

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

**JOÃO VICTOR PERES SILVA
LUÍS FERNANDO MENDES SILVA**

SISTEMA WEB PARA CONTROLE DE CHAMADO EMPRESARIAL

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Carlos Alberto Lucas

FRANCA/SP

2023

SISTEMA WEB PARA CONTROLE DE CHAMADO EMPRESARIAL

João Victor Peres Silva¹
Luís Fernando Mendes Silva²

Resumo

O controle e gerenciamento do processo de comunicação de dados e tramitação de documentos, é considerado como problema em algumas empresas que crescem de forma rápida e que às vezes acabam se perdendo no controle dos próprios meios de comunicação, principalmente quando adotam uma ferramenta externa, como o WhatsApp. Tal ação, eventualmente, pode provocar a perda de documentos, a dificuldade na compreensão entre os funcionários e o bloqueio da realização de atividades, gerando um atraso em horas ou até mesmo dias na troca de dados entre os setores. Prova disto foi o resultado de uma entrevista realizada com o gerente geral da empresa 'cliente', através das técnicas de elicitación, que relatou e apresentou que muitas vezes ocorrem falhas na comunicação dos colaboradores, gerando dados incompletos e que prejudicam o fluxo dos processos da empresa, o que ocasiona uma repetição de várias solicitações para que um mesmo processo seja realizado. Assim, foi desenvolvida uma solução sistêmica que auxiliará nesta empresa, proporcionando que todas as informações fiquem centralizadas em apenas um lugar, permitindo conversas contínuas e imediatas entre os colaboradores através de mensagens, evitando perdas e prejuízos aos processos e dados, consequentemente impactando no rendimento e na qualidade dos serviços realizados aos clientes.

Palavras-chave: Centralização. Comunicação. Dados. Informação.

Abstract

The control and management of the data communication and document processing process is considered a problem in some companies that grow quickly and that sometimes end up losing control of their own means of communication, especially when they adopt an external tool, such as the WhatsApp. Such action may eventually cause the loss of documents, difficulty in understanding among employees and blocking the carrying out of activities, generating a delay of hours or even days in the exchange of data between sectors. Proof of this was the result of an interview carried out with the general manager of the 'client' company, using elicitation techniques, who reported and presented that failures often occur in employee communication, generating incomplete data that impair the flow of company processes, which causes a repetition of several requests for the same process to be carried out. Thus, a systemic solution was developed that will help this company, providing that all information is centralized in just one place, allowing continuous and immediate conversations between employees through messages, avoiding losses and damages to processes and data, consequently impacting the performance and quality of services provided to customers.

Keywords: Centralization. Communication. Data. Information.

¹ Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: [vitorr-joao2222@hotmail.com].

² Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: [lfms20031027@gmail.com].

1 Introdução

Com o desenvolvimento das empresas, financeira e estruturalmente, o processo de comunicação se torna necessário com a máxima qualidade. A falta desta qualidade pode prejudicar o cumprimento de prazos para execução dos projetos e provocar problemas com os clientes. Para minimizar estas questões faz-se necessário o uso das novas tecnologias da informação e comunicação, visando o desenvolvimento de soluções sistêmicas e recursos viáveis, para o controle do fluxo destes dados compartilhados.

Com a intenção de contribuir com uma empresa da cidade de Franca/SP e aplicar os valiosos conhecimentos adquiridos com o curso, solicitamos apoio de alguns docentes que nos indicaram a empresa cliente, referência neste estudo.

A empresa Cliente atua há 35 anos no segmento de bobinas e maquinários de telhas, vem crescendo e regionalmente se tornando a principal no segmento em fabricação de máquinas e processamento de bobinas. Com isso, o seu tamanho em área construída também cresceu, chegando a mais de 15.000m².

No primeiro contato com o gerente geral da empresa 'cliente' observamos que o processo de comunicação e transmissão de dados era algo crônico e que incomodava todos na empresa.

Analisando a alta demanda diária de troca de informações, muitas vezes incompletas ou sem controle, era causado demora além do necessário para a conclusão de tarefas, afetando de forma negativa os demais processos da empresa.

Esses fatores resultavam em atrasos nos pedidos gerando confusão nos setores e a diminuição da produção, além do risco de pagar multas ou ter prejuízos econômicos de grande valor com o cancelamento do pedido.

Dessa forma, a questão é: como fazer com que a comunicação interna da empresa seja mais eficaz, rápida e segura, utilizando os recursos da tecnologia de informação?.

A solução encontrada foi centralizar em um único canal de comunicação os dados e informações tramitadas entre os colaboradores, fortalecendo a relação da equipe, diminuindo a sobrecarga dos supervisores e líderes, além de possibilitar a fácil localização de dados compartilhados com outros setores, criando um fluxo dinâmico entre as diversas demandas da empresa.

De acordo com BRUNO (2019, *online*) "O principal propósito da comunicação interna é assegurar que os funcionários estejam alinhados e motivados em relação

aos objetivos, valores e mensagens da empresa”, visto que esta seria uma necessidade.

O objetivo do projeto é desenvolver uma solução sistêmica através de uma aplicação, de forma que tais informações se centralizem e evitem divergências entre setores e funcionários, pensando em um *software* que tornará a interação mais integrada à cultura da empresa.

Um ponto de destaque que o sistema busca atender é permitir um maior direcionamento de tomada de decisão pelas lideranças, seja em questões técnicas ou voltadas para o monitoramento da maneira como os dados são passados, tudo levando em conta a facilidade de compreensão e o retorno do que foi solicitado.

Por meio dos artefatos de engenharia de *software* foi construída toda a documentação necessária para demonstrar a proposta descrita, permitindo o acompanhamento dos procedimentos internos em andamento, assim como a criação, gerenciamento e segurança dos colaboradores quanto a dados sensíveis.

Conseqüentemente, faz-se necessário dividir o projeto em dois escopos de desenvolvimento, sendo o *front-end* e *back-end*, com aplicação de conceitos de prototipação de tela, *design patterns* e componentização de elementos, mesmo que futuramente possa ser escalável e utilizado em outras corporações.

Por fim, foi de suma importância a implementação do *framework* React.js para a criação de um *software* moderno, que cumpra com as exigências de alta performance, interface e experiência do usuário, além da ferramenta Node.js no *back-end* e banco de dados PostgreSQL para o armazenamento de dados.

1.1 Termo de Abertura do Projeto (TAP)

O documento do Termo de Abertura do Projeto tem como objetivo formalizar o início do projeto e descrever todas as suas etapas, justificativas, premissas, riscos, restrições, marcos e partes interessadas (COUTINHO, 2020, *online*).

Serão abordadas todas as etapas a serem seguidas para o desenvolvimento do *software* e compreensão das necessidades da empresa. Para descrição será feita a análise de cada tópico a fim de justificar sua importância para o projeto e integrante do grupo.

1.1.1 Situação atual

Atualmente, não existe padronização no encaminhamento de solicitações. Como observado no levantamento de questionário aos *stakeholders*, os principais meios de comunicação são o celular, gmail e ramal de ligações o que deixa as informações avulsas e de difícil compreensão. Dessa forma, a definição de metas e desempenho esperado não fica tão claro para os funcionários.

Os prazos definidos, bem como o retorno desses processos, podem prejudicar o andamento de outro setor por sua demora ou falta de dados. Seria necessário então, um maior tempo para confirmar com os responsáveis, tornando todo o processo complexo e demorado.

Um ponto positivo é que a empresa possui um forte espírito de equipe e uma cultura que favorece a implementação de novos procedimentos internos. A boa relação entre os colaboradores pode conduzir a uma mudança mais natural na maneira de estabelecer a resolução de problemas dentro da equipe.

Com a análise das respostas também foi possível notar que às vezes é necessário o deslocamento excessivo dentro do prédio. O que pode ser apenas uma confirmação, acarreta a perda de produtividade ou atraso na entrega de procedimentos por não achar o responsável procurado ou demorar nesta locomoção.

1.1.2 Justificativa do Projeto

A justificativa do projeto se baseia em criar uma solução sistêmica para auxiliar o cumprimento de prazo de entrega de procedimentos e organização de autorizações, gerando maior facilidade de retorno dos dados e no gerenciamento do grande fluxo de informações passado entre colaboradores e setores da empresa.

1.1.3 Propósito do Projeto e Metas

O projeto tem como objetivo otimizar o tempo de resposta entre setores e colaboradores para que ambas as partes se beneficiem desta troca de informações. Essa comunicação deve ter suas informações armazenadas de forma centralizada, ou seja, utilizando o *software* como intermédio para ações de autorização, abertura de procedimento internos e esclarecimentos.

Outro ponto a ser melhorado, diz respeito à forma como as informações são passadas, tornando-as mais claras e de fácil interpretação. Para isso, busca-se o

preenchimento de campos obrigatórios na abertura de um chamado, garantindo que os envolvidos possam realizar o seu retorno dependendo das dúvidas presentes, ou falta de informações.

Com o passar do tempo, esse retorno irá criar um padrão a ser seguido pela empresa ou equipe, dependendo da maneira como é conduzido, nível de prioridade, autorizações a serem confirmadas, definição de metas e prazos.

Por fim, foi observado que uma das grandes barreiras para tornar os procedimentos mais rápidos e eficientes, é a necessidade de deslocamento dos colaboradores para confirmação de dados ou documentos enviados com determinada liderança ou setor. O sistema fornece uma plataforma com diversas opções de comunicação, seja por *chat*, vídeo ou ligação, para que seja evitada a locomoção desnecessária, principalmente para situações de fácil resolução.

O objetivo é tornar a comunicação interna de empresas mais eficiente e clara, com foco na organização de dados (solicitações, arquivos etc.) ou autorizações passadas por meio de banco de dados. Com base nas descrições anteriores as metas visadas são:

- Possibilitar a definição de prioridade do chamado e controle de *status* pelas lideranças.
- Criar sistema de acompanhamento do chamado aberto de forma que priorize o prazo de resolução.
- Garantir segurança e integridade nos dados enviados, principalmente para documentos e autorizações.
- Tornar a comunicação entre os colaboradores e setores diferentes da empresa mais claros e centralizados (*chat*, vídeo e áudio), de forma que se torne uma ferramenta padrão.
- Manter a interação e troca de experiência entre os colaboradores, mesmo evitando o deslocamento pelo prédio da empresa.

1.1.4 Descrição do Projeto

Este projeto está dividido em 5 etapas. A primeira se concentra na Pesquisa de Mercado e seguimento de atuação da empresa Cliente e criação de formulário de perguntas para a melhor compreensão da atual situação dos processos realizados e de pontos a serem trabalhados.

A segunda etapa é responsável pelo registro das respostas fornecidas pelos colaboradores, reconhecimento de fraquezas e forças da empresa, definição de melhorias e validações com a criação do SWOT, 5W2H e BPMN.

Em terceiro é a elicitação de Requisitos Funcionais e Não Funcionais com documentação, bem como a criação de Diagramas de Casos de Uso, Atividade e Sequência, juntamente com suas revisões.

A quarta etapa possui foco na prototipação de telas seguindo os requisitos pré-estabelecidos, escolha do banco de dados e linguagem que melhor se adequem ao desempenho da solução sistêmica, também levando em consideração a experiência na utilização, segurança e produtividade.

Na quinta e última etapa é feita a definição de funcionalidades e a apresentação da proposta comercial no final do projeto.

1.1.5 Premissas

- Premissa de um *software* intuitivo e com ótima operabilidade
- Cumprir com as atribuições e prazos estipulados
- Desenvolver o projeto visando a fácil implantação e manutenção, para auxiliar no cumprimento dos prazos de solicitações ou autorizações
- Validação dos artefatos produzidos com integrantes do projeto e orientador
- Revisão periódica da documentação para seu aprimoramento

1.1.6 Restrições

- Prazo de entrega do projeto
- Capacitação técnica dos colaboradores
- Tempo disponível dos membros de equipe
- Considerar recursos de *software* e *hardware* disponíveis pela empresa

1.1.7 Stakeholders

- Partes interessadas como colaboradores e lideranças da empresa

1.1.8 Riscos

- Segurança e integridade dos dados e grande fluxo de informações

- Concorrência de empresas que já utilizam sistemas de controle de chamados como diferencial
- Falta de acompanhamento do cronograma de entrega dos artefatos
- Falta de qualidade na entrega de artefatos ao decorrer do projeto, com documentação insuficiente
- Falta de coerência nos diferentes registros ao longo do desenvolvimento da aplicação

1.1.9 Marcos

O cronograma foi estipulado para as entregas iniciais, desde sua concepção, até a data de entrega do documento e apresentação do sistema, seguindo a data de início do projeto no dia 13/08/22 e a data de entrega da documentação até o dia 15/11/23.

2 Viabilidade do projeto

A viabilidade do projeto pode ser apresentada por meio do quadro SWOT, abreviação das palavras em inglês (*Strengths, Weaknesses, Opportunities e Threats*), sua função é avaliar o ambiente interno (força e fraqueza) e externo (oportunidades e ameaças) da empresa e definir os fatores a serem melhorados.

Segundo uma publicação da página oficial da *TOTVS* (2005, *online*) “A matriz *SWOT* ajuda as empresas a se planejarem fazendo com que a empresa tome uma melhor decisão de suas ações”. Por meio desse planejamento estratégico, a tomada de decisões se torna mais simples de ser realizada pensando no funcionamento e interação com o sistema, já que existe resultado efetivo por abranger um escopo interno e externo da empresa.

Realizando a descrição da estrutura do quadro, as forças se referem às iniciativas internas da empresa que são vistas positivamente no levantamento, pois apresentam bom desempenho no funcionamento de seus processos e criam pontos que a diferencia dos concorrentes.

As fraquezas continuam com foco interno, porém se referem aos aspectos que tenham desempenho abaixo do esperado. Esta é uma das principais áreas para analisar o que pode ser melhorado e cria oportunidades no desenvolvimento do sistema, de forma a suprir esta fragilidade.

Se tratando de uma visão externa, em uma posição competitiva em relação ao mercado, as oportunidades são uma combinação entre as forças e fraquezas, visando a melhoria dos processos já executados e sendo fundamental para a prevenção das ameaças. Elas, que também são de fatores externos que podem causar adversidades nas atividades realizadas, relevância no cenário corporativo ou queda na produtividade.

Figura 1: Quadro *SWOT*



Fonte: os autores

Outro recurso de importância equivalente ao *SWOT* é a matriz *5W2H*, considerada uma ferramenta administrativa de atividades, prazos e responsabilidades a fim de registrar de maneira clara e eficiente como serão efetuadas as ações e quem participa, quando, onde, por que, como e quanto irá custar para a empresa.

Assim, o planejamento de procedimentos complexos e simples podem ser feitos de forma organizada para a melhoria das futuras ações a serem tomadas e visualização de problemas que necessitam reestruturação. Ao definir estes pontos, será possível criar um plano de ação detalhado e com agilidade a fim de garantir o desenvolvimento das tarefas e cumprimento da demanda.

Figura 2: Matriz 5W2H

Ação	O quê (What?)	Por quê (Why)	Onde (Where?)	Quem (Who?)	Quando (When?)	Como (How?)	Quanto? (How much?)
1	Centralizar principais formas de comunicação da empresa por meio de uma nova aplicação	Padronizar a busca de informações e consulta para determinar um canal de comunicação organizado	Qualquer local dentro empresa com conexão a internet	Empresas de pequeno e médio porte	Utilizar o software em períodos de baixa demanda para testes e familiarização	Pesquisar o Mercado, Implementar testes e monitoramento de desempenho dos profissionais; Utilizar principais meios de comunicação da empresa pelo software	Valor para desenvolvimento e implementação + Manutenção
2	Registrar solicitações e autorizações	Armazenar em Banco de Dados para que, caso necessário, as solicitações anteriores possam ser consultadas, tornando prático a análise de um procedimento que precisa ser revisto	Qualquer local dentro empresa com conexão a internet	Qualquer funcionário que realizar uma solicitação	Assim que o chamado for aberto, seguido de suas respostas para armazenamento	Utilizar de Banco de Dados para armazenar a abertura de chamados; Garantir a segurança dos dados; Autorizar funcionários para o armazenamento de respostas e eventuais conversas	Tempo para registro
3	Otimizar tempo de retorno de informações	Tornar o processo de retorno das solicitações e autorizações menor, de forma que não prejudique o trabalho de outros setores que aguardam a resposta	Determinados setores que realizam requisições a outras equipes e líderes	Qualquer funcionário que realizar uma solicitação	Durante e posterior a abertura do chamado	Utilizar sistema de notificações que sinalize a solicitação pendente de retorno; Classificar nível de urgência da resposta; Definir prazo máximo para resposta; Sinalizar conclusão do chamado para todos os envolvidos	Tempo para registro e retorno + Disposição para definir métricas
4	Reestruturar a forma como a informação é passada, tornando-a mais clara e objetiva	Criar respostas claras e objetivas para evitar falta de compreensão, realização incorreta de procedimentos e demora na conclusão do chamado por dados insuficientes	Qualquer local dentro empresa com conexão a internet e disposição de um aparelho celular ou computador	Qualquer funcionário que realizar uma solicitação	Durante todo o processo de comunicação por meio de abertura de um chamado, da abertura até sua conclusão	Requisitar preenchimento de campos obrigatórios; Fornecer ferramentas para complementar a abertura do chamado, como chat e chamada por vídeo; Criar requisição de novas informações e dados caso a solicitação não for compreendida	Disposição para preenchimento completo dos campos + Tempo de adaptação com no procedimento
5	Facilitar a comunicação entre setores diferentes e em locais distintos da empresa	Promover o fortalecimento da cultura da empresa e integração dos funcionários.	Setores com disponibilidade de smartphones, computadores e outro dispositivo para comunicação	Líderes de cada setor, e colaboradores que tenham alta de necessidade de autorizações para execução de tarefas	No momento de demanda de outros departamento ou requisição de apoio dos líderes	Utilizar dispositivos já conhecidos pelos colaboradores como celulares e computadores para interação entre setores e líderes; Possibilitar o acompanhamento da solicitação de forma organizada; Definir resposta para conclusão do chamado e indicação de retorno.	Disposição para retorno das solicitações + Redefinição na forma como as equipes interagem.

Fonte: os autores

2.1 Canvas de Negócio (*Business Model Canvas – BMC*)

O Business Model Canvas, conhecido em português como Modelo de Negócios, destaca-se como uma ferramenta essencial não apenas para o desenvolvimento de novos empreendimentos, mas também para a avaliação e ajuste contínuo de empresas já estabelecidas em busca de aprimoramento de seus objetivos e estratégias.

Nesse contexto, o *Business Model Canvas*, concebido pelo suíço Alexander Osterwalder com o apoio de uma rede global de 200 consultores, apresenta-se como uma ferramenta visual poderosa para estruturar e avaliar negócios (EJFGV, [s.d.], *online*).

O Modelo de Negócios é composto por um quadro com nove blocos, cada um representando um aspecto crucial do empreendimento. Esses blocos são preenchidos com ideias e informações, proporcionando uma representação visual clara da estrutura e dinâmica do negócio. Em um mundo empresarial em constante evolução, o *BMC* surge como uma peça fundamental na tomada de decisões estratégicas, capacitando empreendedores a adaptarem-se e prosperarem em ambientes desafiadores. (EJFGV, [s.d.], *online*).

2.1.1 Segmento de Clientes

O sistema atende principalmente a empresas de pequeno e médio porte que buscam otimizar a gestão de usuários e chamados, bem como funcionários de

empresas que precisam de uma solução eficaz para gerenciar usuários e solicitar atendimento entre diversos setores da mesma corporação.

2.1.2 Propostas de Valor

Nossa principal proposta de valor é a criação de um sistema de chamados seguro, com interface simples e que ofereça agilidade na automação de processos, centralização de informações, melhor comunicação interna entre departamentos e segurança dos dados com controle de acesso.

2.1.3 Canais

O Discord e WhatsApp foram os principais canais de comunicação, utilizados para reuniões, *feedback* e interação com a equipe e partes interessadas. Neles, foi possível estabelecer uma transparência e melhoria contínua no desenvolvimento do sistema.

2.1.4 Relacionamento

Foi mantido um relacionamento próximo com a empresa Cliente por meio de entrevistas e comunicação contínua para coletar requisitos e *feedback*. Além disso, a transparência em relação aos objetivos e funcionalidades do sistema se manteve constante.

