

Alunos:

Rômulo Moreira Moreto Guisso

Kayky José Kenji Ribeiro

Luiz Alexandre Nishiyama Galvani

**Design de Software:**  
**Sistema de tickets de suporte**

Orientador: Tiago Dutra Galvão

Brasil  
16 de junho de 2025

# Sumário

<b>Sumário</b>	<b>2</b>
<b>0.1 Introdução</b>	<b>4</b>
<b>0.2 BPMN</b>	<b>4</b>
0.2.1 Notificações Automáticas	4
0.2.2 Acompanhamento de Chamados	5
0.2.3 Criação e Atribuição de Tickets	6
0.2.4 Geração de Relatórios Administrativos	7
<b>0.3 Metodologia Scrum</b>	<b>8</b>
0.3.1 Papel dos Integrantes	8
0.3.2 Organização das Sprints	9
0.3.3 Cerimônias Realizadas	9
0.3.4 Ferramentas de Apoio	10
0.3.5 Resultados Esperados	10
<b>0.4 Levantamento de Requisitos</b>	<b>10</b>
0.4.1 Descrição Geral do Sistema	11
0.4.2 Requisitos Funcionais	11
0.4.3 Requisitos Não Funcionais	12
0.4.4 Perspectivas	12
0.4.4.1 Perspectiva do Usuário	12
0.4.4.2 Perspectiva do Técnico	12
0.4.4.3 Perspectiva do Sistema	12
0.4.4.4 Perspectiva do Negócio	13
<b>0.5 Diagramas UML</b>	<b>13</b>
0.5.1 Diagrama de Atividades	13
0.5.2 Diagrama de Casos de Uso	13
0.5.3 Diagrama de Estados	14
0.5.4 Diagrama de Sequência	15
0.5.5 Diagrama de Classes	16
<b>0.6 Definição de Tecnologias</b>	<b>17</b>
0.6.1 JavaScript Fullstack	17
0.6.2 Banco de Dados em Memória	17

## Resumo

Este trabalho apresenta o desenvolvimento de um sistema de gerenciamento de tickets para suporte técnico, que permite abertura, acompanhamento, atribuição e organização de chamados. O sistema inclui notificações automáticas, controle de acesso por níveis de permissão, armazenamento do histórico de interações e geração de relatórios de desempenho. Utiliza BPMN para modelagem de processos e Scrum para gestão ágil, visando otimizar o atendimento e a transparência do serviço.

## 0.1 Introdução

A gestão eficiente de solicitações de suporte técnico é fundamental para garantir produtividade e satisfação dos usuários. Este projeto propõe um sistema de gerenciamento de tickets que organiza, prioriza e acompanha chamados, com controle de acesso e notificações automáticas. O sistema registra o histórico completo e gera relatórios para análise de desempenho. Utilizando BPMN para modelagem e Scrum para o desenvolvimento ágil, a solução busca melhorar a organização e eficiência do suporte técnico.

## 0.2 BPMN

A modelagem dos processos do sistema foi realizada utilizando a notação BPMN (Business Process Model and Notation), que permite uma visualização clara e estruturada dos fluxos de trabalho e interações entre usuários e o sistema.

### 0.2.1 Notificações Automáticas

O diagrama da Figura 1 representa o fluxo de notificações automáticas por e-mail que o sistema realiza para informar usuários e técnicos sobre atualizações nos tickets. Essa automação é fundamental para garantir transparência e agilidade na comunicação durante o atendimento.

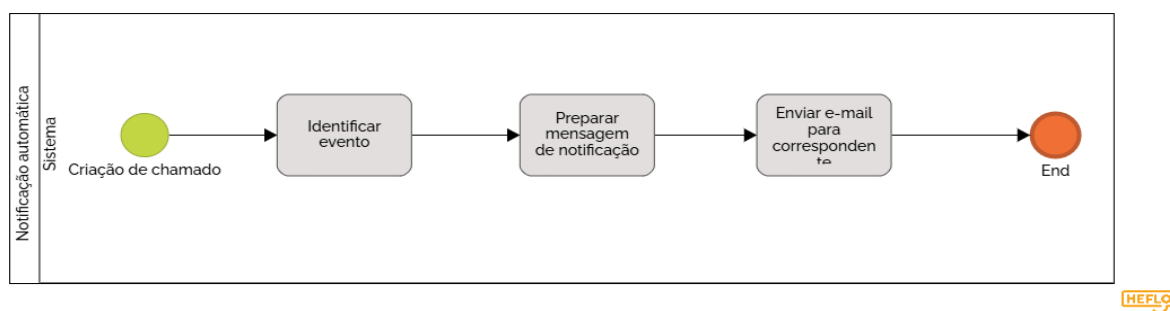


Figura 1 – Fluxo BPMN: Notificações automáticas por e-mail

### 0.2.2 Acompanhamento de Chamados

O ciclo de vida do chamado, apresentado na Figura 2, detalha as etapas desde a abertura até a resolução e encerramento do ticket. Este fluxo contempla os estados do ticket e as ações realizadas por usuários e técnicos, assegurando o acompanhamento em tempo real e registro completo.

