



UNIVERSIDADE PAULISTA

CAUAN RAMOS DE LIMA

LUANN GALLEGO DE SOUZA

MATEUS CAETANO RIBEIRO

SISTEMA INTELIGENTE DE SUPORTE DE TI:  
AUTOMAÇÃO E OTIMIZAÇÃO COM INTELIGÊNCIA  
ARTIFICIAL

SÃO JOSÉ DO RIO PRETO - SP

2025

CAUAN RAMOS DE LIMA  
LUANN GALLEGOS DE SOUZA  
MATEUS CAETANO RIBEIRO

## Sistema Inteligente de Suporte de TI: Automação e Otimização com Inteligência Artificial

Trabalho de projeto integrado multidisciplinar para obtenção do título em graduação em Analise e Desenvolvimento de Sistema apresentado à Universidade Paulista – UNIP. Disciplina de Desenvolvimento de Projeto.

SÃO JOSÉ DO RIO PRETO - SP

2025

# RESUMO

Este trabalho tem como objetivo desenvolver o HelpDesk, um sistema de chamados com integração de Inteligência Artificial (IA), fundamentado nos conhecimentos adquiridos no curso de Análise e Desenvolvimento de Sistemas. A disciplina de Engenharia de Software é essencial para a estruturação do sistema, pois define seus requisitos e assegura a qualidade em todas as etapas do desenvolvimento, desde a concepção até a entrega do produto final. A Análise Orientada a Objetos possibilita a modelagem de casos de uso, a criação de diagramas de classe e a organização eficiente do código, facilitando a manutenção e evolução do sistema. O uso de um Banco de Dados garante o armazenamento seguro e eficiente das informações, preservando sua integridade, confiabilidade e disponibilidade para os usuários. A disciplina de Programação Orientada a Objetos é fundamental para a construção do software, que utilizará a linguagem (C SHARP). Essa abordagem permite a criação de um código mais modular, reutilizável e escalável, garantindo maior flexibilidade para futuras atualizações e aprimoramentos. O Projeto de Interface com o Usuário viabiliza o desenvolvimento de interfaces intuitivas e acessíveis para web, mobile e desktop, proporcionando uma experiência agradável e eficiente para os usuários. Além disso, a disciplina de Economia de Mercado auxilia na análise de custos e na viabilidade financeira do projeto, permitindo um planejamento adequado dos investimentos necessários. Já a Gestão de Recursos Humanos define funções e responsabilidades dentro da equipe de desenvolvimento e suporte, assegurando a eficiência na operação e na manutenção contínua do sistema.

**Palavras-chave:** Sistema de chamados, Engenharia de Software, Análise Orientada a Objetos, Programação Orientada a Objetos, Banco de Dados, Economia de Mercado, Gestão de Recursos Humanos, Interface com o Usuário.

# ABSTRACT

This work aims to develop Digisic, a call system with Artificial Intelligence (AI) integration, based on the knowledge acquired in the Systems Analysis and Development course. The Software Engineering discipline is essential for structuring the system, as it defines its requirements and ensures quality throughout all stages of development, from conception to the final product delivery. Object-Oriented Analysis enables the modeling of use cases, the creation of class diagrams, and the efficient organization of the code, facilitating system maintenance and evolution. The use of a Database ensures secure and efficient storage of information, preserving its integrity, reliability, and availability for users. The Object-Oriented Programming discipline is fundamental for software development, utilizing the (C SHARP) language. This approach allows for the creation of a modular, reusable, and scalable codebase, ensuring greater flexibility for future updates and improvements. User Interface Design enables the development of intuitive and accessible interfaces for web, mobile, and desktop, providing users with a pleasant and efficient experience. Furthermore, the Market Economics discipline assists in cost analysis and the financial viability of the project, allowing for adequate planning of the necessary investments. Human Resources Management defines roles and responsibilities within the development and support team, ensuring efficiency in the operation and continuous maintenance of the system.

**Keywords:** Call System, Software Engineering, Object-Oriented Analysis, Object-Oriented Programming, Database, Market Economics, Human Resources Management, User Interface.

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>7</b>
1.1	<b>Proposta do trabalho</b>	7
1.2	<b>Objetivos</b>	8
1.2.1	Objetivo Geral	8
1.2.2	Objetivos específicos	8
1.3	<b>Justificativa</b>	8
1.4	<b>Regras de Negócios</b>	8
1.5	<b>Glossário do sistema</b>	9
1.6	<b>Ciclo de vida do Software: Kanban</b>	10
<b>2</b>	<b>DESENVOLVIMENTO</b>	<b>12</b>
2.1	<b>Engenharia De Software</b>	12
2.1.1	Requisitos Funcionais	12
2.1.2	Requisitos Não Funcionais	12
2.2	<b>Analise Orientada a Objetos</b>	13
2.2.1	Diagrama de Caso de Uso: Login e Autenticação.	13
2.2.2	Diagrama de Caso de Uso: Registrar Chamado	14
2.2.3	Diagrama de Caso de Uso: Classificar e Encaminhar o chamado	16
2.2.4	Diagrama de Caso de Uso: Consultar status do chamado	17
2.2.5	Diagrama de Caso de Uso: Encerrar o chamado	18
2.2.6	Diagrama de Caso de Uso: Gerenciar usuários e técnicos	20
2.2.7	Diagrama de Caso de Uso: Emitir Relatório	21
2.2.8	Diagrama de Caso de Uso: Gestão de Chamados Inteligente com Encaminhamento por IA	23
2.3	<b>Diagramas de Sequência</b>	25
2.3.1	Diagrama De Sequência: Login e Autenticação	25
2.3.2	Diagrama De Sequência: Registro de Chamados	25
2.3.3	Diagrama De Sequência: Consultar Status de Chamados	26
2.3.4	Diagrama De Sequência: Encerrar Chamados	26
2.3.5	Diagrama De Sequência: Gerenciar usuarios e técnicos	27
2.3.6	Diagrama De Sequência: Emitir Relatório	27
2.4	<b>Diagrama de Classe</b>	28
2.4.1	Diagrama De Classe	28
2.5	<b>Diagrama de Implantação</b>	29
2.5.1	Diagrama De Implantação	29

2.5.2	Estrutura do sistema . . . . .	29
2.5.3	Configuração do Servidor . . . . .	30
2.5.4	Instalação do Banco de Dados . . . . .	30
2.5.5	Instalação dos aplicativos . . . . .	30
2.5.6	Comunicação Entre Componentes . . . . .	31
2.5.7	Manutenção e Atualizações . . . . .	31
<b>2.6</b>	<b>Banco de Dados . . . . .</b>	<b>31</b>
2.6.1	Scripts do Banco de Dados . . . . .	31
2.6.2	Scripts de Testes . . . . .	34
2.6.3	Diagrama Entidade e Relacionamento . . . . .	36
2.6.4	Dicionario de dados: Tabela Tipos Usuarios . . . . .	37
2.6.5	Dicionario de dados: Tabela Usuarios . . . . .	38
2.6.6	Dicionario de dados: Tabela Técnico . . . . .	39
2.6.7	Dicionario de dados: Tabela Administrador . . . . .	39
2.6.8	Dicionario de dados: Tabela Categoria . . . . .	40
2.6.9	Dicionario de dados: Tabela Chamado . . . . .	41
2.6.10	Dicionario de dados: Continuação da Tabela Chamado . . . . .	42
2.6.11	Dicionario de dados: Histórico de Atualização . . . . .	43
2.6.12	Dicionario de dados: Relatório . . . . .	44
2.6.13	Dicionario de dados: Chamado Relatório . . . . .	45
2.6.14	Planilha de Teste . . . . .	45
<b>2.7</b>	<b>Interface Humano Computador . . . . .</b>	<b>47</b>
2.7.1	Protótipo de Tela: Mobile . . . . .	47
2.7.2	Protótipo de Tela: Web . . . . .	50
2.7.3	Protótipo de Tela: Web . . . . .	50
2.7.4	Protótipo de tela: Desktop . . . . .	51
<b>3</b>	<b>CONCLUSÃO . . . . .</b>	<b>54</b>
	<b>REFERÊNCIAS . . . . .</b>	<b>55</b>

# 1 INTRODUÇÃO

## 1.1 Proposta do trabalho

O propósito deste trabalho é aprofundar os conceitos práticos e teóricos abordados nas disciplinas do curso de Análise e Desenvolvimento de Sistemas da Universidade Paulista de São José do Rio Preto, além de aplicar esses conceitos em um estudo de caso em um sistema de chamados para suporte técnico de TI. O principal objetivo é desenvolver um sistema de chamados com integração de inteligência artificial, o qual permite que a equipe de suporte técnico possa responder aos clientes de forma mais rápida e eficiente. Para alcançar esse fim, serão aplicados os conhecimentos adquiridos nas disciplinas de engenharia de software, análise orientada a objetos, banco de dados, interface com o usuário, economia de mercado e gestão de recursos humanos.

Na disciplina de engenharia de software, os alunos exploram os princípios e técnicas fundamentais para o desenvolvimento de sistemas, com foco na definição dos requisitos funcionais e não funcionais que garantem o correto funcionamento do software. Essa base oferece o suporte necessário para o planejamento e a organização deste projeto.

Em programação orientada a objetos, os conceitos de algoritmos e estruturas de dados formam o alicerce da construção do sistema, que será desenvolvido na linguagem C(SHARP) com o framework .NET.

No banco de dados, a estruturação e organização das informações do sistema se baseiam no uso do SQL Server, garantindo armazenamento seguro e eficiente.

Na disciplina de análise orientada a objetos, o uso da UML é aprofundado com foco nos diagramas que representam o funcionamento interno do software, facilitando sua compreensão e manutenção.

interface com o usuário, os conhecimentos adquiridos permitem criar protótipos de telas para desktop, dispositivos móveis e web, promovendo uma experiência mais fluida e intuitiva para o usuário.

Na gestão de recursos humanos, os conceitos ajudam a entender a importância do trabalho em equipe, da liderança e da comunicação eficiente durante o desenvolvimento do projeto, além de contribuir para o gerenciamento das demandas e do tempo da equipe.

## 1.2 Objetivos

### 1.2.1 Objetivo Geral

Este trabalho tem como objetivo aprofundar os conceitos aprendidos no curso de Análise e Desenvolvimento de Sistemas da Universidade Paulista de São José do Rio Preto, além de aplicar na prática os conhecimentos adquiridos ao longo do semestre. A finalidade é construir um sistema de suporte técnico inteligente.

### 1.2.2 Objetivos específicos

Aplicar levantamento de requisitos do sistema.

Desenvolver artefatos UML.

Criação de interfaces gráficas para usabilidade no sistema.

Utilização da inteligência artificial para boas práticas.

## 1.3 Justificativa

Este trabalho tem como objetivo melhorar o gerenciamento de chamados para o suporte técnico, ajudando no controle da entrada e saída de informações. Além disso, permite a aplicação e o aprofundamento dos conhecimentos adquiridos ao longo deste semestre no curso de Análise e Desenvolvimento de Sistemas.

## 1.4 Regras de Negócios

Para que a equipe compreenda adequadamente a solicitação, cada chamado deve conter informações detalhadas sobre o ocorrido, incluindo: assunto, explicação e tipo. O registro de chamados no sistema somente será permitido mediante o fornecimento completo dessas informações por parte do usuário.

A classificação dos chamados será realizada pela IA, que definirá o atendente mais adequado para cada tipo de situação. A prioridade (baixa, média, alta ou crítica) será determinada pelo atendente responsável, com base nas informações disponíveis.

O sistema deve permitir a geração de relatórios contendo dados dos chamados, como quantidade, tempo médio de resolução e categorização.

Relatórios devem ser acessíveis apenas para o administrador e devem manter a confidencialidade dos dados.

O sistema deve garantir que as operações críticas (login, classificação, consulta de status) tenham tempo de resposta inferior a 2 segundos.

A classificação de chamados via IA deve ser realizada em até 5 segundos.

Técnicos não possuem permissão para excluir chamados, sendo autorizados apenas a alterar seu status para “aberto”, “em andamento” ou “resolvido”. Usuários podem registrar novos chamados exclusivamente via aplicativo mobile.

O sistema deve automaticamente classificar o chamado utilizando um mecanismo de Inteligência Artificial.

A classificação deve identificar o tipo de chamado para determinar o técnico mais adequado para atendimento.

O sistema deve encaminhar o chamado para o técnico responsável conforme a classificação. Apenas usuários devidamente cadastrados no sistema poderão registrar chamados, garantindo, assim, a segurança, autenticidade e rastreabilidade dos registros.

## 1.5 Glossário do sistema

**Abertura de Chamado:** Processo inicial de registro de um problema ou solicitação no sistema de suporte técnico.

**Chamado:** Registro formal de um problema ou solicitação feita por um usuário ou cliente para o suporte de TI. Pode ser um incidente ou uma solicitação de serviço.

**Incidente:** Qualquer evento que interrompa o funcionamento normal de um serviço de TI, como falhas de hardware, software ou rede.

**Solicitação de Serviço:** Pedido feito por um usuário para obter um serviço específico, como instalação de software, criação de contas ou modificações em configurações de TI.

**Status do Chamado:** Representa a fase atual do chamado no processo de atendimento, como por exemplo status em andamento, aberto, resolvido.

**Categoria:** Tipo de problema ou serviço solicitado. Exemplos de categorias incluem Hardware, Software, Rede, Suporte ao Usuário, Infraestrutura, entre outros.

**Técnico de Suporte:** Profissional responsável por resolver os chamados registrados.

**Resolução:** Resolve o problema ou atende à solicitação do usuário.

**Histórico do Chamado:** Registro completo de todas as interações, soluções e atualizações realizadas durante o ciclo de vida de um chamado, desde sua abertura até o fechamento.