Para o seguimento do projeto e futuro contato com partes interessadas esse comportamento será importante para mantermos uma relação sólida e que cria um vínculo de confiança com os usuários da aplicação.

2.1.5 Fontes de Receita

A principal fonte de receita será proveniente das mensalidades cobradas das empresas que usam o sistema, sendo essa cobrança escalável de acordo com as necessidades específicas de cada cliente.

As personalizações exigidas pelo clientes serão avaliadas criteriosamente para verificar sua real necessidade e se não impactarão nos fluxos já criados. Com isso, se busca ao mesmo tempo que criar um sistema flexível, manter a proposta inicial e escopo de atuação

2.1.6 Recursos Chave

Para alcançar seus objetivos, o trabalho conta com recursos-chave, como uma equipe de desenvolvimento, ferramentas como React.js, Node.js e TypeScript, e um banco de dados PostgreSQL utilizado em conjunto com Docker para visualização e desenvolvimento inicial, posteriormente migrando para a hospedagem no Render. A utilização de plataformas como Google *Forms*, WhatsApp e Discord desempenhou um papel essencial na comunicação, contribuindo para o alinhamento de resoluções e esclarecimento de dúvidas.

2.1.7 Atividades Chave

No cerne das principais atividades, o cadastro de usuários emerge como o ponto de partida para a criação de uma base de dados robusta, em que através desse processo, os usuários são identificados e diferenciados, permitindo gestão personalizada. Paralelamente, a abertura e acompanhamento de chamados constituem um pilar fundamental, proporcionando a capacidade de comunicação e resolução de questões de forma transparente e eficaz.

A incorporação de comentários aos chamados representa um avanço significativo na interação cliente-empresa, pois essa funcionalidade não apenas amplia o canal de comunicação, mas também abre espaço para um *feedback* valioso como opiniões, sugestões ou preocupações, contribuindo para a melhoria contínua do sistema.

Além disso, o controle de categorias de chamados oferece uma abordagem estratégica na classificação e resolução de demandas, garantindo uma gestão eficiente e organizada das diferentes naturezas de solicitações.

A criação dinâmica de departamentos e cargos complementa esse conjunto, proporcionando uma estrutura flexível e adaptável à hierarquia organizacional, otimizando a distribuição de responsabilidades e a fluidez nos processos internos.

2.1.8 Parcerias Chave

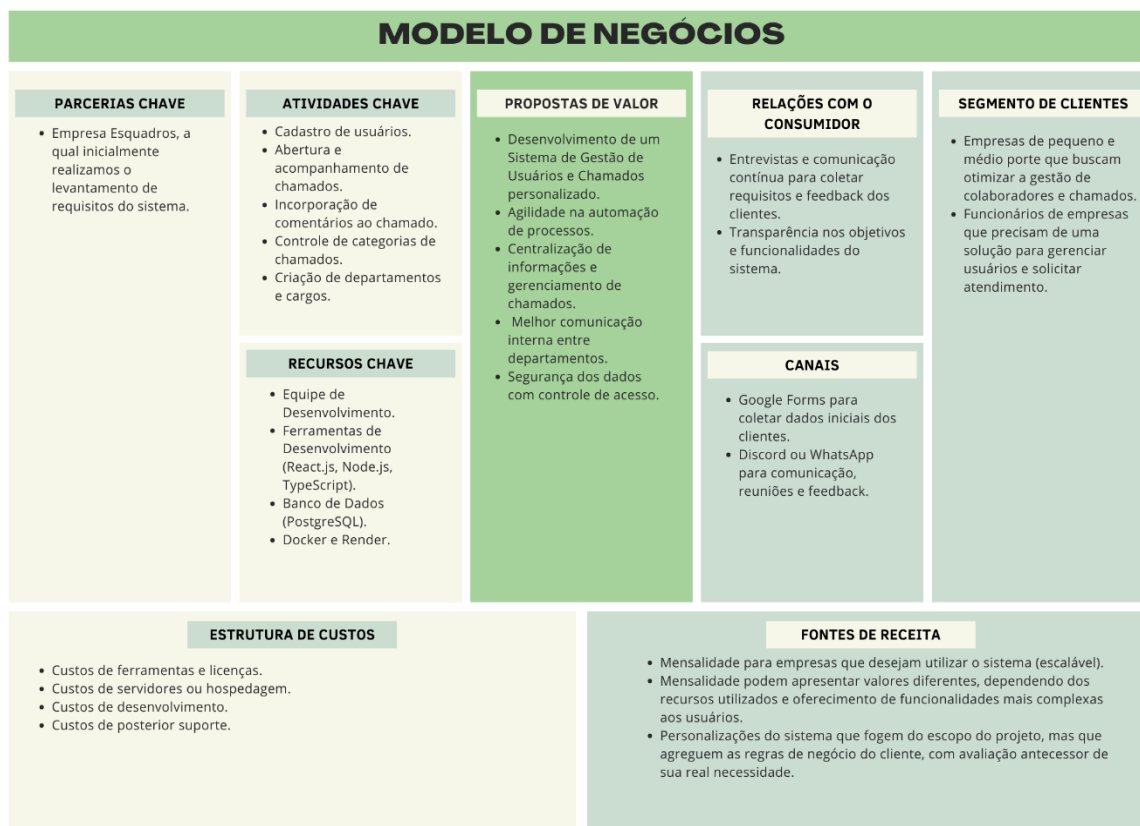
O principal parceiro-chave é a empresa Cliente Esquadros, que foi fundamental para entendimento dos fluxos de trabalho executados e como empresas com uma grande quantidade de departamentos e funções dos colaboradores se organizam para a resolução de problemas.

Também pode-se considerar que ela teve papel determinante para o levantamento de requisitos e confirmação se as diferentes funcionalidades estavam de acordo com as necessidades do *stakeholder*.

2.1.9 Estrutura de Custos

Os custos incluem despesas com ferramentas e licenças, custos de servidores ou hospedagem, desenvolvimento do *software*, suporte e manutenção contínua.

Figura 3: Modelo Canvas de Negócio



Fonte: os autores

3 Levantamento de Requisitos

O Levantamento de Requisitos é essencial para o desenvolvimento do *software*, sendo utilizado para ajudar na estruturação do *software* no momento do desenvolvimento e na compreensão do que é necessário para atingir as expectativas e necessidades do cliente (PIERRI, 2019, *online*). A seguir uma abordagem mais detalhada sobre a metodologia de Elicitação dos Requisitos para atingir o objetivo proposto.

3.1 Elicitação e Especificação dos Requisitos

As técnicas de Elicitação de Requisitos são utilizadas para coletar dados sobre as necessidades do usuário em relação ao sistema, contendo as características ideais

em seu desenvolvimento. Essa coleta pode ser realizada diretamente, através de comunicação com todos os envolvidos no projeto, utilizando questionários, entrevistas, grupos focais, entre outros (PLÍNIO, 2016, *online*).

Além disso, pode ser realizada indiretamente, através de manuais, relatórios, análise de documentos, entre outros. Independentemente da técnica utilizada, a finalidade é a mesma: obter o máximo de informações possíveis para a criação dos requisitos e o entendimento do que deve ser construído.

Para iniciar sua produção, realizou-se uma entrevista com o chefe do departamento de tecnologia, com o objetivo de demonstrar como o projeto seria executado, suas funcionalidades e objetivos.

Posteriormente, foram entrevistados os setores da empresa para coletar seus requisitos, restrições e melhor usabilidade para o contexto e nível de entendimento técnico dos usuários do sistema.

3.2 BPMN

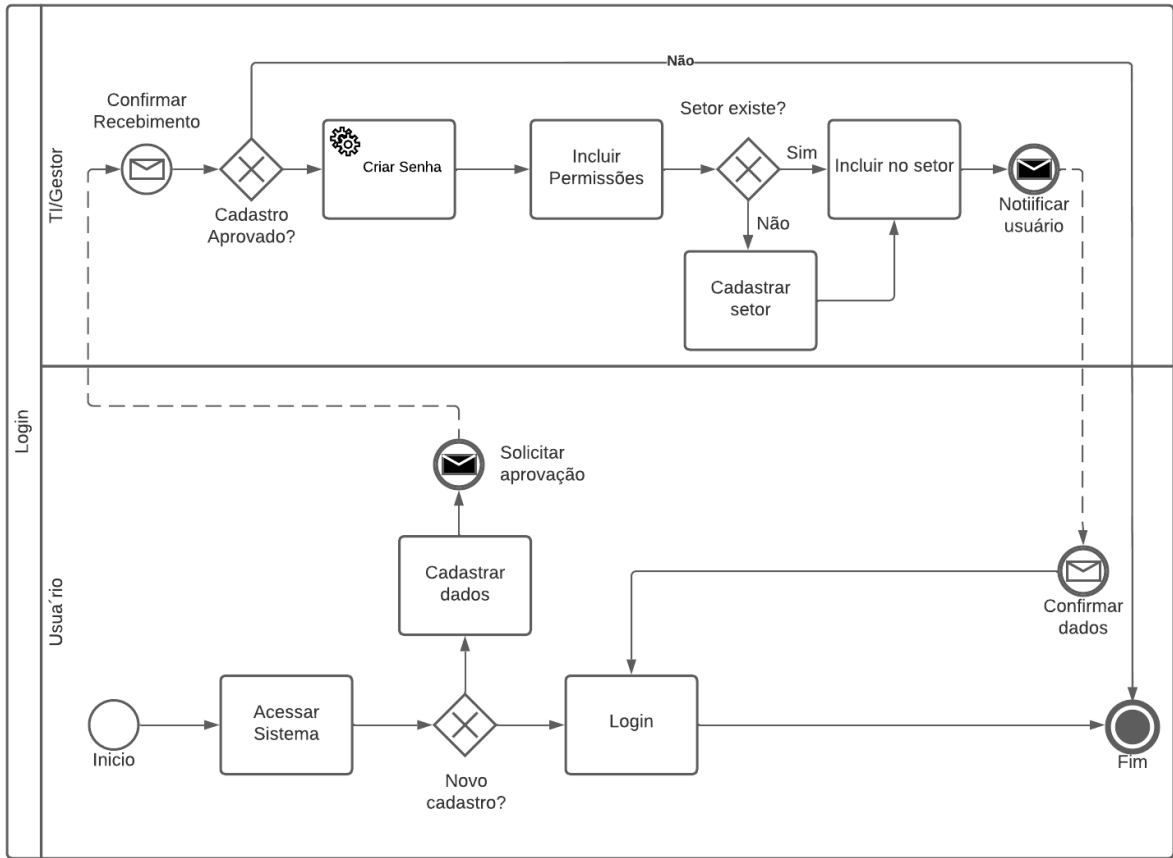
O *BPMN (Business Process Model and Notation)* é outra ferramenta importante para atender às expectativas dos *stakeholders* quanto ao sistema, pois se concentra no mapeamento dos processos de negócios.

Durante esta modelagem são mostradas as atividades e fluxos de informações de forma visual, assim como as pessoas responsáveis por sua implementação em linguagem acessível de áreas mais técnicas ou não.

Por meio da notação BPMN, se busca modelar as etapas de um processo de negócio planejado, representando uma sequência de atividades por meio de elementos (objetos de fluxo, objetos de conexão, raias e piscinas) e indicações de tarefas (LUCIDCHART, [s.d], *online*).

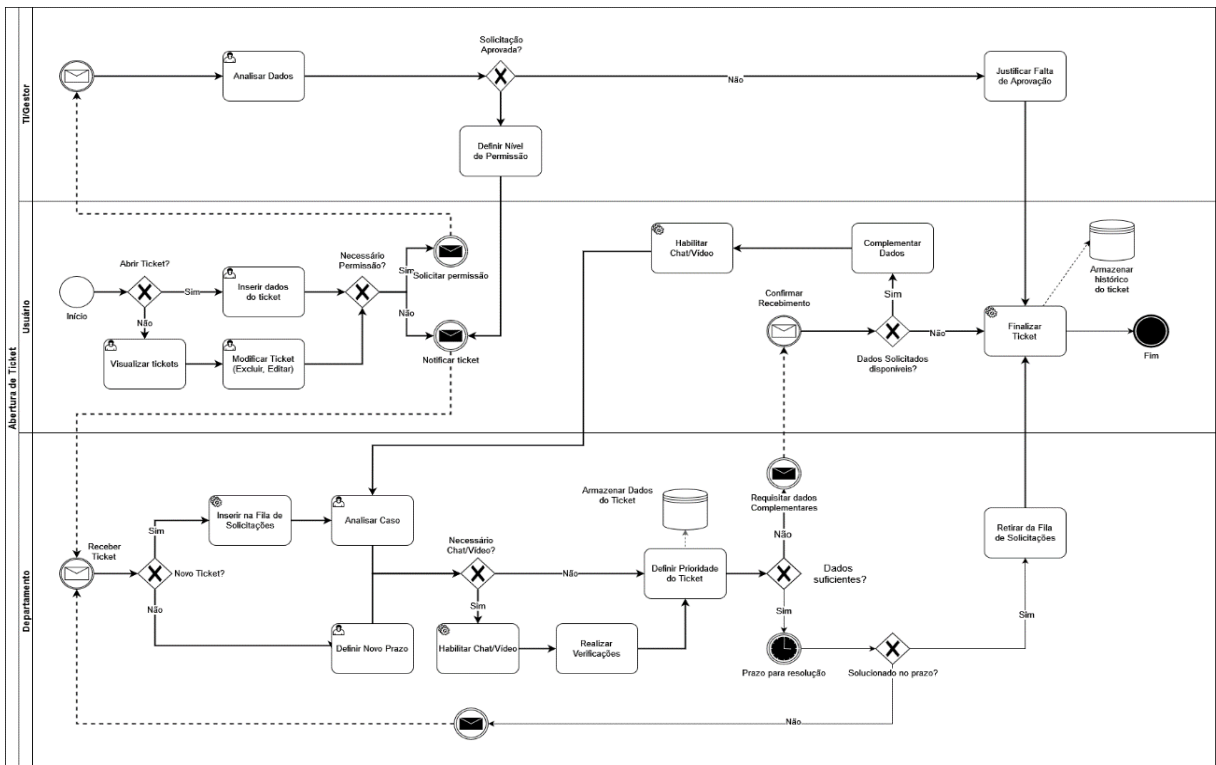
Assim, é possível proporcionar uma visão geral do funcionamento da aplicação desde seu início até o término, o que torna a visualização do fluxo do projeto mais claro para os *stakeholders*. Para demonstrar esta construção foram desenvolvidos dois BPMNs, um para a fluxo de *login* e o outro para a abertura de chamados que o sistema irá oferecer.

Figura 4: BPMN de Login



Fonte: os autores

Figura 5: BPMN de abertura do chamado



Fonte: os autores

3.3 Requisitos Funcionais

Requisitos Funcionais são uma parte do desenvolvimento do *software* voltados para decidir as funcionalidades do sistema, portanto incluí-los na documentação do projeto foi imprescindível para a validação correta da aplicação.

Eles garantem que o sistema contenha todas as funcionalidades esperadas pelo cliente, obtidas por meio de entrevistas, *workshops*, protótipo etc. Podem-se tornar uma base para a criação dos Requisitos Não Funcionais, que se diferem no foco de como o sistema deve operar (SIMÕES; VAZQUEZ, 2017, *online*).

Quadro 1 – Requisitos Funcionais do sistema

RF001- Cadastrar setor	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir o cadastro de departamentos para o vínculo dos usuários.		
RF002- Cadastrar novos usuários	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá permitir o cadastro de usuários, contendo seu nome, CPF, departamento, <i>e-mail</i> , número de celular, cargo e senha.		
RF003- Notificar solicitação de aprovação de cadastro	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá notificar uma solicitação de aprovação do cadastro do usuário ao Gestor ou TI.		
RF004- Gerar senha	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá gerar uma senha aleatória de 8 caracteres para cada usuário aprovado.		
RF005- Definir permissões do usuário	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá notificar uma solicitação de aprovação do cadastro do usuário ao Gestor ou TI.		
RF006- Abrir chamado	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá iniciar o processo de abertura de chamado quando o usuário solicitar.		

RF007- Notificar chamado ao departamento	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá notificar uma solicitação de aprovação do cadastro do usuário ao Gestor ou TI.		
RF008- Adicionar chamados à fila de solicitações	Categoria: <input checked="" type="checkbox"/> Oculto <input type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve adicionar os novos chamados a uma fila.		
RF009- Habilitar <i>chat</i> /vídeo	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve habilitar a opção de interação por <i>chat</i> e vídeo caso solicitado pelo usuário ou departamento.		
RF010- Negar criação de novo usuário	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O TI e o gerente do setor podem negar o pedido de criação de usuário, o excluindo.		
RF011- Transferir responsabilidade do chamado	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O responsável pode transferir o chamado para outra pessoa, setor ou subsetor.		
RF012- Consultar perfil dos usuários	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input type="checkbox"/> Altíssima <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O gerente do setor poderá ter acesso ao perfil de todas as pessoas que estiverem em seu departamento podendo ver todo seu histórico, e o setor de TI terá acesso geral a todos os perfis independentemente do setor.		
RF013- Consultar chamados	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deverá listar os chamados criados pelo usuário.		
RF014- Definir prioridade do chamado	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa
Descrição: O sistema deve tornar obrigatório o departamento escolhido, para definir uma prioridade para os chamados em aberto.		
RF015- Alterar status do chamado	Categoria: <input type="checkbox"/> Oculto <input checked="" type="checkbox"/> Evidente	Prioridade: <input checked="" type="checkbox"/> Altíssima <input type="checkbox"/> Alta <input type="checkbox"/> Média <input type="checkbox"/> Baixa

Descrição: O sistema deverá alterar o status do chamado de acordo com a escolha do usuário, ou caso ultrapasse o tempo limite de resolução.

Fonte: os autores

3.4 Requisitos Não Funcionais

Os Requisitos Não Funcionais são os que definem as restrições e regras no qual o sistema deverá obedecer e não estão diretamente ligadas a suas funcionalidades principais. São atributos que o sistema deverá possuir para que funcione da maneira mais coerente, o que resulta em um sistema que tenha qualidade, desempenho, segurança e outras funções relacionadas ao comportamento do sistema (ANTONIO, 2008, *online*).

Quadro 2 – Requisitos Não Funcionais do sistema

RNF001 Validar login no sistema	O sistema deve permitir o acesso do usuário ao sistema, realizando as restrições conforme parametrização do administrador	Segurança	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF002 Recuperar senha	Quando o usuário solicitar a troca ou a recuperação de senha será enviado um e-mail com um código, no qual o sistema estará liberando a troca.	Evidente	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF003 Definir design intuitivo	O sistema deverá oferecer um design simples e intuitivo.	Usabilidade	(X) Desejável () Obrigatório	() Permanente (X) Transitório
RN004 Validar nível de senha	A senha deverá conter letras maiúsculas, minúsculas e números com no mínimo 8 caracteres.	Segurança	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RN005 Executar backup do sistema	Necessário backup diário em nuvem para os usuários e chamados.	Segurança	() Desejável (X) Obrigatório	(X) Permanente () Transitório

Fonte: os autores

3.5 Regras do Negócio

São definidas como Regras do Negócio as orientações e restrições que ajudam a regularizar as operações da corporação, considerando que um negócio funciona por meio de processos, cada um com seus limites e permissões (RIBEIRO, 2020, *online*).

Este artefato permite definir quem, quando, onde, por que e como as operações devem ser feitas, garantindo que as políticas e objetivos da empresa sejam seguidos, descrevendo cada uma das funções individualmente, seguindo a legislação vigente, inteligível aos colaboradores envolvidos no procedimento de forma a oferecer

possibilidade de acompanhamento juntamente com atualização constante (SYDLE, 2023, *online*).

No desenvolvimento de *software* as Regras de Negócios determinam o fluxo na programação e suas funcionalidades, pois envolve a busca por soluções relacionadas aos requisitos anteriormente destacados. Dessa forma, para seu levantamento serão apresentados a seguir seu registro a partir da identificação e descrição de cada um.

Quadro 3 – Regras de Negócio do sistema.