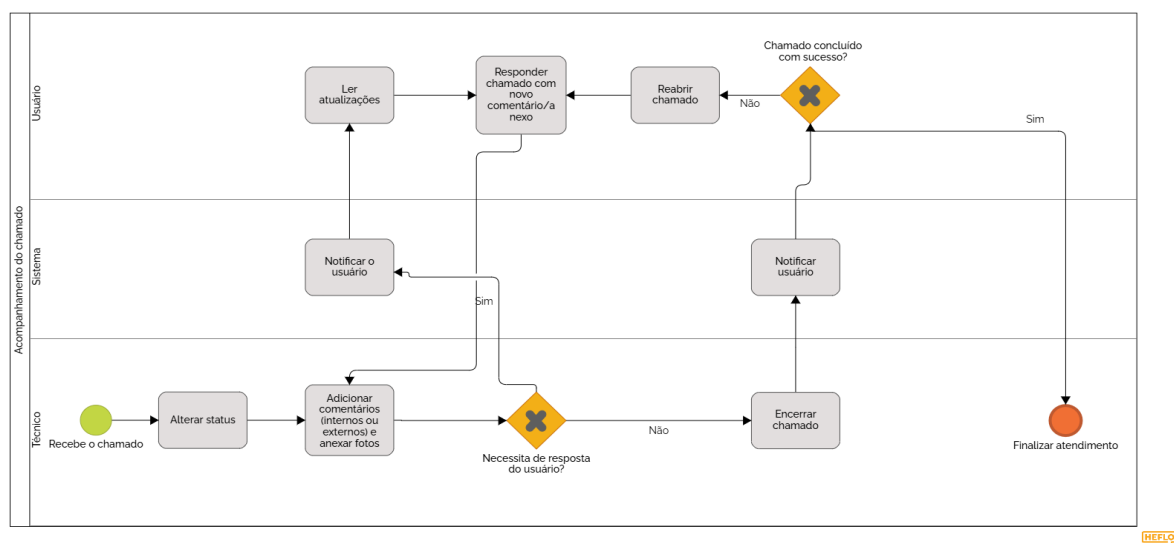
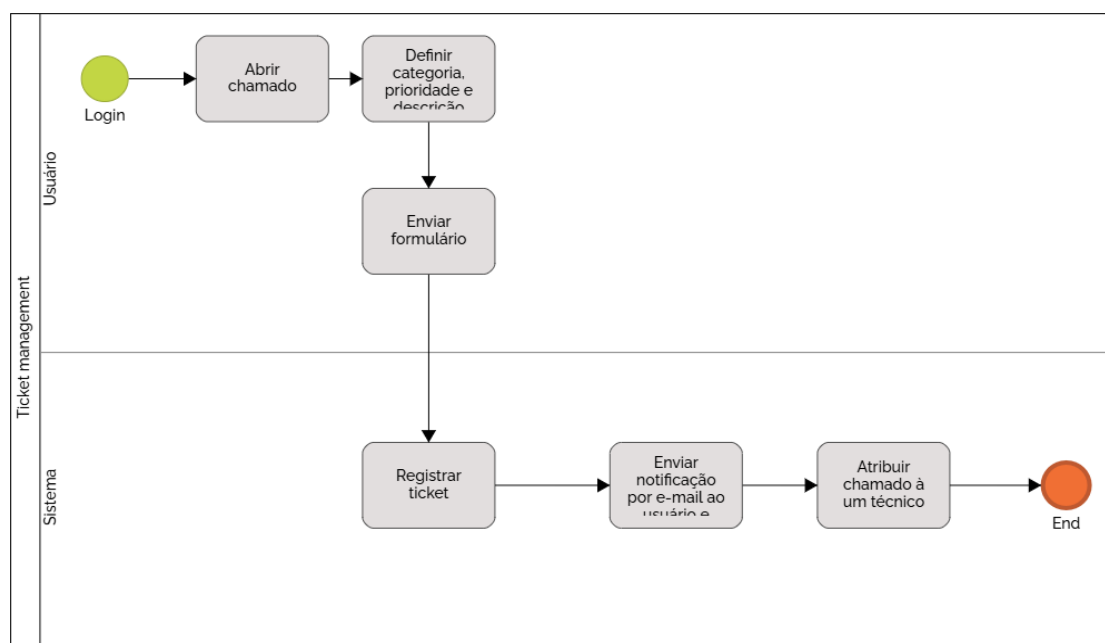