**Escalonamento:** Processo de encaminhar um chamado para um nível mais avançado de suporte, quando não é possível resolvê-lo no nível atual. **d Notificação:** Alerta enviado

ao usuário ou técnico para informar sobre atualizações, status ou resolução do chamado.

## 1.6 Ciclo de vida do Software: Kanban

Foi optado pela utilização da metodologia ágil Kanban para o desenvolvimento do sistema de suporte de TI.

Visualização clara do fluxo de trabalho: o kanban permite visualizar todas as etapas do processo de desenvolvimento deste sistema, como novo, em andamento, teste, concluído, o que facilita a organização e permite saber exatamente o que está sendo realizado.

Identificar gargalos: Facilita a percepção de onde os processos estão travando ou acumulando tarefas, permitindo ações corretivas rápidas.

Transparência para todos os envolvidos: Toda a equipe pode acompanhar o andamento dos chamados, promovendo confiança e clareza.

Adaptação e melhoria contínua: O Kanban é flexível e pode ser ajustado constantemente, conforme as necessidades da equipe e do sistema.

Priorização de chamados: o kanban permite que os chamados mais urgentes sejam atendidos primeiro.

Feedback rápido e constante: Como o processo de desenvolvimento e suporte é transparente, os membros da equipe recebem feedback contínuo sobre o status dos chamados, facilitando a tomada de decisões rápidas e informadas.

Facilidade de gestão e controle: A utilização de Kanban proporciona uma gestão visual e simples, permitindo que líderes de equipe e gestores monitorem a performance do time e o progresso das tarefas em tempo real.

Ajuste contínuo e responsivo: O processo contínuo de ajustes no fluxo de trabalho permite que o Kanban se encaixe perfeitamente no ritmo e nas necessidades da equipe, tornando-se uma ferramenta ágil de fácil adaptação.

Redução do tempo de ciclo: Ao visualizar o progresso e identificar gargalos, o Kanban ajuda a reduzir o tempo de ciclo de cada tarefa, permitindo que os chamados sejam resolvidos mais rapidamente e com mais qualidade.

Como é mostrado abaixo:

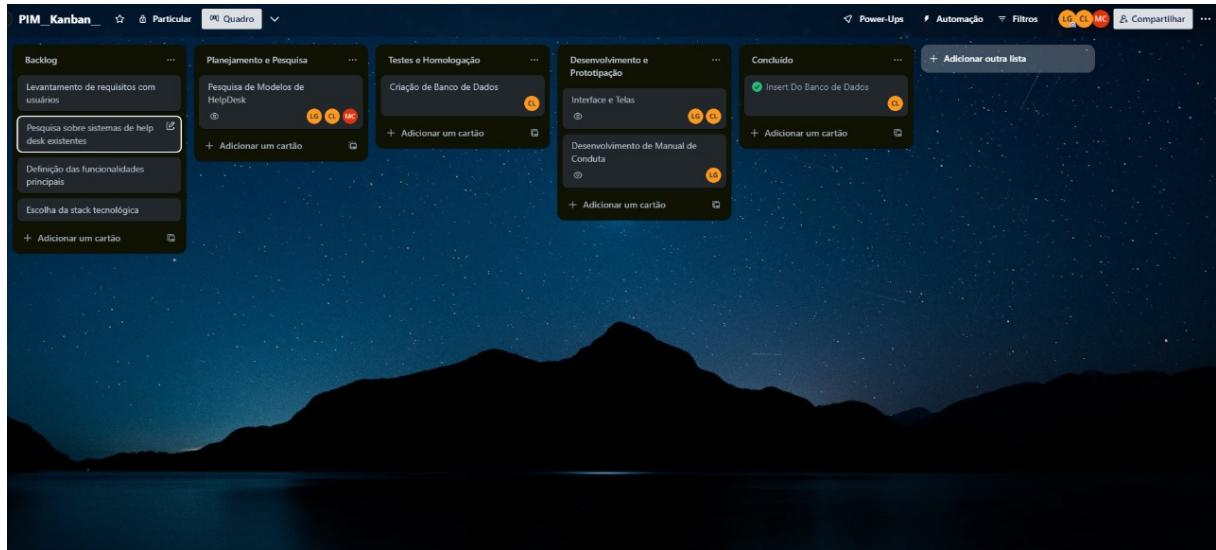


Imagen do quadro kanban

## 2 DESENVOLVIMENTO

### 2.1 Engenharia De Software

#### 2.1.1 Requisitos Funcionais

Login de usuário. O sistema deve permitir que todos os atores (Usuário, Técnico e Administrador) realizem login com suas credenciais pessoais para acessar suas funcionalidades específicas dentro do sistema.

Validação de credenciais. O sistema deve validar as credenciais informadas e permitir ou negar o acesso ao ambiente interno.

Registro de chamados. O sistema deve permitir que o usuário registre um novo chamado via mobile.

Classificação Inteligente de Chamados. O sistema deve utilizar um mecanismo de Inteligência Artificial para classificar automaticamente o chamado e atribuí-lo ao técnico responsável.

Encaminhamento de chamados. O sistema deve encaminhar o chamado classificado ao técnico mais apropriado, com base nos critérios definidos.

Consulta de status de chamado. O sistema deve permitir que o usuário consulte o status atual do seu chamado.

Encerramento de chamado. O sistema deve permitir que o técnico responsável encerre o chamado após sua resolução.

Gerenciamento de usuários e técnicos. O sistema deve permitir ao administrador cadastrar, editar, consultar e excluir usuários e técnicos.

Geração de relatórios. O sistema deve permitir a geração de relatórios com dados sobre os chamados (quantidade, tempo médio de resolução, por categoria, etc.).

#### 2.1.2 Requisitos Não Funcionais

Interface responsiva. A interface do sistema deve ser acessível e responsiva para diferentes dispositivos.

Conformidade com LGPD. O sistema deve seguir integralmente as diretrizes da Lei Geral de Proteção de Dados (LGPD), garantindo segurança e privacidade.

Tempo de Resposta. As operações críticas (login, classificação de chamado, consulta de status) devem ter tempo de resposta inferior a 2 segundos.

Desempenho da IA. O sistema deve realizar a classificação de chamados por IA em até 5 segundos.

Suporte Simultâneo. O sistema deve suportar pelo menos 100 usuários acessando simultaneamente.

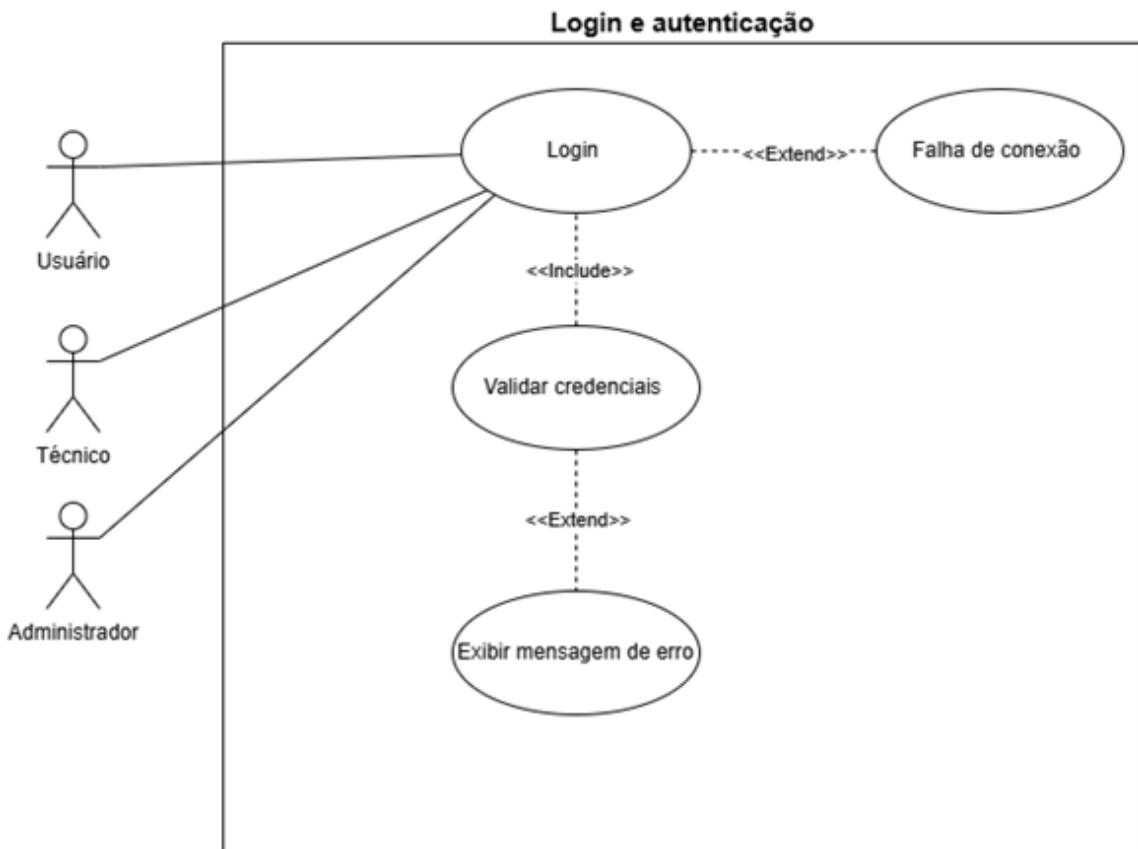
Capacidade de Armazenamento. O sistema deve suportar o gerenciamento de até 1 mil chamados ativos.

Usabilidade. A navegação deve ser intuitiva, com linguagem clara e acessível a usuários de diferentes perfis técnicos.

(ENGENHARIA, )

## 2.2 Analise Orientada a Objetos

### 2.2.1 Diagrama de Caso de Uso: Login e Autenticação.



(DIAGRAMA..., )

Atores: Usuário, técnico e administrador.

Descrição: Permite que o usuário realize login de forma segura no sistema, informando suas credenciais (usuário e senha). O sistema valida os dados e concede acesso

conforme o perfil de permissão.

Fluxo principal: O usuário acessa a tela de login. O sistema solicita usuário e senha. O usuário informa os dados. O sistema valida as credenciais e libera o acesso ao painel correspondente (usuário, técnico ou administrador).

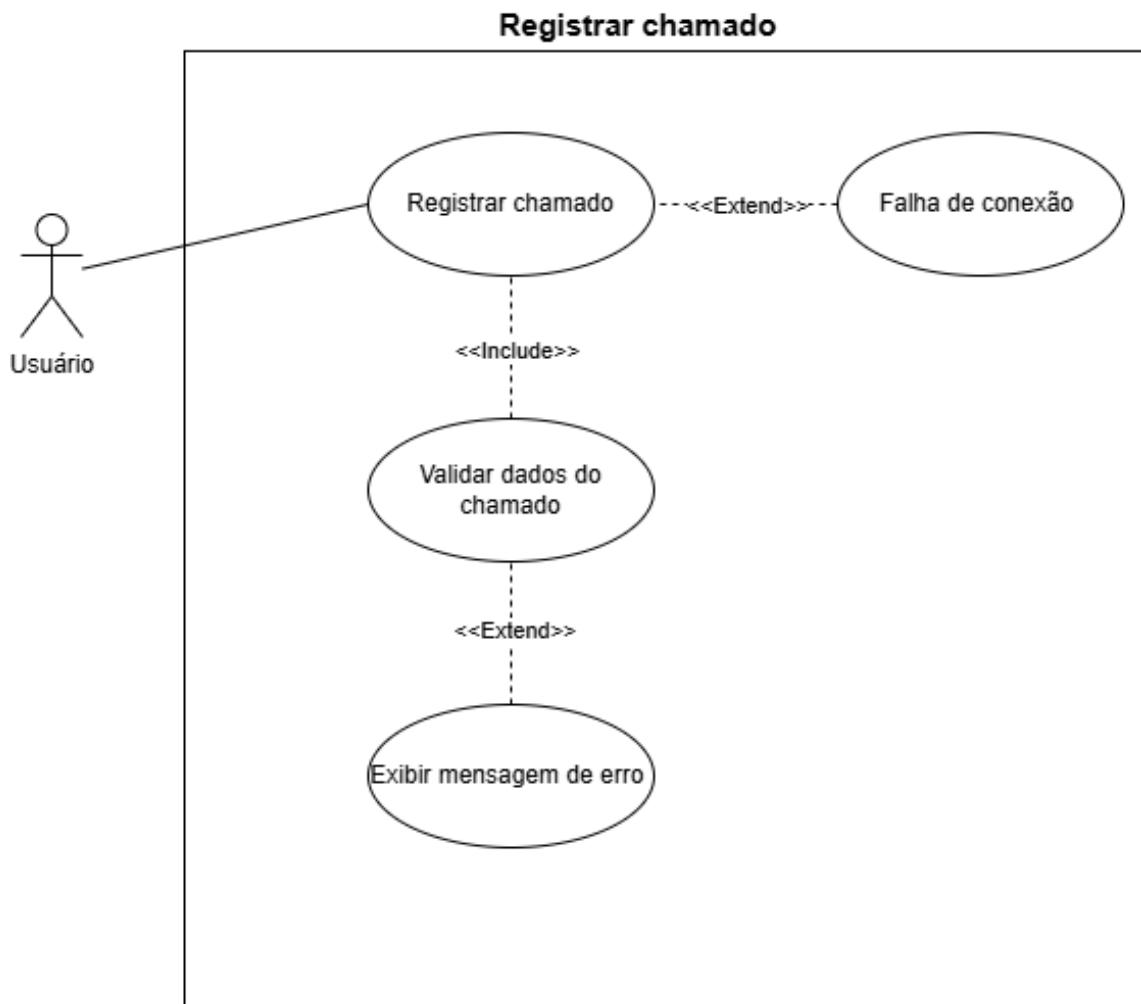
Fluxo alternativo: Dados inválidos Caso os dados estejam incorretos, o sistema exibe mensagem de erro e o usuário pode tentar novamente.