RN001 – Não Repetir Cadastros
Descrição: O sistema não deverá permitir cadastro de um setor caso ele já exista.
RN002 – Cadastrar usuário somente com permissão
Descrição: O sistema deverá permitir o cadastro de um setor ou senha apenas se o usuário for Gestor ou TI.
RN003 – Cadastrar Usuário
Descrição: O sistema deverá permitir o cadastro de um novo usuário.
RN004 – Alterar Dados do Usuário
Descrição: O sistema deverá permitir a alteração dos dados de um usuário
RN005 – Tempo de Notificar Usuário
Descrição: A notificação só poderá sumir da tela depois que o usuário clicar em cima dela.
RN006 – Permissão de Acordo com Usuário
Descrição: O sistema deverá conceder permissões ao usuário conforme a definição do Gestor ou TI.
RN007 – Bloquear de Permissões
Descrição: O sistema não deverá permitir o próprio usuário definir as permissões.
RN008 – Definir Campos Obrigatórios no <i>Ticket</i>
Descrição: O sistema deve definir obrigatoriedade no preenchimento de todos os dados para abertura de chamado.
RN009 – Criar Chamados Únicos por Departamento
Descrição: O sistema deve abrir o chamados diretamente para apenas um departamento.
RN010 – Verificar Nível de Permissão
Descrição: O sistema deve verificar conforme solicitação de usuário se é necessário permissão.
RN011 – Notificar Permissão
Descrição: O sistema deve notificar o Gestor ou TI caso seja necessário permissão.
RN012 – Enviar Chamados Para Gestores
Descrição: O sistema deve enviar o chamado aos responsáveis do departamento.
RN013 – Prazo de Conclusão de Chamado
Descrição: O sistema deve monitorar o prazo para conclusão do chamado.
RN014 – Verificar Tempo de Conclusão de Chamado
Descrição: O sistema deve identificar se o chamado é novo ou está em atraso.
RN015 – Adicionar Chamados na Fila
Descrição: O sistema deverá adicionar os novos chamados em uma fila.
RN016 – Notificar Departamento
Descrição: O sistema deverá notificar o departamento sempre que um chamado entrar na fila.

RN017 – Habilitar Chat/Vídeo
Descrição: O sistema deverá habilitar a opção de <i>chat</i> /video caso o departamento solicite após a análise do <i>ticket</i> e deverá permitir a opção caso o usuário solicite quando o chamado não tiver sido solucionado.
RN018 – Tornar obrigatório Todos Campos de Cadastrar
Descrição: O sistema deverá permitir o cadastro de um novo usuário, alteração dos dados e tornar obrigatório o preenchimento de todos os campos do cadastro como CPF, cargo, departamento, email, nome, número de celular e senha.
RN019 – Alterar Responsável
Descrição: O sistema deverá permitir a alteração de responsável pelo chamado ou setor responsável e deverá ser enviado uma notificação para o seu criador caso seja transferido
RN020 – Registros de Acessos
Descrição: O sistema deverá registrar todas as modificações ou acessos do usuário acessos dentro do sistema.
RN021 – Listar Criador de chamado
Descrição: O sistema deverá listar ao usuário criador do chamado, uma lista ordenada por data, dos chamados criados.
RN022 – Priorizar Chamados
Descrição: O sistema deve solicitar ao departamento a inclusão da prioridade do chamado e notificar o usuário quando o departamento definir um prazo, com seu posterior monitoramento.
RN023 – Definir Status do Chamado
Descrição: O sistema deverá atribuir o status de concluído a um chamado apontado como solucionado, atribuir o status de pendente a um chamado que deve ser analisado para tratativa. Também deverá atribuir o status de em andamento a um chamado que ainda não foi solucionado e o sistema deverá atribuir o status de em atraso a um chamado que ultrapassou seu prazo anteriormente definido.
RN024 – Bloquear Acesso/Permissão Não autorizados
Descrição: O sistema deverá restringir acesso aos usuários conforme parametrização definida pelo Gestor ou TI e deverá bloquear o acesso caso não existir o usuário no banco de dados.
RN025 – Bloquear Modificação de layout
Descrição: O sistema não deverá permitir que o usuário faça modificações de layout.
RN026 – Alterar Senha
Descrição: O sistema deverá permitir o cadastro da nova senha, caso ela não cumpra as regras de senha com no mínimo 8 caracteres ou permissão de seu cargo e departamento, só poderá permitir a alteração através de uma solicitação aos administradores.
RN027 – Permitir Backup
Descrição: O sistema não deverá executar o backup caso o Gestor ou TI não permitam.
RN028 – Padronizar tempo de Troca de Senha
Descrição: Só poderá trocar a senha 5 vezes em um período de de 24 horas.

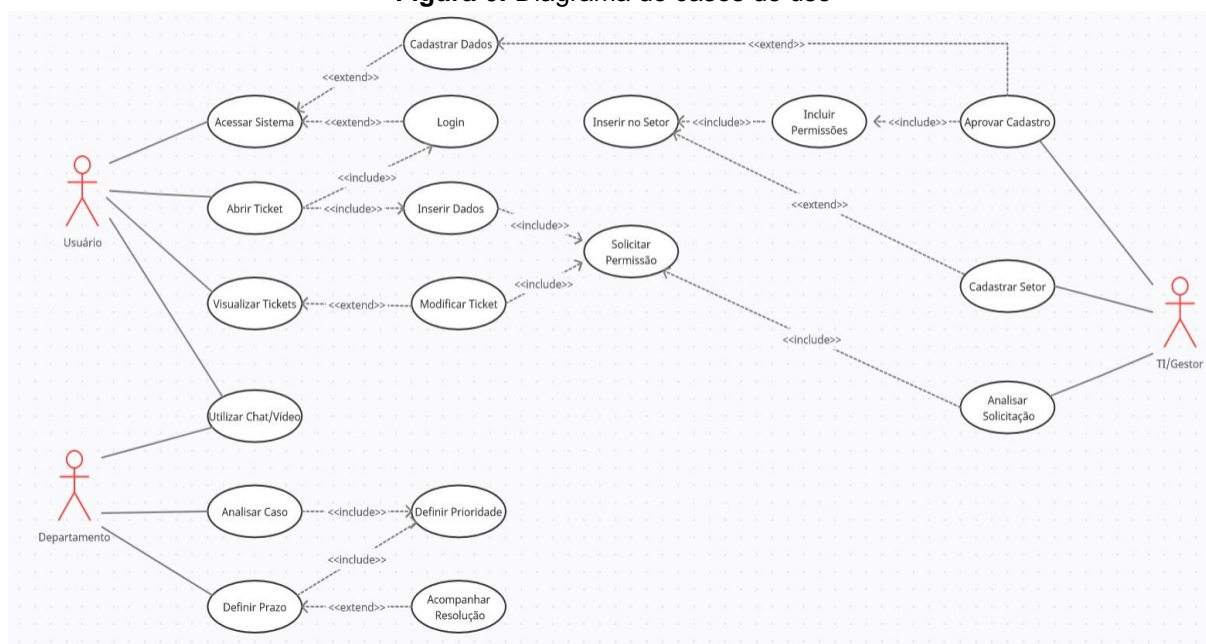
Fonte: os autores

3.6 Casos de Uso

Para se construir um Diagrama de Casos de Uso, inicialmente o analista de sistema irá identificar todos os atores, que no exemplo são o usuário, departamento e o ti/gestor. Neste diagrama é demonstrada a ação que cada um deles pode fazer com a aplicação, como os serviços que são exigidos e quais oferecem ao sistema, desde

o início no local de acesso ao sistema até o momento de criar e finalizar um chamado (LEANDRO, 2012, *online*).

Figura 6: Diagrama de casos de uso



Fonte: os autores

Este diagrama é uma ferramenta que permite visualizar a interação do usuário com o sistema e isso facilita a criação de Requisitos Funcionais, especificação de seu contexto e modelagem do fluxo básico de eventos do caso de uso. Foram usados recursos visuais como atores, representados pelos bonecos palito, que são os tipos de usuário já descritos, ações realizadas e seus relacionamentos.

Sua documentação também demonstra o cenário no qual vai ser usado, juntamente com as regras em que o sistema deverá seguir com outras opções por meio da escolha do usuário. São descritos os eventos necessários para efetuar as ações no cenário principal e alternativo, considerando pré-condições ou aquelas que forem posteriores a sua execução. Novamente são destacados os atores, parâmetros a que devem ser fornecidos, restrições e validações.

Quadro 4 – Documentação de Casos de Uso

Caso de Uso – Acessar Sistema	
ID	UC 001
Descrição	Este caso de uso tem o objetivo de possibilitar a tela de primeiro acesso ou login para um usuário já cadastrado.
Ator Primário	Usuário

Pré-condição	Ser funcionário da empresa.
Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o usuário acessa o link da página do sistema. 2. O sistema carrega a tela de login para preenchimento dos dados. 3. O sistema disponibiliza duas opções para realizar o login ou cadastrar um novo acesso. 4. O usuário seleciona a opção de cadastrar um novo usuário. 5. O usuário seleciona a opção de <i>login</i> 6. O sistema direciona para a próxima opção selecionada.
Pós-condição	Nenhuma
Cenário Alternativo	3a. O usuário opta por não realizar nenhuma das opções e sai do acesso.

Caso de Uso – Cadastrar Dados	
ID	UC 002
Descrição	Este caso de uso tem como o objetivo de criar um cadastro para o usuário.
Ator Primário	Usuário
Pré-condição	Ter realizado o passo 1 e 4 da UC 001
Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o usuário seleciona a opção de criar um cadastro. 2. O sistema solicita o preenchimento dos dados no formulário 3. O usuário preenche os dados no formulário. 4. O usuário confirma os dados. 5. O sistema recebe os dados e direciona para o usuário administrador.
Pós-condição	O usuário deve aguardar a aprovação dos dados e geração de senha pelo usuário administrador.
Cenário Alternativo	3a. O usuário opta por não realizar nenhuma das opções e sai do acesso.

Caso de Uso – Realizar Login	
ID	UC 003
Descrição	Este caso de uso tem como o objetivo realizar o login do cadastro do usuário existente no sistema.
Ator Primário	Usuário
Pré-condição	Ter um cadastro válido no sistema
Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o usuário seleciona a opção de realizar login na tela inicial. 2. O sistema irá carregar e exibir o formulário para preenchimento dos dados. 3. O usuário preenche os dados e confirma.

	<p>4. O sistema consulta na base de dados se o usuário está cadastrado.</p> <p>5. O sistema indica o sucesso na validação.</p> <p>6. O usuário acessa o sistema.</p>
Pós-condição	nenhuma
Cenário Alternativo	<p>1a. O usuário deseja sair da tela e selecionar outra opção.</p> <p>5a. O sistema não encontra o cadastro do usuário na base de dados.</p> <p>5ª. 1 O sistema exibe indicação de dados inválidos</p> <p>5a. 2 O sistema retorna para a tela de preenchimento dos dados.</p> <p>6a. O usuário não foi cadastrado e não acessa o sistema.</p>

Caso de Uso – Aprovar Cadastro	
ID	UC 004
Descrição	Este caso de uso tem como o objetivo aprovar o cadastro dos novos usuários pelo usuário administrador.
Ator Primário	TI/Gestor
Pré-condição	Ter solicitado a criação de um cadastro para o usuário.
Cenário Principal	<p>1. O use case se inicia quando gestor recebe a notificação para criar um cadastro.</p> <p>2. O sistema irá carregar os dados preenchidos pelo usuário na criação do cadastro.</p> <p>3. O sistema irá organizar os dados para visualização.</p> <p>4. O gestor irá realizar a última verificação dos dados e conferir se enquadram com a padronização da empresa.</p> <p>5. O gestor irá aprovar os dados e criação do cadastro.</p> <p>6. O sistema irá criar a senha para o usuário.</p> <p>7. O sistema direciona o gestor para a próxima página do cadastro.</p>
Pós-condição	nenhuma
Cenário Alternativo	<p>5a. O gestor não aprova o cadastro do usuário.</p> <p>5a. 1 O gestor descreve o motivo da não aprovação.</p> <p>6a. A senha não é criada pela desaprovação do usuário.</p> <p>6a. 1 O sistema notifica o usuário para informar a falta de sucesso na criação do cadastro.</p>

Caso de Uso – Incluir Permissão	
ID	UC 005
Descrição	Este caso de uso tem como o objetivo definir as permissões dos novos usuários pelo usuário administrador.

Ator Primário	TI/Gestor
Pré-condição	Ter dados de cadastro aprovados pelo gestor
Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o gestor acessar a página para selecionar as permissões a serem incluídas ao usuário. 2. O gestor seleciona as permissões de acordo como cargo do usuário. 3. gestor confirma a inclusão das permissões. 4. sistema atribui as permissões selecionadas. 5. O sistema direciona o gestor para a próxima etapa do cadastro.
Pós-condição	nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1a. O usuário deseja sair da tela e selecionar outra opção. 5a. O sistema não encontra o cadastro do usuário na base de dados. 5ª. 1 O sistema exibe indicação de dados inválidos 5a. 2 O sistema retorna para a tela de preenchimento dos dados. 6a. O usuário não foi cadastrado e não acessa o sistema.

Caso de Uso – Incluir Setor	
ID	UC 006
Descrição	Este caso de uso tem como o objetivo definir o departamento de atuação dos novos usuários pelo usuário administrador.
Ator Primário	TI/Gestor
Pré-condição	Ter dados de cadastro aprovados pelo gestor
Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o gestor acessar a página para incluir o departamento relacionado a área de atuação do usuário. 2. O gestor seleciona o departamento. 3. O gestor confirma a inclusão do departamento ao usuário. 4. O sistema atribui o departamento selecionado. 5. O sistema registra as atribuições. 6. O sistema notifica o usuário.
Pós-condição	nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1a. O usuário deseja sair da tela e selecionar outra opção. 5a. O sistema não encontra o cadastro do usuário na base de dados. 5ª. 1 O sistema exibe indicação de dados inválidos 5a. 2 O sistema retorna para a tela de preenchimento dos dados. 6a. O usuário não foi cadastrado e não acessa o sistema.

Caso de Uso – Abrir <i>Ticket</i>	
ID	UC 007
Descrição	Este caso de uso tem como o objetivo realizar a abertura do <i>ticket</i> .
Ator Primário	Usuário

Pré-condição	Ter acessado o sistema com usuário válido.
Cenário Principal	1. O use case se inicia quando o usuário seleciona a opção de abrir um <i>ticket</i> . 2. O sistema carrega o formulário de preenchimento do <i>ticket</i> .
Pós-condição	Preencher os dados solicitados no formulário do <i>ticket</i> .
Cenário Alternativo	1a. O usuário poderá retornar a tela inicial do sistema para selecionar outra opção.

Caso de Uso – Inserir Dados	
ID	UC 008
Descrição	Este caso de uso tem como o objetivo inserir os dados necessários para abertura do <i>ticket</i> .
Ator Primário	Usuário
Pré-condição	Ter selecionado a opção de abrir <i>ticket</i>
Cenário Principal	1. O use case se inicia quando o usuário acessa a tela do formulário para preenchimento dos dados. 2. O usuário preenche os dados obrigatórios solicitados. 3. O usuário confirma os dados. 4. O sistema confirma se todos os dados foram preenchidos corretamente. 5. O sistema registra os dados inseridos. 6. O sistema direciona o usuário para próxima tela de preenchimento.
Pós-condição	Solicitar permissão para prosseguir com solicitação.
Cenário Alternativo	1a. O usuário poderá retornar a tela inicial do sistema para selecionar outra opção. 4a. O sistema notifica que existem dados que não foram preenchidos ou estão inseridos de forma incorreta. 4a. 1 O sistema indica quais dados devem ser preenchidos. 4a. 2 O sistema indica quais dados estão preenchidos de forma incorreta. 4a. 3 O usuário faz o preenchimento dos dados novamente e confirma o envio.

Caso de Uso – Visualizar <i>Ticket</i>	
ID	UC 009
Descrição	Este caso de uso tem como o objetivo visualizar os <i>tickets</i> já registrados no sistema.
Ator Primário	Usuário
Pré-condição	1. Ter o <i>ticket</i> cadastrado no sistema. 2. Ter acesso ao sistema com usuário válido.

Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o usuário seleciona a opção de visualizar <i>tickets</i> na tela inicial do sistema. 2. O usuário acessa a tela onde estão todos os <i>tickets</i> cadastrados de acordo com sua permissão. 3. O usuário seleciona um <i>ticket</i>. 4. O sistema carrega os dados armazenados do <i>ticket</i>. 5. O sistema mostra na tela do <i>ticket</i> os dados, comentários internos e histórico de realização
Pós-condição	nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1a. O usuário poderá retornar a tela inicial do sistema. 2a. O usuário poderá retornar à tela inicial do sistema ou a tela de seleção dos <i>tickets</i>. 3a. 1 com a análise das permissões o sistema confere quais dados poderão ser mostrados e poderá não mostrar todos os preenchidos.

Caso de Uso – Modificar <i>Ticket</i>	
ID	UC 010
Descrição	Este caso de uso tem como o objetivo alterar os dados de <i>tickets</i> já registrados.
Ator Primário	Usuário
Pré-condição	Ter acessado o <i>ticket</i> a ser modificado. Ter acesso ao sistema com usuário e permissões válidas.
Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o usuário seleciona a opção de editar dentro da página do <i>ticket</i>. 2. O usuário seleciona o dado que deseja editar. 3. O sistema verifica se usuário tem permissão para editar o campo selecionado. 4. O sistema autoriza a edição. 5. O usuário altera o dado do campo selecionado. 6. O usuário confirma a alteração. 7. O sistema atualiza o campo editado. 8. O sistema registra o horário de alteração.
Pós-condição	Solicitar permissão para prosseguir com as alterações.
Cenário Alternativo	<ol style="list-style-type: none"> 4a. O sistema não autoriza a edição. 4a. 1 O sistema notifica o usuário que só tem permissão de leitura. 4a. 2 O usuário solicita permissão ao usuário administrador e avança para a próxima tela de solicitação.

Caso de Uso – Solicitar Permissão
--

ID	UC 011
Descrição	Este caso de uso tem como o objetivo possibilitar a permissão do usuário administrador para concluir o procedimento solicitado.
Ator Primário	Usuário
Pré-condição	Ter solicitado permissão para usuário administrador. Ter acesso ao sistema com usuário válido.
Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o usuário seleciona a opção de solicitar permissão ao usuário administrador ou realiza alguma ação que necessita de aprovação. 2. O usuário confirma a solicitação. 3. O sistema registrar a solicitação. 4. O sistema envia a solicitação para o usuário administrador.
Pós-condição	O usuário deve aguardar a autorização do usuário administrador.
Cenário Alternativo	<ol style="list-style-type: none"> 1a. O usuário poderá retornar a tela anterior caso estiver modificando um <i>ticket</i> existente. 1b. A solicitação pode partir do próprio usuário. 1b. O usuário deverá preencher o motivo de solicitara permissão. 1b. O usuário confirma a descrição.

Caso de Uso – Analisar Solicitação	
ID	UC 012
Descrição	Este caso de uso tem como o objetivo possibilitar a análise da solicitação pelo usuário administrador a fim de retornar sua confirmação final.
Ator Primário	TI/Gestor
Pré-condição	Ter uma solicitação pendente de análise.
Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o usuário administrador acessa a página de solicitações pendentes. 2. O gestor analisa a solicitação. 3 O gestor aprova a solicitação e permissão para o usuário. 4. O sistema registra a resposta. 5. O sistema retorna a resposta juntamente com as permissões fornecidas.
Pós-condição	nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1a. O usuário poderá retornar a página inicial do sistema. 3a. O gestor não aprova a solicitação. 3a. 1 O gestor especifica por escrito o motivo da solicitação ser negada. 3a. 2 O gestor confirma a descrição.

Caso de Uso – Cadastrar Setor	
ID	UC 013

Descrição	Este caso de uso tem como o objetivo criar departamentos no sistema para inserção de usuários e permissões.
Ator Primário	TI/Gestor
Pré-condição	Ter um usuário administrador válido para realizar o cadastro.
Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o usuário administrador seleciona a opção para cadastrar um novo setor. 2. O sistema carrega a tela de cadastro para preenchimento do formulário. 3. O gestor preenche os dados obrigatórios no formulário e realiza a confirmação. 4. O sistema confirma se todos os dados foram preenchidos corretamente. 5. O sistema direciona o usuário para a próxima telade permissões. 6. O gestor seleciona as permissões do departamento. 7. O gestor confirma as permissões selecionadas. 8. O sistema analisa os dados e conclui o registro. 9. O novo departamento é criado.
Pós-condição	nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1a. O gestor poderá sair da tela de cadastro do novo de partamento. 4a. O sistema informa que existem dados preenchidos incorretamente e não avança para a tela de permissões. <ol style="list-style-type: none"> 4a. 1 O gestor revisa e preenche novamente os dados. 4a. 2 O gestor confirma os novos dados. 4a. 3 O sistema confirma os dados. 7a. O sistema identifica que foram selecionadas permissões de outro departamento. <ol style="list-style-type: none"> 7a. 1 O sistema emite aviso caso o usuário administrativo deseje continuar com a seleção. 7a. 2 O usuário realiza a confirmação.

