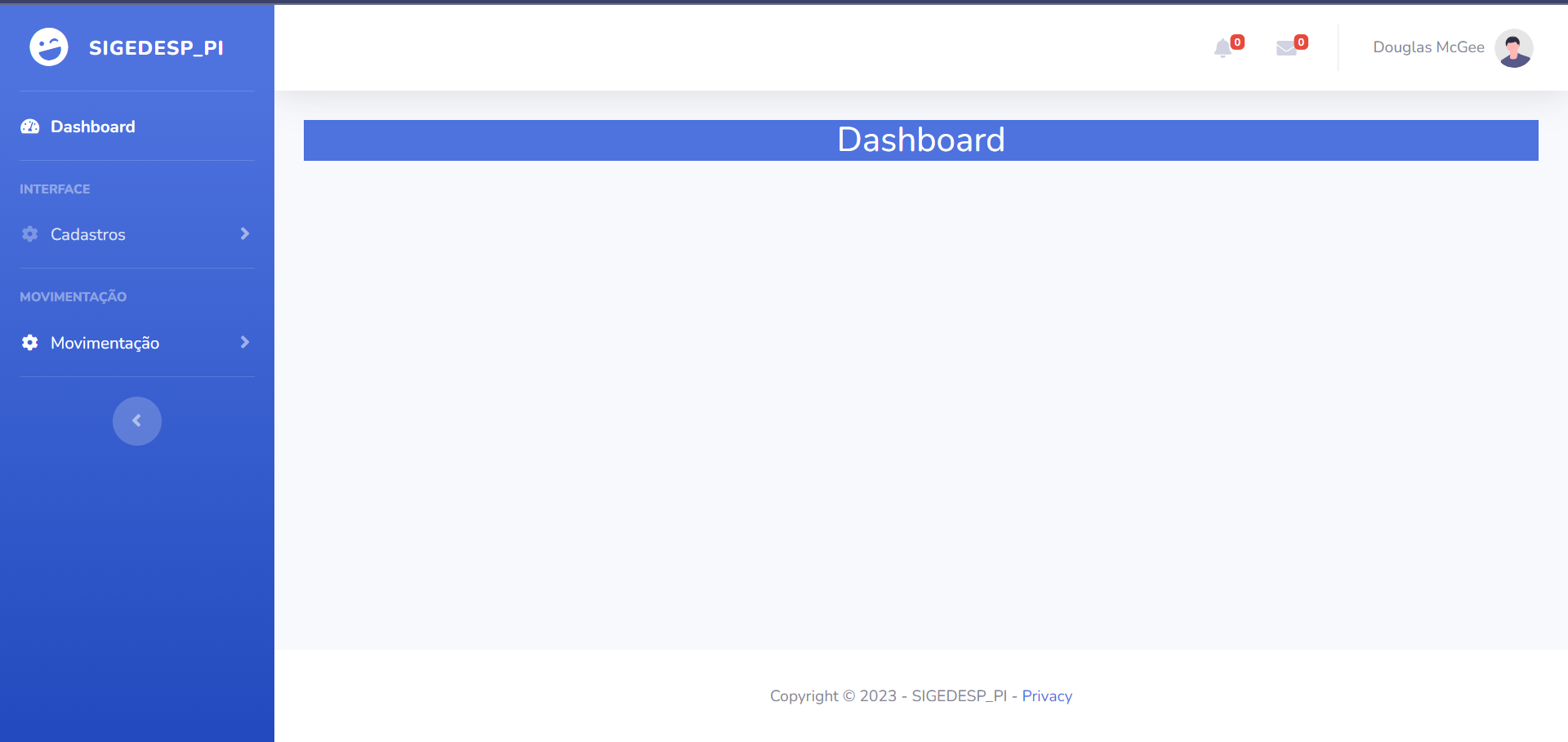
Fonte: GitHub

O GitHub é uma plataforma de hospedagem de código-fonte e colaboração baseada em Git. Ele permite que desenvolvedores trabalhem juntos em projetos, controlando versões, gerenciando código-fonte, revisando alterações e facilitando a colaboração por meio de recursos como pull requests e issues. O GitHub é amplamente utilizado para projetos de código aberto e privados, tornando-se essencial no desenvolvimento de software colaborativo.