Fluxo de exceção: Falha de conexão com o servidor de autenticação o sistema exibe mensagem de erro e orienta o usuário a tentar mais tarde.

Pré-condição: O usuário precisa ter uma conta válida cadastrada no sistema.

Pós-condição: Usuário autenticado com sucesso e tem o acesso liberado conforme seu perfil.

### 2.2.2 Diagrama de Caso de Uso: Registrar Chamado



(DRAWIO, )

Atores: Usuário

Descrição: Permite que o usuário registre um chamado técnico através do aplicativo mobile, descrevendo o problema e anexando informações relevantes.

Fluxo principal:

O usuário acessa o aplicativo mobile e seleciona "Registrar chamado".

O sistema solicita o preenchimento de informações sobre o problema.

O usuário preenche os dados e anexa imagens (se necessário).

O usuário confirma o envio.

O sistema registra o chamado e encaminha para a IA para classificação e encaminhamento ao técnico.

Fluxo alternativo:

Dados incompletos.

O sistema alerta sobre campos obrigatórios não preenchidos.

O usuário preenche as informações faltantes.

Fluxo de exceção:

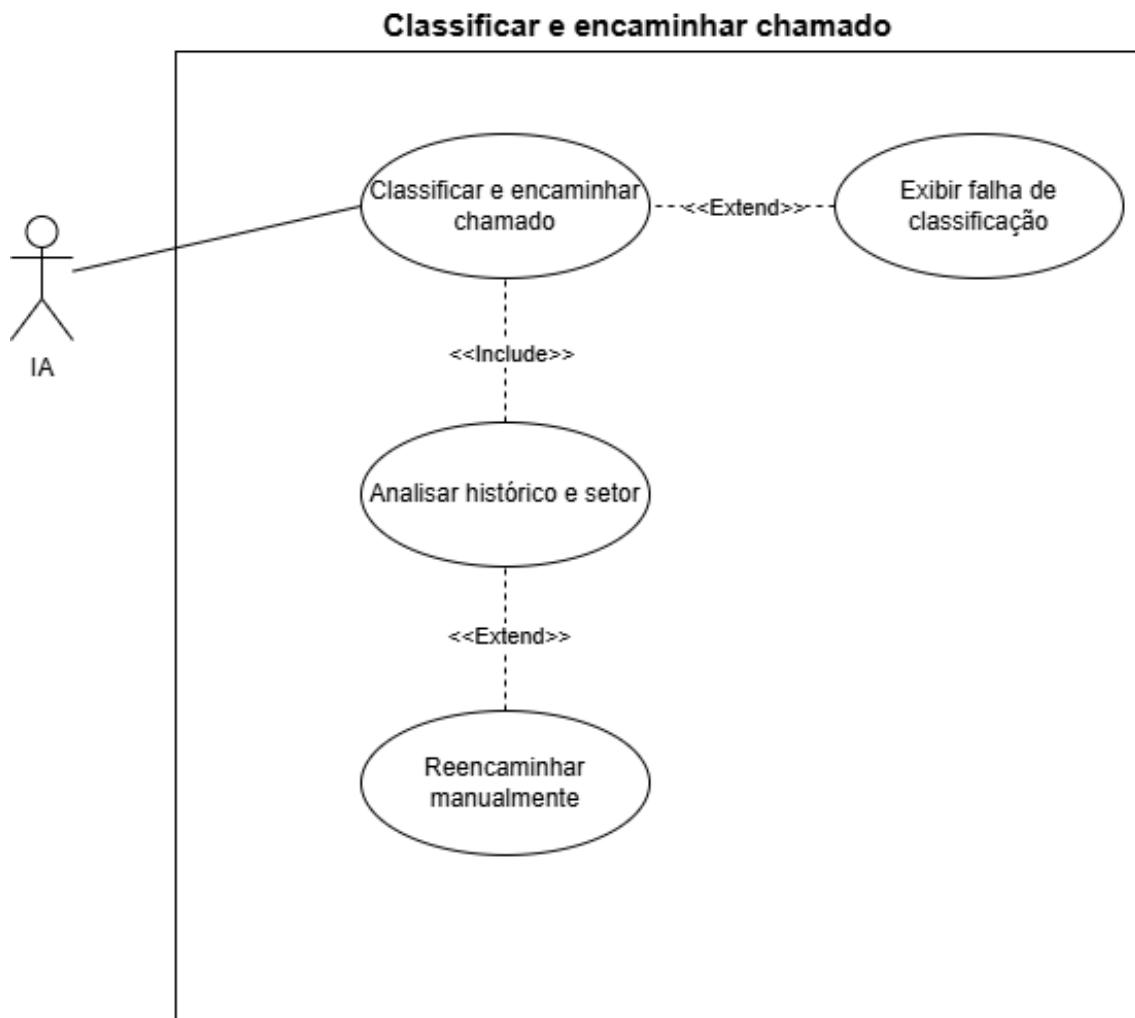
Falha no envio do chamado

O sistema exibe mensagem de erro e orienta o usuário a tentar novamente.

Pré-condição: Usuário autenticado e logado no aplicativo mobile.

Pós-condição: Chamado registrado e encaminhado para classificação pela IA.

### 2.2.3 Diagrama de Caso de Uso: Classificar e Encaminhar o chamado



Atores: IA

Descrição: Após o chamado ser registrado, a IA analisa o conteúdo do chamado e, com base em histórico e complexidade, encaminha automaticamente ao técnico mais adequado.

Fluxo principal:

Chamado é registrado no sistema.

A IA analisa o texto e metadados.

A IA identifica o técnico mais capacitado.

O sistema encaminha automaticamente o chamado ao técnico selecionado.

Fluxo alternativo:

Dúvida na classificação.

Caso a IA não encontre correspondência clara, o chamado é encaminhado para o administrador para ajuste manual.

Fluxo de Exceção:

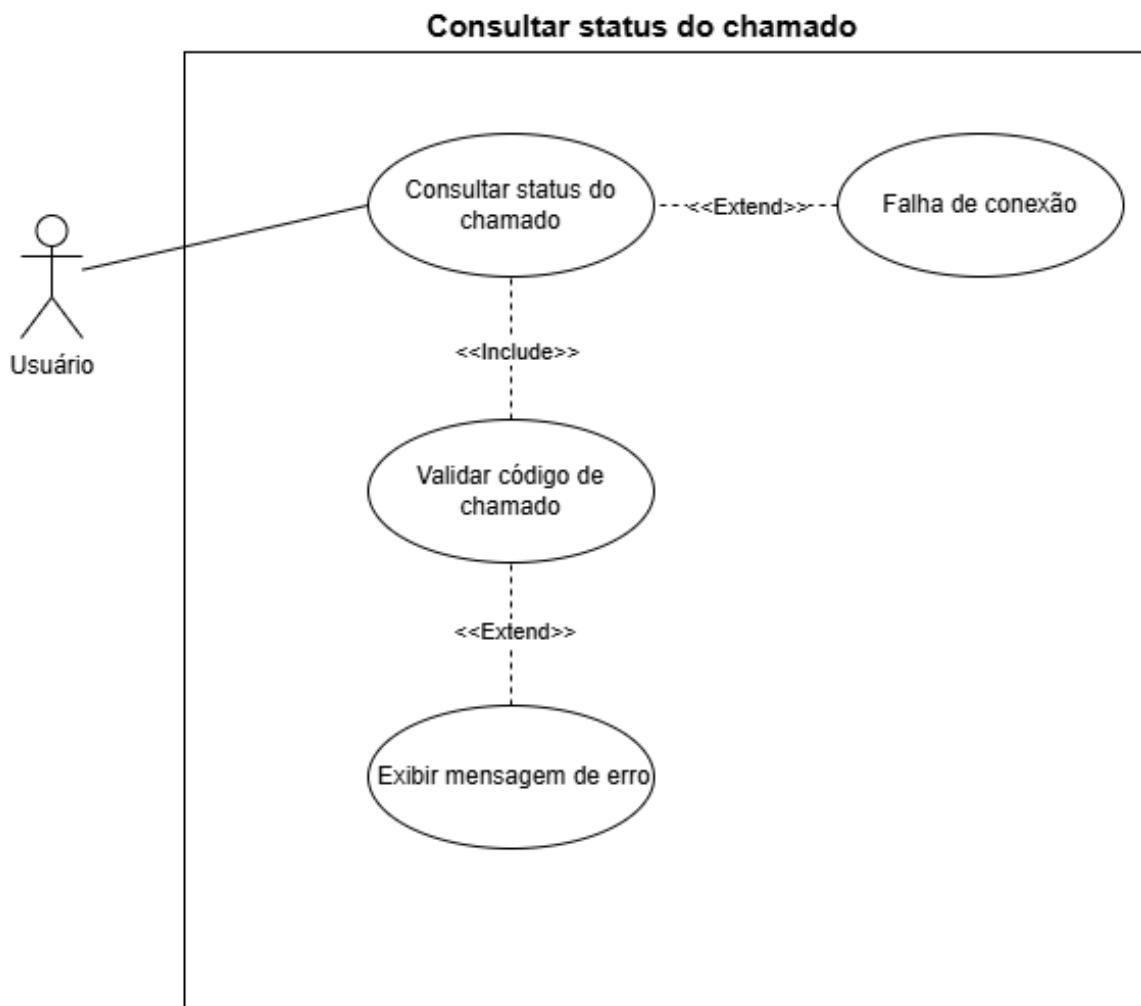
Falha na comunicação com o serviço de IA

O sistema salva o chamado e aguarda o serviço de IA restabelecer para processar novamente.

Pré-condição: Chamado registrado no sistema.

Pós-condição: Chamado atribuído ao técnico responsável.

#### 2.2.4 Diagrama de Caso de Uso: Consultar status do chamado



Atores: Usuário

Descrição: Permite ao usuário consultar, via aplicativo mobile, o status de seus chamados registrados.

Fluxo principal:

O usuário acessa a área "Meus Chamados".

O sistema exibe a lista de chamados com seus respectivos status.

O usuário pode visualizar detalhes de cada chamado. alternativo: Nenhum chamado encontrado

O sistema exibe mensagem indicando que não existem chamados.

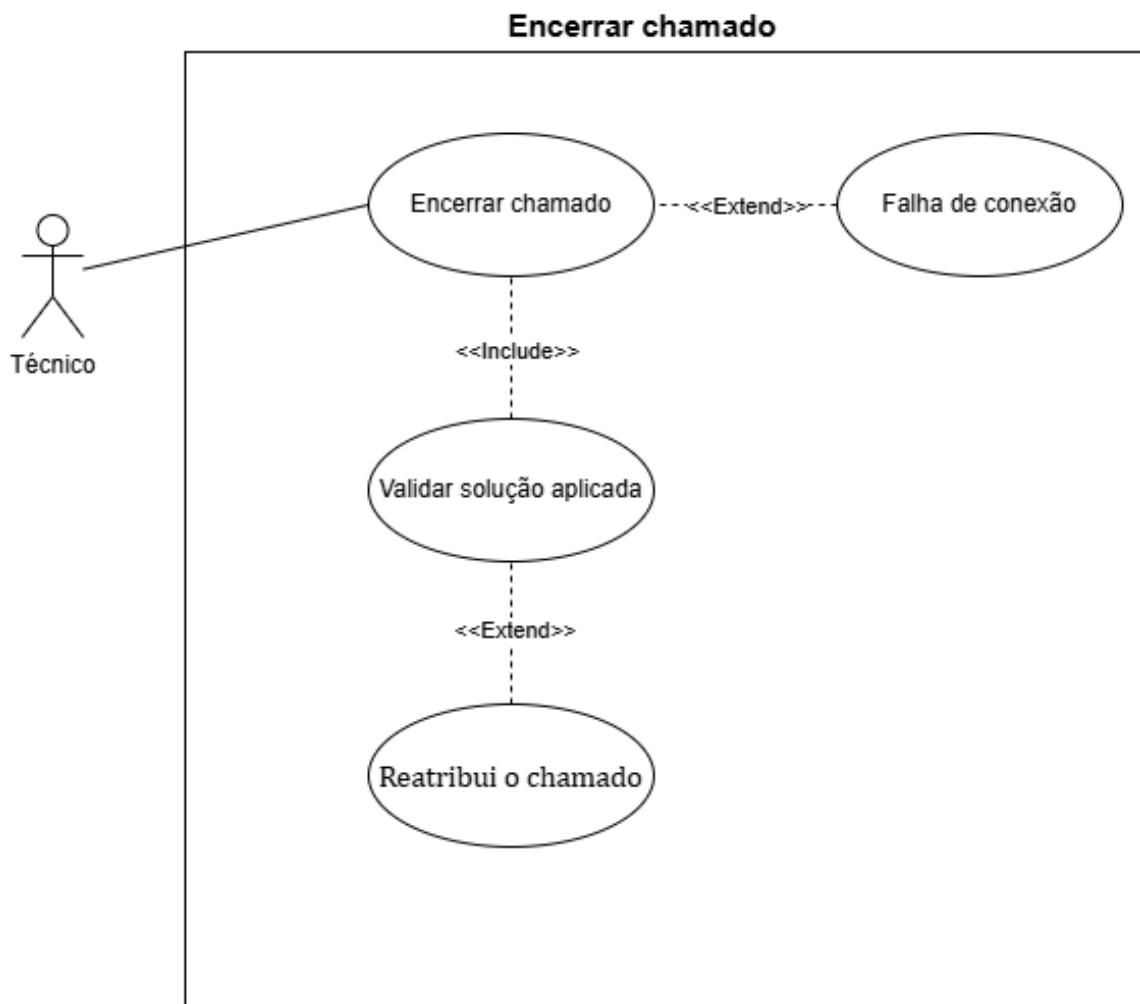
Fluxo de exceção:

Falha de conexão

O sistema exibe mensagem de erro e orienta o usuário a tentar novamente.