Caso de Uso – Utilizar *Chat/Vídeo*

ID	UC 014
Descrição	Este caso de uso tem como o objetivo possibilitar a utilização de <i>chat</i> ou chamada de vídeo para troca de informações e resolução de <i>tickets</i> .
Ator Primário	Usuário, Departamento
Pré-condição	<ol style="list-style-type: none"> 1. Ter acessado o sistema com usuário válido 2. Ter algum <i>ticket</i> em andamento. 3. Ter as opções habilitadas pelo sistema.

Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o usuário ou departamento seleciona a opção de usar o <i>chat</i> ou chamada de vídeo. 2. O sistema mostra para o <i>chat</i> a tela lateral dentro do procedimento em aberto para comunicação. 3. O sistema abre a tela de chamada de vídeo. 4. O usuário e departamento realizam a comunicação necessária com objetivo de complementar a solicitação. 5. Após finalizar a comunicação, o <i>chat</i> e chamada de vídeo são fechados. 6. O texto do <i>chat</i> é armazenado com o <i>ticket</i>. 7. O <i>chat</i> continua habilitado para consulta e comunicação.
Pós-condição	nenhuma
Cenário Alternativo	<ol style="list-style-type: none"> 1a. O usuário ou departamento pode minimizar a tela de <i>chat</i> ou chamada de vídeo a qualquer momento. 7a. O procedimento pode estar encerrado. 7a. 1 O usuário pode reabrir o <i>ticket</i>. 7a. 1.1 Caso o usuário não tenha permissão, a reabertura poderá ser solicitada. 7a. 2 O usuário poderá realizar comentários internos no <i>ticket</i>, mas não serão vinculados ao <i>chat</i>.

Caso de Uso – Analisar o Caso	
ID	UC 015
Descrição	Este caso de uso tem como o objetivo analisar a solicitação para prosseguir com a resolução.
Ator Primário	Departamento
Pré-condição	<ol style="list-style-type: none"> 1. Ter acessado o sistema com usuário válido. 2. Ter algum <i>ticket</i> em andamento.
Cenário Principal	<ol style="list-style-type: none"> 1. O use case se inicia quando o departamento acessa o <i>ticket</i> cadastrado para análise. 2. O sistema carrega os dados registrados no <i>ticket</i>. 3. O departamento analisa o <i>ticket</i>, verificando os meios que podem ser utilizados para resolução. 4. O departamento insere comentários no <i>ticket</i>. 5. O sistema registra o complemento. 6. O sistema avança para a próxima etapa de preenchimento.
Pós-condição	O departamento deve definir a prioridade do <i>ticket</i> .
Cenário Alternativo	1a. O usuário representante do departamento poderá sair da tela do <i>ticket</i> e retornar para a página de visualização geral.

Caso de Uso – Definir Prazo	
ID	UC 016
Descrição	Este caso de uso tem como o objetivo definir o prazo máximo de resolução do <i>ticket</i> .
Ator Primário	Departamento
Pré-condição	1. Ter acessado o sistema com usuário válido. 2. Ter algum <i>ticket</i> em andamento. 3. Ter definido a prioridade do <i>ticket</i> .
Cenário Principal	1. O use case se inicia quando o usuário representante do departamento define o nível de prioridade na tela anterior. 2. O sistema irá calcular o prazo com base na prioridade definida. 3. O sistema informará o prazo definido na página do <i>ticket</i> . 4. O sistema passará a cronometrar o prazo até a conclusão do <i>ticket</i> .
Pós-condição	nenhuma
Cenário Alternativo	1a. O usuário do departamento poderá retornar à opção definida na tela de escolha de prioridade. 2a. O <i>ticket</i> pode não ser novo e já ter um prazo definido. 2a. 1 O prazo pode ser alterado pelo usuário do departamento. 2a. 2 O novo prazo será atualizado no <i>ticket</i> após confirmação.

Caso de Uso – Definir Prioridade	
ID	UC 017
Descrição	Este caso de uso tem como o objetivo selecionar a prioridade de resolução do <i>ticket</i> , o que irá definir seu prazo máximo para novos procedimentos.
Ator Primário	Departamentos
Pré-condição	1. Ter acessado o sistema com usuário válido. 2. Ter algum <i>ticket</i> em andamento.
Cenário Principal	1. O use case se inicia quando o usuário representante do departamento acessa a página de definição da prioridade do <i>ticket</i> . 2. O usuário escolhe o nível de prioridade do <i>ticket</i> . 3. O usuário do departamento confirma a escolha. 4. O sistema registra a prioridade na página do <i>ticket</i> . 5. O sistema calcula o prazo de resolução. 6. O sistema finaliza a criação do <i>ticket</i> . 7. O sistema sinaliza o sucesso na inserção das especificações do <i>ticket</i> na tela. 8. O sistema retorna para a página inicial do <i>ticket</i> .
Pós-condição	nenhuma

Cenário Alternativo	<p>1a. O usuário do departamento pode retornar a qualquer momento para a opção anterior para inserção de dados e comentários no <i>ticket</i>.</p> <p>3a. O usuário pode ter realizado a escolha incorretada prioridade.</p> <p>3a. 1 O usuário do departamento pode redefinir a prioridade selecionando novamente a opção.</p> <p>3a. 2 O sistema irá atualizar a prioridade do <i>ticket</i>.</p>
----------------------------	---

Caso de Uso – Acompanhar resolução	
ID	UC 018
Descrição	Este caso de uso tem como o objetivo acompanhar a resolução do <i>ticket</i> bem como a evolução da tratativa.
Ator Primário	Departamentos
Pré-condição	<p>1. Ter acessado o sistema com usuário válido.</p> <p>2. Ter algum <i>ticket</i> em andamento.</p>
Cenário Principal	<p>1. O use case se inicia quando o usuário representante do departamento consulta um <i>ticket</i> em aberto ou recebe a notificação do sistema indicando alteração em seus dados ou prazo.</p> <p>2. O usuário do departamento acessa o <i>ticket</i>, abrindo sua página.</p> <p>3. O sistema carrega o <i>ticket</i> e suas atualizações.</p> <p>4. O sistema indica quais dados foram alterados, ou adicionados exibindo sua data.</p> <p>5. O sistema indica se houve alteração no prazo de resolução e prioridade.</p> <p>6. O sistema solicita revisão do usuário do departamento.</p> <p>7. O usuário confirma os dados após revisão.</p> <p>8. O sistema salva o horário da revisão e retorna o andamento da resolução.</p>
Pós-condição	nenhuma
Cenário Alternativo	<p>1a. O usuário do departamento pode voltar a tela geral dos <i>tickets</i> a qualquer momento.</p> <p>1a. 1 O sistema registra o horário de acesso ao <i>ticket</i>.</p> <p>7a. 1 O usuário do departamento não confirma a revisão dos dados.</p> <p>7a. 2 O sistema mantém o <i>ticket</i> em sua fila de prioridade específica com indicação a revisar.</p> <p>7b. O usuário do departamento insere novos dados e comentários no <i>ticket</i>.</p> <p>7b. 1 O usuário do departamento confirma a inserção.</p> <p>7b. 2 O sistema registra as alterações no <i>ticket</i> vinculado ao nome do usuário.</p> <p>7c. Ao revisar o usuário do departamento, notifica que existem informações faltantes, necessárias para a resolução do <i>ticket</i>.</p>

	7c. 1 O sistema notifica o usuário ou departamento que abriu o procedimento.
--	--

Fonte: os autores

3.7 Diagrama de Atividades

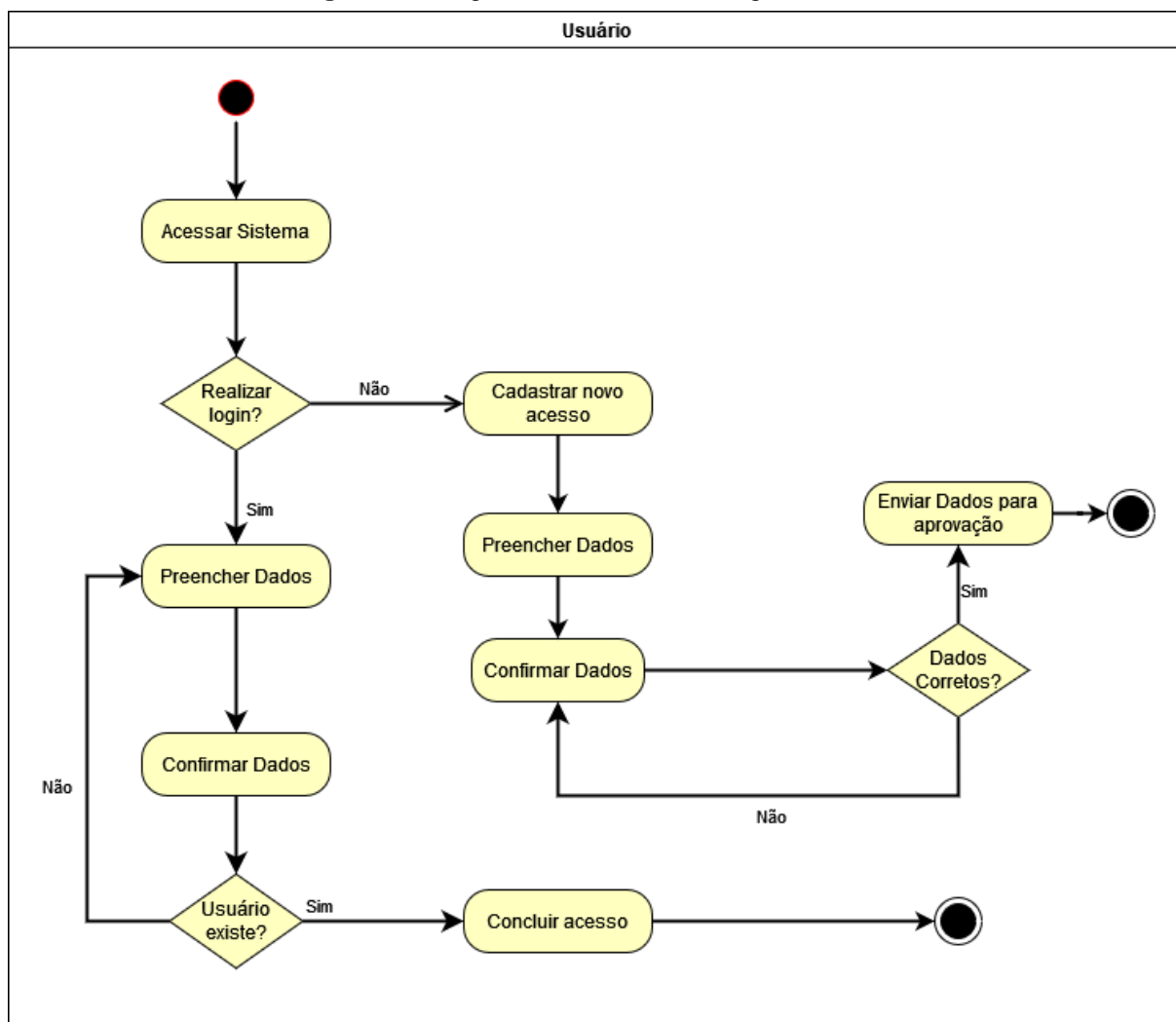
Este diagrama tem como objetivo especificar o fluxo das atividades mais complexas do sistema, especialmente a lógica dos casos de uso que exigem maior conhecimento sobre as interações entre atividades internas.

O Diagrama de Atividade demonstra a natureza dinâmica de um sistema pela modelagem do fluxo de controle de uma atividade para outra, representando uma operação em alguma classe no sistema e que resulta na mudança de estado. Por meio dele, é possível descrever a relação entre os casos de uso e seu esclarecimento, fluxo de trabalho entre usuário e sistema e demonstração da lógica do algoritmo.

3.7.1 Acesso de Login do usuário

Aqui é demonstrado a forma como o usuário poderá fazer acesso ao sistema e o que acontecerá caso não tiver *login*, preenchendo um formulário que deverá conter todo o acesso desejado junto com os dados de cadastro enviados à equipe de TI. Dessa forma, deve aguardar o retorno da aprovação e caso o cadastro não seja aprovado é enviado uma notificação para o usuário explicando o motivo.

Figura 07: Diagrama de Atividade – *Login* Usuário

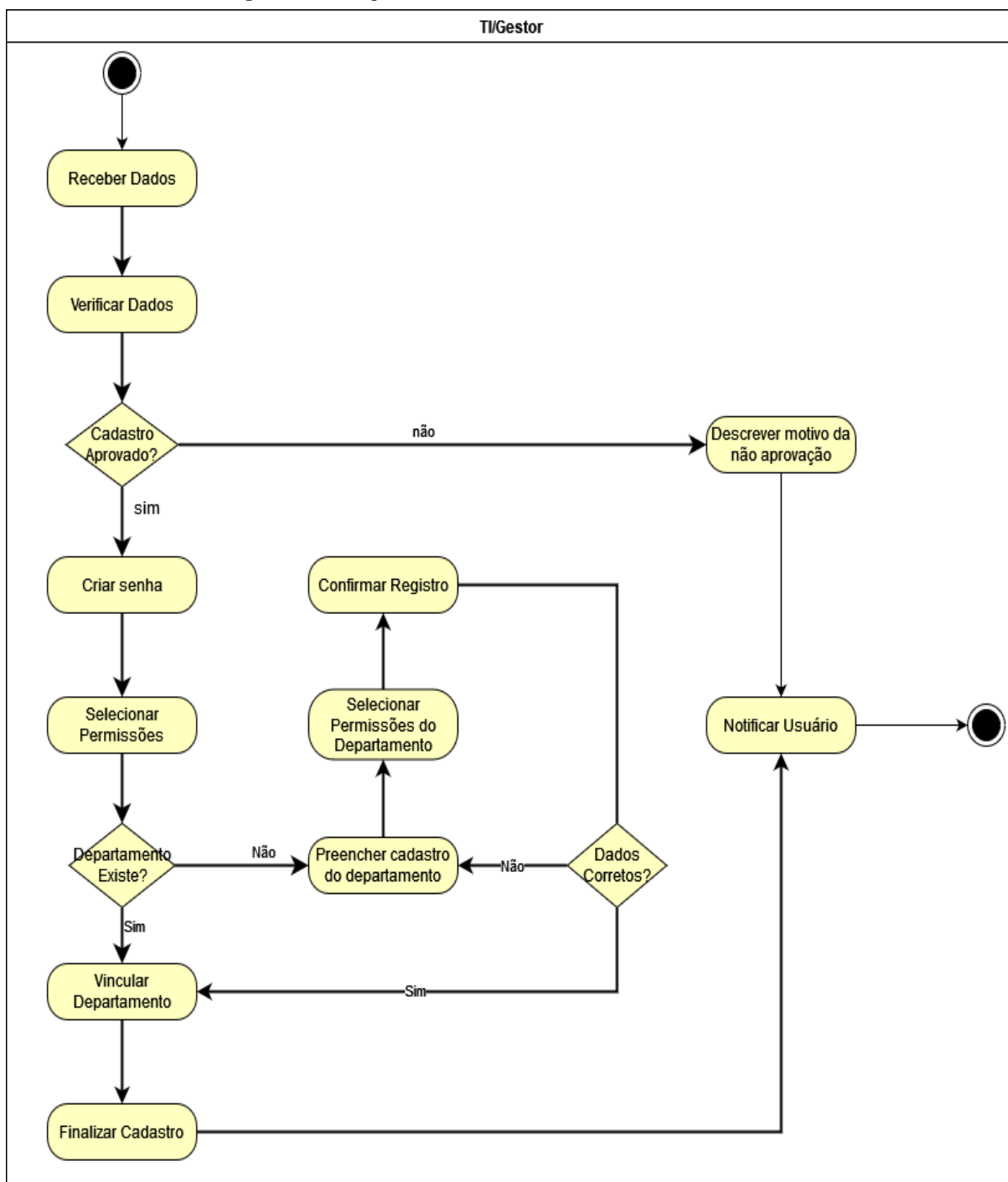


Fonte: os autores

3.7.2 Recebimento de formulário equipe de TI/Gestor

O formulário preenchido pelo usuário na tela de cadastro é analisado, onde no caso de aprovação será criado uma senha de primeiro acesso selecionando as devidas permissões relacionadas ao seu departamento. Se for um novo departamento será cadastrado pela equipe de TI, assim concluindo o cadastro e enviando uma notificação de sucesso ao usuário ou negação após devidas análises.

Figura 08: Diagrama de Atividade – Abertura de *Ticket* - TI/Gestor

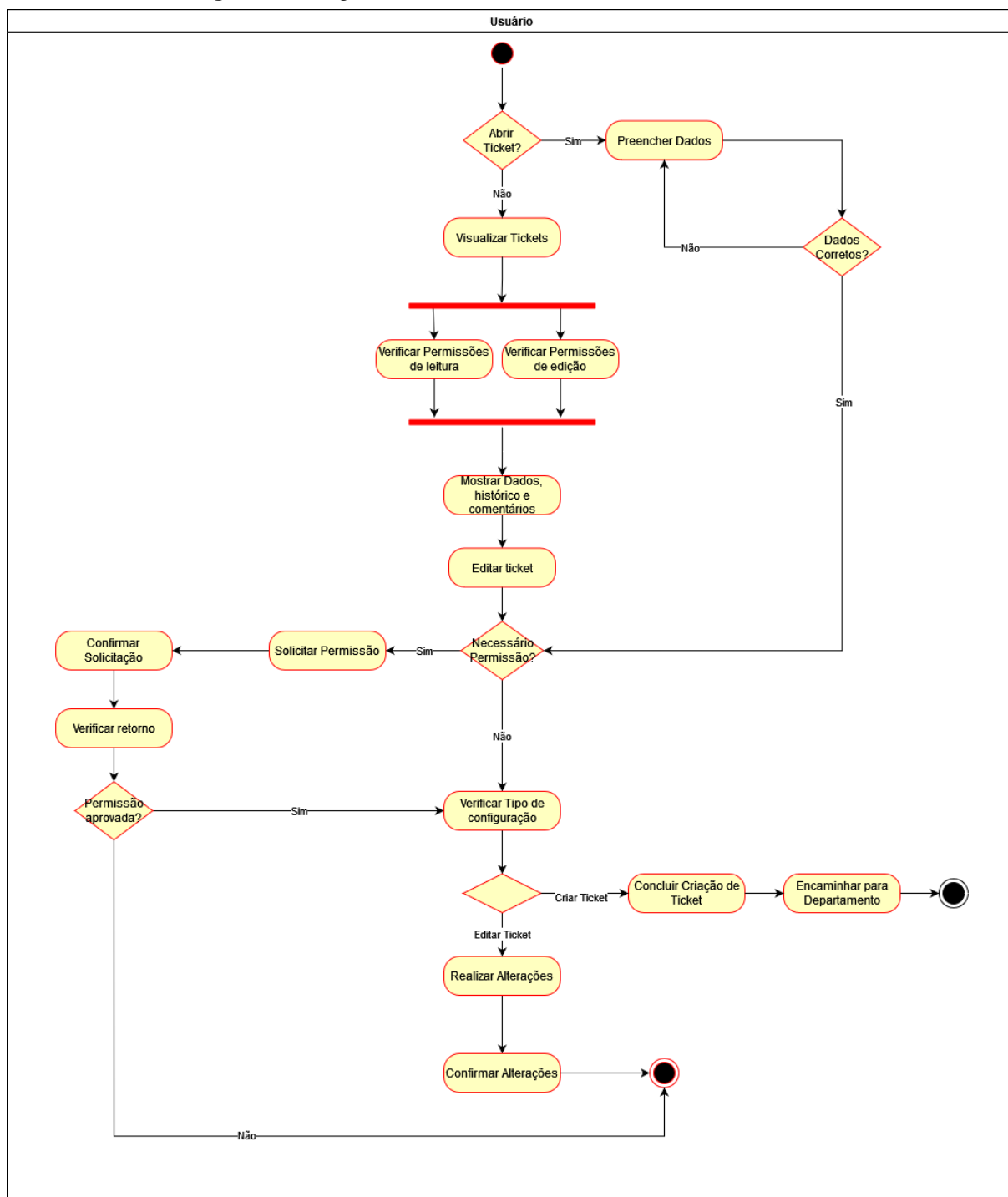


Fonte: os autores

3.7.3 Abertura de *ticket* – Usuário

Depois de logado no sistema, o usuário poderá criar e visualizar os *tickets* e se possuir acesso de gestor poderá definir o nível de permissão para o item ou criar um, em que poderá preencher um formulário com as informações tais como títulos, descrição, setor/usuário para qual será enviado.

