

INSTITUTO FEDERAL DE EDUCAÇÃO CIÊNCIA E TECNOLOGIA DE MINAS  
GERAIS – *CAMPUS* SÃO JOÃO EVANGELISTA  
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Eduardo Antônio Maia Valério; Marcos Antônio da Silva

**ONEFLOW: DESENVOLVIMENTO DE UM SISTEMA WEB PARA GESTÃO DA  
EMPRESA JÚNIOR ONEBIT**

São João Evangelista - MG

2025

EDUARDO ANTÔNIO MAIA VALÉRIO; MARCOS ANTÔNIO DA SILVA

**ONEFLOW: DESENVOLVIMENTO DE UM SISTEMA WEB PARA GESTÃO DA  
EMPRESA JÚNIOR ONEBIT**

Trabalho de conclusão de curso apresentado ao Curso Bacharelado em Sistemas de Informação do Instituto Federal de Minas Gerais – *Campus* São João Evangelista para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Me. Rosinei Soares de Figueiredo  
Coorientador: Prof. Dr. Wesley Gomes de Almeida

São João Evangelista - MG

2025

---

S586o Silva, Marcos Antônio da.  
OneFlow: desenvolvimento de um sistema web para gestão da  
Empresa Júnior OneBit/Marcos Antônio da Silva, Eduardo Antônio  
Maia Valério– 2025.  
55f.: il.

Orientador: Me. Rosinei Soares de Figueiredo.  
Coorientador: Dr. Wesley Gomes de Almeida.  
Trabalho de Conclusão de Curso (bacharelado em Sistemas de  
Informação) – Instituto Federal Minas Gerais. *Campus* São João  
Evangelista, 2025.

1. Sistema de gestão personalizado. 2. Empresa Júnior. 3. Sistema  
web. I. Silva, Marcos Antônio da. II. Valério, Eduardo Antônio Maia.  
III. Instituto Federal de Minas Gerais *Campus* SJE. IV. Título.

CDD 005.1

---


Catálogo: Esther Soares Cunha - CRB-6/4333

EDUARDO ANTÔNIO MAIA VALÉRIO; MARCOS ANTÔNIO DA SILVA

**ONEFLOW: DESENVOLVIMENTO DE UM SISTEMA WEB PARA GESTÃO DA  
EMPRESA JÚNIOR ONEBIT**


Trabalho de conclusão de curso apresentado ao  
Curso Bacharelado em Sistemas de  
Informação do Instituto Federal de Minas  
Gerais – *Campus* São João Evangelista para  
obtenção do grau de Bacharel em Sistemas de  
Informação.

Aprovado em: 06/02/2025 pela banca examinadora:

Documento assinado digitalmente  
 ROSINEI SOARES DE FIGUEIREDO  
Data: 25/02/2025 07:58:03-0300  
Verifique em <https://validar.iti.gov.br>


---

Prof. Me. Rosinei Soares de Figueiredo (Orientador) – IFMG – SJE

Documento assinado digitalmente  
 WESLEY GOMES DE ALMEIDA  
Data: 26/02/2025 18:46:52-0300  
Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Wesley Gomes de Almeida (Coorientador) – IFMG - SJE

Documento assinado digitalmente  
 FABIO RODRIGUES MARTINS  
Data: 26/02/2025 21:03:08-0300  
Verifique em <https://validar.iti.gov.br>

---

Prof. Dr. Fábio Rodrigues Martins (Convidado) – IFMG - SJE

## RESUMO

O presente trabalho trata do desenvolvimento de um sistema de gestão personalizado para a Empresa Júnior OneBit, com o objetivo de centralizar informações e otimizar os processos internos, como o gerenciamento de membros, clientes, projetos, reuniões e *links* para documentos importantes. Para alcançar tal objetivo, foi realizada uma análise das necessidades específicas da empresa, identificando os principais desafios enfrentados na gestão. Observou-se a necessidade de uma solução que automatizasse e organizasse as informações, reduzindo o tempo gasto em tarefas administrativas e aumentando a eficiência dos processos. A metodologia utilizada foi de natureza aplicada e qualitativa, com foco na resolução de problemas práticos e na obtenção de *feedback* direto dos membros da empresa durante o processo de testes. Os resultados obtidos demonstraram que o sistema possui uma interface intuitiva, é fácil de usar e atende às demandas da empresa, facilitando a gestão de informações e o acompanhamento das atividades. O *feedback* recebido também indicou que o sistema tem potencial para se tornar uma ferramenta essencial para a organização e o crescimento da empresa. Por fim, conclui-se que todas as etapas do projeto foram realizadas de forma satisfatória, e o sistema desenvolvido representa uma solução prática e eficiente para os desafios enfrentados pela Empresa Júnior OneBit, com perspectivas de futuras melhorias e expansões.

**Palavras-chave:** Sistema de gestão personalizado. Empresa Júnior. Sistema web.

## **ABSTRACT**

This work deals with the development of a customized management system for the Junior Company OneBit, with the aim of centralizing information and optimizing internal processes, such as managing members, clients, projects, meetings, and links to important documents. To achieve this goal, an analysis of the company's specific needs was conducted, identifying the main challenges faced in management. The need for a solution that would automate and organize information was observed, reducing the time spent on administrative tasks and increasing process efficiency. The methodology used was applied and qualitative in nature, focusing on solving practical problems and obtaining direct feedback from the company's members during the testing process. The results showed that the system has an intuitive interface, is easy to use, and meets the company's demands, facilitating information management and activity tracking. The feedback also indicated that the system has the potential to become an essential tool for the organization and growth of the company. Finally, it is concluded that all stages of the project were carried out satisfactorily, and the developed system represents a practical and efficient solution to the challenges faced by the Junior Company OneBit, with prospects for future improvements and expansions.

**Keywords:** Customized management system. Junior Company. Web system.

## **LISTA DE QUADROS**

Quadro 1 – Requisitos Funcionais e Não Funcionais.....	26
--	----

## LISTA DE FIGURAS

Figura 1 – Diagrama de caso de uso.....	28
Figura 2 – Diagrama entidade relacionamento.....	29
Figura 3 – Tela de Login .....	31
Figura 4 – Tela Principal .....	32
Figura 5 – Tela de Gestão de Usuário .....	33
Figura 6 – Tela de Reuniões .....	33
Figura 7 – Tela de Documentos.....	34
Figura 8 – Tela de Clientes.....	35
Figura 9 – Tela do Setor de Projetos .....	35
Figura 10 – Formulário Adicionar Novo Projeto .....	36
Figura 11 – Formulário Nova Reunião.....	37
Figura 12 – Formulário Adicionar Novo Documento .....	37
Figura 13 – Formulário Adicionar Novo Cliente .....	38
Figura 14 – Formulário Adicionar Novo Usuário .....	39
Figura 15 – Tela de Login no Dispositivo Móvel .....	39
Figura 16 – Tela Principal no Dispositivo Móvel .....	40
Figura 17 – Menu do Sistema no Dispositivo Móvel.....	41

## LISTA DE ABREVIATURAS E SIGLAS

- API - *Application Programming Interface* (Interface de Programação de Aplicação)
- BJ - Brasil Júnior
- CSS - *Cascading Style Sheets* (Folhas de Estilo em Cascata)
- DER - Diagrama Entidade Relacionamento
- EJ - Empresa Júnior
- EJs - Empresas Juniores
- ESSEC - *L'École Supérieure des Sciences Economiques et Commerciales*
- HTML - *Hyper Text Markup Language* (Linguagem de Marcação de Hipertexto)
- IDE - *Integrated Development Environment* (Ambiente de Desenvolvimento Integrado)
- IFMG - SJE - Instituto Federal de Minas Gerais - *Campus São João Evangelista*
- JS - *JavaScript*
- JVM - *Java Virtual Machine* (Máquina Virtual Java)
- MEJ - Movimento Empresa Júnior
- RH - Recursos Humanos
- SQL - *Structure Query Language* (Linguagem de Consulta Estruturada)
- TI - Tecnologia da Informação
- TS - *TypeScript*
- UI - *User Interface* (Interface de Usuário)
- W3C – *World Wide Web Consortium* (Consórcio Web Mundial)

## SUMÁRIO

<b>1 INTRODUÇÃO .....</b>	<b>11</b>
<b>2 FUNDAMENTAÇÃO TEORICA.....</b>	<b>14</b>
<b>2.1 Empresa Júnior .....</b>	<b>14</b>
<b>2.2 Movimento Empresa Júnior .....</b>	<b>14</b>
<b>2.3 OneBit.....</b>	<b>15</b>
<b>2.4 Softwares e Sistemas de Gestão.....</b>	<b>15</b>
<b>2.5 Sistemas Web .....</b>	<b>16</b>
<b>2.6 Ferramentas e Tecnologias .....</b>	<b>17</b>
<b>2.6.1 HTML.....</b>	<b>17</b>
<b>2.6.2 CSS .....</b>	<b>18</b>
<b>2.6.3 JavaScript.....</b>	<b>18</b>
<b>2.6.4 TypeScript .....</b>	<b>19</b>
<b>2.6.5 React.....</b>	<b>19</b>
<b>2.6.6 Java .....</b>	<b>20</b>
<b>2.6.7 Spring Boot .....</b>	<b>21</b>
<b>2.6.8 MySQL .....</b>	<b>21</b>
<b>2.6.9 Trello .....</b>	<b>21</b>
<b>2.6.10 Git e Github.....</b>	<b>22</b>
<b>2.6.11 Figma .....</b>	<b>22</b>
<b>2.6.12 IDE .....</b>	<b>22</b>
<b>2.7 Trabalhos Correlatos .....</b>	<b>23</b>
<b>3 METODOLOGIA.....</b>	<b>25</b>
<b>3.1 Natureza da pesquisa .....</b>	<b>25</b>
<b>3.2 População e amostra.....</b>	<b>25</b>
<b>3.3 Métodos e procedimentos.....</b>	<b>25</b>
<b>3.3.1 Levantamento de requisitos .....</b>	<b>26</b>
<b>3.3.2 Modelagem .....</b>	<b>27</b>
<b>3.3.3 Desenvolvimento .....</b>	<b>29</b>
<b>4 RESULTADOS E DISCUSSÕES .....</b>	<b>31</b>
<b>4.1 Descrição do sistema.....</b>	<b>31</b>
<b>4.2 Testes realizados .....</b>	<b>42</b>
<b>5 CONSIDERAÇÕES FINAIS.....</b>	<b>44</b>

<b>REFERÊNCIAS .....</b>	<b>45</b>
<b>APÊNDICES .....</b>	<b>48</b>

## 1 INTRODUÇÃO

O mundo contemporâneo é marcado por uma acelerada transformação digital, em que a tecnologia está presente em vários aspectos da vida, inclusive no ambiente empresarial. A constante evolução de ferramentas digitais impacta significativamente a forma como as empresas operam, se organizam e se relacionam com o mercado.

As vantagens proporcionadas pelo avanço tecnológico e a transformação digital estão na redução de custos, ganhos em agilidade e identificação de gastos excessivos de recursos. Com o aumento da produtividade e o fluxo de trabalho automatizado, o engajamento da equipe cresce, o retrabalho diminui e a equipe passa a entregar mais com os mesmos recursos. Assim, com o ganho de tempo pode-se investir mais em estratégias e, ao padronizar e automatizar tarefas que antes eram feitas manualmente, os erros diminuem e a qualidade aumenta (MATOS, 2021).

Muitas empresas de pequeno porte ainda não possuem sistemas informatizados, ou seja, várias de suas tarefas e controles são feitos por meio de papeladas intermináveis. Sem um sistema, a empresa acaba estruturando processos que podem gerar retrabalhos, erros e perda de dados, isso porque planilhas e documentos impressos são ferramentas extremamente suscetíveis a erros. Além disso, não funcionam bem em um ambiente de trabalho em equipe (LEÃO, 2024).

O *software* de gestão é uma tecnologia complexa que auxilia na organização dos negócios e na melhoria da eficiência empresarial. Pode variar desde sistemas de gerenciamento de tarefas até *softwares* que ajudam a gerenciar históricos e dados de clientes, bem como departamentos de recursos humanos (RH). Com diferentes tipos de *software* de gerenciamento, está se tornando cada vez mais popular para os gestores a utilização desses recursos para maior produtividade e organização (EQUIPE TOTVS, 2023).

Esse tipo de sistema desempenha um papel essencial no mundo empresarial moderno. Ele auxilia as empresas a acessarem informações de forma rápida e prática, além de oferecer ferramentas úteis para organizar e melhorar a gestão. Com isso, há menos necessidade de atualizações manuais e tarefas repetitivas, permitindo que a equipe foque em atividades mais produtivas (EQUIPE TOTVS, 2023).