Pré-condição: Usuário autenticado w pós-condição: O usuário visualiza as informações de status do chamado.

## 2.2.5 Diagrama de Caso de Uso: Encerrar o chamado



Atores: Técnico

Descrição: Permite que o técnico encerre um chamado após a resolução do problema.

Fluxo principal:

O técnico acessa a lista de chamados atribuídos.

O técnico seleciona o chamado resolvido.

O técnico insere observações finais e conclui o encerramento.

O sistema registra o encerramento e notifica o usuário.

Fluxo alternativo:

Chamado não resolvido

O técnico informa a necessidade de suporte de outro setor e reatribui o chamado.

Fluxo de exceção:

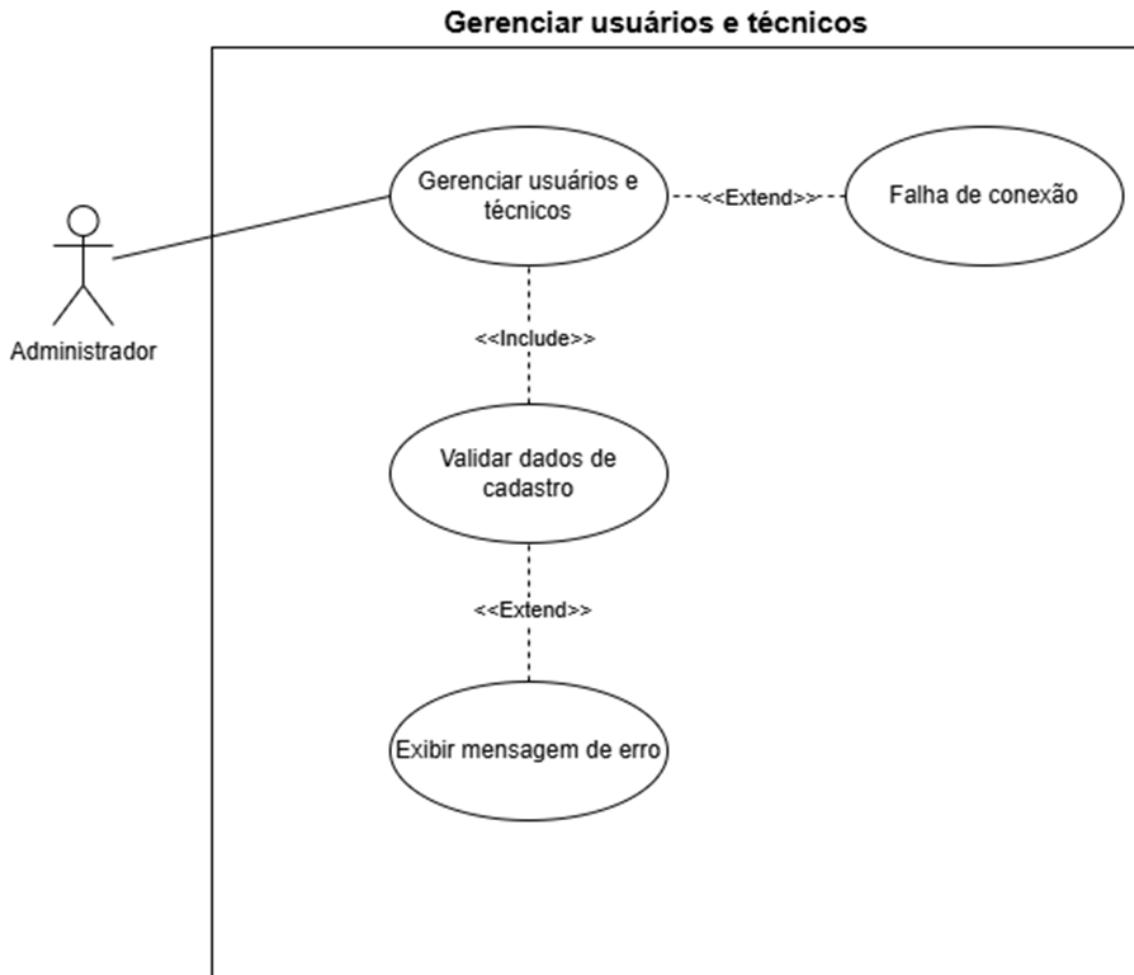
Falha ao encerrar o chamado

O sistema exibe mensagem de erro.

Pré-condição: Chamado atribuído ao técnico.

Pós-condição: Chamado encerrado e usuário notificado.

### 2.2.6 Diagrama de Caso de Uso: Gerenciar usuários e técnicos



Gerenciar usuários e técnicos

Atores: Administrador

Descrição: Permite ao administrador cadastrar, editar e excluir contas de usuários e técnicos.

Fluxo principal:

O administrador acessa o módulo de gestão de contas.

O administrador cria, edita ou remove perfis.

O sistema salva as alterações no banco de dados.

Fluxo alternativo:

Dados incompletos

O sistema alerta sobre campos obrigatórios não preenchidos.

Fluxo de exceção:

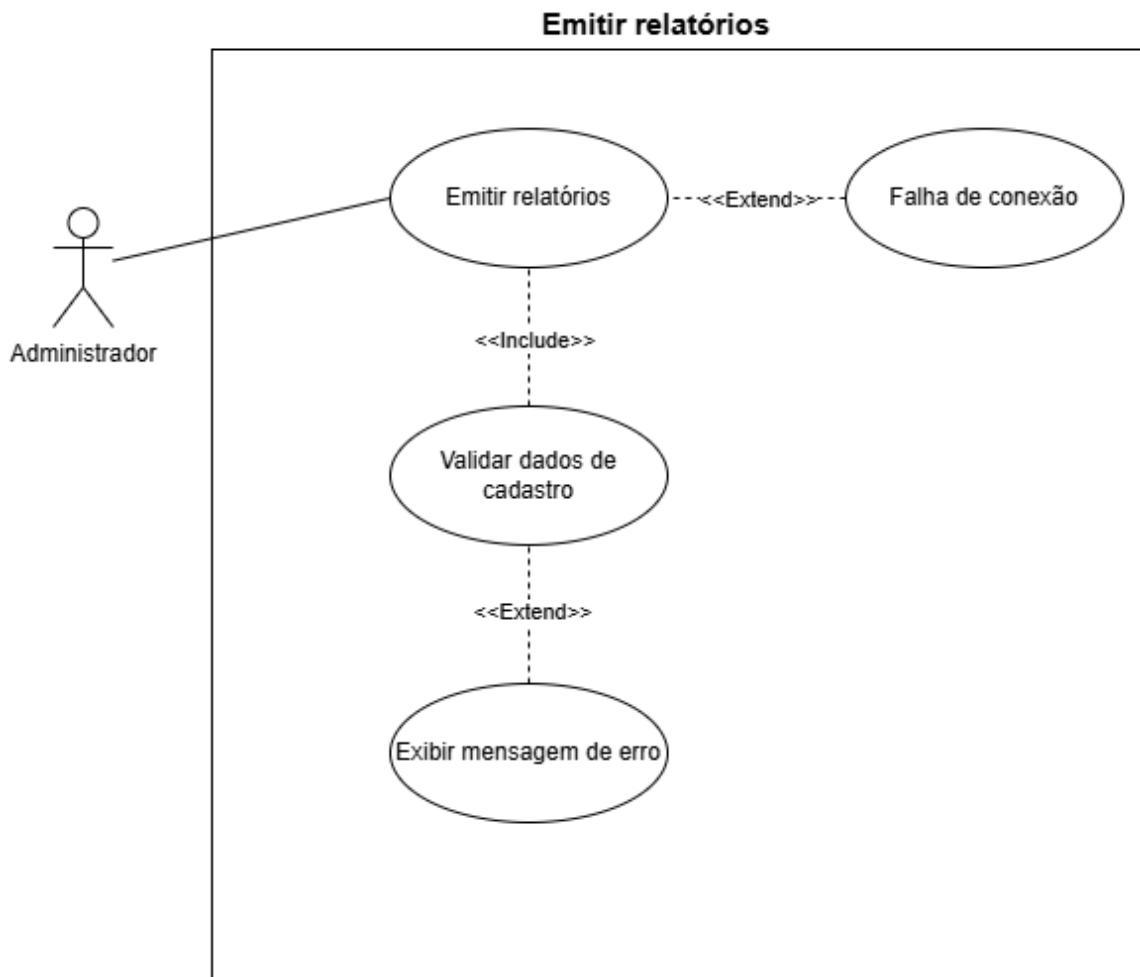
Falha na conexão com o banco de dados

O sistema exibe mensagem de erro.

Pré-condição: Administrador autenticado.

Pós-condição: Usuários e técnicos cadastrados e atualizados no sistema.

### 2.2.7 Diagrama de Caso de Uso: Emitir Relatório



Emitir relatórios

Atores: Administrador

Descrição: Permite ao administrador gerar relatórios detalhados de desempenho, tempo de resposta e volume de chamados.

Fluxo principal:

O administrador acessa o módulo de relatórios.

O sistema oferece filtros e critérios de consulta.

O administrador seleciona as opções desejadas.

O sistema gera e exibe o relatório.

Fluxo alternativo:

Nenhum dado disponível

O sistema informa que não há dados para o período selecionado.

Fluxo de exceção:

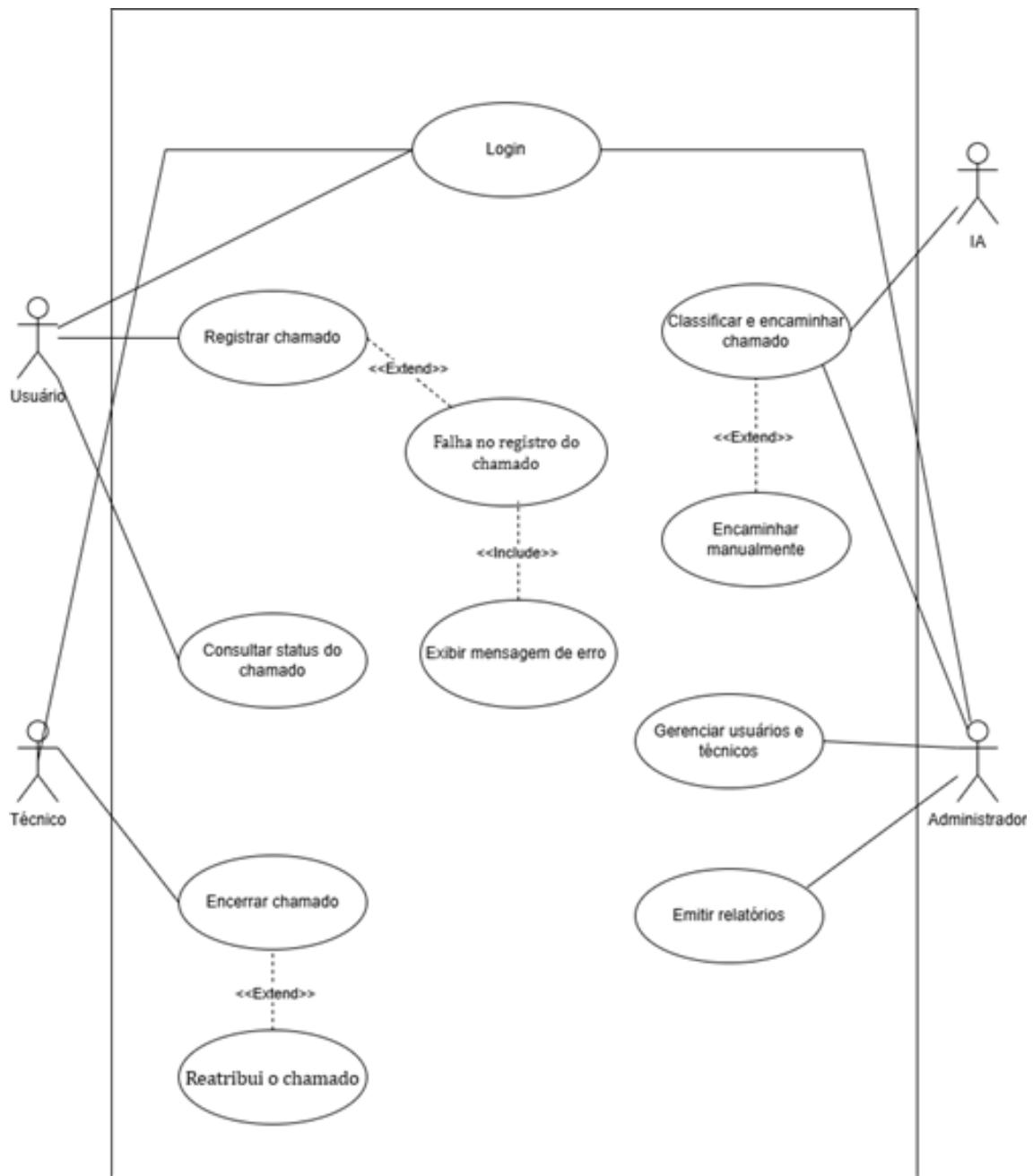
Falha ao gerar relatório

O sistema exibe mensagem de erro e orienta nova tentativa.

Pré-condição: Administrador autenticado.

Pós-condição: Relatório gerado e disponibilizado.