Figura 09: Diagrama de Atividade - Abertura de *Ticket* – Usuário



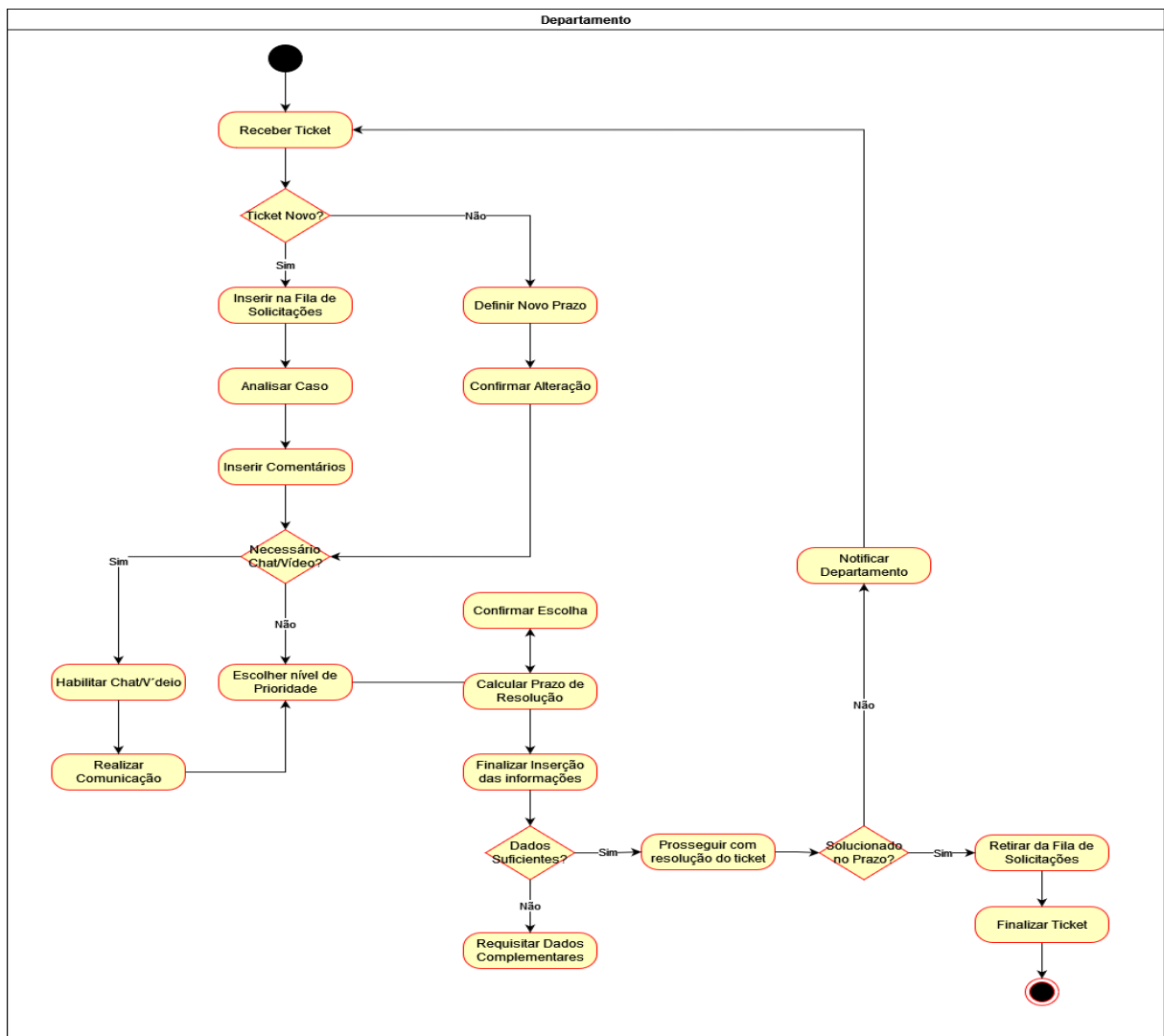
Fonte: os autores

3.7.4 Abertura de *ticket* – Departamento

Depois do envio ou notificação do *ticket* a um departamento, existe verificação se é recent. Para a segunda opção o sistema colocará um prazo para resolução do *ticket* e para um *ticket* novo poderá inserir comentários e mudar sua prioridade na fila, podendo habilitar a opção para *chat*/vídeo no final de cada procedimento com

necessidade de definir a prioridade e revisar se os dados enviados a ele serão suficientes para prosseguir com a resolução e posterior retirada da fila.

Figura 10: Diagrama de Atividade - Abertura de *Ticket* - Departamento



Fonte: os autores

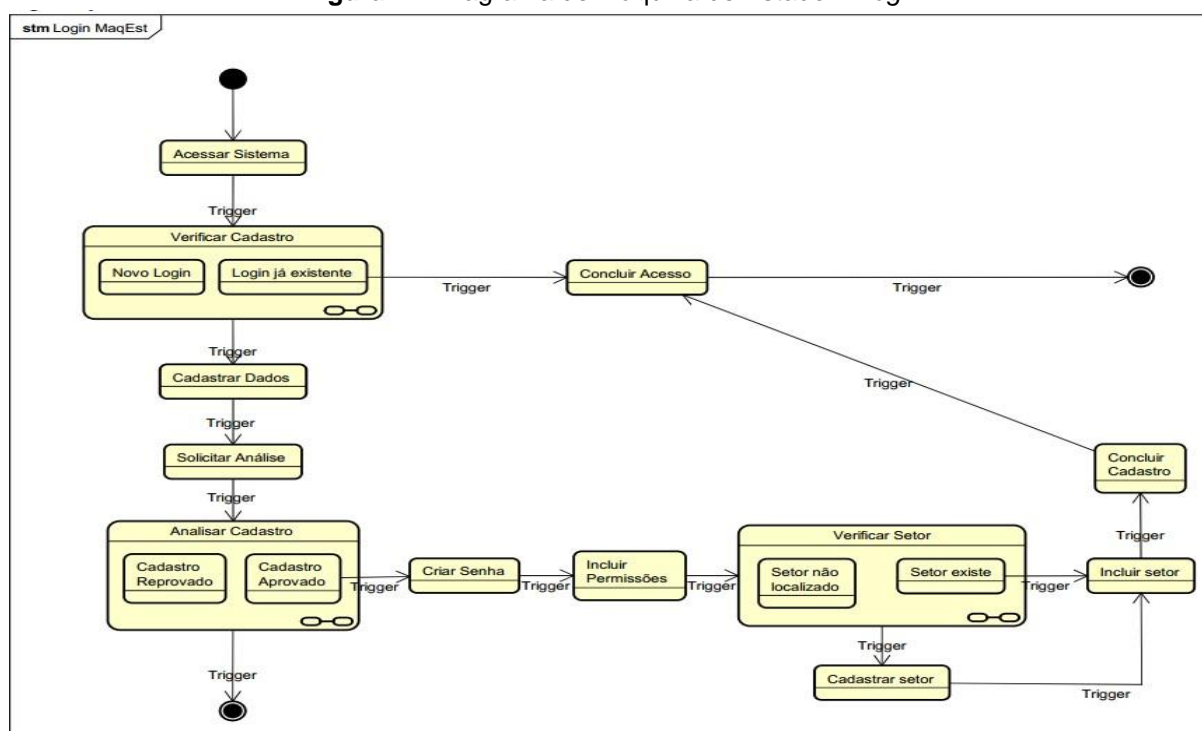
3.8 Diagrama de Estados

É um Diagrama Comportamental na Linguagem de Modelagem Unificada (UML) que mostra estados e transições entre vários objetos com o objetivo de ilustrar cenários de caso de uso no contexto de negócios, o percurso de um objeto em seu tempo de vida e descrição de objetos orientados a eventos por meio da modelagem de seu comportamento dinâmico.

3.8.1 Login

Ao acessar o sistema ocorrerá a verificação da existência do *login*. Para usuários que não possuam conta, o fluxo do sistema irá realizar o direcionamento para o cadastro e análise dos dados, posteriormente seguindo para a criação de senha, que deverá ser alterada no primeiro acesso. Se o usuário fizer parte de um novo setor será feito um cadastro e inclusão do registro. Dessa forma é disparada notificação com o resultado, aprovação ou reprovação.

Figura 11: Diagrama de Máquina de Estado – Login

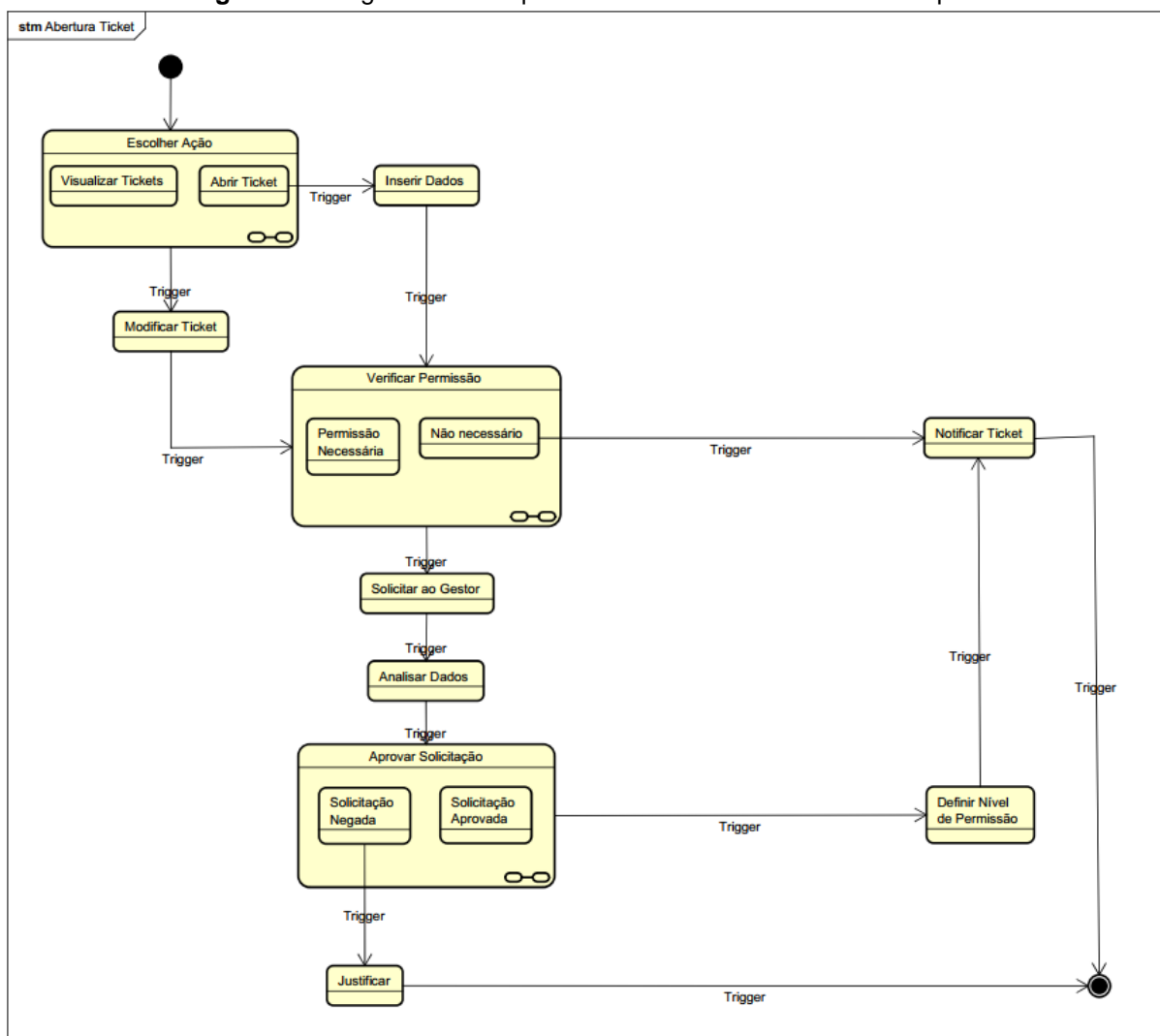


Fonte: os autores

3.8.2 Abertura de *ticket* – Usuário

Depois do acesso o usuário poderá criar ou visualizar um *ticket*, caso for necessário adicionar ou editar algum dado, deverá ter um nível de permissão correspondente à ação que deseja realizar. O usuário também pode solicitar a permissão ao administrador ou gestor que irá analisar a possibilidade de mudança, autorizando ou não, com retorno de justificativa final.

Figura 12: Diagrama de Máquina de Estado – Abertura de *ticket* pelo usuário



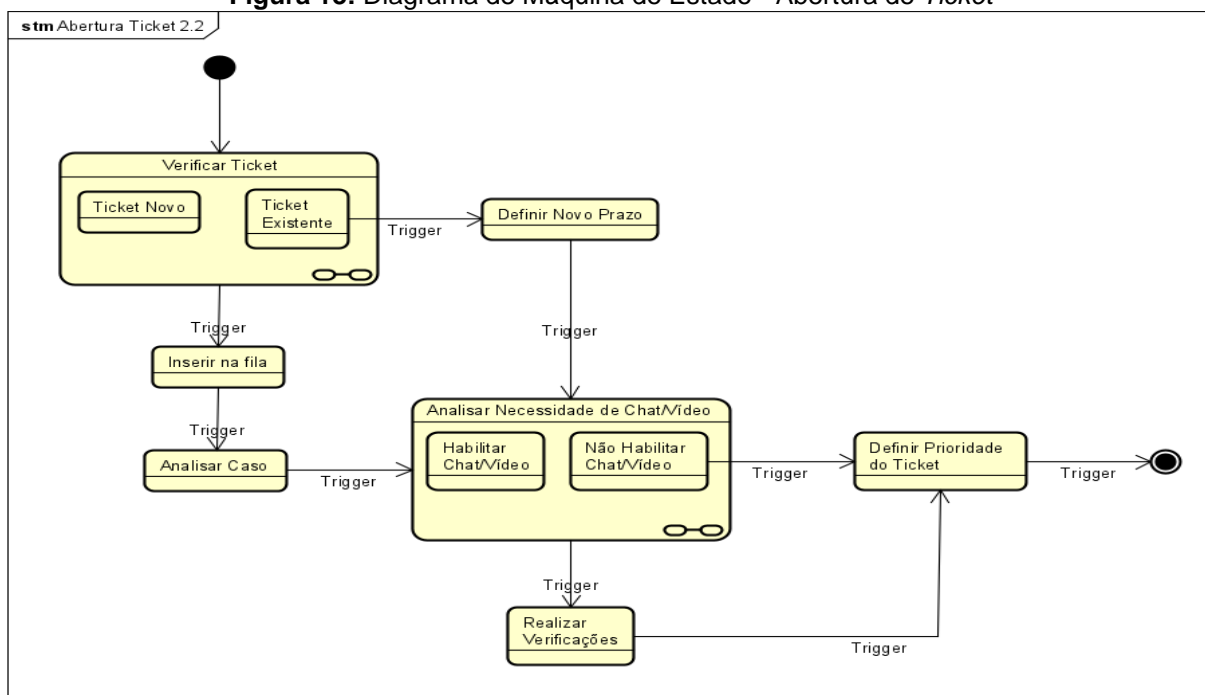
Fonte: os autores

3.8.3 Abertura de *ticket* – Departamento

Neste diagrama pode-se visualizar o fluxo de um *ticket* encaminhado a um departamento da empresa. Se o *ticket* for novo será tratado na fila de solicitações para análise e para aqueles que já foram abertos, mas tem necessidade de nova análise, é definido um novo prazo.

Em seguida será feita validação da necessidade do uso de *chat* ou vídeo para análise das questões relacionadas aos procedimentos por meio de uma comunicação mais direta ou apenas definir a prioridade do *ticket*, respectivamente.

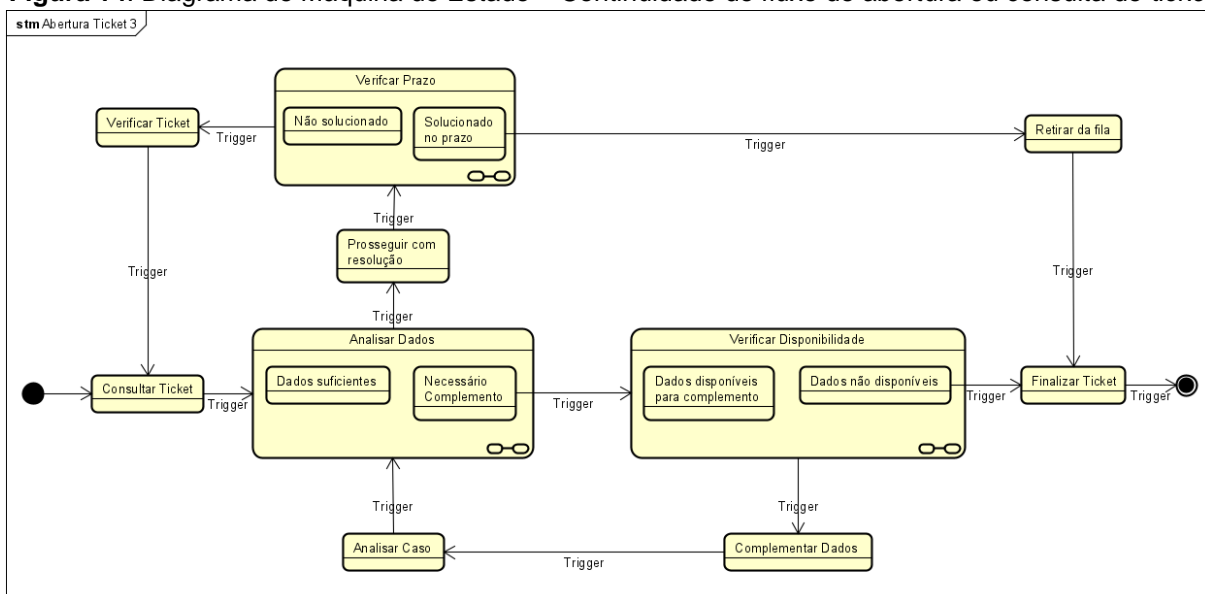
Figura 13: Diagrama de Máquina de Estado - Abertura de *Ticket*



Fonte: os autores

Em seguida o fluxo terá o direcionamento para análise se os dados preenchidos são suficientes e cumprem a obrigatoriedade necessária. No término dessa análise é decidido se o chamado precisa de complemento ou se pode prosseguir com resolução e consulta do tempo estipulado. Também existe um fluxo para retirada do *ticket* da fila se os dados necessários estiverem indisponíveis ou se ele foi solucionado dentro do prazo.