Segundo a TOTVS (2023), com os avanços tecnológicos e as mudanças no mercado, a gestão de empresas tornou-se ainda mais complexa do que no início do século. Os *softwares* acompanham essas necessidades e agregam benefícios como:

- a) **Integração dos setores e informações:** o gerenciamento otimizado das empresas passa por uma comunicação rápida e assertiva entre todos os setores;
- b) **Agilizar a comunicação das equipes:** mesmo quando as informações não estão disponíveis na plataforma de gestão, os colaboradores responsáveis estão lá, portanto, o contato é mais rápido;
- c) **Redução de custos:** acesso a dados importantes e comunicação ágil por si só já representa diminuição de custos, uma vez que nas operações corporativas tempo é dinheiro.

Esses benefícios não só permitem um monitoramento mais acurado, como também geram *insights* mais precisos para aprimorar os processos (EQUIPE TOTVS, 2023). Neste contexto, torna-se evidente a necessidade das empresas, inclusive aquelas de pequeno porte, adotarem sistemas informatizados para otimizar suas operações e promover um ambiente de trabalho mais eficiente e produtivo.

Diante disso, a OneBit é uma Empresa Júnior (EJ) sem fins lucrativos composta por estudantes de graduação do curso de Sistemas de Informação do Instituto Federal de Minas Gerais - *Campus* São João Evangelista (IFMG - SJE). No entanto, apesar de sua missão inspiradora de proporcionar aos alunos uma experiência prática no ambiente empresarial, a OneBit enfrenta desafios comuns a muitas organizações. Um desses desafios é a falta de compartilhamento de informações entre os setores. Essas questões podem afetar negativamente a eficiência operacional e a qualidade dos serviços prestados pela Empresa Júnior.

Diante desse cenário, o objetivo geral deste trabalho é desenvolver um sistema web de gestão personalizado para a OneBit. Este sistema, com o nome de OneFlow, visa suprir as necessidades específicas da empresa, oferecendo uma solução tecnológica para realizar o gerenciamento de membros, clientes, projetos, reuniões e *links* de acesso para documentos importantes.

Para atingir o objetivo geral deste trabalho foram definidos os seguintes objetivos específicos:

- a) Analisar requisitos que serão necessários para a implementação do sistema;
- b) Realizar a modelagem e prototipação do sistema com base nos requisitos;
- c) Desenvolver o sistema baseado nos requisitos previamente adquiridos;
- d) Realizar testes juntamente com os membros da OneBit para garantir o bom funcionamento do sistema.

Ao implementar um sistema de gestão sob medida, espera-se proporcionar à OneBit uma série de benefícios, incluindo integração de informações entre os setores, agilização da comunicação das equipes e redução de custos operacionais. Optar por um sistema sob medida, em vez de utilizar um sistema já pronto, justifica-se pela necessidade de atender às especificidades e particularidades do negócio. Um sistema genérico pode não contemplar as demandas exclusivas da empresa, limitando sua capacidade de otimizar processos e gerar valor. Além disso, sistemas prontos muitas vezes incluem funcionalidades desnecessárias e carecem de flexibilidade, o que pode gerar custos adicionais e dificultar adaptações às mudanças estratégicas ou operacionais. Um sistema sob medida, por outro lado, é projetado para alinhar-se totalmente aos objetivos e fluxos internos da OneBit, garantindo maior eficiência e personalização.

Ao final deste estudo, espera-se não apenas oferecer uma solução tecnológica inovadora para a OneBit, mas também contribuir para o avanço das práticas de gestão em empresas juniores e, conseqüentemente, para o desenvolvimento pessoal e profissional dos alunos envolvidos neste projeto.

## 2 FUNDAMENTAÇÃO TEÓRICA

Esta seção apresentará os conceitos, ferramentas e pesquisas utilizados para o desenvolvimento deste trabalho.

### 2.1 Empresa Júnior

Uma Empresa Júnior é considerada uma associação sem fins lucrativos, formada e administrada apenas por estudantes universitários de graduação. Essa empresa é criada com o objetivo de dar aos estudantes uma experiência prática de uma imersão em negócios, assim, permitindo que eles possam aplicar os conhecimentos adquiridos nas aulas para situações reais do cotidiano.

Apesar das Empresas Juniores (EJs) serem geridas por estudantes que contam com o apoio dos professores, elas trabalham de forma profissional seguindo os princípios de uma empresa convencional. As EJs de tecnologia da informação (TI) por exemplo, oferecem serviços como consultorias, *design* gráfico, desenvolvimento de *software*, entre outros produtos para clientes reais. Além disso, é uma excelente forma de preparar o estudante para o mercado de trabalho (AMBAR, 2023).

### 2.2 Movimento Empresa Júnior

O Movimento Empresa Júnior (MEJ) teve seu início em 1967 em Paris, na França, onde alunos da *L'École Supérieure des Sciences Economiques et Commerciales* (ESSEC), cientes das necessidades de complementarem seus conhecimentos através de uma aplicação prática, acabaram criando a primeira Empresa Júnior chamada de *Junior Enterprise*, essa EJ proporcionou uma experiência empresarial para os alunos (ANDRADE, 2015).

No Brasil, o movimento chegou em 1987, sendo introduzido por João Carlos Chaves, Diretor da Câmara de Comércio Franco-Brasileira. Ele orientava alunos de administração da Fundação Getúlio Vargas de São Paulo e assim fundaram a primeira Empresa Júnior do Brasil, a EJFGV (ANDRADE, 2015).

Em apenas sete anos, de 1988 a 1995, o MEJ no Brasil cresceu rapidamente, com o surgimento de 100 empresas, número este que foi alcançado na França após 19 anos. Atualmente o movimento é representado pela Confederação Nacional de Empresas Juniores, conhecida como Brasil Júnior (BJ), que foi fundada em 2003. O Brasil é reconhecido como

um dos países mais ativos na expansão do movimento, o que levou à criação de Federações e Núcleos de Empresas Juniores em diversos estados, com o objetivo de fortalecer essa rede. Dessa forma, o MEJ conseguiu formalizar seus objetivos de promover a educação empresarial e impulsionar novos negócios no Brasil (ABBAD *et al*, 2014).

### **2.3 OneBit**

A OneBit, uma EJ do curso de Sistemas de Informação do IFMG - SJE, foi fundada em 4 de setembro de 2014 com o objetivo de aproximar os alunos do mercado de trabalho, dando a oportunidade de desenvolverem o empreendedorismo, colocar em prática todo o conhecimento que é adquirido em sala de aula, além de ajudar a otimizar o processo da instituição e das empresas regionais com a prestação de vários serviços (ONEBIT, 2024).

A partir de agosto de 2019, a OneBit deu grandes passos: implantou seu primeiro sistema para um cliente, demonstrando sua capacidade de oferecer soluções práticas e eficazes. Em setembro do mesmo ano, conquistou um marco importante ao ser oficialmente reconhecida pelo MEJ, consolidando sua posição como uma empresa séria e comprometida com o desenvolvimento empresarial (ONEBIT, 2024).

Em dezembro de 2020, alcançou outro feito notável: tornou-se a primeira EJ de Minas Gerais a atingir a métrica de Alto Crescimento no portal Brasil Júnior, cumprindo suas metas de faturamento, número de projetos e membros ativos. Ao longo do tempo até aproximadamente o final de 2021, a OneBit veio adquirindo várias outras conquistas como ser uma EJ de destaque por compartilhamento de ações e de impacto por atingir todas as metas estabelecidas pelo portal Brasil Júnior, demonstrando sua capacidade de fazer a diferença no meio empresarial e na comunidade acadêmica (ONEBIT, 2024).

Atualmente a empresa está no ramo de prestação de serviços de desenvolvimento de sistemas web em diversas áreas como fluxo financeiro, ponto de venda de pequenos e microcomércios varejistas, além de desenvolvimento de sites e criação e gerenciamento de mídias sociais (ONEBIT, 2024).

### **2.4 Softwares e Sistemas de Gestão**

Segundo Sommerville (2011), “*softwares* são programas de computador e documentação associada. Produtos de *software* podem ser desenvolvidos para um cliente

específico ou para o mercado em geral.” O autor ainda menciona alguns atributos essenciais que um bom *software* deve ter como:

- a) **Manutenibilidade:** onde o *software* deve ser escrito de forma que possa evoluir para atender às necessidades dos clientes;
- b) **Confiança e Proteção:** onde um *software* confiável não deve causar prejuízos físicos ou econômicos no caso de falha de sistema;
- c) **Eficiência:** onde o *software* não deve desperdiçar os recursos do sistema, como memória e ciclos do processador;
- d) **Aceitabilidade:** onde o *software* deve ser aceitável para todo tipo de usuário para o qual foi projetado.

Aroneza (2023) define sistemas de gestão como *softwares* que permitem gerenciar diferentes tarefas, processos e informações que são executados em uma empresa. Dessa maneira, o sistema é como uma plataforma inteligente que atua realizando automatização de vários processos operacionais centralizando os dados gerados pelos setores da empresa.

O sistema pode ser usado para vários propósitos específicos dependendo para o qual ele foi desenvolvido, alguns desses propósitos são por exemplo para um controle financeiro da empresa, realizar o controle de estoque de algum estabelecimento, gerenciar tarefas e atividades além de fazer o controle de clientes.

Ter um sistema de gestão em uma empresa sendo ela de pequeno, médio ou grande porte, pode oferecer diversas vantagens não só financeiramente, mas também com auxílios nos processos internos proporcionando a integração entre atividades setoriais, e graças as automações fornecidas por esse tipo de ferramenta, é bem provável que haja um aumento na produtividade das equipes e nos resultados dos negócios (ARONEZA, 2023).

## 2.5 Sistemas Web

O sistema web pode ser entendido como uma aplicação que é hospedada em algum servidor, e pode ser acessada por meio de um navegador web. Isso torna possível o acesso remoto, possibilitando então que qualquer usuário possa acessar o *software* e visualizar, manipular e enviar dados que serão processados pelo servidor (CUNHA, 2022).

Segundo ainda Cunha (2022), a vantagem de se utilizar esse tipo de abordagem ao invés dos tradicionais sistemas *desktop* (instalados diretamente na máquina do usuário), é a facilidade de acesso ao mesmo, além de poder ser acessado de qualquer dispositivo, seja ele móvel ou de mesa, desde que haja conexão à internet. Além disso, um sistema web é um

sistema que oferece uma boa segurança, pois por se tratar de um *software* remoto, *backups* são realizados automaticamente e regularmente, evitando perda de dados caso ocorra algum problema que não tenha sido previsto.

De acordo com a Redação XP Educação (2022), quando se fala em desenvolvimento de sistemas web existem algumas áreas que o compõem, que são elas:

- a) **Frontend**: consiste no processo de desenvolver tudo aquilo que pode ser visualizado dentro de uma página web enquanto usuário, isso abrange todos os elementos visuais existentes: disposição, tipos de letra, paleta de cores, imagens, vídeos, *chatbots*, formulários, menus e outros componentes;
- b) **Backend**: nessa parte é onde fica toda a regra de negócio de uma aplicação, ou seja, tudo aquilo que é invisível aos olhos do usuário, isso varia desde integrações com API's de terceiros até armazenamento e proteção de dados internos e externos.

Um sistema web se apresenta como uma boa opção para quem busca uma solução prática e personalizada, com um ótimo custo-benefício. É fundamental que as empresas se adequem à abordagem vivida nos dias atuais com a nova era digital, e visto que o mercado está cada vez mais competitivo, quem não se adaptar às novas tendências correm o risco de ficar atrás de seus concorrentes diretos (CUNHA, 2022).

## 2.6 Ferramentas e Tecnologias

Esta seção tem por objetivo apresentar as ferramentas e tecnologias que serão necessárias para o desenvolvimento do sistema web.

### 2.6.1 HTML

Desenvolvida por Tim Berners-Lee, o *Hyper Text Markup Language* ou Linguagem de Marcação de Hipertexto (HTML) é uma linguagem de marcação, usada para estruturar uma página web e seu conteúdo. Ela utiliza o conceito de *tags* para criar um elemento na tela e definir a estrutura do conteúdo a ser exibido dentro de uma página web. Usando do HTML, as páginas web podem ser organizadas de diversas formas como parágrafos, uma lista de marcações, ou adicionando imagens e tabelas (MDN Web Docs, 2024).

Segundo ainda o MDN Web Docs (2024), as principais partes de um elemento HTML são:

- a) **Tag de abertura:** consiste no nome do elemento, envolvido em parênteses angulares de abertura e fechamento. Isso indica onde o elemento começa, ou onde o efeito que ele causa na página se inicia;
- b) **Tag de fechamento:** utiliza do mesmo conceito que a *tag* de abertura, mas se difere na barra que é incluída antes do nome do elemento. Ela basicamente indica onde o efeito do elemento se encerra;
- c) **Conteúdo:** dentro das *tags* de abertura e fechamento é onde entra o conteúdo, sendo estes de vários tipos como texto, elementos de mídia como imagens e faixas de áudio ou até mesmo outros elementos HTML.