Figura 2 – Fluxo BPMN: Ciclo de vida do chamado

### 0.2.3 Criação e Atribuição de Tickets

A Figura 3 mostra o processo de criação do ticket, onde o usuário informa categoria, prioridade e descrição, seguido da atribuição automática ou manual para o técnico responsável. Esse fluxo é essencial para organizar o atendimento e distribuir as demandas conforme critérios definidos.

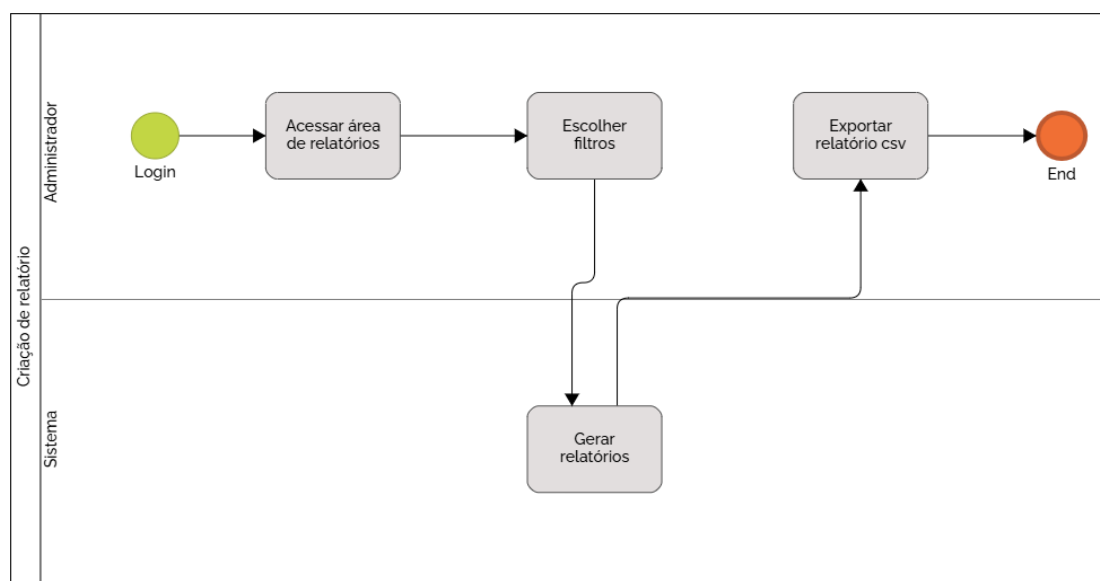


MEFLO

Figura 3 – Fluxo BPMN: Criação e atribuição de tickets

#### 0.2.4 Geração de Relatórios Administrativos

Por fim, o processo ilustrado na Figura 4 demonstra a geração de relatórios de desempenho e estatísticas administrativas, permitindo a análise da eficiência da equipe e o cumprimento dos acordos de nível de serviço (SLAs).



HEFLO

Figura 4 – Fluxo BPMN: Geração de relatórios

### 0.3 Metodologia Scrum

A equipe adotou a metodologia ágil **Scrum** para condução do desenvolvimento do sistema de geração e gerenciamento de tickets de suporte. Essa escolha fundamenta-se na capacidade do framework de promover entregas incrementais, facilitar a adaptação a mudanças de requisitos e estimular a colaboração contínua entre os membros do time. O Scrum permite organizar o projeto em ciclos curtos e iterativos, denominados **sprints**, com entregas funcionais ao final de cada ciclo.

#### 0.3.1 Papel dos Integrantes

No contexto deste projeto, os papéis do Scrum foram distribuídos entre os integrantes da equipe da seguinte forma:

- **Product Owner (PO):** Responsável por definir e priorizar os requisitos do sistema, manter o backlog do produto e assegurar que as funcionalidades entregues estejam alinhadas às necessidades dos usuários finais. Atuou como ponte entre os desenvolvedores e os stakeholders (professor e possíveis usuários).