## 6.2 Telas do sistema

Neste segmento, serão delineadas as interfaces elaboradas para o Sistema de Gerenciamento de Despesas (SIGEDESP).

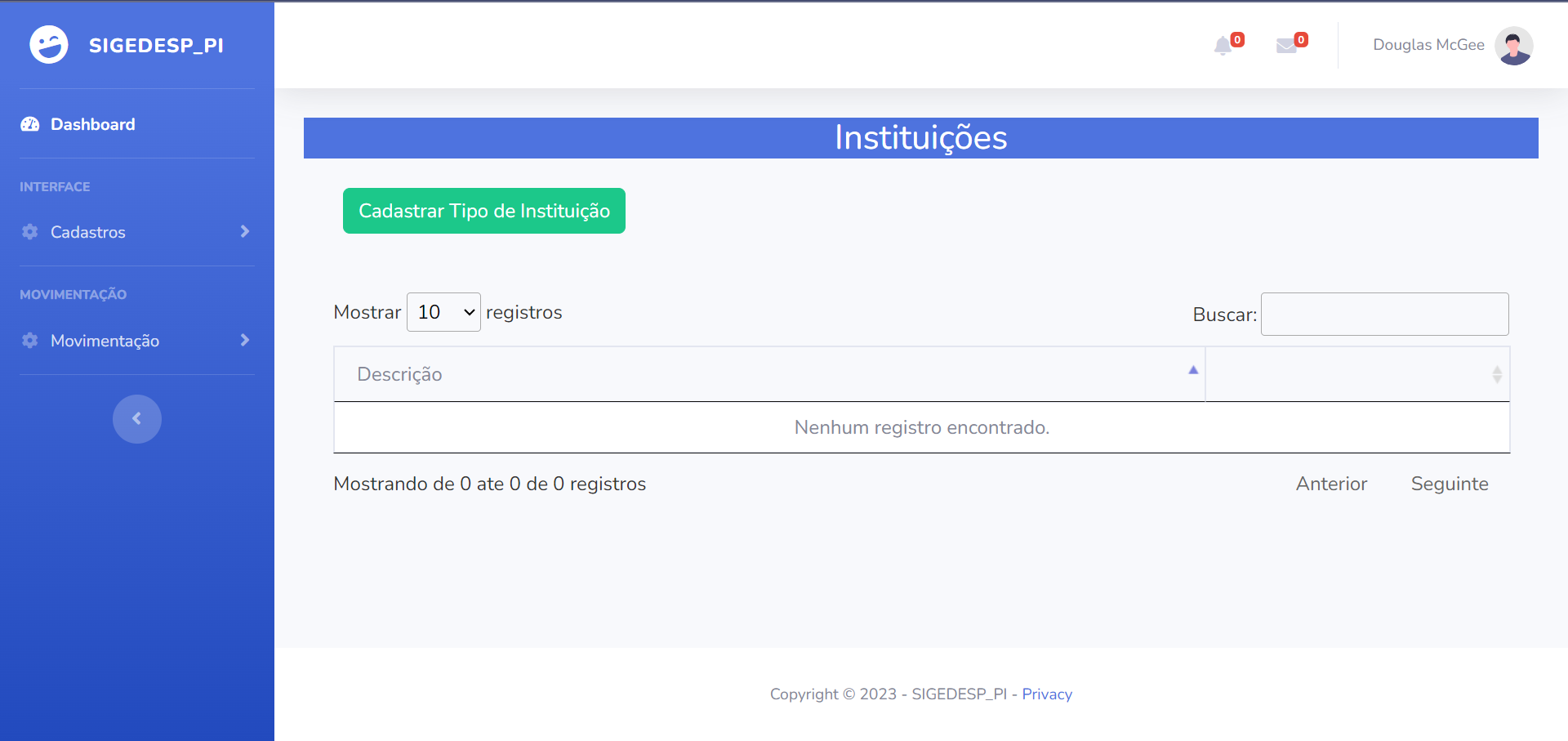
Figura ? – Tela Principal Administrador



Fonte: Elaborado pelos autores.