### 2.2.8 Diagrama de Caso de Uso: Gestão de Chamados Inteligente com Encaminhamento por IA



Atores: Usuário, IA, técnico e administrador

Descrição:

Permite que o usuário registre um chamado via aplicativo mobile.

O sistema utiliza Inteligência Artificial para analisar o conteúdo do chamado e encaminhá-lo automaticamente ao técnico mais adequado, considerando o histórico, setor e complexidade do problema.

O técnico realiza o atendimento e encerra o chamado. O usuário é notificado sobre a solução.

O administrador pode supervisionar, ajustar atribuições e gerar relatórios.

Fluxo Principal:

O usuário acessa o aplicativo e realiza o login.

O usuário registra o chamado.

O sistema armazena o chamado.

A IA analisa o conteúdo e classifica o chamado.

A IA encaminha o chamado ao técnico mais adequado.

O técnico resolve o problema e encerra o chamado.

O sistema notifica o usuário sobre a conclusão.

Fluxo alternativo:

Classificação Incerta

Se a IA não tiver certeza sobre o técnico adequado, o chamado é encaminhado ao administrador para atribuição manual.

Chamado reatribuído

O técnico pode reatribuir o chamado para outro técnico, caso identifique necessidade.

Fluxo de exceção:

Falha no Registro do Chamado

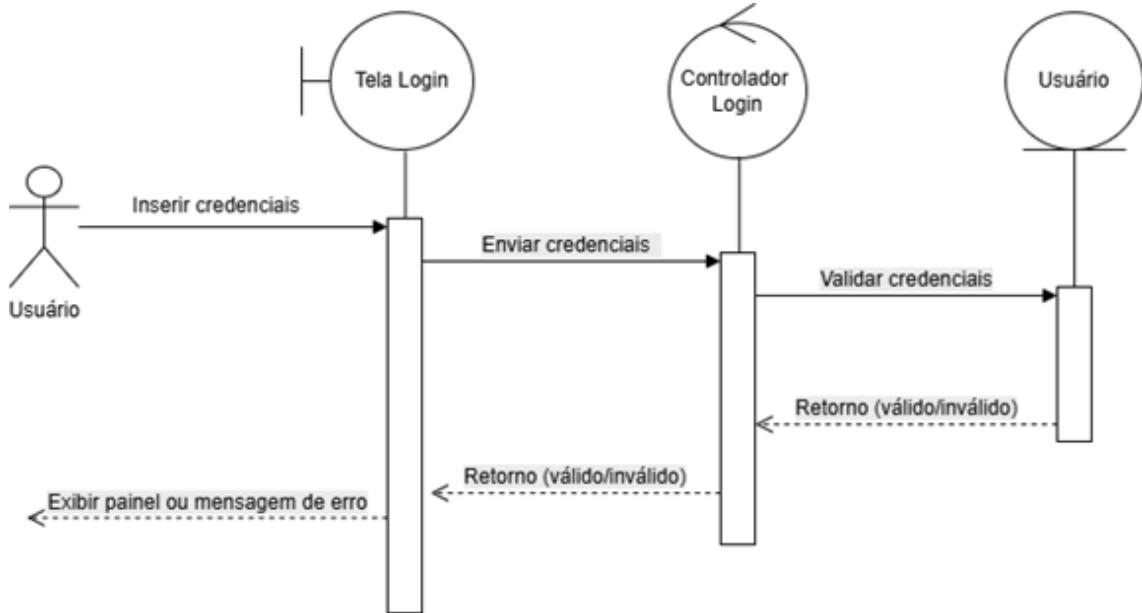
O sistema exibe uma mensagem de erro e orienta o usuário a tentar novamente.

Pré-condição: Usuário autenticado no sistema.

Pós-condição: Chamado classificado, atribuído ao técnico responsável e finalizado com notificação ao usuário.

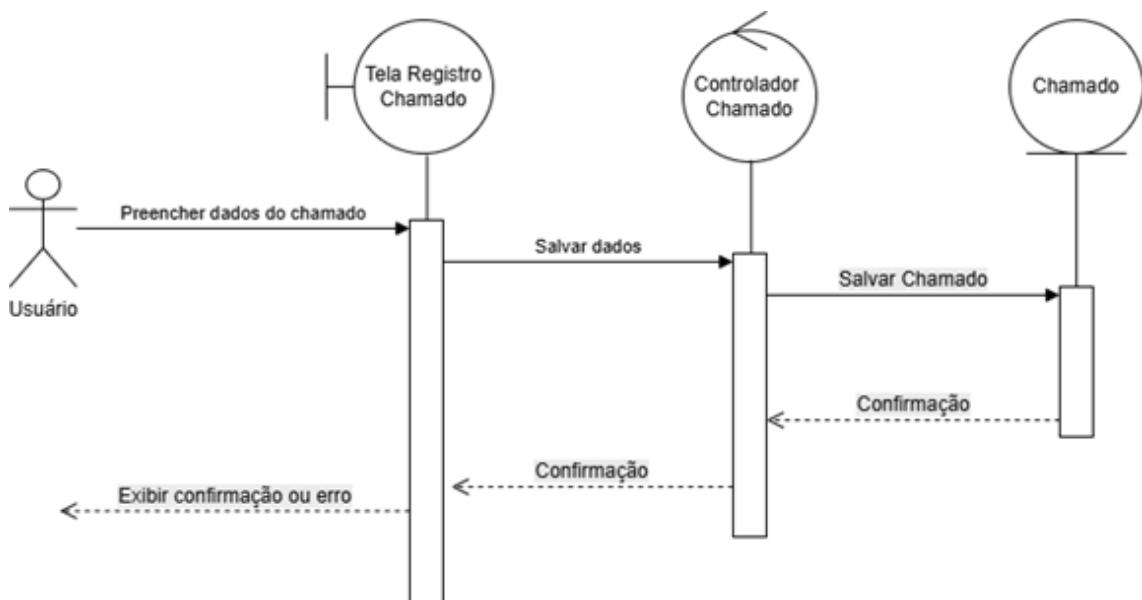
## 2.3 Diagramas de Sequência

### 2.3.1 Diagrama De Sequência: Login e Autenticação



Descrição: Este diagrama representa o fluxo de autenticação de um usuário. A interface de login coleta as credenciais e as envia para o controlador, que consulta a entidade de usuários para validar os dados. Em caso de sucesso, o sistema concede acesso conforme o perfil; caso contrário, exibe mensagens de erro ou falha de conexão.

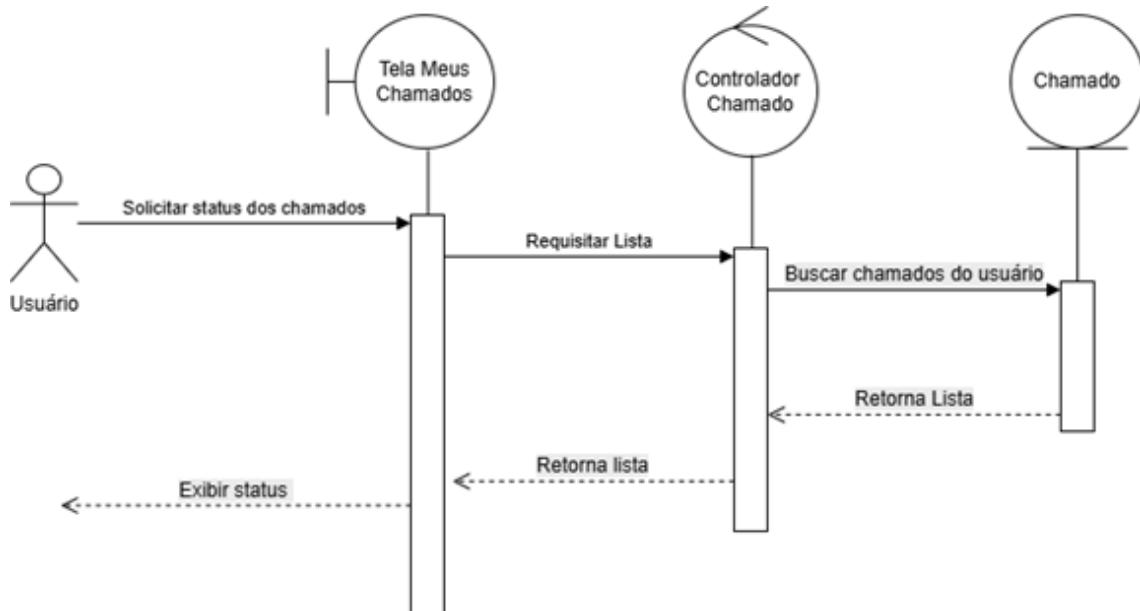
### 2.3.2 Diagrama De Sequência: Registro de Chamados



Descrição: Este diagrama ilustra o processo de registro de um chamado técnico

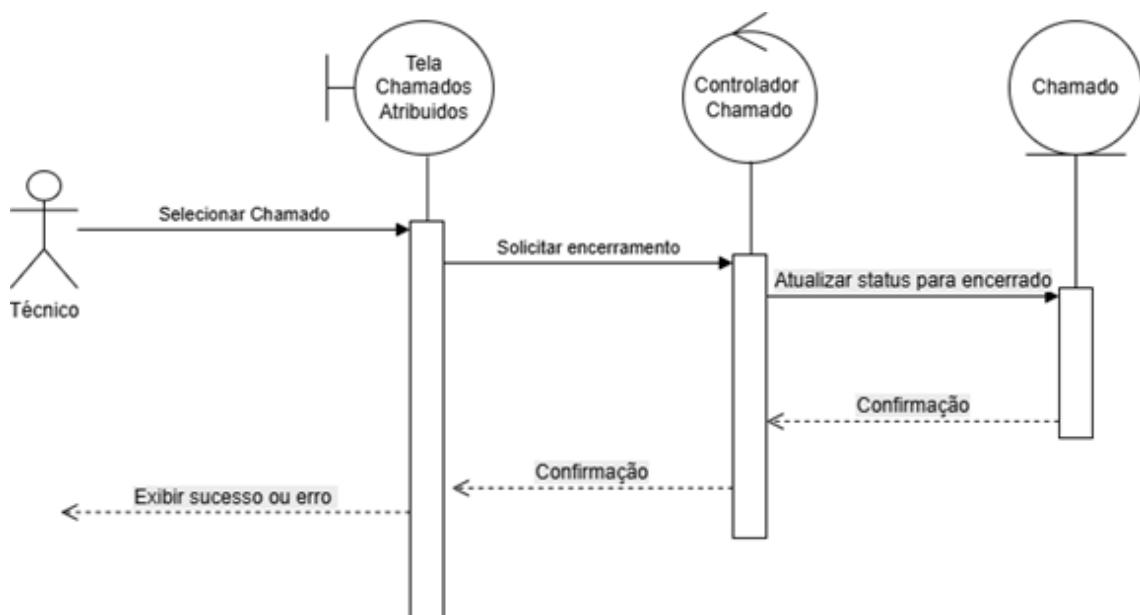
via aplicativo mobile. O usuário preenche os dados, e o controlador valida os dados, encaminhando-os à entidade de chamado. O sistema armazena o chamado e aciona a IA para classificação automática.

### 2.3.3 Diagrama De Sequência: Consultar Status de Chamados



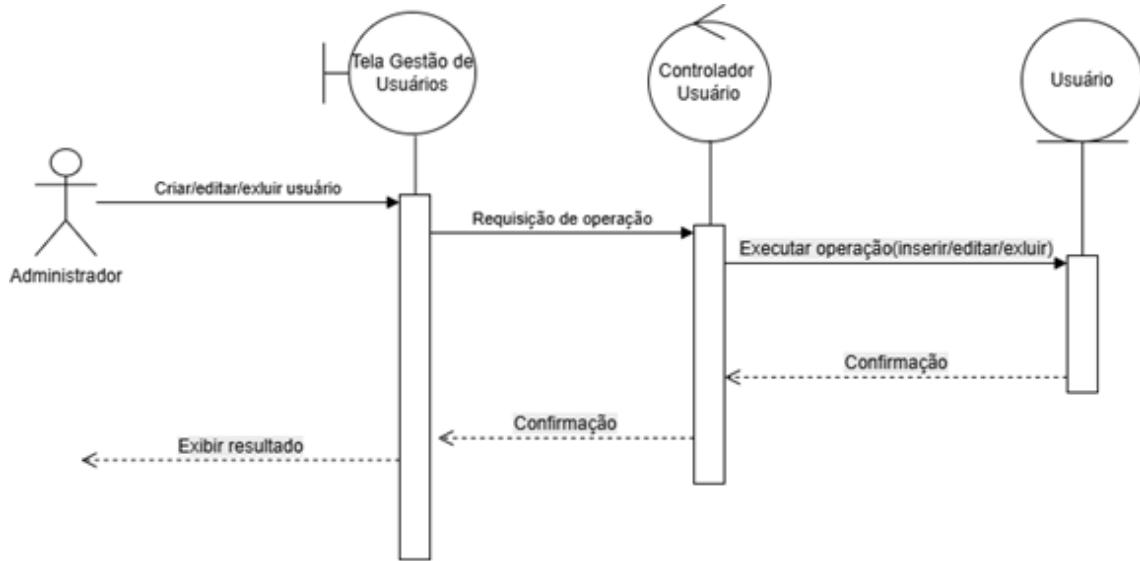
Descrição: Este diagrama descreve como o usuário verifica o status dos chamados registrados. A interface solicita a lista de chamados do controlador, que recupera os dados da entidade de chamado e retorna os detalhes ao usuário.