- **Scrum Master:** Responsável por garantir que a equipe siga os princípios e práticas do Scrum, removendo impedimentos e promovendo a melhoria contínua do processo. Facilitou as cerimônias e promoveu a comunicação eficiente entre os membros.
- **Time de Desenvolvimento:** Equipe multidisciplinar encarregada de implementar as funcionalidades do sistema, realizar testes e garantir a qualidade das entregas. As decisões técnicas, como escolha de tecnologias e arquitetura do sistema, também foram tomadas coletivamente por este grupo.

### 0.3.2 Organização das Sprints

O desenvolvimento foi estruturado em **sprints semanais**, cada uma com duração de sete dias. Ao final de cada sprint, eram entregues funcionalidades incrementais do sistema, proporcionando visibilidade contínua do progresso e possibilidade de ajustes rápidos.

- **Sprint 1:** Planejamento do escopo inicial, definição dos requisitos básicos, criação da estrutura do projeto e do banco de dados relacional.
- **Sprint 2:** Desenvolvimento do módulo de criação e submissão de tickets, com campos essenciais como categoria, prioridade e descrição.
- **Sprint 3:** Implementação do acompanhamento em tempo real dos chamados e envio de notificações automáticas por e-mail.
- **Sprint 4:** Geração de relatórios administrativos e construção da interface para equipe de suporte.
- **Sprint 5:** Integração de todos os módulos, testes funcionais e ajustes finais com base nos feedbacks.

### 0.3.3 Cerimônias Realizadas

Durante o ciclo de desenvolvimento, foram realizadas as seguintes cerimônias do Scrum:

- **Daily Scrum (Reunião Diária):** Realizada todos os dias úteis, com duração média de 15 minutos. Os integrantes compartilharam o que foi feito no dia anterior, o que seria feito no dia atual e os impedimentos enfrentados.
- **Sprint Planning (Planejamento da Sprint):** Realizada no início de cada sprint para definição das tarefas a serem desenvolvidas, estimativas de esforço e divisão de responsabilidades.
- **Sprint Review (Revisão da Sprint):** Ao final de cada sprint, foi feita uma apresentação das funcionalidades implementadas, validando o que havia sido planejado e discutindo eventuais desvios.

- **Sprint Retrospective (Retrospectiva da Sprint):** Reunião interna da equipe para reflexão sobre o que funcionou bem, o que poderia ser melhorado e quais ações seriam tomadas para a próxima sprint.

#### 0.3.4 Ferramentas de Apoio

A gestão das tarefas foi realizada utilizando a ferramenta **Trello**, na qual o backlog foi estruturado em colunas que representam os estágios do fluxo de trabalho: *Product Backlog*, *To Do*, *In Progress*, *Code Review*, *Done*. Cada card representava uma tarefa ou funcionalidade, permitindo visibilidade clara e controle do andamento do projeto.

Além disso, foram utilizados outros recursos para comunicação e acompanhamento das atividades, como:

- **GitHub:** Para controle de versão e integração contínua do código-fonte.
- **WhatsApp:** Para comunicação informal e tomada de decisões rápidas.
- **TS/Discord/Teams:** Para comunicação informal chamada de vídeo para reuniões online.
- **Google Docs:** Para registro de atas de reunião e documentação auxiliar.

#### 0.3.5 Resultados Esperados

Com a aplicação do Scrum, a equipe buscou garantir:

- Entregas frequentes e incrementais com valor funcional;
- Facilidade na identificação e correção de falhas;
- Maior organização e autonomia da equipe;
- Melhoria contínua ao longo do desenvolvimento;
- Maior adaptação às mudanças de escopo e requisitos.

### 0.4 Levantamento de Requisitos

Este capítulo apresenta o levantamento detalhado dos requisitos do sistema de gerenciamento de tickets de suporte técnico, contemplando as funcionalidades esperadas, restrições e necessidades dos usuários, além dos aspectos técnicos e de negócio.

### 0.4.1 Descrição Geral do Sistema

O sistema de gerenciamento de tickets tem como objetivo permitir a abertura, acompanhamento, organização e resolução de chamados de suporte técnico. Para isso, o sistema deve:

- Permitir abertura e acompanhamento de tickets pelos usuários;
- Organizar os chamados por prioridade, categoria e status;
- Atribuir chamados automaticamente ou manualmente a técnicos;
- Enviar notificações automáticas sobre atualizações nos tickets;
- Armazenar o histórico completo das interações;
- Gerar relatórios de desempenho e cumprimento de SLAs;
- Controlar o acesso por níveis de permissão;
- Permitir comentários e anexos em tickets para interação entre usuários e técnicos.