### 2.6.2 CSS

O *Cascading Style Sheets* ou Folhas de Estilo em Cascata, mais conhecido como CSS, é uma linguagem de estilo desenvolvida em 1996 pelo *World Wide Web Consortium* (W3C), e seu propósito era complementar o HTML. O CSS é uma linguagem que determina a aparência de uma página web. Ela permite alterar elementos como cor do texto, tamanho de fonte e espaçamento entre blocos de uma página, além da criação de efeitos, animações em tela, profundidade e páginas interativas (TOTVS, 2020).

### 2.6.3 JavaScript

O *JavaScript* (JS) é uma linguagem de programação interpretada, leve e muito utilizada como linguagem de *script* para páginas web. A maioria dos sites modernos utiliza *JavaScript* e todos os navegadores modernos possuem interpretadores que são os motores, fazendo com que a linguagem se tornasse uma das mais utilizadas do mundo. Ela vem para complementar o HTML e o CSS, criando assim a tríade de tecnologias que todos os desenvolvedores web deveriam conhecer (FLANAGAN, 2013).

O *JavaScript* foi criado em 1996 por Brendan Eich que, além de ser o responsável pela criação da linguagem também foi um dos fundadores da *Mozilla Corporation*. Ele desenvolveu a linguagem quando ainda fazia parte da *Netscape Communications Corporation* e seu objetivo era facilitar os processos dentro de uma página web, tornando a programação mais simples (ESTRELLA, 2023).

Segundo o mesmo autor, o JS é uma linguagem que à princípio foi criada para desenvolver aplicações no lado do cliente (*client-side*) mas graças ao advento do *Node.js* fez com que a linguagem pudesse ser utilizada para desenvolver aplicações no lado do servidor (*server-side*) também. Além disso, o JS também pode ser utilizado para o desenvolvimento de aplicações para dispositivos móveis, sistemas *desktop* e até mesmo para criação de jogos.

#### 2.6.4 *TypeScript*

O *TypeScript* (TS) é uma linguagem de programação de código aberto desenvolvida pela Microsoft. Como um superconjunto sintático estrito de *JavaScript*, o TS adiciona tipagem estática opcional à linguagem. Tipos fornecem uma maneira de descrever a forma de um objeto, oferecendo melhor documentação e permitindo que o compilador valide se o código está funcionando corretamente. Por ser um superconjunto de *JavaScript*, programas já escritos nessa linguagem são também válidos em TS (WIKIPEDIA, 2024).

Anders Hejlsberg, renomado arquiteto do C# e criador de *Delphi* e *Turbo Pascal*, teve um papel significativo no desenvolvimento do TS. A linguagem é amplamente utilizada para construir aplicações tanto no lado do cliente quanto no servidor (usando *Node.js* ou outra *runtime* de *JavaScript*). Há diversas opções para a conversão, incluindo o verificador padrão do TS ou o uso do compilador *Babel*. O TS foi classificado como a 4ª linguagem "mais amada" pelo público em uma pesquisa do *Stack Overflow* em 2018 e está entre as 15 mais populares, segundo a *RedMonk* (WIKIPEDIA, 2024).

#### 2.6.5 *React*

De acordo com Coutinho (2023), o *React* é uma biblioteca *JavaScript* criada pelo Facebook (atual Meta) para auxiliar no desenvolvimento de aplicações web ou para dispositivos móveis. A biblioteca contém um conjunto de trechos de código *JavaScript* para serem reaproveitados para a criação de interfaces de usuário (UI).

O *React* possui diversos elementos que o compõem, juntamente com suas características, sendo elas:

- a) **Componentes:** é responsável por separar as interfaces de usuários em diversas partes de código isoladas e reutilizáveis.
- b) **Props:** os componentes também apresentam entradas arbitrárias conhecidas como *props* (que vem de propriedade). Essas propriedades comportam-se como

objetos com diversas informações que apresentam variados tipos, como *string*, número e função;

c) **State**: os *states* são informações criadas dentro de um componente. Assim, diferentemente dos *props*, os *states* são mutáveis, ou seja, seu valor pode ser alterado a depender do que o desenvolvedor estabelecer;

d) **JSX**: é uma extensão que tem o intuito de ajudar na sintaxe do *JavaScript*, muito parecido com o próprio HTML. Esse elemento é utilizado para descrever as interfaces de usuário. Dessa maneira, o JSX permite que códigos JS possam ser misturados com a sintaxe do HTML, possibilitando que a escrita dos componentes seja armazenada de forma mais fácil em apenas um arquivo de código.

O *React* é amplamente utilizado por várias empresas ao redor do mundo, como Netflix e Instagram, que são empresas de grande renome, o que significa que é uma tecnologia com recursos poderosos e fáceis de se utilizar, além de ter uma comunidade bastante ativa ao redor do mundo. Além disso, é mantido por uma grande empresa, o que passa confiabilidade no quesito de receber suporte e atualizações frequentemente (COUTINHO, 2023).

### 2.6.6 Java

Desenvolvida pela *Sun Microsystems* e posteriormente adquirida pela *Oracle*, o *Java* é uma linguagem de programação orientada a objetos e multiplataforma. Diferentemente de outras linguagens que são compiladas para código nativo, o *Java* é compilado para *bytecode*, que dessa forma é interpretado por uma máquina virtual. As aplicações que são feitas em *Java* podem ser executadas em qualquer plataforma que possua a *Java Virtual Machine* (JVM) instalada.

O *Java* é também uma linguagem de programação muito poderosa e está em constante crescimento, sendo bem popular entre vários desenvolvedores, além de ser gratuita, que foi um dos fatores que proporcionou a popularização da linguagem entre programadores e empresas que utilizam a linguagem para o desenvolvimento de seus sistemas (GUEDES, 2018).

### 2.6.7 *Spring Boot*

O *Spring Boot* é uma ferramenta que foi criada a partir do *Spring*, que é um *framework* desenvolvido para a plataforma Java, que se baseia em padrões de projetos, inversão de controle e injeção de dependência. Apesar do *Spring framework* possuir o intuito de simplificar as configurações de aplicações web, ele não conseguiu atender às expectativas do mercado pois as configurações ainda eram grandes e muito complexas.

Dessa forma, a fim de abstrair toda a complexidade de configurações, surgiu o *Spring Boot*, que fornece a maioria dos componentes necessários para aplicações em geral de maneira pré-configurada, possibilitando aplicações de estarem em produção rapidamente, com o esforço mínimo de configuração e implantação, permitindo que o desenvolvedor tenha o foco na codificação da aplicação (GEEKHUNTER, 2022).

### 2.6.8 *MySQL*

Segundo Pisa (2012), o *MySQL* é um sistema gerenciador de banco de dados relacional de código aberto usado na maioria das aplicações gratuitas para gerir suas bases de dados. Ele utiliza a linguagem *Structure Query Language* ou Linguagem de Consulta Estruturada (SQL), que é bem popular e comumente usada para realizar inserções, acessos e gerenciamento dos dados armazenados em um banco de dados.

O *MySQL* foi desenvolvido para processar grandes bancos de dados de forma rápida e ser usado em ambientes de produção altamente exigentes por vários anos, alguns dos principais benefícios que ele proporciona são: facilidade de uso, confiabilidade, escalabilidade, desempenho, segurança e flexibilidade (ORACLE, 2024).

### 2.6.9 *Trello*

O *Trello* é uma ferramenta que, através de quadros interativos possibilita o gerenciamento de qualquer tipo de projeto, fluxo de trabalho e monitoramento de tarefas (TRELLO, 2024).

Essa ferramenta é perfeitamente adequada para ser utilizada com base no *Kanban*, que é uma metodologia de gestão que permite o acompanhamento de projetos e tarefas, sendo possível também visualizar atividades pendentes, em andamento e as que foram concluídas (PEREIRA, 2023).

### 2.6.10 *Git e Github*

*Git* é um sistema de controle de versão que rastreia alterações em arquivos de forma inteligente. O *Git* é particularmente útil quando um grupo de pessoas está fazendo alterações nos mesmos arquivos ao mesmo tempo, permitindo que os desenvolvedores vejam todo o cronograma de suas mudanças, decisões e progressão de qualquer projeto em um só lugar. A partir do momento em que acessa o histórico de um projeto, o desenvolvedor tem todo o contexto necessário para entendê-lo e começar a contribuir.

*GitHub* é uma plataforma baseada em nuvem que foi construída sobre o *software* de código aberto *Git*, em que se pode armazenar, compartilhar e trabalhar junto com outras pessoas para escrever código. Com o *GitHub* é possível acompanhar e gerenciar alterações no código ao longo do tempo, deixar que outras pessoas revisem o código, façam sugestões de melhorias e também colaborem em repositórios compartilhados, sem se preocupar com o impacto das alterações no trabalho dos colaboradores (GITHUB, 2024).

### 2.6.11 *Figma*

*Figma* é uma ferramenta de *design* colaborativa e poderosa para equipes, ela permite a criação de protótipos realistas e que simplifique o desenvolvimento de produtos com sistemas de *design*. A interface intuitiva e os recursos do *Figma* foram desenvolvidos para a web, permitindo que especialistas do setor e *designers* amadores criem *designs* com facilidade e com a colaboração em tempo real (FIGMA, 2024).

### 2.6.12 *IDE*

Uma *Integrated Development Environment* ou Ambiente de Desenvolvimento Integrado (IDE), é um *software* para criar aplicações que combina ferramentas comuns de desenvolvedor em uma única interface de usuário gráfica (GUI). As IDEs geralmente consistem em:

- a) **Editor de código-fonte:** é um editor de texto que auxilia na criação de código de *software* por meio de funcionalidades como destaque da sintaxe com indicadores visuais, recurso de preenchimento automático específico da linguagem e verificação de erros durante o desenvolvimento;

b) **Automação de compilação local:** são utilitários que automatizam tarefas simples e repetíveis durante a criação de uma compilação local do *software* usada pelo desenvolvedor. São tarefas como compilação de código-fonte em código binário, criação de pacotes de código binário e execução de testes automatizados;

c) **Debugger:** é um programa usado para testar outros programas e mostrar graficamente a localização do *bug* no código original.

O objetivo das IDEs é ajudar os desenvolvedores a programarem novas aplicações de forma rápida, já que os vários utilitários não precisam ser ajustados e integrados manualmente durante a configuração. Os desenvolvedores também não precisam passar horas aprendendo a usar cada uma das diferentes ferramentas, isso é especialmente útil quando eles são novos no projeto. Eles podem contar com a IDE para se atualizar em relação às ferramentas e fluxos de trabalho da equipe. Na verdade, o objetivo da maior parte das funcionalidades é a economia de tempo, pois o preenchimento inteligente e a geração automática de código, por exemplo, eliminam a necessidade de digitar sequências inteiras (REDHAT, 2023).

## 2.7 Trabalhos Correlatos

O trabalho desenvolvido por Fonseca (2023) consiste no desenvolvimento de um sistema web para auxiliar na gestão de empresas juniores, visando otimizar a gestão e organização interna dessas entidades, facilitando processos e melhorando a eficiência operacional. Foi desenvolvido com as linguagens Vue.js para o *frontend* e Nest.js para o *backend*, funcionando em um ambiente web.

O sistema tem funcionalidades específicas para diferentes áreas de uma empresa júnior (comercial, projetos, administrativo/financeiro e presidência), com isso ele reflete as necessidades práticas de uma EJ e se destaca pelas funcionalidades específicas que otimizam suas operações cotidianas.

As semelhanças com o presente trabalho são algumas funcionalidades como cadastro de membros, projetos e clientes, além de ser um sistema web. O projeto desenvolvido apresenta alguns diferenciais em relação ao trabalho de Fonseca, dentre eles vale à pena citar que será específico para a Empresa Júnior OneBit, além das linguagens utilizadas não serem as mesmas.

O trabalho desenvolvido por Oliveira (2022) consiste em um sistema web para gerenciar informações relevantes de empresas juniores, centralizando dados de membros,

projetos e relatórios, a fim de facilitar o gerenciamento e reduzir a dependência de documentos descentralizados, como planilhas. O sistema foi projetado para ser usado por diretores e membros autorizados de empresas juniores, com funcionalidades como cadastro de projetos, gerenciamento de membros, controle de acesso e armazenamento de *links* importantes. Ele utiliza Node.js para o *backend* do projeto e Vue.js para o *frontend*.

As semelhanças entre o sistema citado anteriormente e o OneFlow são as funcionalidades como o cadastro de membros, projetos, controle de acesso e armazenamento de *links* importantes pertinentes à EJ. Como diferenciais, é válido apontar que o sistema proposto conta com uma separação dos membros da empresa júnior em seus respectivos setores, além de possuir uma funcionalidade a mais, que seria o cadastro de reuniões, onde o presidente e os diretores poderiam cadastrar o horário e a data em que ela ocorrerá.