Figura 14: Diagrama de Máquina de Estado – Continuidade do fluxo de abertura ou consulta do *ticket*



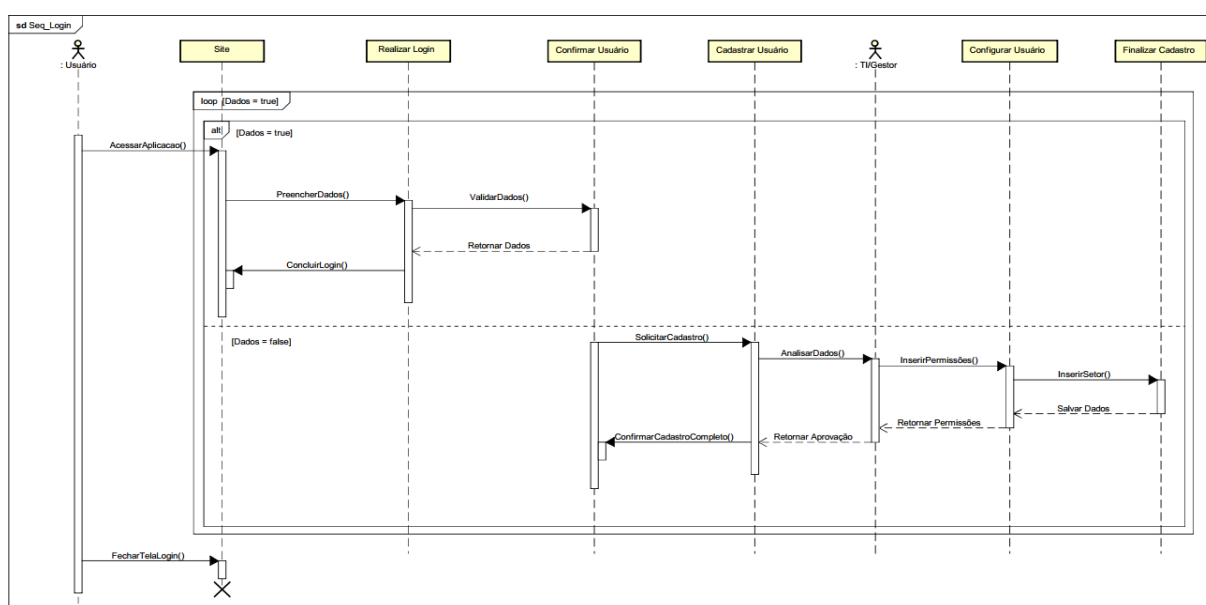
Fonte: os autores

3.9 Diagrama de Sequência

O Diagrama de Sequência modela as interações entre objeto, ilustrando como as diferentes partes do sistema interagem entre si para realizar uma função como as mensagens entre participantes e objetos, e a ordem em que as interações ocorrem quando um determinado caso de uso é executado ao longo do tempo.

No acesso do sistema o usuário deve efetuar o *login* preenchendo os dados para confirmação. Porém se for feito um novo cadastro é necessária uma averiguação por parte do gestor e somente após seu retorno possibilitar o acesso.

Figura 15: Diagrama de Sequência - Abertura de Login

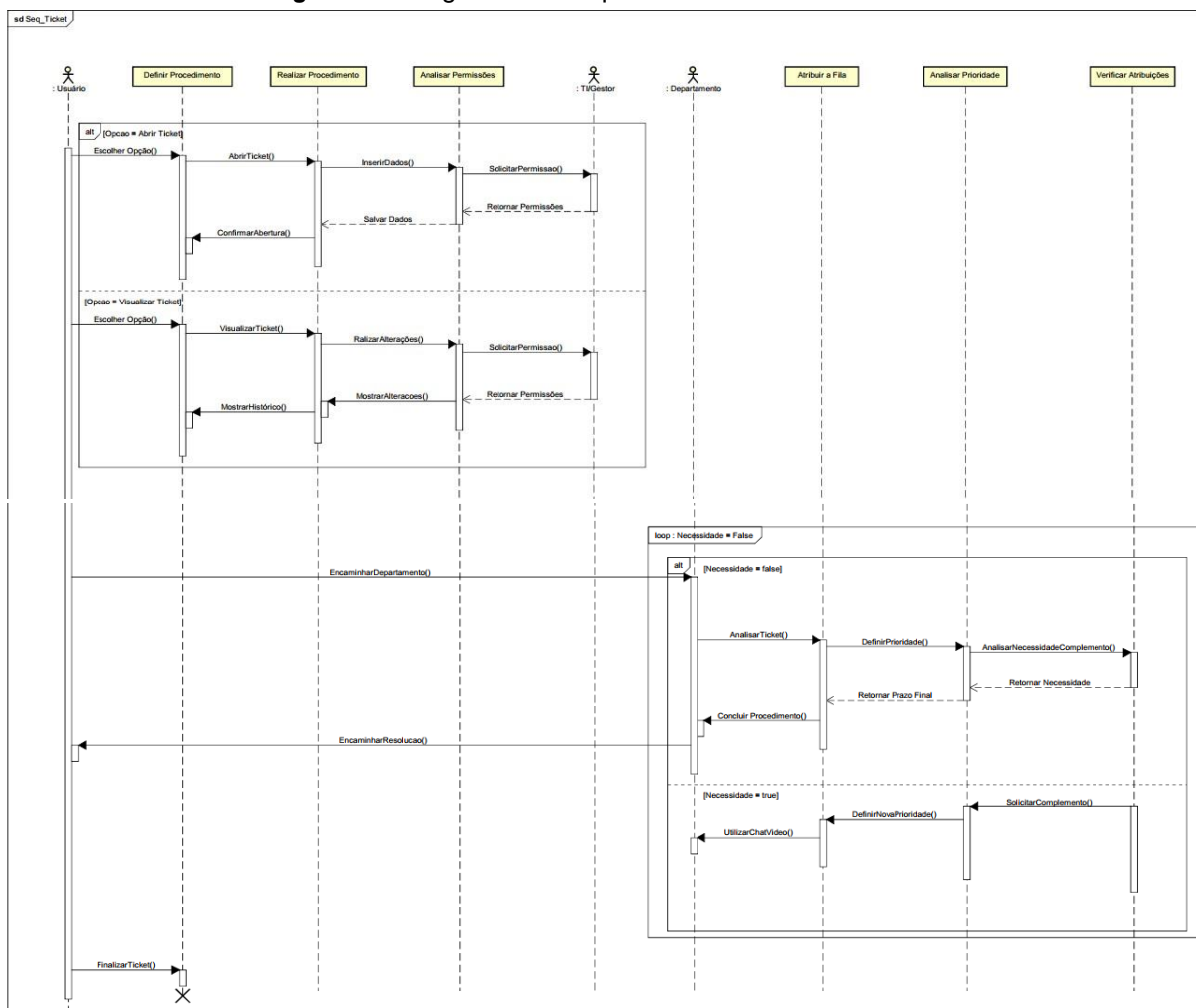


Fonte: os autores

De forma semelhante, a abertura de *ticket* também espera uma sequência de passos definida, incluindo cenários alternativos. Aqui, o usuário escolherá primeiro se deseja acessar um *ticket* existente ou criar um novo, em que a primeira opção carrega os dados vinculados ao chamado específico e a segunda abre uma nova página para preenchimento da solicitação.

Na abertura também são consideradas as permissões do usuário ou cargo, oferecendo após está confirmação ações como editar, visualizar ou excluir campos presentes. Na sequência das alterações o *ticket* passa por um acompanhamento do prazo para conclusão, bem como a ativação de ferramentas de *chat* ou vídeo conforme prioridade ou necessidade da equipe responsável pela solicitação. Este fluxo pode ser visualizado de forma completa a seguir.

Figura 16: Diagrama de Sequência – Abertura de *Ticket*



Fonte: os autores

3.11 Diagrama Entidade-Relacionamento

O Diagrama de Entidade e Relacionamento (DER ou ERD, do inglês *Entity Relationship Diagram*) é a forma usada para descrever as entidades do banco de dados de um sistema para se definir o relacionamento entre as tabelas.

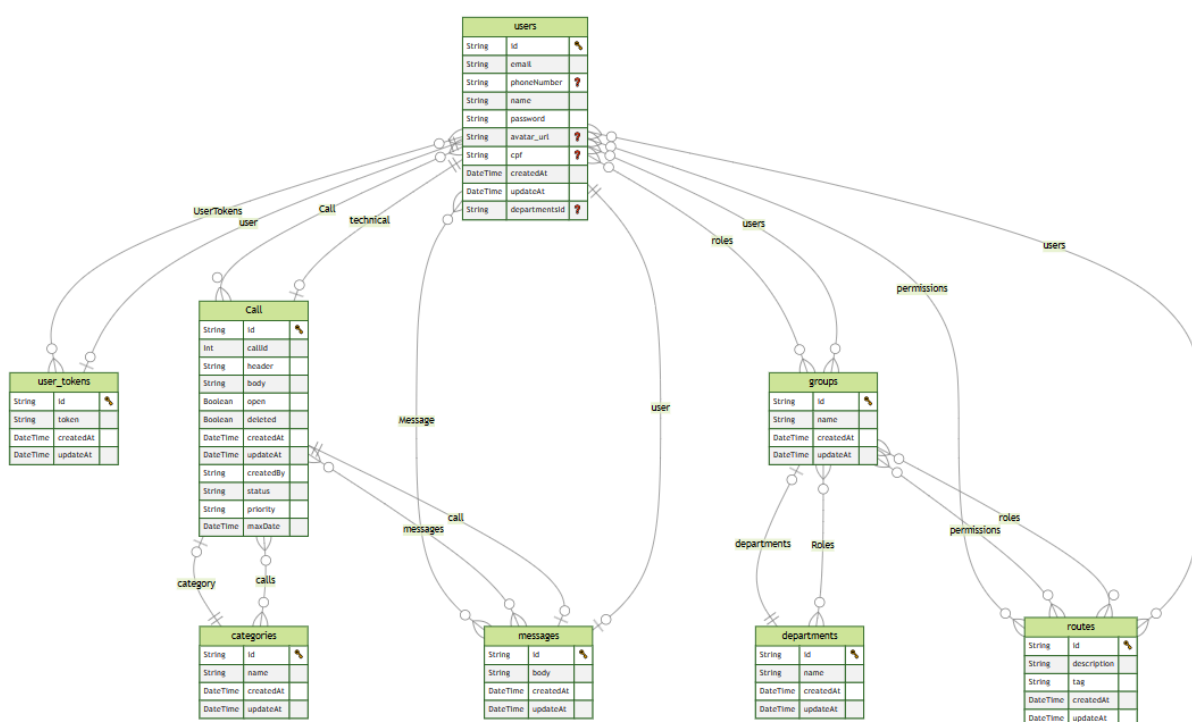
Existem três tipos principais para se relacionar uma tabela podendo ser elas um para um, um para muitos e muitos para muitos, em que esses relacionamentos são representados por linhas que ligam uma tabela a outra e através dos atributos da entidade, representam objetos e conceitos da realidade como usuário, clientes, produtos etc. Cada uma das entidades contém os atributos como por exemplo, um usuário que possui nome, idade, sexo, número de telefone, endereço, entre outros dados relevantes para sua identificação.

Os DER são frequentemente usados para o desenvolvimento de *software* e projetos de banco de dados para que se possa modelar da melhor forma possível e

permitir uma compreensão do projeto por todas as partes envolvidas, assim evitando as redundâncias e melhor visualizar do escopo do projeto. Assim, foi demonstrado o funcionamento do banco de dados e suas relações de cadastrado dos usuários, posteriormente gerando uma chave primaria ID fazer a identificação de *login* e na criação de um *ticket*, possibilitando que fiquem relacionados através desse código.

Este tipo de identificação também pode ser usado para criação de um *chat*, onde é criado um vínculo entre conversas de contatos diferentes, assim como os atributos da entidade de cargo e departamento também seguem o mesmo padrão.

Figura 17: Diagrama DER



Fonte: os autores

4 Ferramentas e Métodos ou Desenvolvimento

Durante o desenvolvimento da aplicação, várias ferramentas foram utilizadas para a prototipação de telas, criação de diagramas, gerenciamento da documentação, controle do banco de dados e no código, por meio de *frameworks* e ferramentas que otimizassem o tempo de produção juntamente com garantia de qualidade do *software*.

Por meio das descrições de recursos aplicados é possível compreender também a construção do sistema de forma técnica, reflexo dos levantamentos iniciais e que atingissem de forma satisfatória todos os requisitos propostos.

Quanto aos métodos, foi realizada uma abordagem descrevendo a evolução do projeto e como os diferentes artefatos da engenharia de *software* foram usados para

suprir as necessidades de desenvolvimento. Neste tópico, tanto ferramentas quanto metodologia estão relacionados no atendimento das necessidades do cliente e partes interessadas.

4.1 Ferramentas utilizadas para desenvolvimento:

No projeto utilizamos ferramentas para facilitar e agilizar etapas de desenvolvimento, desde a prototipação até a codificação do sistema, garantindo ao mesmo tempo qualidade dos requisitos propostos. A seguir será apresentado os recursos utilizados, com descrição de seus conceitos e enfatizando sua importância em relação a área aplicadas.

4.1.1 Visual Studio Code:

Versão: 1.77.3

Licença: Livre

Sobre: O visual Studio Code é um *software* leve e rápido que suporta vários tipos de linguagem de programação, podendo ser executado em diversos sistemas operacionais, pois possui extensões que ajudam no desenvolvimento junto com a integração do Git para o controle de versão do projeto.

4.1.2. Microsoft Office:

Versão: 365

Licença: Acesso via *e-mail* institucional

Sobre: O Microsoft Office conta com muitos recursos nos quais ajudam no desenvolvimento da documentação como o Word, que é um dos melhores editores de texto junto do Excel para desenvolvimento de planilhas.

No Word foi utilizado sua versão office 365, em que oferece correção ortográfica, pontuação de texto, dicas de palavras, formatação e maior flexibilidade no controle da documentação criada.

4.1.3 Git/GitHub:

Versão: 2.41.0

Licença: Livre

Sobre: O Git é um sistema para controle de versões do projeto, possibilitando acompanhar todas as alterações feitas no código. Ele permite que cada

desenvolvedor tenha uma versão local em seu computador e com isso podem desenvolver uma parte do sistema sem interferir nos demais versões, *branches*, depois podendo ser unidas em um repositório mestre para unificação do código.

O GitHub também permite a hospedagem de projetos que não utilizam muitos recursos, podendo ser usado em equipe para controlar fluxo do trabalho. Contudo somente foi feito o seu uso para controle do código em dois repositórios, um para *front-end* e outro para *back-end*.

4.1.4 Bpmn.io:

Versão: 2.0

Licença: *Web*, Livre. Baseado nas bibliotecas bpmn-js, dmn-js, form-js e cmmn-js, fornecidas pela Camunda Services GmbH, regido pela licença bpmn.io License.

Sobre: É um sistema web que permite criar digramas de processos usando o *BMPN*, possuindo uma interface interativa e de fácil uso, podendo também compartilhar os digramas em tempo real com outros usuários. O sistema oferece recursos avançados na criação dos diagramas, mesmo sendo um *software* gratuito.

4.1.5 Draw.io:

Licença: *Web*, Livre.

Sobre: O draw.io uma aplicação *web open source* que permite criar digramas, fluxogramas, mapa mentais entre outros tópicos, podendo ser feita de forma fácil por suas integrações com o GitHub, GitLab, Google Drive, DropBox, OneDrive e oferecendo a alternativa de salvar os projetos na nuvem. Conta com seu código fonte livre podendo ser modificado pelo usuário para melhorias, se assemelhando com o Bpmn.io.

4.1.6 Figma:

Versão: 127.2

Licença: Gratuita, mas possui planos pagos.

Sobre: O Figma é uma aplicação *web* para criação de interfaces gráficas interativas para aplicativos e sites, que em sua versão gratuita possibilita compartilhar em até três usuários diferentes, para modificação do protótipo em tempo real. É uma

aplicação muito utilizada para criações de interfaces para diversos dispositivos como celulares, computadores, *tablets* etc.

As telas desenvolvidas puderam ser facilmente modificadas quando necessário, tornando o Figma uma das ferramentas mais utilizadas durante o desenvolvimento do projeto.

4.1.7 LucidChart:

LucidChart Free: É a versão gratuita que permite criar diagramas básicos e colaborar com um número limitado de usuários.

Sobre: É uma ferramenta de criação de diagramas, fluxogramas e organogramas *online*, possuindo várias bibliotecas para criação de projetos com elementos gráficos. Uma das suas grandes vantagens é sua integração com o Google Drive, Microsoft Office e DropBox.

4.1.8 Astah:

Versão 8.4.1

Sobre: Uma ferramenta usada para criar modelos *UML (Unified Modeling Language)* com vários recursos avançados, como diagramas de classes, sequência, casos de uso, atividade estado etc.

Sendo uma aplicação para *desktop* tanto para *Windows* como para *MacOS* e *Linux*. Ela permite gerar códigos e integração com *softwares* e *IDEs (Integrated Development Environments)* a partir de modelos *UML*, acompanhada com o recurso de compartilhamento para vários usuários.

4.1.9 Node.js:

Versão: 18.16.0

Sobre: O Node é uma plataforma feita sobre o interpretador V8 do Google, permitindo executar JavaScript fora de um navegador. É uma ferramenta utilizada na criação de aplicativos de alta performance, ainda mais para aplicações que exigem comunicação em tempo real como *chat*, jogos, *streaming* etc.

Uma de suas grandes vantagens é o uso de módulos nativos HTTP, acesso a arquivos de sistema, comunicação com banco de dados, arquitetura assíncrona orientada a eventos e constante atualização com forte apoio da comunidade (NASCIMENTO, 2021, *online*).

Possuindo um gerenciador próprio chamado de *npm (Node Package Manager)*, permite ao desenvolvedor instalar, gerenciar e compartilhar pacotes de código. O node possui suporte para todos os sistemas operacionais e é compatível com diversas arquiteturas de *hardware* incluindo *x86, ARM e IBM power*. No projeto foi implementado para a criação de *APIs* com comunicação com o *front-end* desenvolvido em *React.js* e banco de dados *PostgreSQL*.

4.1.10 Node Express:

Versão: 4.18.2

Sobre: Node Express é um *framework* de desenvolvimento *web* feito para o *Node.js* para simplificar o processo de criação de *APIs* com recursos como o roteamento de *URLs*, gerenciamento de sessão, integração com banco de dados, *middleware* etc.

A criação de uma *API* com poucas linhas de código pode ser configurada pelo desenvolvedor da melhor forma, independente do projeto, sendo compatível com diversos outros módulos do *Node*.

4.1.11 Prisma:

Versão: 4.18.2

Sobre: O *prisma* é um *ORM (Object Relational Mapping)* de código aberto para ajudar na conexão com o banco de dados que simplifica o gerenciamento e o acesso dos dados, facilitando o desenvolvimento do *back-end* na construção de *queries*.

4.1.12 Nodemon:

Versão: 2.0.22

Sobre: O *Nodemon* é uma ferramenta utilizada por desenvolvedores *Node.js* que monitora o código para detectar mudanças, reiniciando automaticamente o servidor quando necessário. Também é possível ser configurado para ignorar determinados arquivos ou pastas, opção de executar comandos personalizados entre várias outras.

4.1.13 Dntenv:

Versão: 16.0.3

Sobre: Sendo uma biblioteca do *Node.js* o *Dotenv* ajuda a gerenciar as variáveis de ambiente dentro da aplicação, que são usadas para armazenar as informações confidenciais ou de configuração, como senhas de banco de dados, chaves de *APIs* e demais dados sensíveis, tudo que não poderá ser mostrado no código. Ele possibilita criar um arquivo *.env* na raiz do projeto onde será definido as variáveis de ambiente, inclusive criando valores padrões.

4.1.14 ESLint:

Versão: 8.38.0

Sobre: O *ESLint* é uma ferramenta de *linting* para JavaScript que ajuda a corrigir os erros de sintaxe do código, estilos e práticas incorretas de programação. Após análise, dispara mensagens de erros relatando problema, conforme as regras e personalização definidas. Pode ser integrado em ferramentas de edição de códigos como *Visual Studio Code*, *Sublime Text* etc.

4.1.15 Insomnia:

Versão: 2023.2.0

Sobre: O *Insomnia* é um aplicativo *desktop* que permite o teste de *APIs* de maneira rápida, eficaz e organizada. Por meio de uma interface com fácil entendimento o usuário pode criar, visualizar e enviar solicitações *HTTP* e *HTTPS* para teste das *APIs*, salvar as solicitações em vários formatos como json, xhtml, html entre vários outros. Ele pode gerenciar diversas solicitações em um único projeto possuindo suporte com outras ferramentas como *Git* e *Swagger* e é compatível em diversos sistemas operacionais como *Windows*, *MacOS* e *Linux*.

4.1.16 React.js:

Versão: 18.2.0

Sobre: O *React.js* é uma biblioteca feita em *JavaScript*, criado pelo *Facebook* e código aberto, possuindo uma grande comunidade ativa para melhorar a tecnologia. Ele é utilizado para criar aplicações *web* com atualização em tempo real por meio de componentes, sendo partes de código reutilizáveis e compartilhadas em diferentes locais do sistema.