### 2.3.4 Diagrama De Sequência: Encerrar Chamados



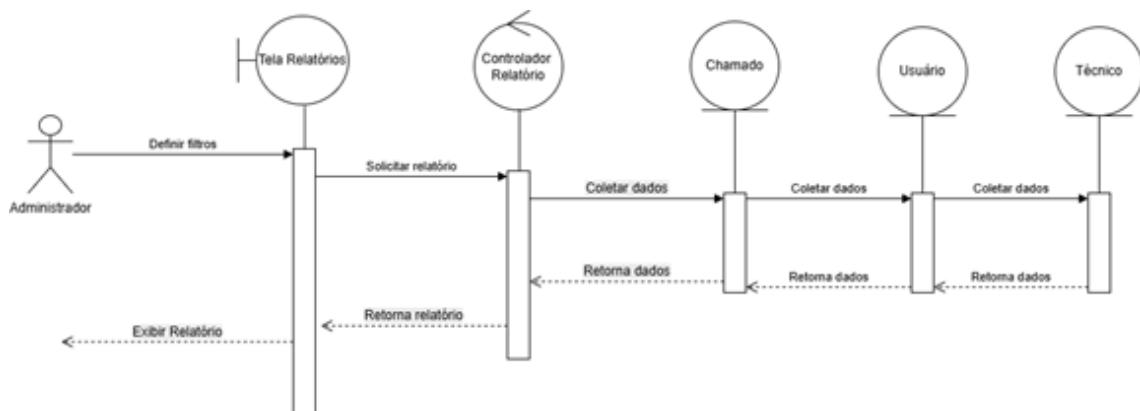
Descrição: Este diagrama mostra o fluxo em que o técnico encerra um chamado após resolver o problema. O técnico acessa os chamados atribuídos, seleciona o resolvido e envia os dados finais. O controlador atualiza a entidade de chamado e o sistema notifica o usuário.

### 2.3.5 Diagrama De Sequência: Gerenciar usuarios e técnicos



Descrição: Este diagrama descreve como o administrador gerencia perfis de usuários e técnicos. A interface administrativa envia comandos ao controlador, que consulta e atualiza a entidade de usuário conforme os dados inseridos ou alterados.

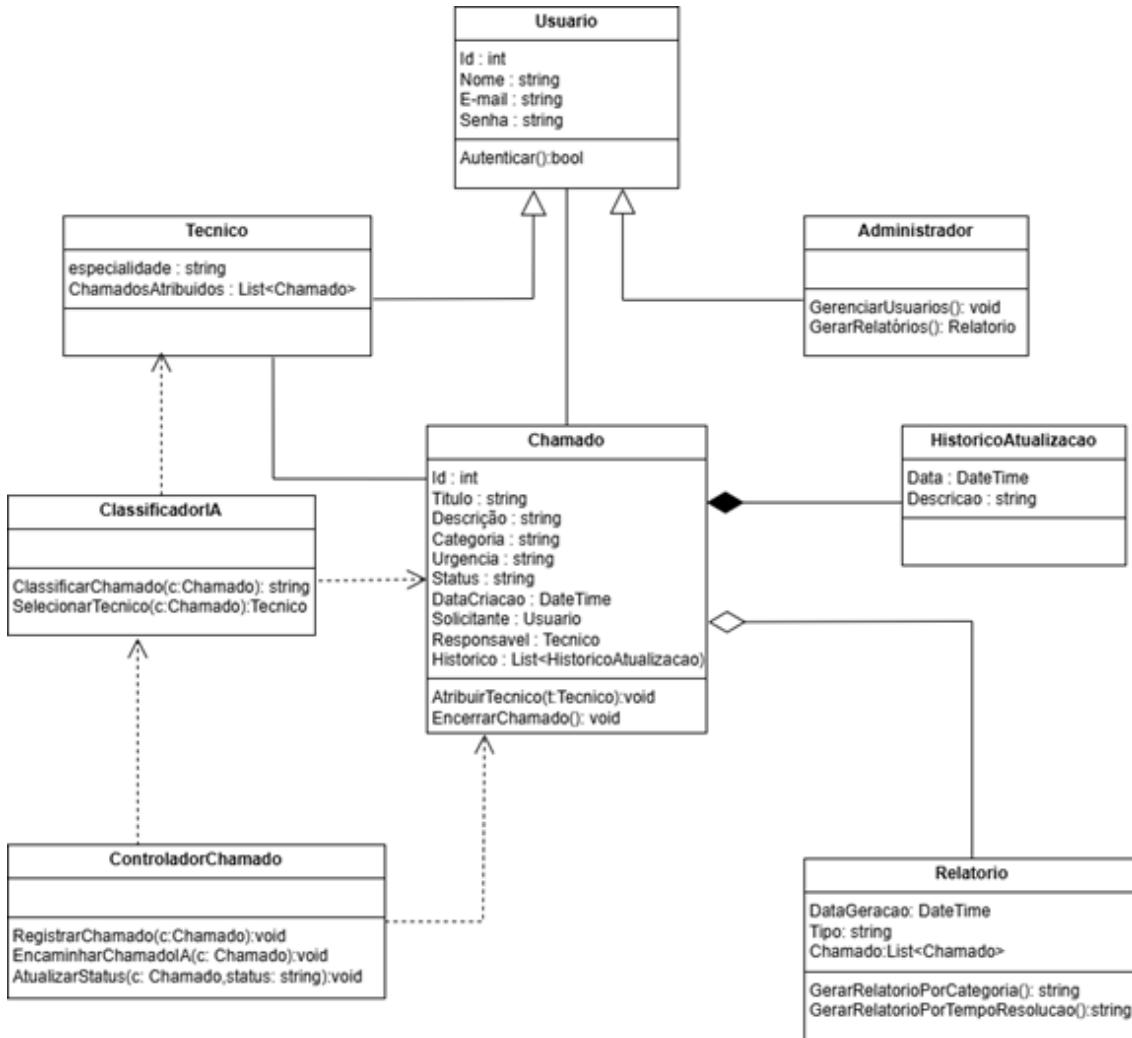
### 2.3.6 Diagrama De Sequência: Emitir Relatório



Descrição: Este diagrama mostra o processo de emissão de relatórios gerenciais. O administrador define os filtros, o controlador coleta os dados das entidades e o sistema gera o relatório com os resultados solicitados.

## 2.4 Diagrama de Classe

### 2.4.1 Diagrama De Classe



(DIAGRAMA, )

O diagrama de classes apresentado descreve a estrutura principal do sistema de gestão de chamados com suporte à inteligência artificial. Nele, estão representadas as principais classes envolvidas nas funcionalidades do sistema, como login, registro e encerramento de chamados, classificação automática por IA, acompanhamento de status, geração de relatórios e gestão de usuários.

Cada classe define seus atributos, que representam as informações que ela armazena, e seus métodos, que representam as ações ou comportamentos que ela executa. As relações entre as classes também foram modeladas para demonstrar como os elementos do sistema interagem entre si. Dentre os principais relacionamentos modelados, destacam-se:

Herança: Utilizada para indicar que Administrador e Técnico são especializações da classe Usuario, herdando seus atributos e comportamentos, mas podendo ter funcionalidades

específicas.

**Associação:** Indicada entre classes que precisam se comunicar, como Chamado associado a Usuario e Técnico, pois cada chamado possui um solicitante e um responsável.

**Composição:** Representada entre Chamado e HistoricoAtualizacao, indicando que um histórico de atualizações só existe dentro de um chamado.

**Dependência:** O ControladorChamado depende das classes Chamado e ClassificadorIA para gerenciar e classificar chamados, enquanto o ClassificadorIA depende das classes Chamado e Tecnico para realizar a análise e encaminhamento adequado do atendimento.

Esse diagrama oferece uma visão clara e estruturada dos elementos centrais do sistema, sendo essencial para o entendimento da lógica do projeto e servindo de base para a implementação em linguagem de programação orientada a objetos, como C(SHARP).

## 2.5 Diagrama de Implantação

### 2.5.1 Diagrama De Implantação



### 2.5.2 Estrutura do sistema

O sistema é composto por três interfaces principais:

Aplicativo mobile: Utilizado pelos usuários para registrar novos chamados.

Aplicativo web: Utilizado pelos técnicos para visualizar, atender e encerrar chamados.

Aplicativo desktop: Utilizado pelo administrador para gerenciar usuários, técnicos e relatórios.

Todos esses aplicativos se conectam ao servidor principal, onde estão instalados:

O sistema (backend).

O banco de dados (Microsoft SQL Server).

A comunicação entre os componentes ocorre através dos seguintes protocolos:

HTTP/HTTPS: Para a troca de informações entre os aplicativos e o servidor.

TCP/IP: Para garantir a conectividade entre os dispositivos na rede.

### 2.5.3 Configuração do Servidor

Para funcionar corretamente, o servidor deve conter:

Um sistema operacional compatível;

O Microsoft SQL Server instalado para armazenar os dados;

O backend do sistema instalado e configurado, preparado para receber e processar requisições dos usuários.

### 2.5.4 Instalação do Banco de Dados

No servidor, após a instalação do SQL Server, deve-se:

Criar ou importar a estrutura do banco de dados conforme o projeto do sistema;

Definir as tabelas, colunas e relacionamentos;

Configurar as permissões básicas de acesso para os usuários do sistema.

### 2.5.5 Instalação dos aplicativos

Após a configuração do servidor e banco de dados, os aplicativos devem ser instalados:

Aplicativo Mobile: Instalado em smartphones via loja de aplicativos ou pacote instalador (APK).

Aplicativo Web: Acessado diretamente via navegador de internet utilizando o endereço do servidor (via HTTPS).

Aplicativo Desktop: Instalado em máquinas administrativas usando um instalador apropriado.

Durante o processo de instalação ou no primeiro acesso, será necessário configurar os aplicativos com o endereço do servidor (IP ou domínio) para que possam se conectar corretamente.

### 2.5.6 Comunicação Entre Componentes

A comunicação entre os aplicativos e o servidor é feita por meio de:

HTTPS: Protocolo seguro para envio e recebimento de dados.

TCP/IP: Protocolo de rede que permite a conexão entre dispositivos da rede local (intranet) ou da internet.

### 2.5.7 Manutenção e Atualizações

Após a instalação e funcionamento do sistema, é necessário realizar manutenções periódicas, como:

Monitoramento do servidor;

Aplicação de atualizações de sistema;

Correções de erros;

## 2.6 Banco de Dados

### 2.6.1 Scripts do Banco de Dados

```
CREATE DATABASE SuporteTIv3;
```

```
USE SuporteTIv3;
```

```
CREATE TABLE TiposUsuarios (
    TipoID INT IDENTITY(1,1) PRIMARY KEY,
    NomeTipo NVARCHAR(50) NOT NULL UNIQUE
);
```

```
CREATE TABLE Usuarios (
    UsuarioID INT IDENTITY(1,1) PRIMARY KEY,
    Nome NVARCHAR(100) NOT NULL,
    Email NVARCHAR(100) NOT NULL UNIQUE,
    Telefone NVARCHAR(20),
    TipoUsuarioID INT NOT NULL,
    FOREIGN KEY (TipoUsuarioID) REFERENCES TiposUsuarios(TipoID)
);
```

GO

```
CREATE TABLE Tecnico (
    UsuarioID INT PRIMARY KEY,
    Especialidade NVARCHAR(100),
    FOREIGN KEY (UsuarioID) REFERENCES Usuarios(UsuarioID)
);
```

GO

```
CREATE TABLE Administrador (
    UsuarioID INT PRIMARY KEY,
    FOREIGN KEY (UsuarioID) REFERENCES Usuarios(UsuarioID)
);
```

GO

```
CREATE TABLE Categorias (
```

```
CategoriaID INT IDENTITY(1,1) PRIMARY KEY,  
NomeCategoria NVARCHAR(100) NOT NULL UNIQUE  
);  
GO
```

```
CREATE TABLE Chamado (  
Id INT IDENTITY(1,1) PRIMARY KEY,  
Titulo NVARCHAR(200) NOT NULL,  
Descricao NVARCHAR(MAX),  
CategoriaID INT,  
Urgencia NVARCHAR(50),  
Status NVARCHAR(50),  
DataCriacao DATETIME DEFAULT GETDATE(),  
SolicitanteId INT NOT NULL,  
ResponsavelId INT,  
FOREIGN KEY (CategoriaID) REFERENCES Categorias(CategoriaID),  
FOREIGN KEY (SolicitanteId) REFERENCES Usuarios(UsuarioID),  
FOREIGN KEY (ResponsavelId) REFERENCES Tecnico(UsuarioID)  
);  
GO
```

```
CREATE TABLE HistoricoAtualizacao (  
Id INT IDENTITY(1,1) PRIMARY KEY,  
Data DATETIME DEFAULT GETDATE(),  
Descricao NVARCHAR(MAX),
```

```
ChamadoId INT NOT NULL,  
FOREIGN KEY (ChamadoId) REFERENCES Chamado(Id)  
);  
GO
```

```
CREATE TABLE Relatorio (  
Id INT IDENTITY(1,1) PRIMARY KEY,  
DataGeracao DATETIME DEFAULT GETDATE(),  
Tipo NVARCHAR(50)  
);  
GO
```

```
CREATE TABLE ChamadoRelatorio(ChamadoId INT,  
RelatorioId INT,  
PRIMARY KEY (ChamadoId, RelatorioId),  
FOREIGN KEY (ChamadoId) REFERENCES Chamado(Id),  
FOREIGN KEY (RelatorioId) REFERENCES Relatorio(Id)  
);  
GO
```