O trabalho desenvolvido por Silva e Araújo (2015) consiste em um sistema *desktop* desenvolvido para gerenciar e armazenar informações sobre clientes, empresas parceiras, membros, projetos e tarefas da Empresa Júnior OneBit, buscando melhorar sua eficiência e organização. Este sistema inicialmente seria desenvolvido utilizando a linguagem *Java*, mas por questões técnicas os autores decidiram alterar para a linguagem *C#*.

As semelhanças com o presente trabalho são algumas das funcionalidades oferecidas pelo sistema como o gerenciamento de membros, clientes e projetos, já as diferenças estão no tipo do sistema, que é voltado para a plataforma *desktop* sendo necessário a instalação no computador dos usuários e a linguagem de programação utilizada.

Diante dos trabalhos correlatos analisados, percebe-se que nenhum deles aborda exatamente a proposta do presente trabalho. O sistema OneFlow é uma solução web específica para a Empresa Júnior OneBit, contemplando funcionalidades já exploradas em outros trabalhos, como o gerenciamento de membros, clientes e projetos, porém com diferenciais importantes, entre eles, destaca-se a separação detalhada dos membros por setor, a inclusão do cadastro de reuniões e a utilização de tecnologias específicas para melhor atender às necessidades da empresa. Dessa forma, o sistema proposto busca oferecer uma solução mais integrada, eficiente e alinhada à realidade da EJ.

### **3 METODOLOGIA**

Esta seção tem por objetivo apresentar a natureza da pesquisa e seu caráter, a população e amostra, métodos e procedimentos e o cronograma de atividades para o desenvolvimento do trabalho.

#### **3.1 Natureza da pesquisa**

A natureza da pesquisa do presente trabalho é aplicada, a pesquisa aplicada interessa-se pela aplicação, utilização e consequências práticas dos conhecimentos. Destina-se a aplicar os conhecimentos científicos para a solução dos mais variados problemas individuais ou coletivos. Concretiza-se por meio das “ciências aplicadas” e “tecnológicas” (ASSIS, 2009). No contexto deste trabalho, o desenvolvimento de uma solução tecnológica para melhorar a gestão de membros, projetos e outras atividades da Empresa Júnior, alinha-se diretamente com a definição de pesquisa aplicada.

A metodologia de pesquisa utilizada no presente trabalho apresenta caráter qualitativo. A pesquisa qualitativa pode ser caracterizada como uma tentativa de se explicar em profundidade o significado e as características do resultado das informações obtidas através de reuniões, sem a mensuração quantitativa de características ou comportamento (OLIVEIRA, 2008). Essa abordagem qualitativa permitirá realizar uma análise para entender se o sistema desenvolvido atendeu aos requisitos solicitados, avaliando a percepção dos usuários sobre a usabilidade, eficiência e adequação da solução às necessidades da OneBit.

#### **3.2 População e amostra**

Este trabalho tem como população as Empresas Juniores do Instituto Federal de Minas Gerais - *Campus* São João Evangelista. A amostra dessa população será a Empresa Júnior OneBit do curso Bacharelado em Sistemas de Informação do IFMG-SJE.

#### **3.3 Métodos e procedimentos**

Esta seção tem por objetivo apresentar os métodos e procedimentos que foram utilizados para a realização do trabalho.

### 3.3.1 Levantamento de requisitos

Primeiramente foi feito o levantamento e análise dos requisitos do sistema, estes que foram adquiridos por meio de uma reunião com os membros da OneBit. Vale destacar que os autores do trabalho faziam parte da empresa durante o desenvolvimento do mesmo.

Após essa etapa, os requisitos são divididos em funcionais e não funcionais, que de acordo com Sommerville (2011):

Os requisitos de *software* são frequentemente classificados como requisitos funcionais e requisitos não funcionais: 1. Requisitos funcionais. São declarações de serviços que o sistema deve fornecer, de como o sistema deve reagir a entradas específicas e de como o sistema deve se comportar em determinadas situações. Em alguns casos, os requisitos funcionais também podem explicitar o que o sistema não deve fazer. 2. Requisitos não funcionais. São restrições aos serviços ou funções oferecidas pelo sistema. Incluem restrições de *timing*, restrições no processo de desenvolvimento e restrições impostas pelas normas. Ao contrário das características individuais ou serviços do sistema, os requisitos não funcionais, muitas vezes, aplicam-se ao sistema como um todo (SOMMERVILLE, 2011, p.59).

No Quadro 1, é apresentada a lista dos requisitos funcionais e não funcionais do sistema.

Quadro 1 – Requisitos Funcionais e Não Funcionais

<b>Requisitos Funcionais (RF)</b>	
<b>Código</b>	<b>Descrição</b>
<b>RF 01</b>	O sistema deve permitir gerenciar os membros, com funcionalidades para cadastrar, visualizar, atualizar e excluir dados.
<b>RF 02</b>	O sistema deve permitir gerenciar os clientes, com funcionalidades para cadastrar, visualizar, atualizar e excluir dados.
<b>RF 03</b>	O sistema deve permitir gerenciar os projetos, com funcionalidades para cadastrar, visualizar, atualizar, adicionar membros no projeto e excluir os dados.
<b>RF 04</b>	O sistema deve permitir gerenciar as reuniões, com funcionalidades para cadastrar, visualizar, atualizar e excluir dados.
<b>RF 05</b>	O sistema deve permitir gerenciar os <i>links</i> de acesso aos documentos importantes, com funcionalidades para cadastrar, visualizar, atualizar e excluir os dados.
<b>RF 06</b>	O sistema deve permitir que os usuários realizem o login no sistema, informando o e-mail e senha.
<b>RF 07</b>	O sistema deve ter um controle de diferentes níveis de permissão para os usuários, controlando o acesso a funcionalidades específicas (por exemplo, apenas presidentes e diretores podem excluir projetos ou documentos).
<b>RF 08</b>	O sistema deve permitir que os usuários redefinam sua senha em caso de esquecimento.

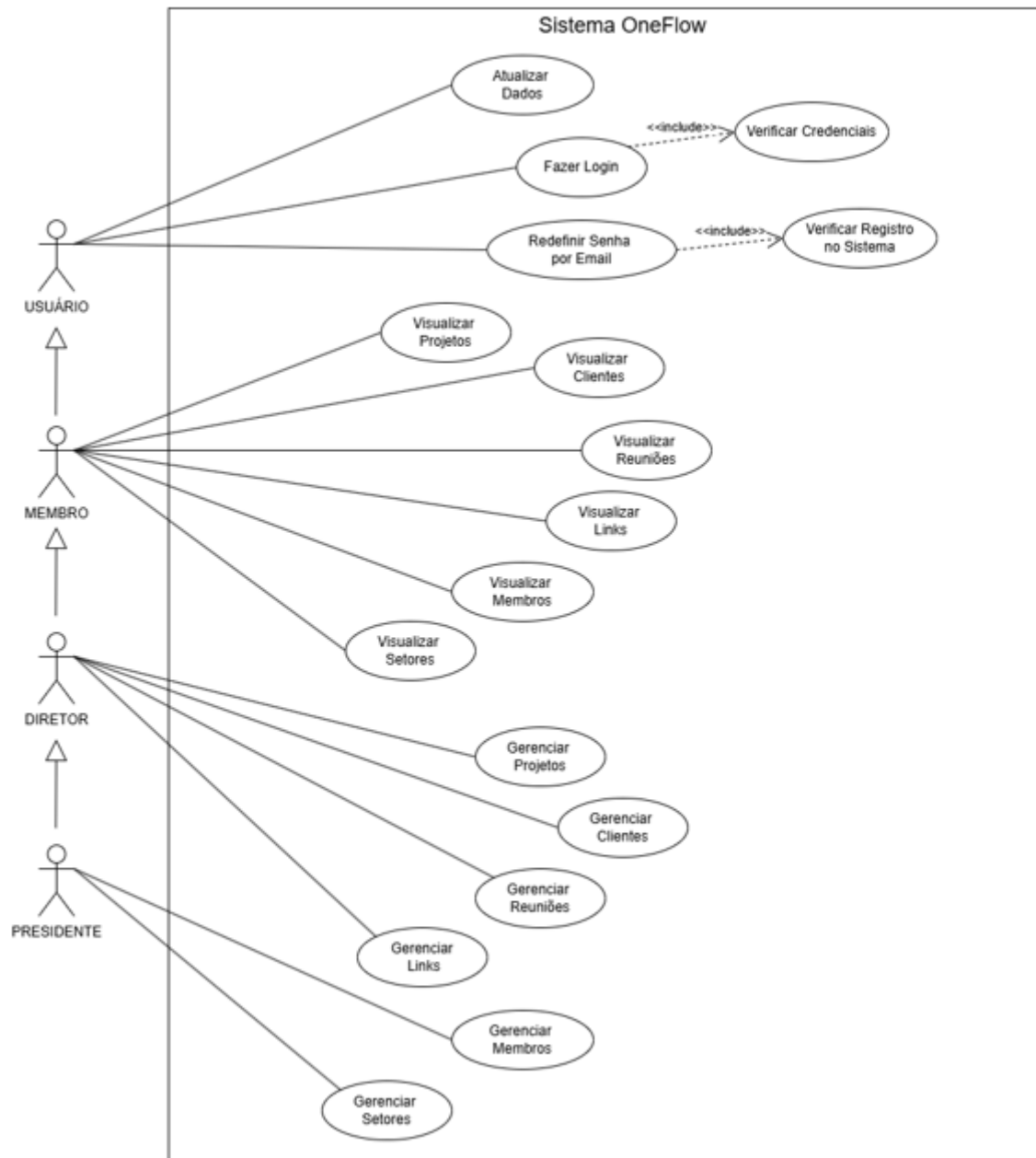
<b>Requisitos Não Funcionais (RNF)</b>	
<b>RNF 01</b>	A senha dos usuários deverá ser salva criptografada no banco de dados.
<b>RNF 02</b>	O sistema deverá ter uma interface simples e intuitiva para os usuários.
<b>RNF 03</b>	O sistema deverá ter um modo responsivo para funcionar em vários navegadores, dispositivos móveis e <i>tablets</i> .
<b>RNF 04</b>	O sistema deve ter verificações para que os dados como e-mail, CPF e CNPJ sejam únicos.
<b>RNF 05</b>	O sistema deve realizar a validação de campos obrigatórios para o armazenamento correto dos dados.

Fonte: Elaborado pelos autores, 2025.

### **3.3.2 Modelagem**

Após ter sido feita a definição dos requisitos, iniciou-se a modelagem do sistema com a construção de um diagrama de caso de uso, que tem como objetivo demonstrar as maneiras que o usuário pode interagir com um sistema. Na Figura 1 é apresentado o diagrama de caso de uso do sistema.

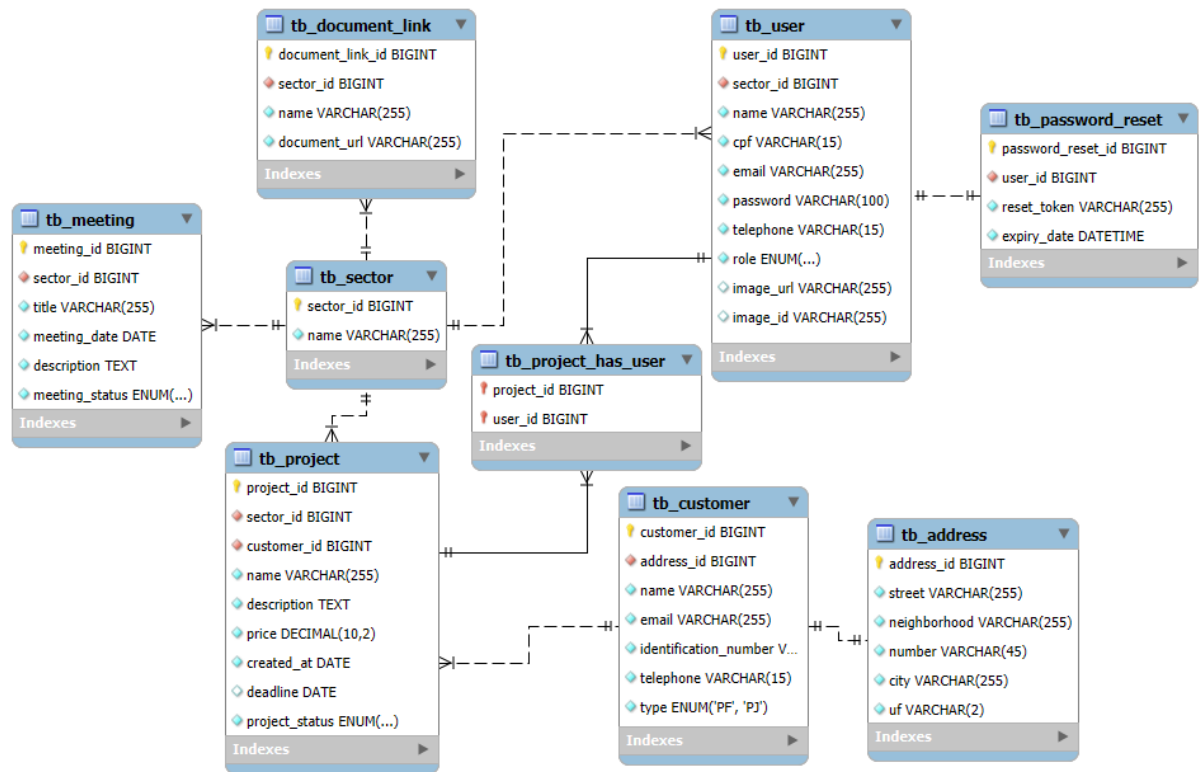
Figura 1 – Diagrama de caso de uso



Fonte: Elaborado pelos autores, 2025.