Esta foi umas das principais ferramentas utilizadas ao longo do desenvolvimento do projeto, tornando o *front-end* rápido, mais flexível a alterações e com um

código organizado pelas diferentes funcionalidades separadamente. Com o *React.js*, também foi possível a utilização de diversas bibliotecas para otimização do tempo produção do Material UI, que permite o uso de componentes prontos, mas ao mesmo tempo permitindo modificações de maneira simples.

O exemplo de uso da biblioteca e ao *framework* nos proporcionou maior velocidade na conclusão da interface, importante para que também dedicássemos tempo no *back-end* e manipulação dos dados cadastrados e principais funcionalidades.

4.1.17 Styled-Components:

Versão: 5.3.10

Sobre: É uma biblioteca para estilizar páginas da *web* feitas em *React.js*, permitindo que se escreva código CSS no JavaScript. Uma de suas grandes vantagens é poder utilizá-la sem se preocupar com interferência de nomes de classes ou gerenciamento de estilos em arquivos CSS separados, pois permite criar estilos dinâmicos baseados no estado ou propriedade dos componentes, bem como a forma mais comum que seria pelo nome da *tag* em HTML.

Sua capacidade de compilar o estilo no código faz com que tenha um melhor desempenho, considerando que os componentes são renderizados juntamente com a página, já que apenas é carregado o necessário CSS.

4.1.18 ReactDOM:

Versão: 18.2.0

Sobre: É uma biblioteca do *React.js* que permite criar interfaces de usuário, fazer sua renderização no navegador e manipular o *DOM (Document Object Model)*. Dessa forma, pode-se considerar o ReactDOM como uma ponte entre o *React.js* e o navegador considerando que apenas são renderizados nas alterações necessárias na página para manter a interface da aplicação atualizada ao seu estado.

4.1.19 TypeScript:

Versão: 5.0.2

Sobre: TypeScript é uma linguagem de programação de código aberto desenvolvida pela Microsoft como um superconjunto do JavaScript, incluindo a adição de tipagem estática, interfaces, definição de tipos de variáveis, parâmetros de função de

objetos, herança e outros recursos avançados, o que auxilia no tratamento e prevenção de erros durante o desenvolvimento de *software*.

Dessa forma, a linguagem permite *feedbacks* mais precisos, facilidade na faturação do código, autocomplete e a possibilidade de utilização gradual. No caso deste projeto, a linguagem foi utilizada desde o início do desenvolvimento, não sendo necessário migração de JavaScript para TypeScript e no uso do React.js e Node.js.

4.1.20 Material UI

Sobre: O Material UI também é uma biblioteca de componentes para React.js com código aberto e implementação do design do Google. Além de oferecer rapidez em sua utilização, seus componentes permitem que o desenvolvimento do sistema seja otimizado, já que sua implementação é simples. Juntamente com a biblioteca também foram usados os seguintes pacotes *mui/joy* na versão 5.0.0-alpha.78, uma dependência relacionada ao projeto Material-UI, que fornece componentes adicionais para criar interfaces de usuário mais ricas e interativas, *mui/material* na versão 5.12.3 que contém a implementação dos componentes básicos para criar interfaces de usuário com o design do *Material Design* e *mui/x-data-grid* na versão 6.3.1, mais uma dependência do *Material-UI* que fornece uma grade de dados (*data grid*) avançada para exibir e manipular grandes quantidades de dados em uma interface de usuário.

4.1.21 Vite:

Versão: 4.3.2

Sobre: Um empacotador de módulos para o desenvolvimento de aplicações *front-end*. Ele oferece um ambiente de desenvolvimento rápido, criando uma divisão do código fonte e dependências, e é eficiente para projetos baseados em React.js, utilizando ECMAScript nativamente, *build* sob demanda com tempo de carregamento mais rápido e substituição de módulos sem atualizar a página.

4.1.22 Axios

Versão: 1.4.0

Sobre: Uma biblioteca de cliente HTTP baseado em *promises* para Node.js que permite fazer solicitações para servidores remotos e tratar respostas. É comumente usado para interagir com APIs em projetos React.js, transformando dados para JSON automaticamente com suporte contra XSRF no lado do cliente.

4.1.23 React-quilljs

Versão: 1.3.3

Sobre: Uma biblioteca *React* para integrar o editor de texto *Quill* em um aplicativo *React*. O *Quill* é um editor de texto rico e personalizável que permite a edição de conteúdo formatado e possibilitando criar o local de preenchimento das mensagens presentes no chamado. Dessa forma, o usuário pode inserir sua solicitação e a incorporação de imagens e vídeos de forma fácil.

4.1.24 Docker

Versão: 4.19.0

Sobre: É uma tecnologia de containerização, se assemelhando a máquinas virtuais modulares em ambiente isolados. Conhecidos como *containers*, cada unidade inclui o código, bibliotecas e dependências necessárias para executar o aplicativo, oferecendo maior flexibilidade para a sua manipulação em diferentes cenários.

Em nosso projeto, o *Docker* facilitou a construção do *back-end* e *front-end* ao mesmo tempo, pois não era necessário instalar o banco de dados na máquina de todos os computadores, somente do responsável pela construção do *back-end*, por exemplo. Todo o código também foi armazenado no container, o que nos deu mais segurança para incluir as modificações.

4.1.25 PostgreSQL

Versão: 15.4

Sobre: É um gerenciador de bancos de dados relacional com implantação da linguagem SQL na criação da estrutura. Com flexibilidade para operar com diferentes linguagens, padrões de conformidade, otimização, segurança por meio de controle de acesso e autenticação sólida, robustez com grande fluxo de dados e garantia de sua integridade, o PostgreSQL garantiu que as expectativas com o sistema desenvolvido fossem alcançadas. A escolha também levou em consideração a experiência e conhecimento da ferramenta em comparação com as demais, como MySQL ou *MongoDB*.

4.1.26 Discord

Sobre: O Discord é uma plataforma de comunicação em grupos por *chat*, vídeo e voz. Utiliza-se essa ferramenta em de reuniões periódicas, compartilhamento de arquivos, alinhamento do que era registrado na documentação e se o código do sistema seguia as boas práticas de organização e objetividade, sem redundância nas diferentes funcionalidades criadas.

4.1.27 Sonner

Versão: 0.7.0

Sobre: Sonner é um componente do React feito especificamente para a criação de animações de mensagens. Ele oferece várias opções de estilos predefinidos, tornando-o uma ferramenta ideal para adicionar um toque de dinamismo e interatividade às mensagens exibidas em seu aplicativo. Além disso, o Sonner é totalmente personalizável, permitindo adaptar seu estilo de acordo com as necessidades específicas do projeto com essa flexibilidade pode criar experiências de usuário únicas e envolventes que se destacam e cativam seu público.

4.1.28 Render

Sobre: Render é uma plataforma unificada na nuvem que simplifica a criação e execução de aplicativos e sites, oferecendo certificados *TLS* gratuitos, uma *CDN* global, proteção contra-ataques, *DDoS*, redes privadas e implantações automáticas a partir do *Git*.

4.1.29 React-Html-Parser

Versão: 2.0.2

Sobre: *React-Html-Parser* é uma biblioteca do *react* que é capaz de renderizar de forma dinâmica *html* como texto formatado, descrições de produtos ou conteúdo gerado pelo usuário, sem arriscar ataques de injeção de código (XSS) por meio de uma renderização insegura.

4.1.30 Verificação JWT

Versão: 9.0.0

Sobre: A verificação *JWT* (*JSON Web Token*) é o processo de garantir a validade e segurança de *tokens JWT*. Isso envolve verificar a assinatura digital, a validade temporal e as reivindicações do *token*. Com isso, foi feito o armazenamento do *token*

na *localStorage* e posteriormente passando para cada rota do sistema. A verificação inclui a decodificação do *token*, verificação da assinatura, garantia de que não tenha expirado e confirmar se as reivindicações são válidas por meio do acesso com e-mail e senha na aplicação.

5 Desenvolvimento

A seguir serão apresentadas as telas desenvolvidas do sistema e suas respectivas funcionalidades. O objetivo inicial do projeto é desenvolver um sistema eficiente, capaz de suprir as necessidades dos *stakeholders* para a comunicação e abertura de chamados internos em uma empresa.

Este objetivo foi bem-sucedido baseando-se na construção e arquitetura do software por meio dos diagramas, requisitos, casos de uso e regras de negócio da empresa, o que proporciona um direcionamento das restrições e funcionalidades necessárias ao modelo de negócio da corporação. Como identificação do projeto, foi escolhido o nome Astra, pela simplicidade e fácil pronúncia.

Em uma visão mais técnica o sistema foi criado utilizando arquitetura *MVC* (*Model-View-Controller*) com criação das *models*, *controllers* e *views*. Optou-se por esse tipo de estrutura devido a uma maior organização do código com divisão de responsabilidades e maior facilidade em sua manutenção. Dessa forma a divisão é feita em três camadas: a camada de interação do usuário (***view***) em React.js, a camada de manipulação dos dados (***model***) e a camada de controle (***controller***). O Node.js foi o responsável por se relacionar com o banco de dados PostgreSQL por meio do envio de dados, gravação e na criação das requisições necessárias para as ações de criar, visualizar, deletar e editar os dados.

Como o banco de dados está hospedado na plataforma Render, foi necessário inserir as credenciais de acesso em um arquivo seguro de extensão *.env* e fazer sua manipulação com o ORM Prisma mapeando objetos do código-fonte para tabelas em um banco de dados relacional ao invés de criarmos as consultas em SQL manualmente. O Prisma proporciona mais robustez ao sistema no momento da criação do SQL, pois evita violações de segurança como o *SQLi* (*Structured Query Language Injection*).

Uma demonstração pode ser feita no código para a criação de um *ticket*, em que começa importando o *prismaClient* e o tipo *CreateCall* de um arquivo em um diretório específico (usando uma notação de caminho relativo). Em seguida, ele define

uma classe chamada *CreateCallService* que possui um método chamado *execute* que é assíncrono. Este método recebe um objeto desestruturado da requisição contendo vários parâmetros relacionados à criação de uma chamada de serviço, como *body*, *technicalId*, *createdBy*, *categoryId*, *header*, *maxDate*, *priority* e *status*.

Dentro do método *execute*, o código utiliza o Prisma para realizar algumas operações relacionadas ao banco de dados, buscando um usuário específico com base no *createdBy* usando *prismaClient.user.findUnique()*, em seguida, cria uma chamada de serviço com os dados fornecidos usando *prismaClient.call.create()*.

O serviço *CreateCallService* desempenha um papel semelhante ao modelo (*Model*), pois está preocupado com a lógica de negócios relacionada à criação de chamadas de serviço e à interação com o banco de dados, encapsulando essa lógica em um serviço reutilizável. Essa mesma estrutura foi usada para os diferentes serviços do sistema e é interessante para mostrar a interação de diferentes ferramentas utilizadas. Pode-se destacar as seguintes telas:

Figura 18: Código da Tela de *login*

```
import prismaClient from "../../prisma"
import { CreateCall } from "./schema"

class CreateCallService {
  async execute({ body, technicalId, createdBy, categoryId, header, maxDate, priority, status }: CreateCall) {
    const userCreated = await prismaClient.user.findUnique({
      where: {
        id: createdBy,
      },
    })
    const call = await prismaClient.call.create({
      data: {
        body,
        technicalId,
        createdBy,
        categoryId,
        header,
        maxDate,
        priority,
        status,
      },
    })

    return { call, userCreated }
  }
}

export { CreateCallService }
```

Fonte: os autores

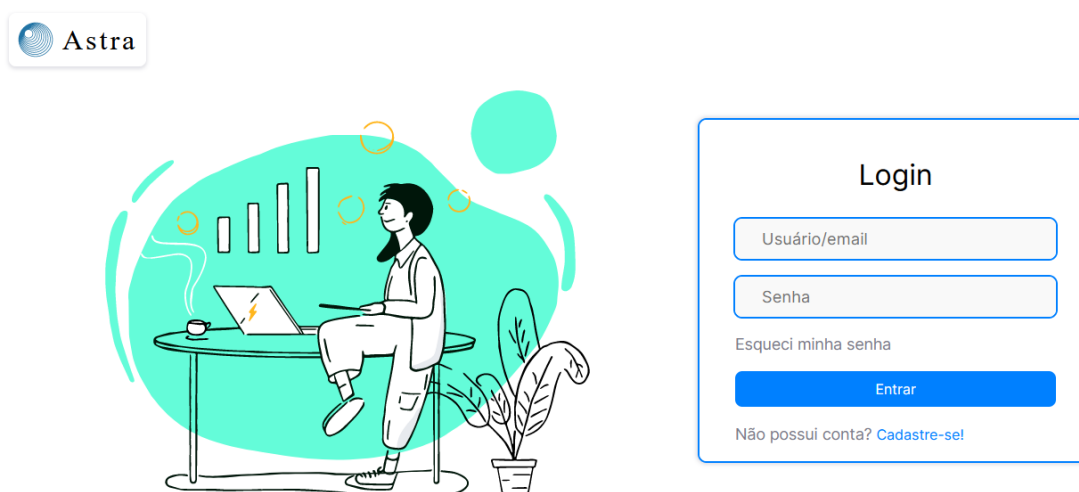
5.1 Tela de *Login*

Primeira tela onde o usuário irá fazer o acesso com o *e-mail* e senha passado pelo gestor, no momento de seu cadastro. Caso ele já tenha um *login* apenas deve

preencher os campos obrigatórios e não será necessário clicar no botão de cadastro para uma nova solicitação.

Assim que a validação for feita com sucesso, será gerado um *token* e encaminhado para armazenamento local no navegador. For informado no tópico de ferramentas ele será passado para as rotas do sistema de forma segura nas funcionalidades, pois apenas usuários cadastrados poderão fazer uso dos recursos. Assim que o *token* for expirado, em um prazo de 30 minutos, ou o próprio usuário sair do programa o mesmo código não pode ser mais usado, bloqueando novamente a navegabilidade, portanto é necessário fazer um novo acesso.

Figura 19: Tela de *login*



Fonte: os autores

Considerando essa página no desenvolvimento do *front-end* em React, foram utilizados componentes que ajudam na reutilização do HTML e CSS já criados. Um exemplo seria a logo e imagem *svg* à esquerda na tela descrita:

Figura 20: Código em React.js da página de *Login*

```

<IntroContainer>
  <IntroHeader>
    <SVGlogo />
    <p>Astra</p>
  </IntroHeader>
  <IntroMain>
    <img src={SVGdraw} alt="Ilustração com colaboradora trabalhando" />
  </IntroMain>
</IntroContainer>

```

Fonte: os autores

Com o uso de *Style Components* foram criados componentes que reutilizam os estilos de CSS e sua estrutura *HTML*. No código, todas as *tags*, com exceção de *img* foram criados seguindo esse conceito e para descrever mais sua praticidade existe o componente *IntroContainer*, que ao ser importado de outro arquivo carrega os estilos CSS e cria um uma *tag div* por meio da propriedade *styled*, seguindo a seguinte estrutura:

Figura 21: Componente estilizado do bloco principal da tela de *login*

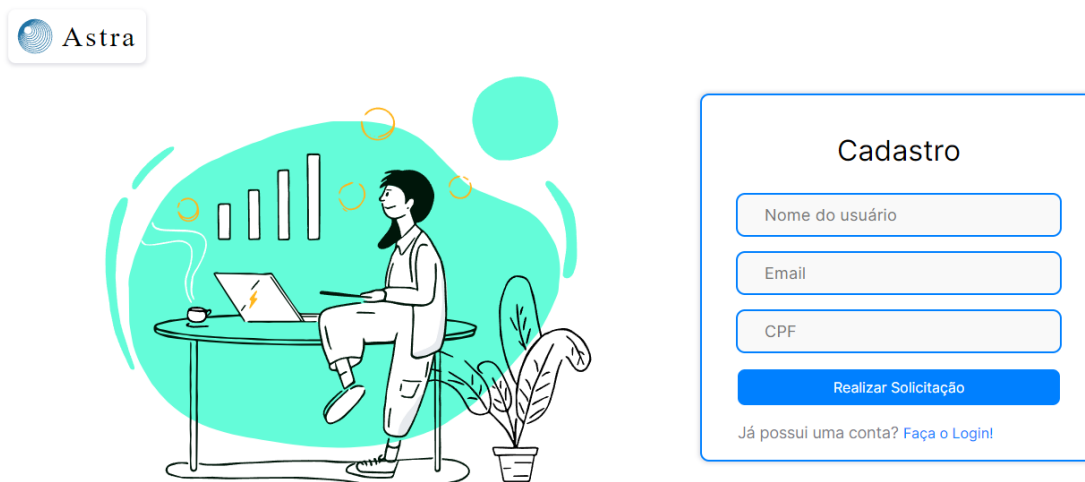
```
const IntroContainer = styled.div`  
  display: flex;  
  flex-direction: column;  
  justify-content: space-between;  
  padding: 30px 0;  
  box-sizing: border-box;  
`;  
;
```

Fonte: os autores

Isso é feito para as demais telas, destacando que nem sempre será necessária sua utilização, por motivos que uma *tag* simples já pode satisfazer os objetivos do sistema e criar a função desejada, porém em casos mais complexos e que necessitem de reutilização do mesmo código, serão empregados.

Caso o usuário tenha clicado no botão para se cadastrar, será necessário preencher os campos de nome, CPF e *e-mail* para que a solicitação seja feita. Depois de todos os campos serem preenchidos a solicitação é enviada para análise e criação do cadastro pelo gestor.

Figura 22: Tela de cadastro



Fonte: os autores

5.2 Tela de Usuários

A página mostrará uma tabela com todos os usuários do sistema demonstrando as suas respectivas informações de nome, *e-mail*, CPF, celular, departamento e cargo, juntamente com as opções de editar, excluir e criar um usuário. Todas as opções serão exploradas mais a fundo nas telas seguintes.

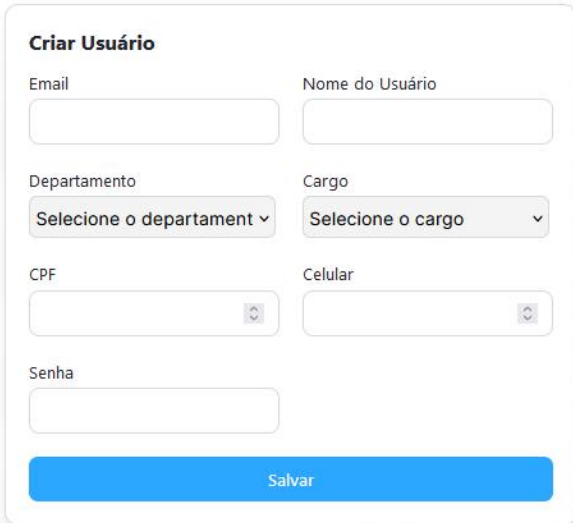
Figura 23: Tela de usuário cadastrados

Criado em	Nome	Email	CPF	Celular	Ação
23/09/2023, 06:46:40	João Victor	joao@admin.com.br	73385491509	1699980	+ Editar Excluir
23/09/2023, 06:48:49	Silas	silas@rh.com.br	09838957089	1697894	+ Editar Excluir
23/09/2023, 08:08:51	Thiago	thiago@admin.com.br	15030598049	16974125	+ Editar Excluir
23/09/2023, 19:00:50	Douglas	douglas@admin.com.br	98190067087	1697854	+ Editar Excluir
23/09/2023, 19:01:35	Ana	ana@admin.com.br	86920588072	1697856	+ Editar Excluir
23/09/2023, 19:02:15	Luis	luis@admin.com.br	35906318054	1695236	+ Editar Excluir
23/09/2023, 19:11:35	Samuel	samuel@admin.com.br	34365647024	1697854	+ Editar Excluir

Fonte: os autores

O botão de criar usuário abre um modal para preenchimento dos dados de e-mail, nome, CPF, celular e senha, assim como a escolha do departamento e cargo. Após ser salvo, o registro é mostrado na tabela de usuários para visualização.