### 0.4.2 Requisitos Funcionais

Os principais requisitos funcionais levantados para o sistema são:

- RF01 – O usuário deve poder criar uma conta e realizar login com segurança;
- RF02 – O usuário deve abrir chamados informando categoria, prioridade, descrição e anexos;
- RF03 – O técnico deve alterar o status dos chamados para atualizar o acompanhamento;
- RF04 – O sistema deve atribuir chamados automaticamente com base em critérios pré-definidos;
- RF05 – Usuários e técnicos devem poder buscar e filtrar chamados para facilitar a localização;
- RF06 – O sistema deve enviar notificações por e-mail para atualizações nos tickets;
- RF07 – Técnicos devem poder adicionar comentários internos e externos, além de anexar arquivos;
- RF08 – Administradores devem gerar relatórios de desempenho, tempo médio de atendimento e SLAs;
- RF09 – Administradores devem exportar dados dos chamados em formato CSV para análise externa.

### 0.4.3 Requisitos Não Funcionais

Os requisitos não funcionais do sistema, classificados conforme a norma ISO/IEC 25010, são:

Tabela 1 – Requisitos Não Funcionais do Sistema

Requisito	Categoria ISO/IEC 25010	Subcaracterística
Criptografia dos dados	Segurança	Confidencialidade
Controle de permissões	Segurança	Autenticação e autorização
Tempo de resposta até 1 segundo	Desempenho e eficiência	Eficiência de tempo
Uptime de 99,9%	Confiabilidade	Disponibilidade
Interface amigável	Usabilidade	Estética, acessibilidade e operacionalidade
Compatibilidade com navegadores e mobile	Portabilidade	Portabilidade e adaptabilidade
Backups diários automáticos	Confiabilidade	Recuperabilidade

### 0.4.4 Perspectivas

#### 0.4.4.1 Perspectiva do Usuário

- O usuário deve poder abrir tickets para relatar problemas;
- Visualizar e acompanhar o status dos chamados em tempo real;
- Receber notificações por e-mail quando houver atualizações nos tickets.

#### 0.4.4.2 Perspectiva do Técnico

- Alterar status dos tickets, inserir comentários internos e externos;
- Anexar arquivos e encerrar tickets;
- Visualizar todos os tickets atribuídos ou pendentes.

#### 0.4.4.3 Perspectiva do Sistema

- Permitir cadastro de usuários com diferentes perfis e níveis de permissão;
- Registrar, categorizar e priorizar tickets com prazos definidos;
- Manter histórico completo dos tickets para auditoria;
- Suportar notificações automáticas por e-mail baseadas nas ações realizadas.

#### 0.4.4.4 Perspectiva do Negócio

- Contribuir para redução do tempo de resolução dos chamados;
- Permitir mensuração do desempenho da equipe de suporte via relatórios;
- Garantir atendimento dentro dos prazos estipulados nos SLAs;
- Assegurar que todas as interações sejam auditáveis, garantindo rastreabilidade e segurança.

## 0.5 Diagramas UML

Esta seção apresenta os principais diagramas UML que modelam o sistema de gerenciamento de tickets, ilustrando diferentes aspectos do projeto, como comportamento, estrutura e interações.

### 0.5.1 Diagrama de Atividades

O diagrama de atividades representa o fluxo dinâmico das ações no sistema, detalhando os processos envolvidos na criação, acompanhamento e encerramento dos tickets.