A tela principal do administrador é como o centro de controle, mostrando de forma clara e visualmente atraente as informações cruciais sobre as despesas do sistema. Nela, você encontrará gráficos que revelam o orçamento disponível, a quantidade de despesas cadastradas e o status delas.

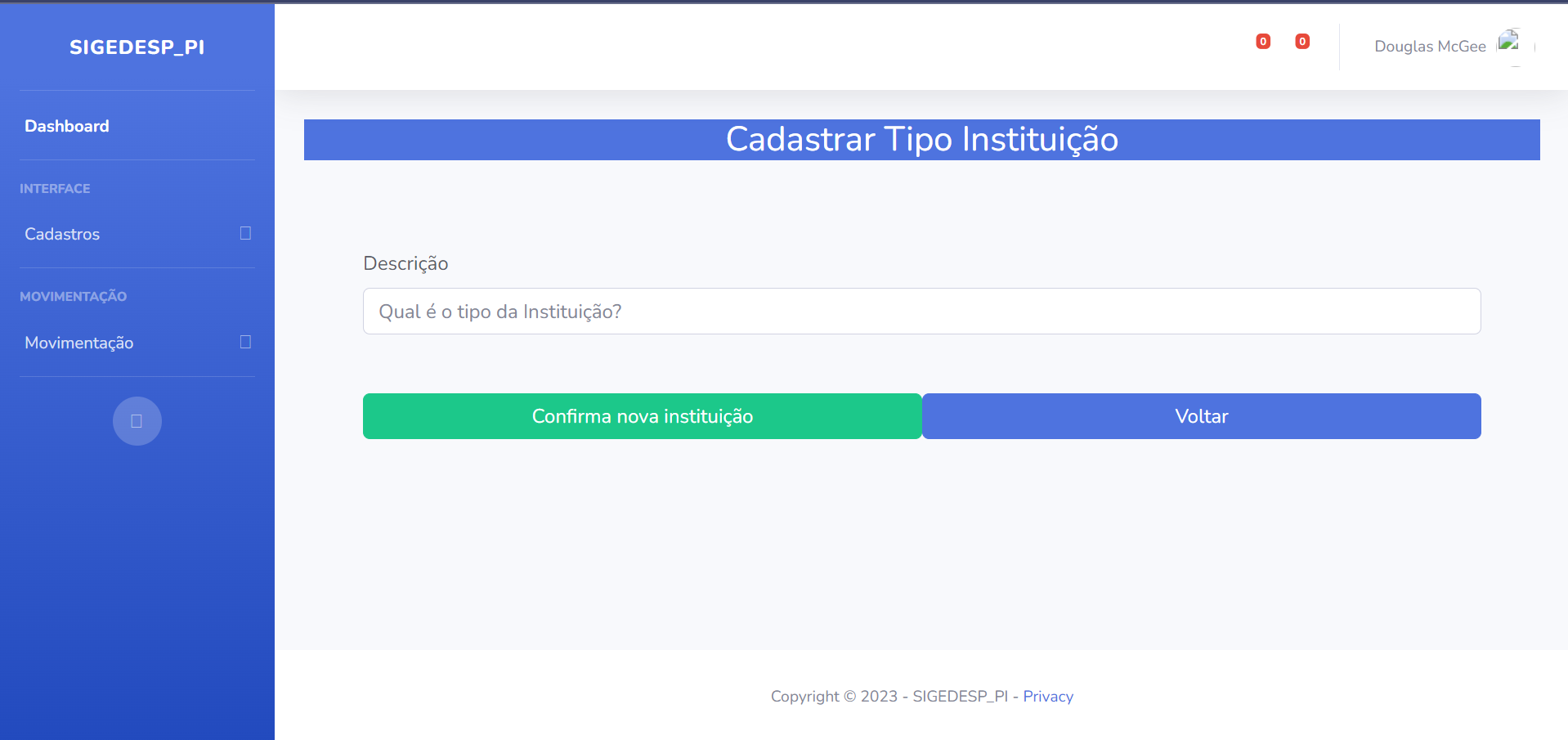
Figura ? – Tela Listagem



Fonte: Elaborado pelos autores.