Na sequência, foi criado o Diagrama Entidade Relacionamento (DER) para facilitar a visualização das tabelas e entidades que estão no banco de dados da aplicação. Na Figura 2 é apresentado o DER do sistema.

Figura 2 – Diagrama entidade relacionamento



Fonte: Elaborado pelos autores, 2025.

### 3.3.3 Desenvolvimento

No processo de desenvolvimento, o Trello foi utilizado como uma ferramenta fundamental para a gestão das tarefas. Um quadro específico foi criado nessa ferramenta para o projeto, onde foram adicionados cartões que representam cada tarefa a ser realizada. Esses cartões são organizados em listas que representam as diferentes etapas do desenvolvimento, como “*Backlog (Pendências)*”, “*Todo (A fazer)*”, “*Doing (Fazendo)*”, “*Testing (Testando)*” e “*Done (Concluído)*”. Cada cartão corresponde a uma tarefa ou conjunto de tarefas relacionadas, e contém informações como:

- Descrição da Tarefa:** um breve resumo do que precisa ser feito;
- Responsável:** o desenvolvedor que será responsável pela tarefa;
- Prazos:** datas de início e término previstas para a execução da tarefa;
- Checklist:** um conjunto de subtarefas que precisam ser concluídas para que a tarefa principal seja considerada finalizada.

Para o versionamento de código e manter o controle das alterações, foram utilizados o *Git* e a plataforma *Github*, onde foram criados os repositórios para salvar os códigos do *frontend* e *backend* do sistema.

O *frontend* foi desenvolvido no editor de código *Visual Studio Code* utilizando tecnologias como HTML, CSS, *JavaScript* e *TypeScript*, juntamente com a biblioteca *React*, para garantir uma interface de usuário dinâmica e responsiva. As telas prototipadas na ferramenta *Figma* serviram como guia para a implementação, garantindo que a experiência do usuário final esteja alinhada com os requisitos previamente definidos.

O *backend* foi implementado na IDE *Intellij Idea* utilizando a linguagem *Java* e o *framework Spring Boot*, sendo responsável por processar as regras de negócio e gerenciar a comunicação com o banco de dados MySQL. Durante essa etapa, a *Application Programming Interface*, ou Interface de Programação de Aplicação (API), foi criada para permitir a integração do *frontend* com o *backend*, garantindo a troca segura e eficiente de dados.

Dessa forma, o sistema foi desenvolvido seguindo as etapas citadas acima e os testes iniciais feitos pelos próprios desenvolvedores, posteriormente foram realizados os testes com os membros da OneBit. Nesses testes, eles executaram cadastros, edições, exclusões e consultas de dados para verificar se o sistema está funcionando corretamente.

## 4 RESULTADOS E DISCUSSÕES

Nesta seção serão apresentados os resultados que foram obtidos com o desenvolvimento do trabalho.

### 4.1 Descrição do sistema

A primeira tela que é visualizada quando o usuário acessa o sistema é a tela de *login*, que pode ser vista na Figura 3. Ela exibe uma mensagem de boas-vindas com instruções para o usuário informar suas credenciais, acompanhada de campos para e-mail e senha, sendo que este último possui a opção de exibir a senha para facilitar a conferência. Além disso, há um botão para recuperação de senha, caso o usuário tenha esquecido suas credenciais é possível redefini-la via e-mail. A principal função dessa tela é garantir a segurança do sistema, permitindo acesso apenas a usuários autorizados.

Figura 3 – Tela de Login



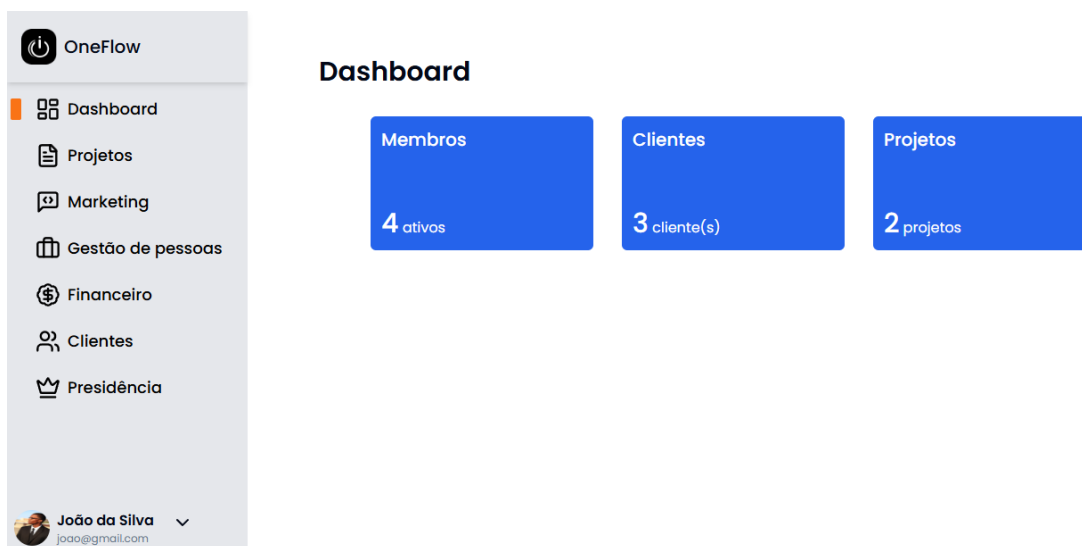
A imagem mostra a interface de login de um sistema web. No topo, há um ícone de energia (um círculo com uma barra vertical no meio). Abaixo dele, o texto "Bem vindo" em negrito, seguido por "Por favor informe suas credenciais". O formulário contém dois campos de entrada: "E-mail" e "Senha". O campo "Senha" possui um ícone de olho no canto inferior direito para alternar a visibilidade. Abaixo dos campos, há um botão azul com o texto "Entrar". Na base do formulário, há um link azul "Esqueci minha senha".

Fonte: Elaborado pelos autores, 2025.

Após o usuário realizar o login no sistema, ele é direcionado para a tela principal, que é o *dashboard*, como mostra a Figura 4. O *dashboard* contém três painéis destacando o número total de membros ativos, clientes cadastrados e projetos, permitindo um acompanhamento rápido dessas métricas essenciais.

No lado esquerdo, há um menu de navegação vertical com ícones e opções como Projetos, Marketing, Gestão de Pessoas, Financeiro, Clientes e Presidência, facilitando o acesso aos dados de cada setor da empresa. Na parte inferior do menu, estão exibidos o nome e o e-mail do usuário logado, reforçando a personalização e identificação do acesso.

Figura 4 – Tela Principal



Fonte: Elaborado pelos autores, 2025.

A Figura 5 apresenta a tela de gestão de usuários, acessada na seção Presidência do sistema. Essa tela permite a administração dos membros cadastrados no sistema, exibindo informações organizadas em uma tabela com colunas para nome, CPF, e-mail, contato, cargo e setor. Cada linha representa um usuário cadastrado, com opções de ações para editar ou excluir os registros, garantindo um controle eficiente e dinâmico.

Na parte superior da tela, há três botões principais, o "Adicionar" que permite o cadastro de novos usuários, "Reuniões" que permite o cadastro de reuniões gerais para participação de todos os membros, e "Documentos", para cadastro de *links* com acesso a arquivos ou informações relevantes. Essa funcionalidade é essencial para o gerenciamento centralizado de informações dos membros, permitindo à presidência monitorar e atualizar os dados de forma prática e segura.

Figura 5 – Tela de Gestão de Usuários

**Gestão de Usuários** + Adicionar Reuniões Documentos

Nome	CPF	E-mail	Contato	Cargo	Setor	Ações
Anna Miranda	123.456.789-12	anna@gmail.com	(33) 99999-8888	Membro	Projetos	Editar Deletar
João da Silva	123.456.789-11	joao@gmail.com	(33) 99999-9999	Diretor	Projetos	Editar Deletar
John Doe	123.456.789-10	john@gmail.com	(33) 91234-5678	Presidente	Presidência	Editar Deletar
Joyce Costa	123.456.789-13	joyce@gmail.com	(33) 99999-7777	Diretor	Marketing	Editar Deletar

< Anterior 1 Próxima >

João da Silva  
joao@gmail.com

Fonte: Elaborado pelos autores, 2025.

A tela de reuniões, como mostra a Figura 6, apresenta uma interface voltada para o gerenciamento de encontros corporativos da Empresa Júnior. Nela, é possível visualizar uma lista de reuniões organizadas em colunas que incluem título, descrição, data, status, setor responsável e ações disponíveis. O usuário com as devidas permissões pode adicionar novas reuniões por meio de um botão destacado no topo da página, bem como editar ou excluir reuniões existentes utilizando os botões de ação ao lado de cada item listado.

Figura 6 – Tela de Reuniões

**Reuniões** + Adicionar

Título	Descrição	Data	Status	Setor	Ações
Reunião	Assuntos importantes	16/01/2025	Finalizada	Projetos	Editar Deletar
Reunião de alinhamento	Reunião para fazer o alinhamento da equipe e saber o andamento das atividades	21/01/2025	Pendente	Projetos	Editar Deletar

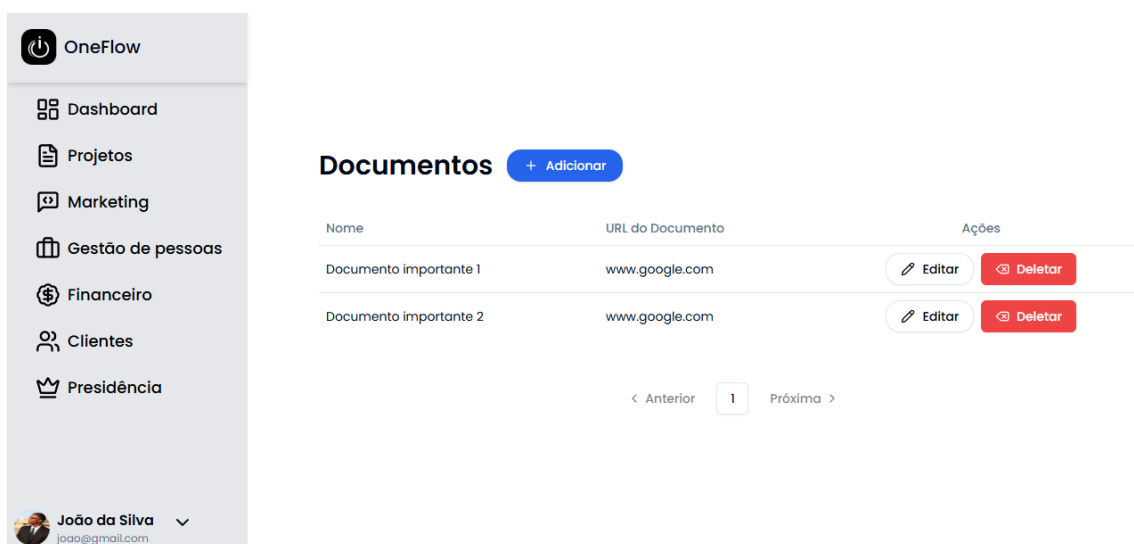
< Anterior 1 Próxima >

João da Silva  
joao@gmail.com

Fonte: Elaborado pelos autores, 2025.

A tela de documentos, apresentada na Figura 7, permite o gerenciamento de *links* de arquivos relevantes para a EJ. Ela exibe uma lista organizada com o nome dos documentos e seus respectivos *links*, possibilitando acesso rápido às informações. No topo da página, há um botão que possibilita a inclusão de novos documentos. Além disso, para cada item listado, há botões de ação que permitem editar ou excluir o registro. Para acessar a tela de documentos basta clicar no botão “Documentos” que se encontra na parte superior à direita da tela de cada setor.

Figura 7 – Tela de Documentos



Fonte: Elaborado pelos autores, 2025.