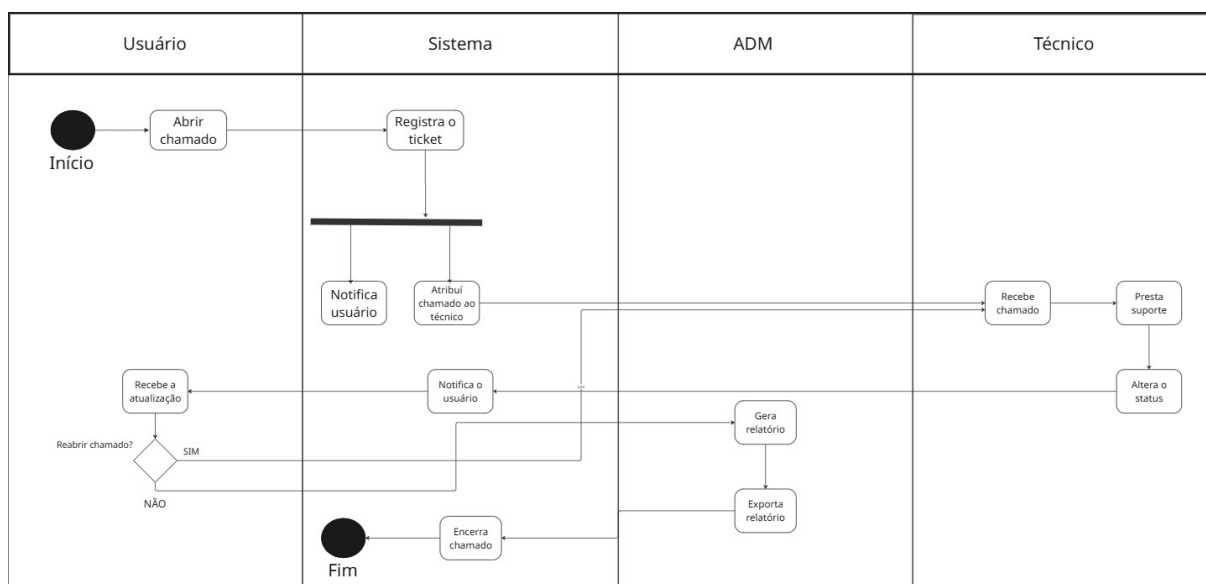


Figura 5 – Diagrama de Atividades do Sistema

### 0.5.2 Diagrama de Casos de Uso

Este diagrama demonstra as funcionalidades do sistema do ponto de vista dos atores envolvidos, destacando as principais interações entre usuários, técnicos e administradores.