Figura 24: Modal de criação do usuário

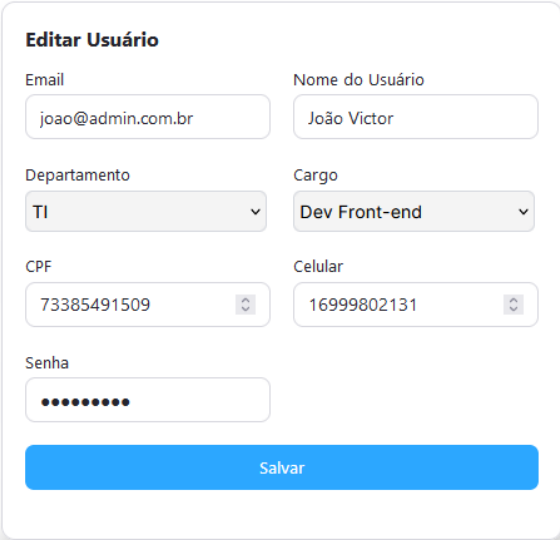


O modal de criação de usuário, intitulado "Criar Usuário", apresenta os seguintes campos e elementos:

- Campos de texto para "Email" e "Nome do Usuário".
- Menu suspenso para "Departamento" com o texto "Selecione o departament".
- Menu suspenso para "Cargo" com o texto "Selecione o cargo".
- Campos de texto para "CPF" e "Celular", cada um com um ícone de máscara de teclado.
- Campo de texto para "Senha".
- Botão azul "Salvar" na base do modal.

Fonte: os autores

Depois de selecionar a opção de editar, o usuário poderá escolher quais campos serão alterados. Esta funcionalidade oferece a edição de todos os campos preenchidos na criação, inclusive inserindo uma nova senha. Por questões de segurança, o valor da senha não é informado na tela de editar, mas é possível sua mudança ao inserir um novo valor.

Figura 25: Modal de edição de usuário

O modal de edição de usuário, intitulado "Editar Usuário", apresenta os seguintes campos:

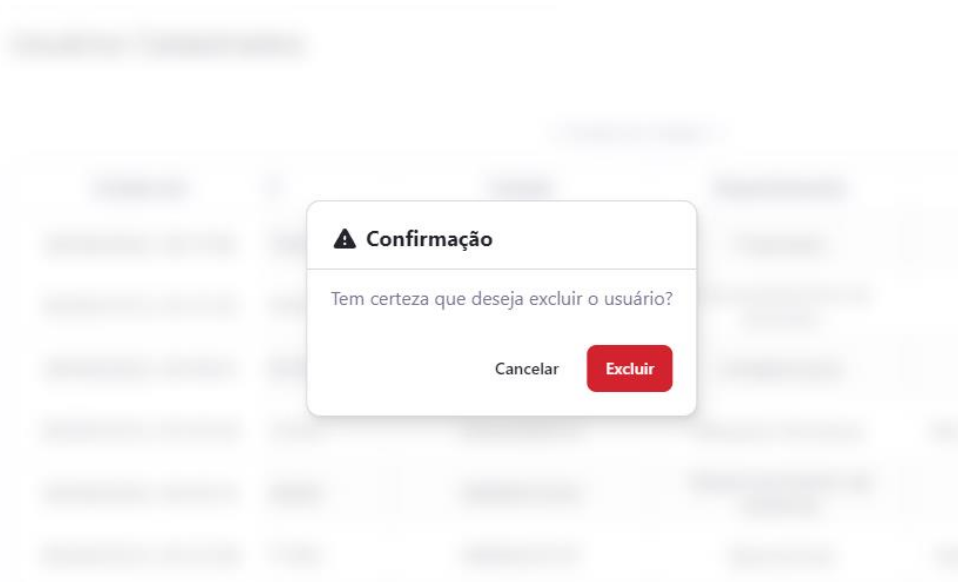
- Email: joao@admin.com.br
- Nome do Usuário: João Victor
- Departamento: TI (menu suspenso)
- Cargo: Dev Front-end (menu suspenso)
- CPF: 73385491509 (campo com máscara)
- Celular: 16999802131 (campo com máscara)
- Senha: campo oculto com pontos

Um botão azul "Salvar" está localizado na base do modal.

Fonte: os autores

Para deletar um cadastro, abrirá uma caixa de diálogo para confirmar ou cancelar a ação. Em ambos os casos, será feito retorno para a tela de usuários cadastrados após conclusão do procedimento.

Figura 26: Modal de exclusão do usuário



Fonte: os autores

5.3 Tela de *Tickets*:

A página oferece a funcionalidade de visualizar, criar, excluir ou editar os *tickets*, além de dados totais de *tickets* criados por status e prioridade. Na tabela se concentram informações do código de identificação, data em que foi criado, prazo máximo de resolução, *status*, prioridade, título, assunto, atribuição, departamento responsável pela tratativa e categoria designada.

Figura 27: Tela de *tickets* cadastrados

Astra
23/09/2023 Sábado

Serviços
Tickets
Usuários

Usuário
Configurações

Sair

Pesquisar

João Victor

Tickets Cadastrados

Total 5

Pendente Concluído 2 0

Em andamento Atrasado 2 1

Baixa Alta 1 1

Média Urgente 2 1

+ Criar Ticket

Filtrar por...

— Direção de rolagem —

Prioridade	Código	Status	Título	Assunto	Ação
Média	#1	Pendente	Contratar Desenvolvedor	Contratação de desenvolvedor para o cargo de Back-end.	Visualizar Excluir
Média	#2	Em andamento	Criar template da nova plataforma operacional.	Criar layout com time front-end. Seguir padrão do Figma.	Visualizar Excluir
Baixa	#3	Pendente	Problema de Conexão Wi-Fi	Não consigo me conectar à rede Wi-Fi em casa. Preciso de ajuda para resolver esse problema.	Visualizar Excluir
Alta	#4	Em andamento	Dúvida sobre Pagamento	Gostaria de esclarecer uma dúvida relacionada ao pagamento da minha fatura.	Visualizar Excluir
Urgente	#5	Atrasado	Suporte Técnico para Software	Estou enfrentando problemas ao usar um software e preciso de suporte técnico para solucionar isso.	Visualizar Excluir

Fonte: os autores

Ao clicar em editar é feito acesso à página do *ticket* relacionado e suas opções, oferecendo possibilidade de alterar todos os dados da tabela e adicionar ou excluir comentários na tela lateral. Os comentários somente podem ser criados ao seu respectivo *ticket* informando a data e hora de cadastro de forma automática, portanto a opção não pode ser usada no momento da abertura do procedimento. Nele são permitidos a inclusão de textos com formatação, *links*, imagens e vídeos.

Figura 28: Tela do *ticket*

Astra
23/09/2023 Sábado

Serviços
Tickets
Usuários

Usuário
Configurações

Sair

Q Pesquisar

João Victor

Título
Contratar Desenvolvedor

Assunto
Contratação de desenvolvedor para o cargo de Back-end.

Categoria
Departamento Pessoal

Departamento
RH

Atribuição
Silas

Prazo Máximo
07 / 10 / 2023

Status
Pendente

Prioridade
Média

Salvar

Publicar

João Victor
23/09/2023, 19:54:22

Descrição da Vaga de Desenvolvedor Front-End na Astra

Empresa: Astra Tech

Localização: Franca, SP

A Astra Tech está em busca de um talentoso Desenvolvedor Front-End para se juntar à nossa equipe em Franca. Como Desenvolvedor Front-End na Astra, você será responsável por projetar e desenvolver interfaces de usuário atraentes e eficazes, bem como otimizar a experiência do usuário em nossas aplicações web.

Responsabilidades:

1. Colaborar com a equipe de design e desenvolvimento para traduzir designs de UI/UX em código funcional e responsivo.
2. Desenvolver e manter aplicações web de alto desempenho, garantindo que elas sejam escaláveis e eficientes.
3. Otimizar aplicações para garantir a melhor experiência do usuário possível, incluindo o carregamento rápido de páginas e a compatibilidade com dispositivos móveis.
4. Identificar e resolver problemas de usabilidade e desempenho.

Fonte: os autores

Caso queira deletar o *ticket* a opção excluir da tabela pode ser utilizada, em que ao clicar, será aberto um modal de confirmação com dois botões, um para confirmar a exclusão e outro para cancelamento, ambos com retorno à tela de *tickets* cadastrados.

Figura 29: Tela de exclusão do *ticket*

Confirmação

Tem certeza que deseja excluir o ticket?

Cancelar Excluir

Fonte: os autores

Na criação de um novo *ticket*, a página tem o mesmo padrão demonstrado na tela de editar, com diferença de que os campos estão vazios para preenchimento. É importante destacar que as opções de departamento, atribuição e categoria do *ticket*

realizada uma busca nos registros da própria aplicação, e nas telas de configurações e usuários é possível adicionar e modificar cada um.

5.4 Tela de Configurações:

Na página de configurações o usuário poderá cadastrar, editar e excluir departamentos, cargos e categorias conforme sua necessidade, todos seguindo a mesma estrutura. Se um departamento possuir cargos vinculados não é possível excluí-lo.

Isso foi feito devido a associação de ambos no momento de cadastro do cargo, porém essa alteração também é pode de ser feita para atender o organograma da empresa. A opção de categorias permite criar repartições nos *tickets* e proporcionar um maior direcionamento da solicitação, por conta que os departamentos são usados para criar cargos, que por sua vez serão aplicados aos usuários e estes aos *tickets*. Já as categorias são introduzidas diretamente no *ticket*.

Figura 30: Tela de configurações, cargos

Nome	Departamento	Ação
Recrutamento e seleção	RH	Editar Excluir
Infraestrutura	TI	Editar Excluir
Suporte Técnico	Atendimento ao Cliente	Editar Excluir
Dev Front-end	TI	Editar Excluir
Dev Back-end	TI	Editar Excluir
Gerente de Contas	Financeiro	Editar Excluir

Fonte: os autores

6 Resultados e Discussão

A princípio foi realizada uma pesquisa sobre o seguimento de atuação da empresa, pois o conhecimento da área é fundamental para estruturar o projeto e pontuar as principais necessidades e melhorias que podem ser feitas nos processos já usados.

Para coletar dados, utilizamos a ferramenta Google Forms, que nos permitiu fazer perguntas simples aos funcionários da empresa, analisando o tempo gasto e as perdas de arquivos. Essa ferramenta nos permitiu coletar os pontos necessário para

início do projeto. Após criar o quadro *SWOT* e o *5W2H*, foi possível determinar as principais dificuldades da empresa e onde deveríamos focar nossos esforços.

Através de uma entrevista com os funcionários da empresa e a análise de suas respostas, foi possível a construção da Matriz *SWOT* (Forças, Oportunidades, Fraquezas e Ameaças), com a ferramenta Excel. A principal função da análise *SWOT* é avaliar resultados do ambiente externo e interno da empresa, obtendo informações necessárias para identificação dos pontos fracos e fortes. Através dela, desenvolvemos com uso do LucidChart a Estrutura Analítica do Projeto (LUCIDCHART, 2019).

Em seguida, se construiu o Termo de Abertura do Projeto (TAP), onde destaca-se a situação atual, justificativas do projeto, objetivos e metas, descrição, premissas, restrições, *stakeholders*, riscos, custos e prazos. Nele, se propõe uma resolução sistêmica para os problemas e descrição das etapas necessárias para que esse projeto seja construído.

A Elicitação de Requisitos, definido como: definir, tornar explícito, obter o máximo de informações sobre o objeto em questão, é a primeira atividade no processo de engenharia de requisitos, na qual se busca entender quais são as necessidades do usuário que devem ser atendidas pelo *software* que será desenvolvido.

Na criação do BPMN, novamente foi usado o LucidChart em conjunto com o Astah, programa feito para criação de mapas, gráficos, porém voltado para a modelagem de diagramas. O desenvolvimento da documentação continuou com a criação do Diagrama de Caso de Uso, que demonstra os objetivos e interações entre os usuários e o sistema, assim como os processos exigidos pelo *software* para executar esses objetivos.

Paralelamente, outros diagramas desenvolvidos foram o Diagrama de Atividade, representando o fluxo de relacionamento entre os processamentos, o Diagrama de Máquina de Estado, que representa o estado em que o objeto se encontra durante o processo de execução do sistema, e o Diagrama de Sequência, que representa o fluxo dos processos e a comunicação entre eles. Em todos os diagramas foram usadas as ferramentas LucidChart ou Astah.

Para melhor entendimento do usuário, a prototipação das telas foi construída por meio da ferramenta Figma. Nesta etapa, o usuário já consegue visualizar melhor todo o projeto e ter uma noção de como está a implementação de *UI/UX* e da identidade visual construída. Nesta fase, se inicia a escolha dos frameworks, pré-

processadores e demais ferramentas que seriam utilizadas no *front-end* para seu futuro desenvolvimento em linhas de código.

Após sua finalização, deu-se sequência à documentação da Matriz de Rastreabilidade, pois ela interliga os elementos dos Requisitos Funcionais, Casos de uso e Regras de Negócio, evidenciando em caso de modificação do projeto, o que poderia ser afetado e o que deve ser alterado em conjunto.

A Matriz de Portabilidade também foi importante para representar os equipamentos, infraestrutura, características de configuração, descrição dos usuários e recursos humanos e licenças utilizados pela empresa, sendo um indicativo para atribuir os requisitos mínimos para que o *software* funcione como o esperado.

Partindo para outra etapa dos artefatos de engenharia de *software*, foi definido as métricas com base no Diagrama de Classe e em uma prévia do banco de dados a ser estruturado, com o registro do valor da hora trabalhada, tipo de sistema e linguagem utilizada. Aqui, o Excel se mostrou uma excelente ferramenta, tornando esta tarefa simples.

Em seguida, foi debatido a forma como o projeto deveria ser realizado, fazendo esboços e rascunhos. Para tomar decisões de forma eficiente, foi usado a ferramenta Discord, que permitiu interações mesmo em locais diferentes. Nesse processo, decidiu-se realizar os diagramas utilizando programas como LucidChart e Astah.

Antes do desenvolvimento do *software*, fizemos uma prototipagem no Figma, um sistema *web* que permite a criação de protótipos funcionais, evitando erros futuros ou inconformidade com o projeto tratado. Após essa etapa, foi organizado o tempo e a divisão de trabalhos, aprofundando em conhecimentos das linguagens de programação escolhidas.

Assim, deu-se início à fase de desenvolvimento, utilizando o Visual Studio Code como editor de código, com uso de React.js e Node.js, juntamente com TypeScript, como tecnologias principais de programação. Para o banco de dados, optamos pelo PostgreSQL devido ao seu bom relacionamento com o Node.js, juntamente com a divisão responsabilidades entre a equipe no *back-end* e *front-end*.

Durante o desenvolvimento do *software*, o código do *front-end* e *back-end* foi mantido em dois repositórios diferentes, de forma que o *back-end* também se manteve armazenado em um container do *Docker*. Isso possibilitou que inicialmente fosse criado uma estrutura da página e em paralelo as funcionalidades do sistema, testando os *CRUDs* através do Insomnia.

A medida em que o *front-end* se tornava mais completo, com uso de bibliotecas como o MUI e *framework React.js*, o *back-end* começou a ser integrado e sua API ser consumida. Foram realizados testes com todas as funcionalidades criadas a fim de assegurar que o funcionamento esperado estava ocorrendo e caso contrário, retornávamos as regras de negócio e requisitos criados para alinhar e corrigir os erros.

Considerações finais

O projeto teve como objetivo central a criação de um sistema para a criação e controle de chamados internos da empresa. Com esse sistema os usuários poderão, de forma simples e prática, centralizar suas demandas em um único lugar evitando perdas de dados, informações e arquivos. Esta centralização é controlada pelo setor ou funcionário.

Cada chamado será acompanhado por um *chat*, permitindo o envio de textos e arquivos entre os setores, garantindo que os documentos sempre cheguem aos destinatários corretos. Esses *chats* permitem interação de mais de um usuário, com o objetivo de evitar a perda de arquivos caso ocorra alguma modificação no chamado, permanecendo registrado na barra de notificações dos usuários envolvidos.

Assim, essa solução visa resolver problemas comuns em grandes empresas, como perda de tempo e desentendimentos entre os funcionários. Com o sistema em funcionamento, será possível registrar o andamento de todos os documentos e melhorar a comunicação interna.

Além do desenvolvimento *web*, consideramos no futuro a possibilidade de criar uma versão *mobile*, tornando o sistema acessível a funcionários que estão em locais distantes. Especialmente em empresas de pequeno e médio porte, essa abordagem permitirá o uso em várias plataformas, como celulares e relógios. Será uma solução efetiva para empresas em que nem todos os funcionários têm acesso a computadores a qualquer momento.

Acredita-se que o resultado do projeto e o percurso de sua produção resultaram em uma aplicação bem-sucedida, funcional e que atendesse as expectativas dos stakeholders. Demonstrando a metodologia desde o princípio, até a implementação de tecnologias que até então não tínhamos prática, foi preciso estudo e aprofundamento para que pudéssemos garantir a qualidade definida nos requisitos da documentação.

Dessa forma, foi possível conciliar uma aplicação de interface simples com um *back-end* seguro e organizado, o que nos possibilita a implementação de futuras funcionalidades ao escalar o projeto para outras empresas que desejam melhorar sua comunicação e gerenciamento operacional por meio desta solução sistêmica.

Referências

ANTONIO. **Artigo Engenharia de Software 3 - Requisitos Não Funcionais**, 2008. Disponível em: <<https://www.devmedia.com.br/artigo-engenharia-de-software-3-requisitos-nao-funcionais/9525>> Acesso em: 07 ago. 2023.

BRUNO, Elias. **O que é comunicação interna? Aprenda as 14 estratégias para fazer a da sua empresa!**. 07/02/2019. Disponível em: <<https://rockcontent.com/br/blog/comunicacao-interna/>>. Acesso em: 07 ago. 2023.

COUTINHO, T. **Termo de abertura do projeto: saiba o que é e como fazer**. Disponível em: <<https://www.voitto.com.br/blog/artigo/o-que-e-termo-de-abertura-do-projeto>>. Acesso em: 14 ago. 2023.

EJFGV. **Business Model Canvas: o que é e qual a sua importância?** Disponível em: <<https://ejfgv.com/business-model-canvas/>>. Acesso em: 9 nov. 2023.

LEANDRO. **O que é UML e Diagramas de Caso de Uso: Introdução Prática à UML**, 2012. Disponível em: <<https://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408>>. Acesso em: 04 ago. 2023.

LUCIDCHART. **O que é BPMN?**. Disponível em: <<https://www.lucidchart.com/pages/pt/o-que-e-bpmn>>. Acesso em: 25 jul. 2023.

NASCIMENTO, Felipe. **Javascript ou Typescript?** 24/06/2021. Disponível em: <https://www.alura.com.br/artigos/javascript-ou-typescript?gclid=CjwKCAjwj-MiiBhA4EiwAZe6jQ9Y2g-Fhul9HYvleNHrla7PvoYJxiPCE1VcPb-GIsf5hXwa9Tslr9mBoC6loQAvD_BwE>. Acesso em: 04 ago. 2023.

PIERRI, Renato. **Qual é a diferença entre requisitos funcionais e requisitos não funcionais na engenharia de software?** 01/11/2019. Disponível em: <<https://netsnuts.com.br/qual-e-a-diferenca-entre-requisitos-funcionais-e-requisitos-nao-funcionais-na-engenharia-de-software/>>. Acesso em: 27 jul. 2023.

PLÍNIO, Ventura. **O que é Requisito Funcional**. 11/05/2016. Disponível em: <<https://www.ateomomento.com.br/o-que-e-requisito-funcional>>. Acesso em: 18 agosto 2023.

RIBEIRO, Renato. **Entenda o que são as regras de negócio e como ajudam seu software!**, 06/10/2020. Disponível em: <<https://www.iugu.com/blog/regras-de-negocio>>. Acesso em: 4 ago. 2023.

SIMÕES, Guilherme S; VAZQUEZ, Carlos E. **What are the levels of granularity of functional requirements and why this is important**, 21/02/2017. Disponível em: <<https://re-magazine.ireb.org/articles/functional-requirements-and-their-levels-of-granularity/>>. Acesso em: 7 ago. 2023.

SYDLE. **O que são Regras de Negócio? Qual a importância? Veja exemplos**, 01/08/2023. Disponível em: <<https://www.sydle.com/br/blog/regras-de-negocio-5f6333be1e43744c69d995e0>>. Acesso em: 08 agosto 2023.

TOTVS, E. **Análise SWOT: o que é, para que serve, benefícios e como fazer**. Disponível em: <<https://www.totvs.com/blog/negocios/analise-swot/#:~:text=A%20an%C3%A1lise%20SWOT%20%C3%A9%20uma>>. Acesso em: 10 jul. 2023.