A tela de clientes, exibida na Figura 8, apresenta uma lista completa de todos os clientes cadastrados, permitindo que o usuário visualize, adicione, edite ou exclua informações sobre cada um. A tabela exibe dados como nome, e-mail, contato, tipo (pessoa física ou jurídica), documento (CPF ou CNPJ), endereço e oferece botões para realizar ações sobre cada cliente.

Figura 8 – Tela de Clientes

**Clientes** + Adicionar

Nome	E-mail	Contato	Tipo	Documento	Endereço	Ações
Cliente 1	cliente1@gmail.com	(33) 99999-9999	PF	147.258.369-15	👁	<a href="#">✎ Editar</a> <a href="#">🗑 Deletar</a>
Cliente 2	cliente2@gmail.com	(33) 99999-9999	PF	147.258.369-16	👁	<a href="#">✎ Editar</a> <a href="#">🗑 Deletar</a>
Cliente 3	cliente3@gmail.com	(33) 99999-9999	PJ	99.999.999/9999-99	👁	<a href="#">✎ Editar</a> <a href="#">🗑 Deletar</a>

< Anterior 1 Próxima >

João da Silva  
joao@gmail.com

Fonte: Elaborado pelos autores, 2025.

A tela do setor de projetos, representada na Figura 9, ilustra o gerenciamento de atividades relacionadas ao setor de projetos. Ela permite visualizar informações como nome, descrição, preço, prazo final, status, cliente associado e usuários envolvidos. Assim como na tela de documentos, existem botões de ação para editar ou excluir projetos, além de um botão para adicionar novos projetos e um botão de acesso rápido às reuniões associadas ao setor.

É importante destacar que as funcionalidades apresentadas nesta tela são replicadas para os demais setores da empresa, como Marketing, Gestão de Pessoas e Financeiro. Isso garante uma experiência consistente para os usuários em diferentes áreas da plataforma.

Figura 9 – Tela do Setor de Projetos

**Projetos** + Adicionar Reuniões Documentos

Nome	Descrição	Preço	Prazo final	Status	Cliente	Usuários	Ações
Sistema 1	Desenvolvimento do sistema	R\$ 2.000,00	30/01/2025	Em Andamento	Cliente 1	👁	<a href="#">✎ Editar</a> <a href="#">🗑 Deletar</a>
Sistema 2	Desenvolvimento do sistema	R\$ 5.000,00	19/06/2025	Pendente	Cliente 3	👁	<a href="#">✎ Editar</a> <a href="#">🗑 Deletar</a>

< Anterior 1 Próxima >

João da Silva  
joao@gmail.com

Fonte: Elaborado pelos autores, 2025.

Na Figura 10 é apresentado o formulário para adicionar um novo projeto, permitindo ao usuário realizar o cadastro de projetos de forma simples e organizada. Esse formulário é composto por campos como nome, descrição, preço, prazo final e status. Além disso, o formulário permite a seleção do cliente relacionado ao projeto, apresentando uma lista com todos os clientes cadastrados no sistema, e também, permite selecionar os usuários que irão atuar no projeto. Após o preenchimento dos campos, o botão "Criar Projeto" realiza o salvamento das informações no sistema. É importante ressaltar que o formulário segue o mesmo padrão para os outros setores do sistema, como Marketing, Gestão de Pessoas e Financeiro.

Figura 10 – Formulário Adicionar Novo Projeto

Nome	Descrição
Sistema 1	Desenvolvimento do sistema
Sistema 2	Desenvolvimento do sistema

Cliente	Usuários	Ações
Cliente 1		<a href="#">Editar</a> <a href="#">Deletar</a>
Cliente 3		<a href="#">Editar</a> <a href="#">Deletar</a>

Fonte: Elaborado pelos autores, 2025.

O formulário para adicionar uma nova reunião apresentado na Figura 11, permite que o usuário faça o cadastro de reuniões no sistema, os campos disponíveis nessa interface permitem que o usuário insira informações importantes relacionadas à reunião como o título, descrição, data da reunião e status. Assim como no formulário de adicionar novo projeto, essa estrutura é padrão para os outros setores.

Figura 11 – Formulário Nova Reunião

Fonte: Elaborado pelos autores, 2025.

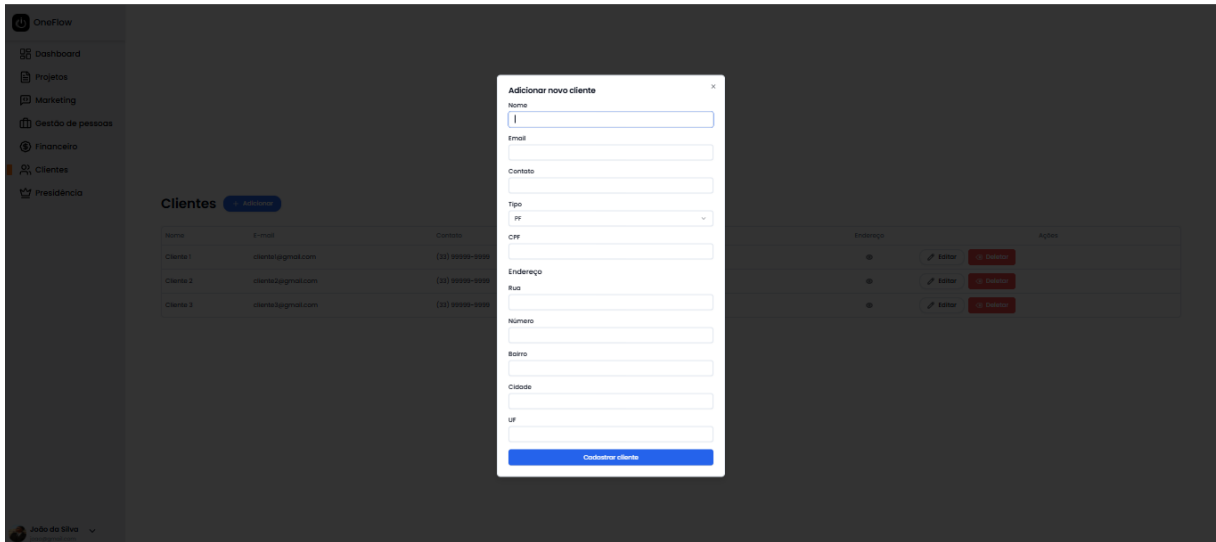
O formulário apresentado na Figura 12 corresponde à funcionalidade de adicionar um novo documento. Trata-se de um dos formulários mais simples do sistema, contando apenas com dois campos, um para o nome do documento e outro para a URL de acesso. Essa funcionalidade permite armazenar os *links* de documentos importantes da empresa, facilitando o acesso posterior por parte dos demais membros da OneBit. O formulário também é padrão e utilizado nos outros setores.

Figura 12 – Formulário Adicionar Novo Documento

Fonte: Elaborado pelos autores, 2025.

Para realizar o cadastro de um novo cliente no sistema, é utilizado um formulário como mostra a Figura 13. Os campos de preenchimento disponíveis como nome, e-mail, contato, tipo (pessoa física ou pessoa jurídica), CPF ou CNPJ e campos para o endereço, permitem que o usuário insira informações completas sobre o cliente, garantindo a organização e a facilidade de consulta posterior.

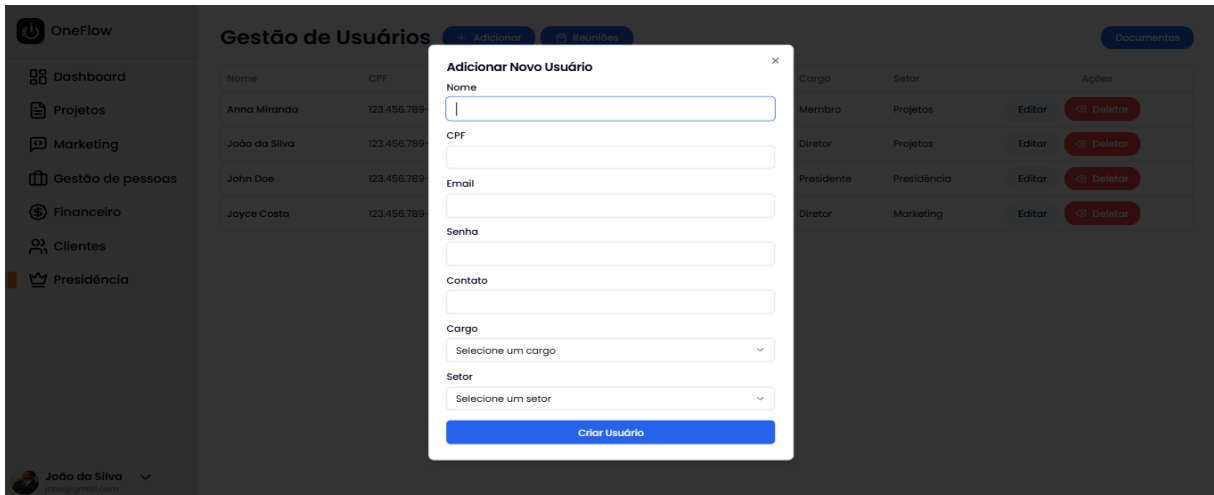
Figura 13 – Formulário Adicionar Novo Cliente

The image shows a screenshot of a web application interface. On the left, there is a dark sidebar with navigation icons for 'Dashboard', 'Projetos', 'Marketing', 'Gestão de pessoas', 'Financeiro', 'Clientes', and 'Presidência'. The main content area is titled 'Clientes' and contains a table with columns for 'Nome', 'E-mail', and 'Contato'. A modal window titled 'Adicionar novo cliente' is open in the center, featuring several input fields: 'Nome', 'E-mail', 'Contato', 'Tipo' (a dropdown menu), 'CPF', 'Endereço' (with sub-fields for 'Rua', 'Número', 'Bairro', 'Cidade', and 'UF'), and a blue 'Cadastrar cliente' button at the bottom.

Fonte: Elaborado pelos autores, 2025.

A funcionalidade de adicionar novos usuários ao sistema, realizada apenas pelo usuário com cargo de presidente, permite o cadastro de informações essenciais, como nome, CPF, e-mail, senha, contato, cargo e setor do qual o novo membro fará parte. Esse cadastro garante que cada usuário tenha o nível adequado de autorização para interagir com o sistema. Essa funcionalidade é fundamental para gerenciar a equipe e assegurar que apenas pessoas autorizadas possam acessar e realizar ações específicas dentro da plataforma. O formulário para o cadastro de novos usuários é apresentado logo abaixo na Figura 14.

Figura 14 – Formulário Adicionar Novo Usuário



Nome	CPF	Cargo	Sector	Ações
Anna Miranda	123.456.789	Membro	Projetos	Editar <a href="#">Deletar</a>
João da Silva	123.456.789	Diretor	Projetos	Editar <a href="#">Deletar</a>
John Doe	123.456.789	Presidente	Presidência	Editar <a href="#">Deletar</a>
Joyce Costa	123.456.789	Diretor	Marketing	Editar <a href="#">Deletar</a>

Fonte: Elaborado pelos autores, 2025.

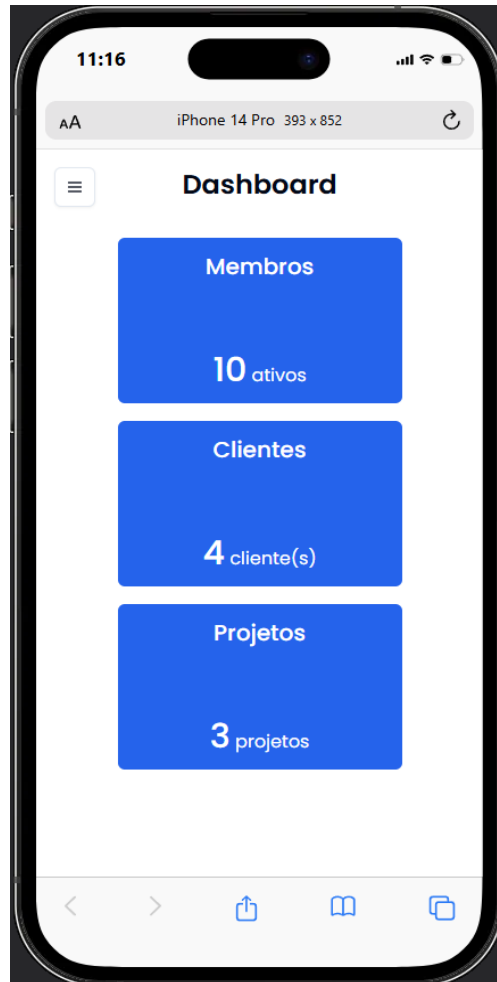
As próximas telas exemplificam o funcionamento do sistema em dispositivos móveis, como pode ser visto nas Figuras 15, 16 e 17.