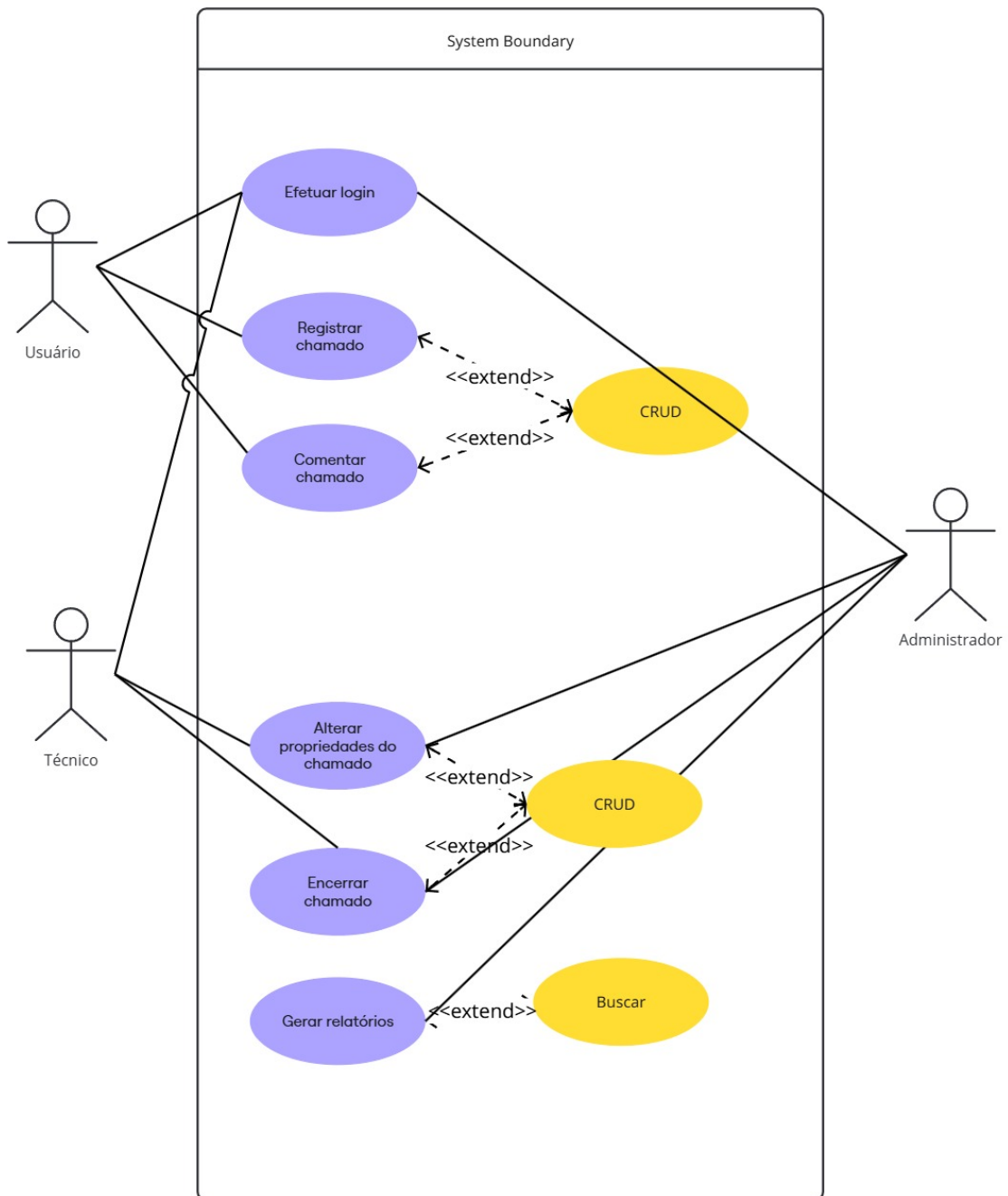


Figura 6 – Diagrama de Casos de Uso

### 0.5.3 Diagrama de Estados

O diagrama de estados detalha as possíveis situações pelos quais um ticket pode passar durante seu ciclo de vida, incluindo estados como aberto, em atendimento, pendente e encerrado.

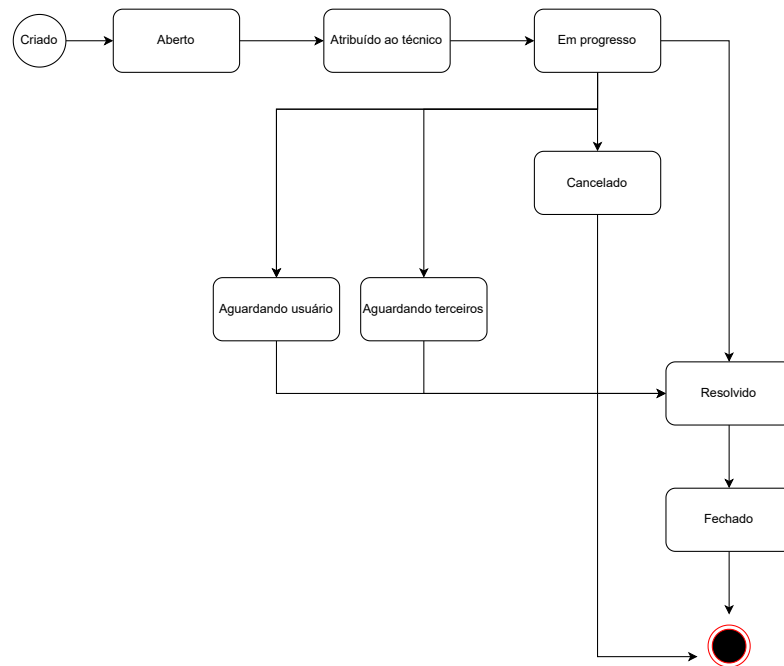


Figura 7 – Diagrama de Estados do Ticket

#### 0.5.4 Diagrama de Sequência

Este diagrama exemplifica a interação entre objetos ao longo do tempo durante o processo de criação e atualização de um ticket, mostrando a ordem das mensagens trocadas.

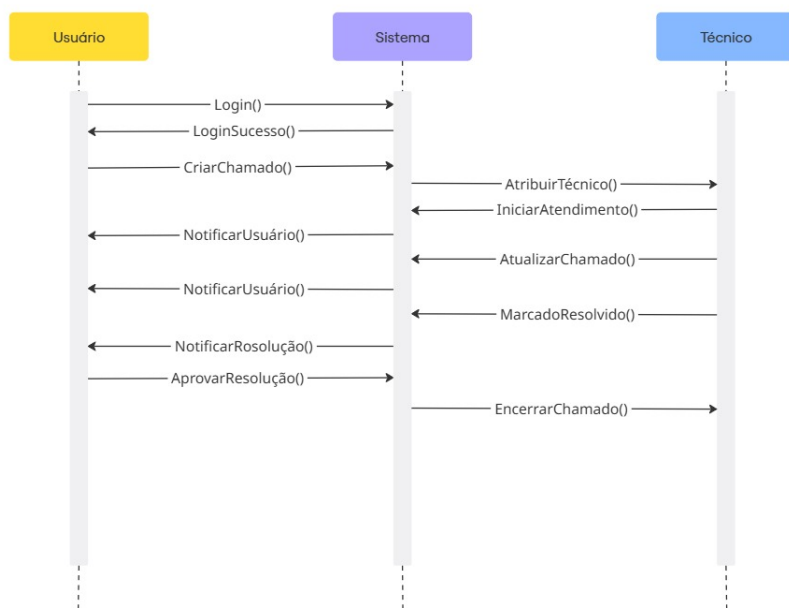


Figura 8 – Diagrama de Sequência para o Processo de Ticket

### 0.5.5 Diagrama de Classes

O diagrama de classes expõe a estrutura estática do sistema, com as principais classes, atributos, métodos e seus relacionamentos.