A tela que antecede o cadastro é uma etapa prévia importante. Ela permite que o administrador ou funcionário confira se o registro desejado já está no sistema. Se não estiver, eles podem acessar a tela de cadastro para adicioná-lo.

Figura ? – Tela Cadastro



Fonte: Elaborado pelos autores.

A tela de cadastro é a porta de entrada para novos elementos no sistema. Até o momento, desenvolvemos telas para cadastrar informações como Tipo de Instituição, Tipo de Despesa, Unidade de Medida e Unidade Consumidora. Essas telas simplificam o processo, pedindo informações específicas e contribuindo para a organização eficiente do sistema.

# CONCLUSÃO

Em síntese, este trabalho de graduação representa um marco significativo na jornada de desenvolvimento do nosso sistema de gestão de despesas. Ao longo deste estudo, pudemos aprimorar a eficiência e a usabilidade, tornando-o uma ferramenta valiosa para monitorar e gerenciar despesas de forma precisa.

Olhando para o futuro, nossa equipe está comprometida em levar este projeto a novos patamares no próximo ano. Planejamos expandir a implementação, introduzindo outras telas e funcionalidades que aprimorarão ainda mais a experiência do usuário. Além disso, dedicaremos esforços para aprofundar nosso conhecimento em APIs, visando uma transição eficiente do sistema para esse modelo, o que promoverá uma integração mais flexível e expansível.

A meta final é instituir nosso projeto na prefeitura, proporcionando aos funcionários a capacidade de utilizar o sistema para realizar a gestão de suas despesas de maneira eficaz. Estamos cientes da importância de fornecer uma solução que atenda às necessidades específicas da administração municipal, e continuaremos trabalhando arduamente para garantir que nosso sistema contribua significativamente para otimizar os processos financeiros e administrativos.

Assim, este não é apenas o encerramento de um trabalho de graduação, mas o início de um compromisso contínuo com a evolução e implementação prática de soluções inovadoras. Estamos ansiosos para os desafios e conquistas que nos aguardam no próximo capítulo deste emocionante projeto.

# REFERÊNCIAS

Resultados Digitais. **Persona: o que é, como definir e por que criar uma para sua empresa [+ exemplos práticos e um gerador]** – Siqueira, F. (2022). Disponível em: <https://resultadosdigitais.com.br/marketing/persona-o-que-e/> Acesso em: 27, 11, 2023.

Miro. **O que é UML?** (2023). Disponível em: <https://miro.com/pt/diagrama/o-que-e-uml/> Acesso em: dia, mês, ano.

Lucidchart. **Diagrama de caso de uso UML: O que é, como fazer e exemplos (2023).** Disponível em: <https://www.lucidchart.com/pages/pt/diagrama-de-caso-de-uso-uml> Acesso em: 28, 11, 2023.

Guedes, **R. UML 2: Uma abordagem prática** – 3° edição. 2018.

Kringer, J. **Wireframe: O QUE É, QUAL SUA FINALIDADE E QUAIS OS TIPOS? (2021).** Disponível em: <https://kenzie.com.br/blog/wireframe/> Acesso em: 29, 11, 2023.

Visure Solutions**. O que são Requisitos Não Funcionais: Exemplos, Definição, Guia Completo (2022).** Disponível em: <https://visuresolutions.com/pt/blog/non-functional-requirements/> Acesso em: 15, 11, 2023.

Sommerville, I. **Engenharia de Software** – 9° edição. 2011.

MACHADO, F. N. R. **Banco de Dados: Projeto e Implementação**. 4. ed. São Paulo: Érica, 2020.

LUCIDCHART. **O que é um modelo de Banco de Dados?** 2023. Disponível em: <https://www.lucidchart.com/pages/pt/o-que-e-um-modelo-de-banco-de-dados#:~:text=Alguns%20dos%20mais%20comuns%20s%C3%A3o%3A%201%20Modelo%20de,Modelo%20entidade-atributo-valor%208%20Esquema%20em%20estrela%20Mais%20itens> Acesso em: 30 nov. 2023.