Figura 15 – Tela de Login no Dispositivo Móvel



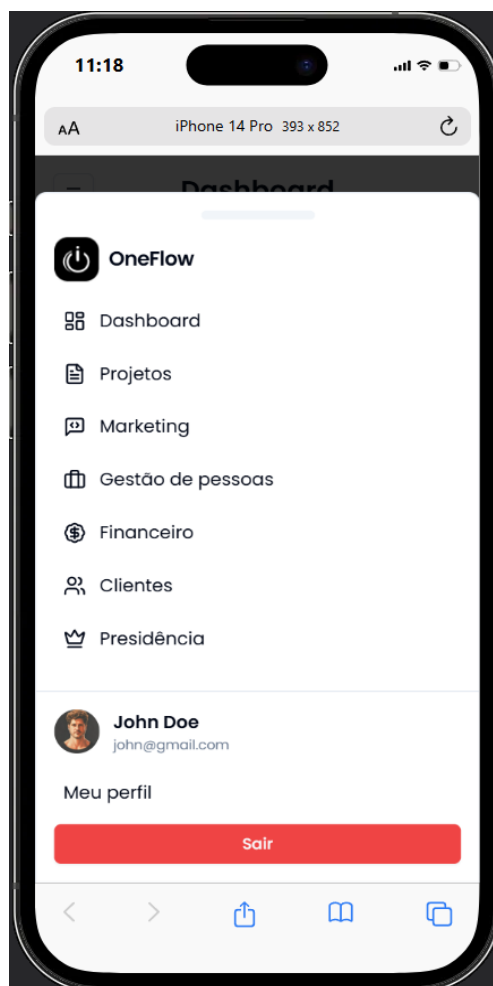
Fonte: Elaborado pelos autores, 2025.

Figura 16 – Tela Principal no Dispositivo Móvel



Fonte: Elaborado pelos autores, 2025.

Figura 17 – Menu do Sistema no Dispositivo Móvel



Fonte: Elaborado pelos autores, 2025.

Os requisitos funcionais e não funcionais apresentados anteriormente, definem as bases para o desenvolvimento e operação do sistema, enquanto as telas mostram como esses requisitos foram implementados na prática. Por exemplo, o requisito funcional que exige o gerenciamento de membros (RF 01) está diretamente refletido na tela de gestão de usuários, em que é possível cadastrar, visualizar, editar e excluir membros. Da mesma forma, o requisito que determina o gerenciamento de reuniões (RF 04) é atendido pela tela específica de reuniões, que oferece funcionalidades para cadastrar, visualizar, editar e excluir encontros corporativos. O controle de diferentes níveis de permissão (RF 07) também é evidenciado nas restrições de acesso apresentadas em diversas telas, garantindo que apenas usuários com determinados cargos possam realizar ações específicas, como excluir projetos ou documentos.

Além disso, os requisitos não funcionais também são atendidos nas telas e funcionalidades descritas. Por exemplo, o requisito de que o sistema deve ter uma interface simples e intuitiva (RNF 02) é claramente observado no design das telas, como no *dashboard*

e nos formulários, que apresentam um *layout* organizado e fácil de usar. A necessidade de responsividade (RNF 03) também foi contemplada, garantindo que o sistema funcione adequadamente em diversos dispositivos, como navegadores e *tablets*. Essa correlação demonstra que os requisitos estabelecidos no Capítulo 3 guiaram todo o desenvolvimento técnico e visual do sistema, resultando em uma ferramenta alinhada às necessidades da Empresa Júnior OneBit e cumprindo os objetivos propostos no trabalho.

Foi criada uma documentação básica para o *frontend* e o *backend* do sistema, com o objetivo de auxiliar futuros desenvolvedores e membros da OneBit interessados em dar continuidade ao projeto, seja na manutenção do sistema ou na implementação de novas funcionalidades. Essa documentação está disponível na seção APÊNDICE.

## 4.2 Testes realizados

Durante o processo de desenvolvimento do sistema, os desenvolvedores realizaram os primeiros testes, realizando operações como cadastros, atualizações e exclusões, para garantir a correta comunicação entre o *frontend* e o *backend* da aplicação.

Após concluída a implementação, o sistema foi hospedado em uma plataforma web para que pudesse ser realizado os testes pelos membros da empresa. Esses testes tiveram como objetivo avaliar a funcionalidade de cada módulo, verificar a usabilidade da interface e identificar possíveis falhas ou inconsistências no sistema. Os membros foram orientados a explorar todas as funcionalidades disponíveis, como cadastro de usuários, gestão de projetos, acesso ao *dashboard* e uso das demais ferramentas do sistema.

Após os testes terem sido realizados, foi realizada uma conversa com os membros da empresa para coletar *feedback* sobre o sistema. Durante essa conversa, os membros destacaram que o sistema se mostrou altamente intuitivo e fácil de usar, com uma interface clara e organizada, facilitando a navegação e o entendimento das funcionalidades. Eles também enfatizaram que as ferramentas disponibilizadas, como o *dashboard*, a gestão de usuários e a administração de projetos, atenderam plenamente às necessidades operacionais da empresa, tornando o sistema uma solução prática e eficiente.

Os membros relataram que o sistema permitirá um gerenciamento mais centralizado e ágil das informações, contribuindo para a organização e o acompanhamento das atividades diárias. Além disso, eles reconheceram que funcionalidades como a gestão de reuniões e documentos são diferenciais que irão otimizar a comunicação e a colaboração interna. O *feedback* positivo recebido reforçou o alinhamento do sistema às demandas da

empresa, validando sua implementação como uma ferramenta indispensável para a gestão empresarial.

## 5 CONSIDERAÇÕES FINAIS

Este trabalho teve como objetivo o desenvolvimento de um sistema de gestão personalizado para a Empresa Júnior OneBit, com foco em otimizar os processos internos, centralizar informações e facilitar o gerenciamento das atividades. Ao longo do projeto, foi realizada uma análise das necessidades específicas da empresa, seguida pela implementação de um sistema funcional e intuitivo, que inclui funcionalidades como gestão de usuários, projetos, clientes, reuniões e acesso a dados importantes da EJ.

Após a conclusão do desenvolvimento, o sistema foi submetido a testes pelos membros da empresa, que avaliaram sua usabilidade e eficácia. O *feedback* recebido indicou que o sistema cumpriu com todos os objetivos propostos, apresentando uma solução prática e de fácil utilização, com potencial para melhorar significativamente a gestão e a organização da empresa.

Este projeto demonstra como a tecnologia pode ser uma aliada poderosa na gestão de empresas juniores, oferecendo ferramentas personalizadas e eficientes. Como trabalhos futuros, sugere-se a possibilidade de adicionar novas funcionalidades, como relatórios automáticos, filtros de pesquisa, notificações das reuniões e uma aplicação nativa para dispositivos móveis, ampliando ainda mais o alcance e a utilidade do sistema.

Conclui-se, portanto, que o desenvolvimento do sistema de gestão OneFlow foi bem-sucedido e representa uma contribuição significativa para a modernização e eficiência da Empresa Júnior OneBit, fornecendo uma base sólida para futuras melhorias e expansões.

## REFERÊNCIAS

- AMBAR, Ingrid Carpim. **O conceito de empresa júnior**. Sorocaba: UNESP, 2023. Disponível em: <https://www.sorocaba.unesp.br/#!/empresa-junior227/parceiros/>. Acesso em: 05 abr. 2024.
- ANDRADE, Giovana. **Conhecendo o mej**: livro 1. [S. I.]: UFPEL, 2015. Disponível em: <https://wp.ufpel.edu.br/crej/files/2012/09/DNAJu%CC%81nior-Livro-I-Conhecendo-o-MEJ.pdf>. Acesso em: 08 abr. 2024.
- ARAÚJO, Kênia Alves Pereira; SILVA, André Luiz Gomes da. **Sistema de gerenciamento para a empresa júnior OneBit**. São João Evangelista: Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais – *Campus São João Evangelista*. 2015.
- ARONEZA, Juan. **Sistemas de Gestão**: conheça seus diferentes tipos e benefícios para empresas. 4cinco, [2023?]. Disponível em: <https://4cinco.com/sistema-de-gestao/>. Acesso em: 06 abr. 2024.
- ASSIS, Maria Cristina de. **Metodologia do trabalho científico**. Linguagens: usos e reflexões. 3. ed. João Pessoa: Editora Universitária UFPB. 2009.
- CAMPOS, Elziane Bouzada Dias; ABBAD, Gardênia da Silva; FERREIRA, Camila Zeitouni e NEGREIROS, João Luis Xavier Magalhães de. Empresas juniores como espaços de apoio à formação profissional de estudantes universitários brasileiros. Florianópolis: **Revista Psicologia Organizações e Trabalho**, v. 14, n. 4, p. 452-463, 2014. Disponível em: [http://pepsic.bvsalud.org/scielo.php?pid=S1984-66572014000400011&script=sci\\_arttext](http://pepsic.bvsalud.org/scielo.php?pid=S1984-66572014000400011&script=sci_arttext). Acesso em: 05 abr. 2024
- COUTINHO, Thiago. **ReactJS**: entenda o que é e como essa biblioteca pode ser aplicada. Voitto, 2023. Disponível em: <https://www.voitto.com.br/blog/artigo/o-que-e-reactjs>. Acesso em: 17 abr. 2024.
- CUNHA, Fernando. **Sistema web**: o que é e como funciona? *Mestres da web*, 2022. Disponível em: <https://www.mestresdawe.com.br/tecnologias/sistema-web-o-que-e-e-como-funciona>. Acesso em: 17 abr. 2024.
- EQUIPE TOTVS. **O que é CSS? Conheça benefícios e como funciona**. Totvs, 2020. Disponível em: <https://www.totvs.com/blog/developers/o-que-e-css/>. Acesso em: 17 abr. 2024.
- EQUIPE TOTVS. **Software de gestão**: o que é, funções e benefícios. Totvs, 2023. Disponível em: <https://www.totvs.com/blog/negocios/software-de-gestao/>. Acesso em: 18 mar. 2024.
- ESTRELLA, Carlos. **O que é JavaScript e para que serve na programação web**. Hostinger, 2023. Disponível em: <https://www.hostinger.com.br/tutoriais/o-que-e-javascript>. Acesso em: 17 abr. 2024.

FIGMA. **Figma**. Figma, 2024. Disponível em: <https://www.figma.com/design/>. Acesso em: 15 abr. 2024.

FLANAGAN, David. **JavaScript: o guia definitivo**. 6. ed. Porto Alegre: Bookman, 2013. Disponível em: [https://github.com/free-educabooks/blob/main/books/JavaScript%20%20Guia%20Definitivo\\_v2.pdf](https://github.com/free-educabooks/blob/main/books/JavaScript%20%20Guia%20Definitivo_v2.pdf). Acesso em: 17 abr. 2024.

FONSECA, Gabriel Viggiano. **Desenvolvimento de um sistema web para apoio ao gerenciamento de empresas juniores**. TCC (Graduação em Sistemas de Informação). Serra: Instituto Federal do Espírito Santo, 2023. Disponível em: <https://repositorio.ifes.edu.br/handle/123456789/4356>. Acesso em: 01 ago. 2024.

GEEKHUNTER. **Spring Boot: tudo que você precisa saber**. Blog Geekhunter, 2022. Disponível em: <https://blog.geekhunter.com.br/tudo-o-que-voce-precisa-saber-sobre-o-spring-boot/>. Acesso em: 07 abr. 2024.

GITHUB. **Sobre o Github e o Git**. Docs Github, 2024. Disponível em: <https://docs.github.com/pt/get-started/start-your-journey/about-github-and-git>. Acesso em: 15 abr. 2024.

GUEDES, Marylene. **O que é e como começar com Java**. Treinaweb, 2018. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-e-como-comecar-com-java>. Acesso em: 07 abr. 2024.

LEÃO, Thiago. **Problemas de não ter um ERP na sua fábrica**. Nomus, 2024. Disponível em: <https://www.nomus.com.br/blog-industrial/nao-ter-um-erp/>. Acesso em: 05 abr. 2024.

MATOS, Évilin. **Avanço da tecnologia nos últimos 10 anos: de casa ao trabalho**. Blog Runrun.it, 2021. Disponível em: <https://blog.runrun.it/avanco-da-tecnologia/>. Acesso em: 05 abr. 2024.

MDN Web Docs. **HTML básico**. Developer Mozilla, 2024. Disponível em: [https://developer.mozilla.org/pt-BR/docs/Learn/Getting\\_started\\_with\\_the\\_web/HTML\\_basics](https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/HTML_basics). Acesso em: 17 abr. 2024.