## 2.6.2 Scripts de Testes

```
INSERT INTO TiposUsuarios (NomeTipo) VALUES  
(‘Administrador’),  
(‘Tecnico’),  
(‘Solicitante’);
```

```
INSERT INTO Usuarios (Nome, Email, Telefone, TipoUsuarioID) VALUES  
(‘João Silva’, ‘joao@email.com’, ‘11999999999’, 1),  
(‘Maria Souza’, ‘maria@email.com’, ‘11988888888’, 2),  
(‘Carlos Pereira’, ‘carlos@email.com’, ‘11977777777’, 3);
```

```
INSERT INTO Tecnico (UsuarioID, Especialidade) VALUES  
(2, ‘Suporte de Rede’);
```

```
INSERT INTO Administrador (UsuarioID) VALUES  
(1);
```

```
INSERT INTO Categorias (NomeCategoria) VALUES  
(‘Hardware’),  
(‘Software’),  
(‘Rede’);
```

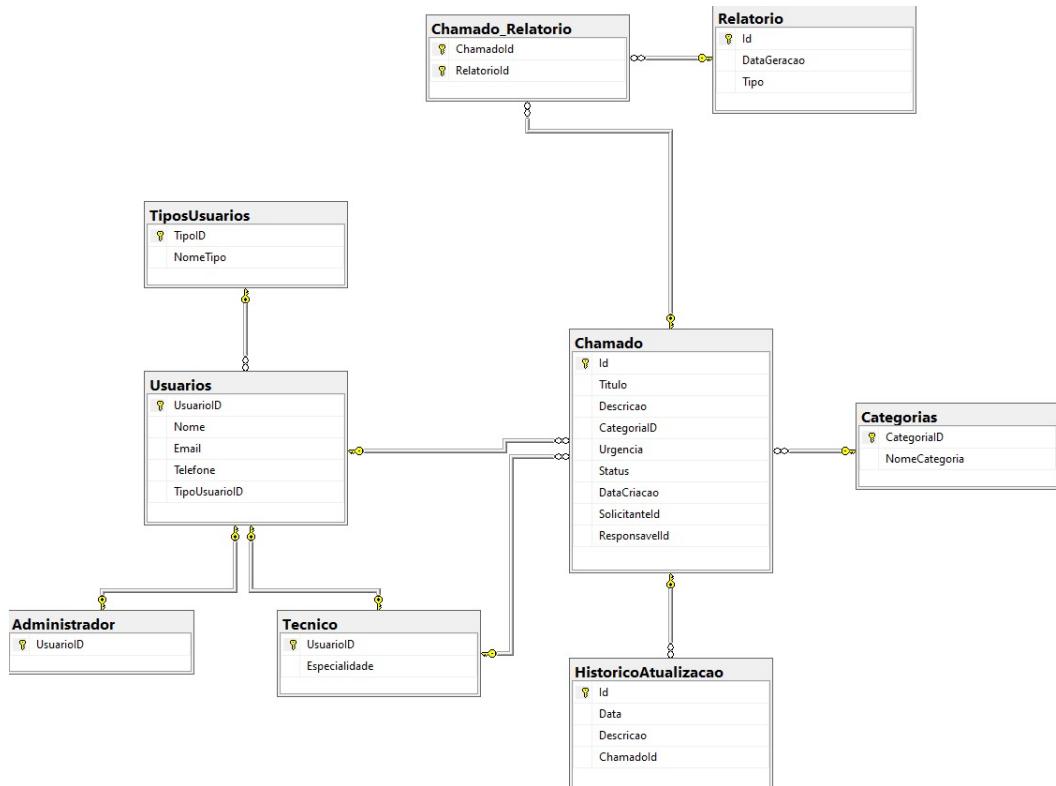
```
INSERT INTO Chamado (Titulo, Descricao, CategoriaID, Urgencia, Status, SolicitanteId,  
ResponsavelId) VALUES  
(‘Computador não liga’, ‘O computador do escritório não está ligando.’, 1, ‘Alta’, ‘Aberto’,  
3, 2);
```

```
INSERT INTO HistoricoAtualizacao (Descricao, ChamadoId) VALUES  
(‘Chamado aberto pelo solicitante.’, 1);
```

```
INSERT INTO Relatorio (Tipo) VALUES (‘Resumo Semanal’);
```

```
INSERT INTO ChamadoRelatorio(ChamadoId, RelatorioId) VALUES(1, 1);
```

### 2.6.3 Diagrama Entidade e Relacionamento



(MODELO..., )

#### 2.6.4 Dicionário de dados: Tabela Tipos Usuarios

Campo	Tipo	Descrição
TipolD	INT (PK, IDENTITY)	Identificador único do tipo de usuário, auto-incrementado.
NomeTipo	NVARCHAR(50)	Nome do tipo de usuário (ex: Administrador, Técnico, Solicitante). Deve ser único.

### 2.6.5 Dicionário de dados: Tabela Usuarios

Campo	Tipo	Descrição
UsuarioID	INT (PK, IDENTITY)	Identificador único do usuário, auto-incrementado.
Nome	NVARCHAR(100)	Nome completo do usuário.
Email	NVARCHAR(100)	Email do usuário, deve ser único.
Telefone	NVARCHAR(20)	Número de telefone do usuário (opcional).
TipoUsuarioID	INT (FK)	Chave estrangeira para TiposUsuarios.TipoID, indica o tipo do usuário.

### 2.6.6 Dicionario de dados: Tabela Técnico

Campo	Tipo	Descrição
UsuarioID	INT (PK, FK)	Identificador do usuário que é técnico, referencia Usuarios.UsuarioID.
Especialidade	NVARCHAR(100)	Área de especialidade do técnico (ex: Suporte de Rede).

### 2.6.7 Dicionario de dados: Tabela Administrador

Campo	Tipo	Descrição
UsuarioID	INT (PK, FK)	Identificador do usuário que é administrador, referencia Usuarios.UsuarioID.

### 2.6.8 Dicionário de dados: Tabela Categoria

Campo	Tipo	Descrição
CategoriaID	INT (PK, IDENTITY)	Identificador único da categoria, auto-incrementado.
NomeCategoria	NVARCHAR(100)	Nome da categoria (ex: Hardware, Software, Rede). Deve ser único.

### 2.6.9 Dicionário de dados: Tabela Chamado

Campo	Tipo	Descrição
Id	INT (PK, IDENTITY)	Identificador único do chamado, auto-incrementado.
Titulo	NVARCHAR(200)	Título ou assunto do chamado.
Descricao	NVARCHAR(MAX)	Descrição detalhada do problema ou solicitação.
CategorialD	INT (FK)	Referência à categoria do chamado (Categorias.CategorialD).
Urgencia	NVARCHAR(50)	Grau de urgência (ex: Alta, Média, Baixa).

### 2.6.10 Dicionario de dados: Continuação da Tabela Chamado

Status	NVARCHAR(50)	Status atual do chamado (ex: Aberto, Fechado).
DataCriacao	DATETIME	Data e hora da criação do chamado, padrão GETDATE().
SolicitanteId	INT (FK)	Usuário solicitante do chamado (Usuarios.UsuarioID).
ResponsavelId	INT (FK)	Técnico responsável pelo chamado (Tecnico.UsuarioID). Pode ser NULL.

### 2.6.11 Dicionario de dados: Histórico de Atualização

Campo	Tipo	Descrição
Id	INT (PK, IDENTITY)	Identificador único da atualização, auto-incrementado.
Data	DATETIME	Data e hora da atualização, padrão GETDATE().
Descricao	NVARCHAR(MAX)	Descrição da atualização feita no chamado.
Chamadold	INT (FK)	Referência ao chamado atualizado (Chamado.Id).

### 2.6.12 Dicionario de dados: Relatório

Campo	Tipo	Descrição
Id	INT (PK, IDENTITY)	Identificador único do relatório, auto-incrementado.
DataGeracao	DATETIME	Data e hora em que o relatório foi gerado, padrão GETDATE().
Tipo	NVARCHAR(50)	Tipo ou título do relatório (ex: Resumo Semanal).

### 2.6.13 Dicionario de dados: Chamado Relatório

Campo	Tipo	Descrição
Chamadoid	INT (PK, FK)	Identificador do chamado, referencia Chamado.Id.
RelatorioId	INT (PK, FK)	Identificador do relatório, referencia Relatorio.Id.

### 2.6.14 Planilha de Teste

Planilha de teste é um documento usado para registrar e acompanhar testes feitos em sistemas ou processos, mostrando o que foi testado, os resultados esperados, os resultados obtidos e o status do teste.

Query SQL	Descrição da Ação	Etapa	Resultado Esperado
<pre>INSERT INTO TiposUsuarios (NomeTipo) VALUES ('Administrador'), ('Tecnico'), ('Solicitante');</pre>	Inserir tipos de usuários padrão para categorizar os perfis de acesso.	Configuração Inicial	3 registros adicionados na tabela TiposUsuarios com os nomes especificados.
<pre>INSERT INTO Usuarios (Nome, Email, Telefone, TipoUsuarioID) VALUES ('João Silva', 'joao@email.com', '11999999999', 1), ('Maria Souza', 'maria@email.com', '11988888888', 2), ('Carlos Pereira', 'carlos@email.com', '11977777777', 3);</pre>	Adicionar usuários iniciais ao sistema, vinculando-os aos seus tipos.	População Inicial	3 usuários cadastrados, cada um associado a um TipoUsuarioID válido.
<pre>INSERT INTO Tecnico (UsuarioID, Especialidade) VALUES (2, 'Suporte de Rede');</pre>	Registrar o usuário com ID 2 como técnico, com sua especialidade.	População Inicial	Um registro na tabela Tecnico com o UsuarioID 2 e especialidade definida.
<pre>INSERT INTO Administrador (UsuarioID) VALUES (1);</pre>	Registrar o usuário com ID 1 como administrador do sistema.	População Inicial	Um registro na tabela Administrador com o UsuarioID 1.
<pre>INSERT INTO Categorias (NomeCategoria) VALUES ('Hardware'), ('Software'), ('Rede');</pre>	Adicionar categorias para classificação dos chamados.	Configuração Inicial	3 categorias adicionadas na tabela Categorias.
<pre>INSERT INTO Chamado (Titulo, Descricao, CategoriaID, Urgencia, Status, SolicitanteId, ResponsavelId) VALUES ('Computador não liga', 'O computador do escritório não está ligando.', 1, 'Alta', 'Aberto', 3, 2); INSERT INTO HistoricoAtualizacao (Descricao, ChamadoId) VALUES ('Chamado aberto pelo solicitante.', 1);</pre>	Criar um chamado com detalhes do problema e associação ao solicitante e técnico responsável.	Teste funcional	Chamado criado com status 'Aberto' e vinculado às categorias e usuários corretos.
<pre>INSERT INTO Relatorio (Tipo) VALUES ('Resumo Semanal');</pre>	Criar um tipo de relatório padrão para geração de resumos semanais.	Configuração Inicial	Registro criado na tabela Relatorio com o tipo 'Resumo Semanal'.
<pre>INSERT INTO ChamadoRelatorio (ChamadoId, RelatorioId) VALUES (1, 1);</pre>	Associar o chamado 1 ao relatório 1 para incluir no resumo semanal.	Relacionamento	Registro criado na tabela de associação ChamadoRelatorio.
<pre>SELECT * FROM Chamado WHERE Status = 'Aberto';</pre>	Listar todos os chamados que estão com status aberto.	Consulta para validação	Retornar os chamados cujo status seja exatamente 'Aberto'.

## 2.7 Interface Humano Computador

### 2.7.1 Protótipo de Tela: Mobile



Tela em que o usuário vai entrar preencher os campos e acessar o aplicativo

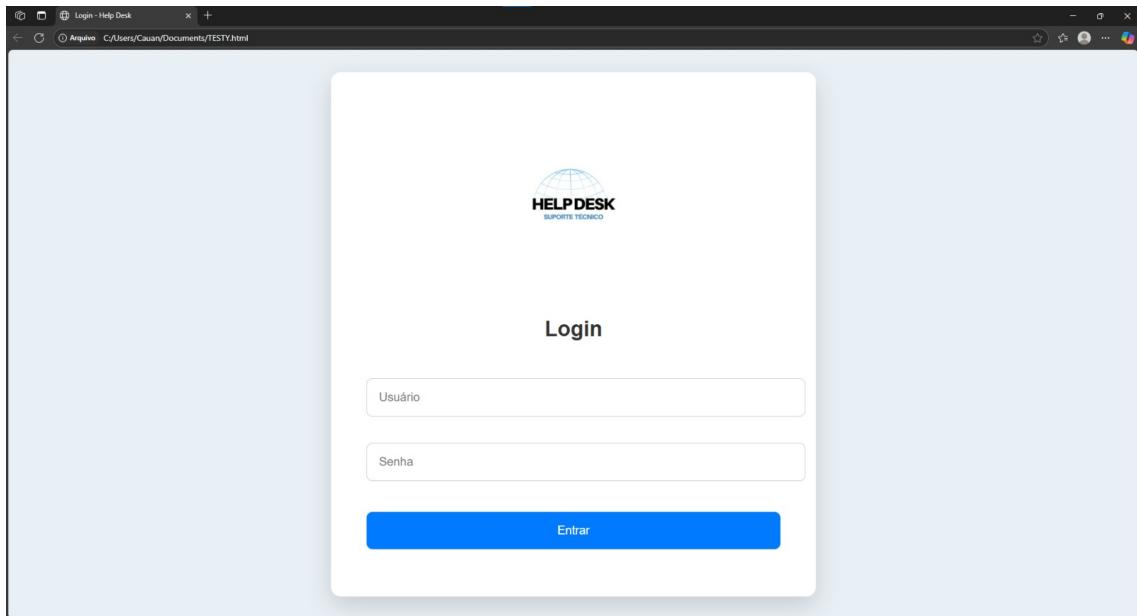


Tela em que o usuário vai inserir o problema e a prioridade e logo após será encaminhado para o especialista