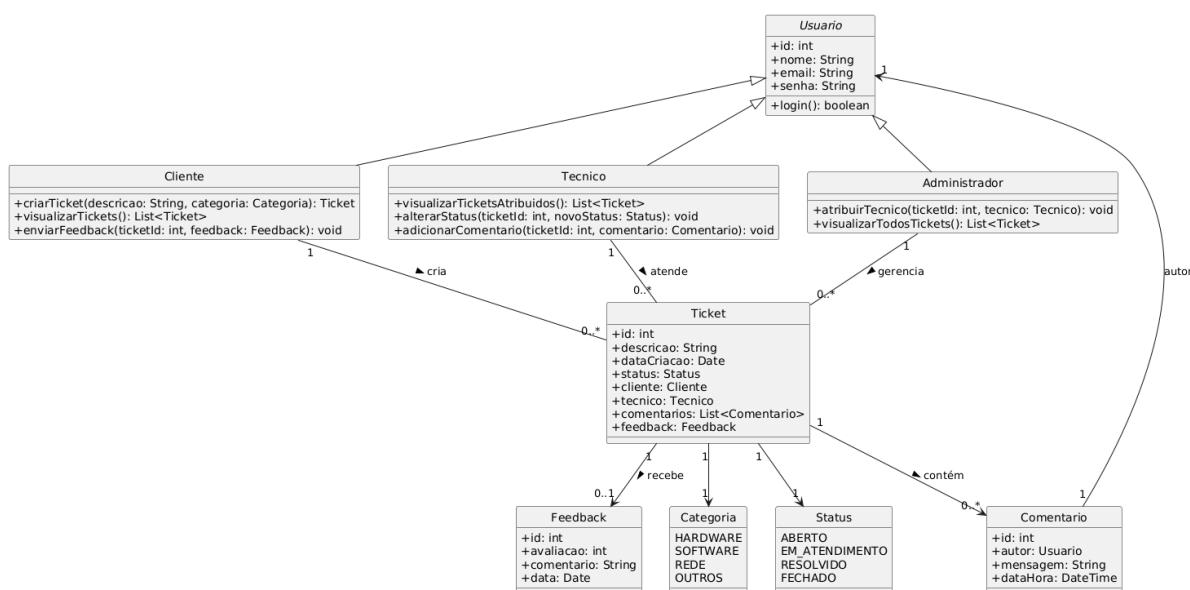


Figura 9 – Diagrama de Classes do Sistema



## 0.6 Definição de Tecnologias

Para o desenvolvimento do sistema de gerenciamento de tickets de suporte, foram consideradas duas possíveis abordagens tecnológicas, ambas amplamente utilizadas no mercado de software. A escolha final levará em conta fatores como familiaridade da equipe, requisitos técnicos e infraestrutura disponível.

### 0.6.1 JavaScript Fullstack

- **Node.js:** Ambiente de execução JavaScript no servidor, que possibilita a criação de aplicações escaláveis e rápidas.
- **Express.js:** Framework minimalista para Node.js, utilizado para construir APIs RESTful de forma simples e organizada.
- **HTML/CSS/JavaScript:** Tecnologias básicas para a criação da interface do usuário, que permitem estruturar, estilizar e implementar a lógica no front-end.
- **Fetch API:** API do navegador para realizar requisições HTTP assíncronas, utilizada para comunicação entre o front-end e o back-end.

### 0.6.2 Banco de Dados em Memória

- **Estrutura em memória:** Para fins de prototipagem e testes, os dados dos tickets são armazenados temporariamente em uma estrutura na memória do servidor, sem persistência em banco externo.

Ambas as tecnologias utilizadas no projeto (Node.js com Express no back-end e HTML/CSS/JS no front-end) possibilitam o desenvolvimento rápido e eficiente do sistema de gerenciamento de tickets. O uso do armazenamento em memória foi uma escolha para simplificar a implementação inicial, podendo ser substituído futuramente por um banco de dados persistente conforme a evolução do projeto.