OLIVEIRA, Lucas Anthony Ferreira de. **Um sistema para gerenciamento de empresas juniores**. (Trabalho de Conclusão de Curso - Artigo). Curso de Bacharelado em Ciência da Computação, Centro de Engenharia Elétrica e Informática. Paraíba: Universidade Federal de Campina Grande - Brasil, 2022. Disponível em: <http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/29267>. Acesso em: 10 ago. 2024.

OLIVEIRA, Maria Marly de. **Como Fazer Pesquisa Qualitativa**. 3. ed. Petrópolis: Vozes, 2008.

ONEBIT. **Onebit**. OneBit, 2024. Disponível em: <https://onebitjr.com.br/>. Acesso em: 05 abr. 2024.

ONEBIT. **Carta de Serviço**. OneBit, 2024. Disponível em: <https://drive.google.com/file/d/1GXm53RA6WxA4HrJ59KLLj6-FsM7Uohyw/view?usp=sharing>. Acesso em: 05 abr. 2024.

ORACLE. **O que é o MySQL?** Oracle, [200-]. Disponível em: <https://www.oracle.com/br/mysql/what-is-mysql/>. Acesso em: 07 abr. 2024.

PEREIRA, Gabriel. **O que é kanban? Veja 6 dicas para aplicar a metodologia ágil no Trello**. Techtudo, 2023. Disponível em: <https://www.techtudo.com.br/listas/2023/02/o-que-e-kanban-veja-6-dicas-para-aplicar-a-metodologia-agil-no-trello-edsoftwares.ghtml>. Acesso em: 07 abr. 2024.

PISA, Pedro. **O que é e como usar o MySQL?** Techtudo, 2012. Disponível em: <https://www.techtudo.com.br/noticias/2012/04/o-que-e-e-como-usar-o-mysql.ghtml>. Acesso em: 07 abr. 2024.

REDHAT. **O que é IDE (Ambiente de desenvolvimento integrado)?** Redhat, 2023. Disponível em: <https://www.redhat.com/pt-br/topics/middleware/what-is-ide>. Acesso em: 15 abr. 2024.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011. Disponível em: <https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>. Acesso em: 06 abr. 2024.

TRELLO. **Trello**. Trello, 2024. Disponível em: <https://trello.com/tour>. Acesso em: 07 abr. 2024.

VELOSO, Ricardo Silva. **TypeScript**. WIKIPEDIA, 2024. Disponível em: <https://pt.wikipedia.org/wiki/TypeScript>. Acesso em: 17 ago. 2024.

XPEDUCAÇÃO. **Desenvolvimento web: tudo o que você precisa saber**. Xpedução, 2022. Disponível em: <https://blog.xpeducacao.com.br/desenvolvimento-web/>. Acesso em: 17 abr. 2024.

## APÊNDICES

### APÊNDICE A – Documentação do *frontend*

#### TECNOLOGIAS UTILIZADAS

- Framework: React
- Linguagem de Programação: TypeScript
- Gerenciamento de Estado: useState, Context API
- Estilização: Tailwind CSS, Shadcn/UI
- Comunicação Assíncrona: Fetch API
- Roteamento: React Router DOM
- Ferramenta de Build: Vite

#### ARQUITETURA E ESTRUTURA DO CÓDIGO

O arquivo **App.tsx** é o arquivo principal do projeto, ele configura as rotas e o comportamento da aplicação. Ele usa o **react-router-dom** para gerenciar a navegação entre as páginas e o **createHashRouter** para permitir navegação baseada em *hash*. As rotas são protegidas por um componente **ProtectedRoute**, garantindo que apenas usuários autenticados possam acessar páginas como o *dashboard* e as de gestão interna.

Além disso, o **AuthProvider** envolve toda a aplicação, fornecendo o contexto de autenticação, e o **Toaster** exibe notificações de sucesso ou erro para o usuário. O *layout* principal da aplicação é gerenciado pelo componente **DefaultLayout**, que organiza o conteúdo das páginas internas.

#### PROCESSO DE INSTALAÇÃO E CONFIGURAÇÃO

- Clonar o repositório: `git clone https://github.com/edumaia85/oneflow-frontend.git`.
- `cd oneflow-frontend`.
- Instalar as dependências: `npm install`.
- Inicializar o projeto: `npm run dev`.

O sistema estará acessível em **http://localhost:5173** (ou conforme configuração específica).

#### ESTRUTURA DE PASTAS E ARQUIVOS

- `public`: Recursos públicos como imagens e fontes.
- `src`: Contém o código-fonte principal.

- `assets`: Recursos estáticos específicos do projeto.
- `components`: Armazena componentes reutilizáveis, organizados por domínio e responsabilidade dentro da aplicação.
- `hooks`: Contém *hooks* customizados para encapsular lógica reutilizável relacionada a estados e efeitos colaterais.
- `pages`: Cada arquivo neste diretório corresponde a uma página navegável da aplicação e contém a lógica específica da interface correspondente.
- `utils`: Conjunto de funções auxiliares, como formatação de dados, manipulação de *strings*, operações matemáticas e outras funções genéricas utilizadas em múltiplas partes do sistema.
- `layouts`: Contém estruturas reutilizáveis de *layout* da aplicação, como barras de navegação, rodapés e containers de conteúdo. Esses *layouts* garantem a consistência visual entre diferentes páginas.
- `contexts`: Armazena os arquivos responsáveis por gerenciar estados globais utilizando *context* API, permitindo que informações como autenticação, temas e configurações sejam acessadas de forma centralizada em toda a aplicação.
- `package.json`: Dependências e *scripts* do projeto.
- `tsconfig.json`: Configuração do *TypeScript*.
- `vite.config.ts`: Configuração do *Vite*.

## APÊNDICE B – Documentação do *backend*

### TECNOLOGIAS UTILIZADAS

- Linguagem: Java versão 17.
- Framework: Spring Boot.
- Banco de dados: MySQL.
- Gerenciador de dependências: Maven.

### PROCESSO DE INSTALAÇÃO E CONFIGURAÇÃO

- Clonar o repositório: `git clone https://github.com/MarcosSilva13/backend-oneflow.git`.
- Necessário ter o Java na versão 17 e o banco de dados MySQL instalados em sua máquina.
- Configure o banco de dados carregando o *script sql*.
- O projeto pode ser executado tanto no Visual Studio Code ou no IntelliJ Idea.
- Ajuste o arquivo **application-dev.properties** com suas configurações de banco de dados e credenciais.
- No arquivo **application.properties**, certifique de que a linha 2 esteja da seguinte forma **spring.profiles.active=dev**, dessa maneira a aplicação será executada com o ambiente de desenvolvimento.
- Abra a classe **OneflowApplication.java** para executar a aplicação, se tudo estiver corretamente configurado, a aplicação estará disponível no endereço: **http://localhost:8080**.
- Para visualizar as todas as rotas documentadas e quais são os dados necessários e códigos de erro, acesse o endereço: **http://localhost:8080/swagger-ui/index.html** onde será carregada a documentação gerada pelo *swagger*.

### ESTRUTURA DE PASTAS E ARQUIVOS

A estrutura de pastas do projeto segue um padrão organizado e modularizado, utilizando boas práticas do *Spring Boot* e **arquitetura em camadas**. A organização está dividida em três grandes pacotes:

- **api**: Responsável pela comunicação externa (controladores, DTOs, mapeadores e documentação).
- **domain**: Contém as regras de negócio, incluindo entidades, repositórios e serviços.

- **infra**: Responsável pela infraestrutura da aplicação, como configurações, segurança e tratamento de erros.

### **PACOTE API**

- Esse pacote é responsável pela entrada e saída de dados da aplicação.
- **controller**: Contém as classes que expõem *endpoints* da API REST, recebendo e respondendo requisições HTTP.
- **documentation**: Contém arquivos de documentação da API, utilizando *Swagger* para descrever os *endpoints*.
- **dto (Data Transfer Object)**: Contém classes usadas para transferência de dados entre o *frontend* e o *backend*, evitando expor diretamente as entidades do banco de dados.
- **mapper**: Contém classes que fazem a conversão entre DTOs e entidades, utilizando o *MapStruct* para mapear as conversões.

### **PACOTE DOMAIN**

- Esse pacote representa o domínio da aplicação, contendo as regras e abstrações principais.
- **entity**: Contém as classes que representam tabelas do banco de dados (modelo de domínio).
- **enums**: Contém enumerações usadas na aplicação para valores fixos, como tipos de status e cargos dos usuários.
- **repository**: Contém as interfaces para acesso ao banco de dados, estendendo a *JpaRepository*.
- **service**: Contém as classes com a lógica de negócios da aplicação, intermediando entre controladores(*controller*) e repositórios(*repository*).
- **util**: Contém métodos e constantes auxiliares, que são utilizados por várias classes do sistema.

### **PACOTE INFRA**

- Esse pacote lida com configurações técnicas da aplicação.
- **configuration**: Contém classes de configuração do *Spring Boot*, como permissões de acesso, CORS, serviço de imagem e e-mail, entre outros.
- **exception**: Contém classes para tratamento de erros e exceções personalizadas, melhorando o controle sobre falhas.

- `security`: Contém configurações relacionadas à segurança, como autenticação e autorização, utilizando o *Spring Security* e a autenticação JWT.

## **PACOTE RESOURCES**

- Esse pacote contém arquivos de configuração da aplicação.
- `application.properties`: Configuração padrão do *Spring Boot*. Define porta do servidor, credenciais de banco de dados, perfil atual de execução, entre outros.
- `application-dev.properties`: Configurações para o ambiente de desenvolvimento (exemplo: banco de dados local).
- `application-prod.properties`: Configurações para o ambiente de produção (exemplo: banco de dados em nuvem).
- `secrets.properties`: Armazena as chaves sensíveis, como *tokens* e credenciais de APIs de serviços externos.

O arquivo **`pom.xml`** que se encontra na pasta raiz do projeto é o arquivo responsável por todas as dependências da aplicação.

Foram utilizados alguns serviços externos como a api do **Mailjet** utilizada para envio de e-mails para redefinição de senha. O **Cloudinary** foi utilizado para o armazenamento em nuvem das imagens de perfil dos usuários.

## APÊNDICE C – Rotas da api no *swagger*

Abaixo são exibidas as rotas da api geradas e configuradas com a ferramenta *swagger*. Em cada uma delas é possível clicar para abrir e visualizar os dados necessários para enviar uma requisição, os dados retornados de uma resposta e também os possíveis retornos com o *status* http do erro.

### Rota de Usuários

Usuários		^
GET	/api/users	Lista todos os usuários
PUT	/api/users	Atualiza os dados de um usuário
PATCH	/api/users	Atualiza a senha do usuário
PUT	/api/users/role-sector	Atualiza o cargo e setor de um usuário
POST	/api/users/reset-password	Reseta a senha do usuário
POST	/api/users/forgot-password	Cria o token de redefinição de senha e envia por email
PATCH	/api/users/remove-image	Remove a imagem do usuário
PATCH	/api/users/image	Atualiza a imagem do usuário
GET	/api/users/{userId}	Retorna um usuário
DELETE	/api/users/{userId}	Deleta um usuário

### Rota de Setores

Setores		^
GET	/api/sectors/{sectorId}	Retorna um setor
PUT	/api/sectors/{sectorId}	Atualiza os dados de um setor
GET	/api/sectors	Lista todos os setores

### Rota de Projetos

Projetos		^
PUT	/api/projects/{projectId}	Atualiza os dados de um projeto
DELETE	/api/projects/{projectId}	Deleta um projeto
GET	/api/projects	Lista todos os projetos de um setor
POST	/api/projects	Registra um novo projeto

### Rota de Reuniões

Reuniões		^
PUT	/api/meetings/{meetingId}	Atualiza os dados de uma reunião
DELETE	/api/meetings/{meetingId}	Deleta uma reunião
GET	/api/meetings	Lista todas as reuniões de um setor
POST	/api/meetings	Registra uma nova reunião

## Rota de Links de Documentos

Links de Documentos		^
PUT	/api/documents/{documentLinkId} Atualiza os dados de um link	🔒 ↓
DELETE	/api/documents/{documentLinkId} Deleta um link	🔒 ↓
GET	/api/documents Lista todos os links de um setor	🔒 ↓
POST	/api/documents Registra um novo link	🔒 ↓

## Rota de Clientes

Clientes		^
GET	/api/customers/{customerId} Retorna um cliente	🔒 ↓
PUT	/api/customers/{customerId} Atualiza os dados de um cliente	🔒 ↓
DELETE	/api/customers/{customerId} Deleta um cliente	🔒 ↓
GET	/api/customers Lista todos os clientes	🔒 ↓
POST	/api/customers Registra um novo cliente	🔒 ↓

## Rota de Autenticação

Autenticação		^
POST	/api/auth/register Registra um novo usuário	🔒 ↓
POST	/api/auth/login Realiza login	↓

## Rota do Dashboard

Dashboard		^
GET	/api/dashboard Retorna a quantidade de membros, projetos e clientes do sistema	🔒 ↓