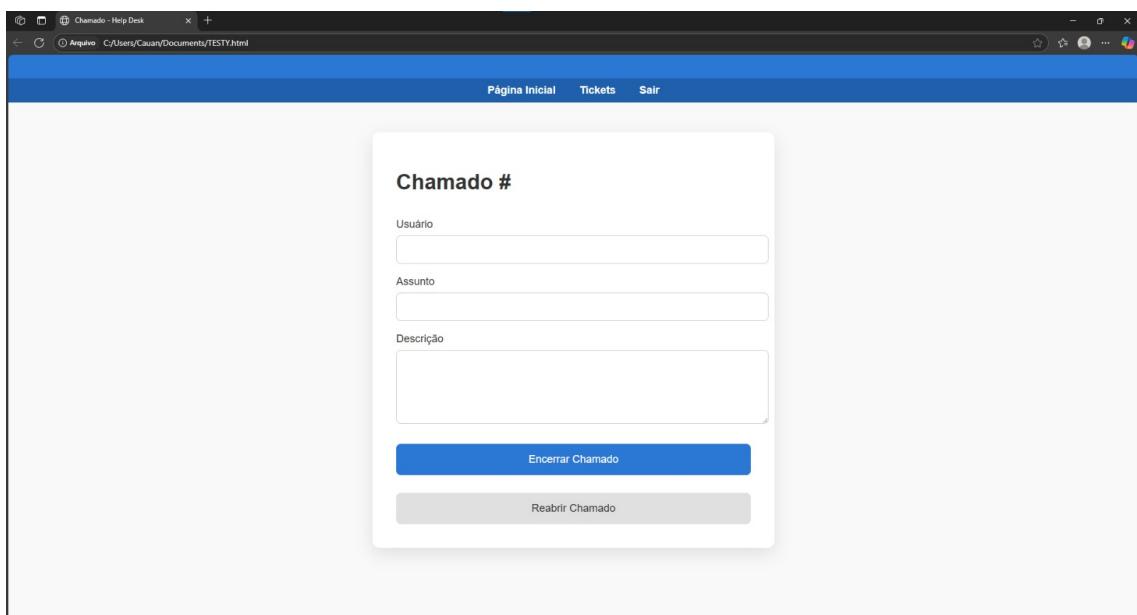
Tela que Registra os chamados feitos e mostra o andamento do chamado pelo usuário como forma de histórico

### 2.7.2 Protótipo de Tela: Web



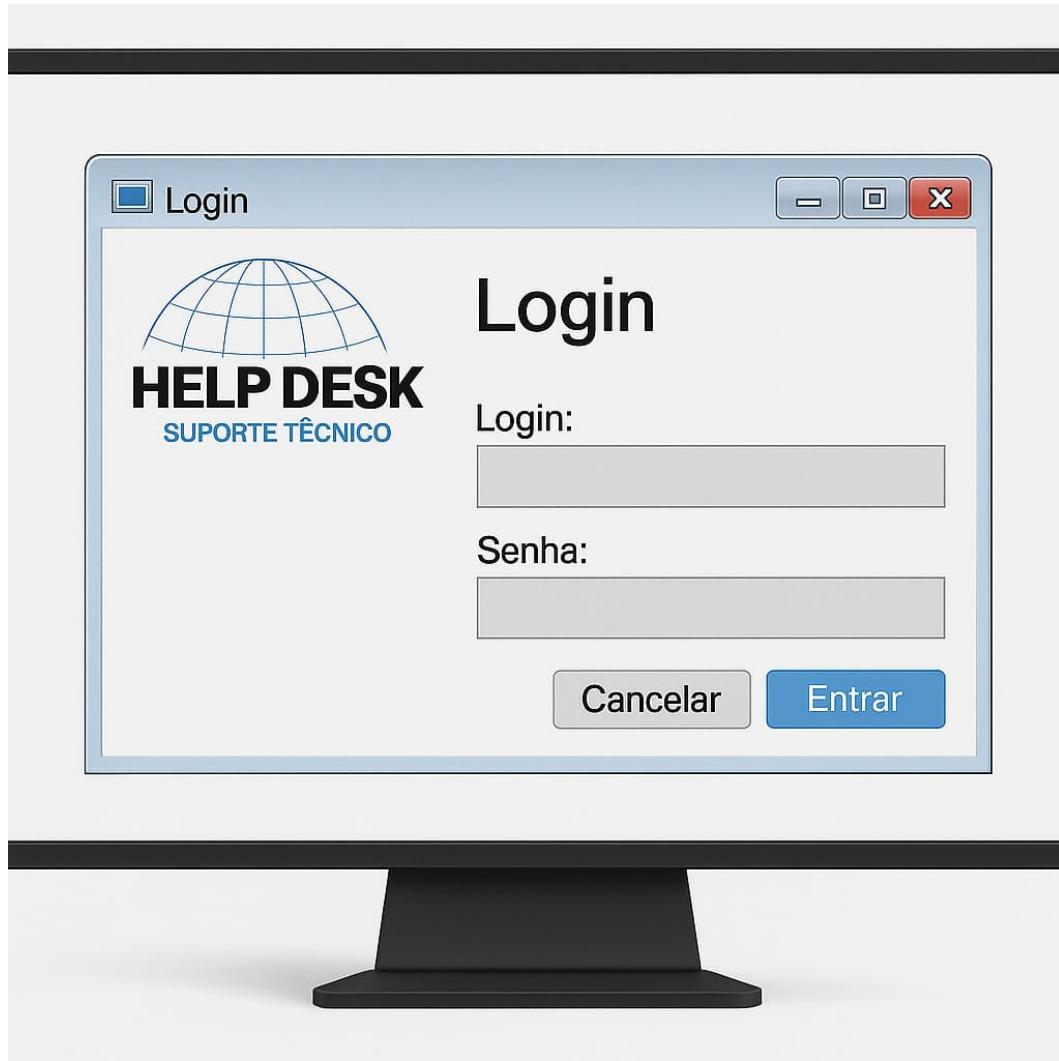
Tela onde os técnicos realizam login

### 2.7.3 Protótipo de Tela: Web

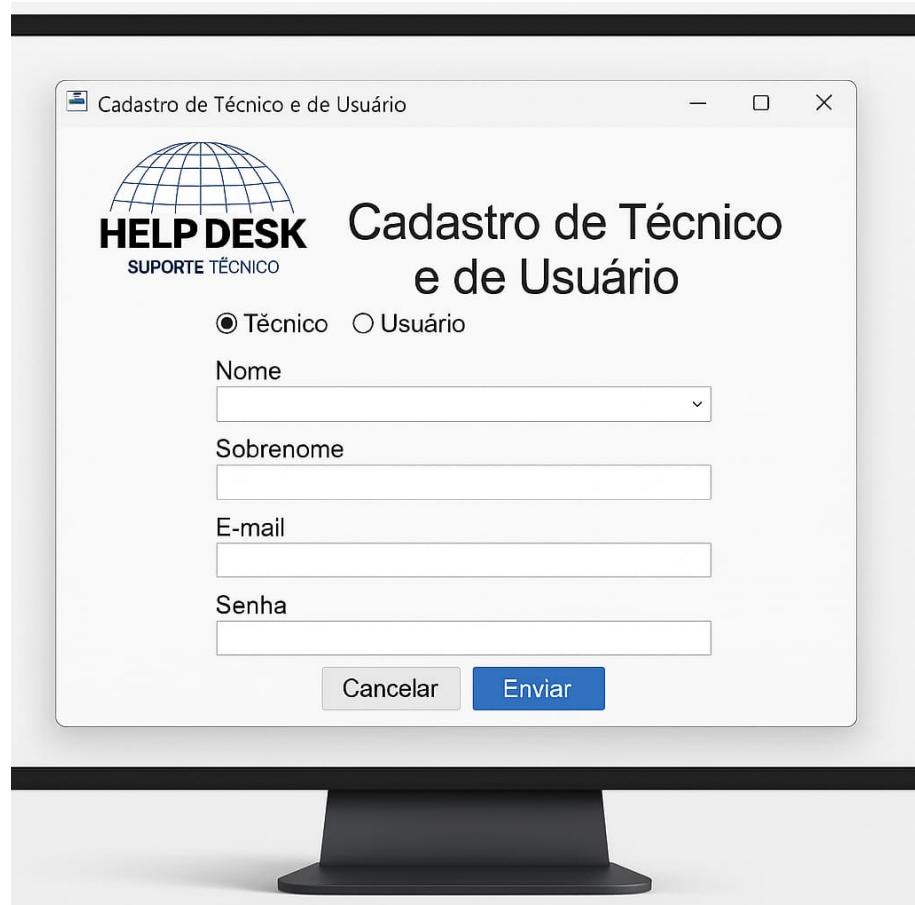


Tela que será utilizada pelos técnicos para preenchimento dos campos e encerramento dos chamados.

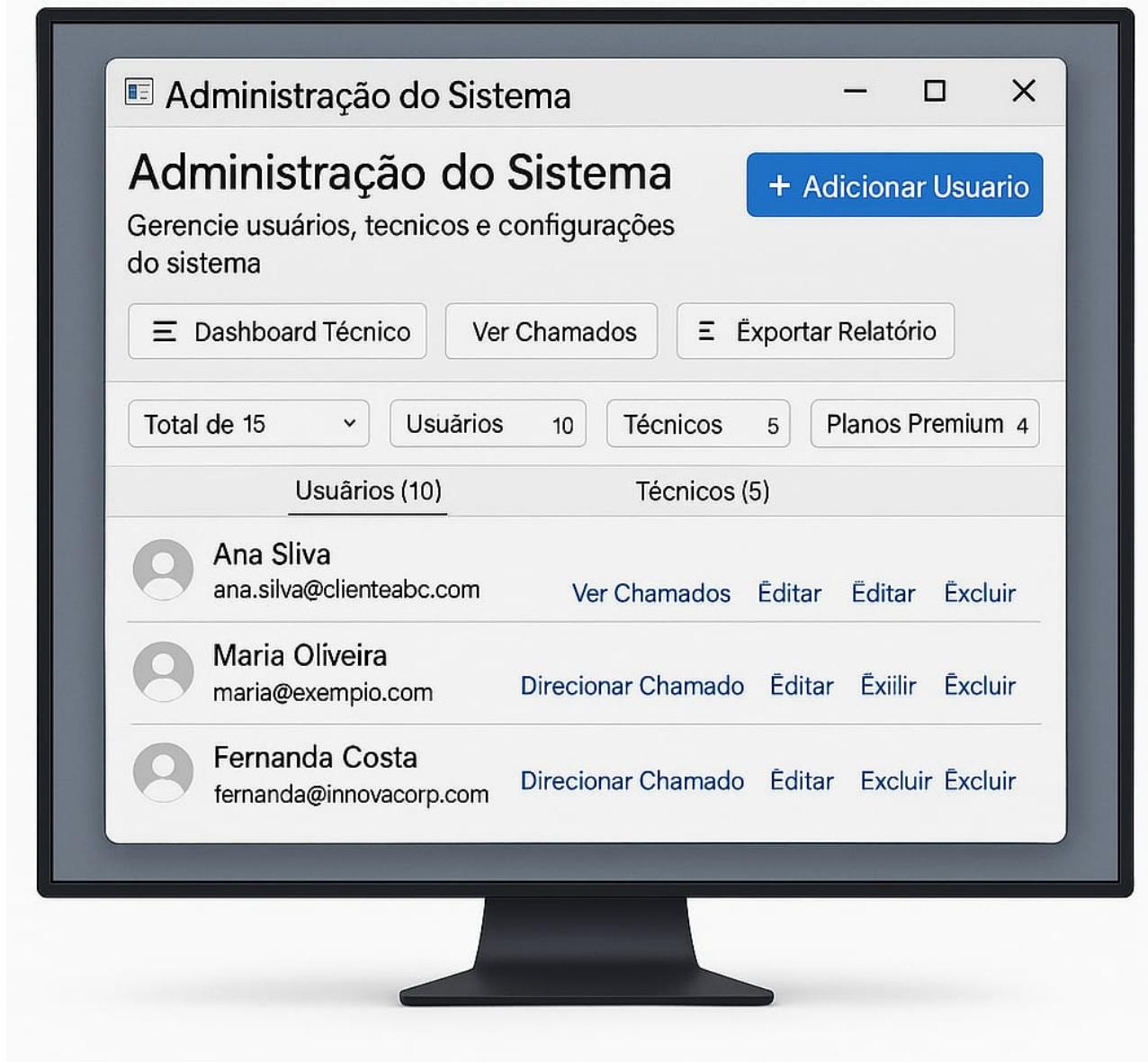
### 2.7.4 Protótipo de tela: Desktop



Tela de Login Desktop



Tela de Cadastro Desktop



Tela de Administração de clientes e técnicos, com a opções de ver clientes, técnicos e direcionar chamados manualmente

### 3 CONCLUSÃO

No início da fase de testes, foram elaborados os diagramas de caso de uso, diagramas de e diagramas de sequência, com o objetivo de representar de forma estruturada os requisitos e o comportamento do sistema. Além disso, foram desenvolvidos os scripts iniciais do banco de dados, visando a criação da estrutura necessária para o armazenamento das informações. Também foram criados os protótipos das interfaces nas versões desktop, web e mobile, permitindo a validação da experiência do usuário em múltiplas plataformas.

# REFERÊNCIAS

DIAGRAMA. [Https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams](https://www.ibm.com/docs/pt-br/rsas/7.5.0?topic=structure-class-diagrams). Citado na página 28.

DIAGRAMA UML. [Https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case](https://www.ibm.com/docs/pt-br/rsm/7.5.0?topic=diagrams-use-case). Citado na página 13.

DRAWIO. <<https://www.drawio.com>>. (Accessed on 20/05/2025). Citado na página 14.

ENGENHARIA. [Https://www.estrategiaconcursos.com.br/blog/requisitos-funcionais-nao-funcionais-cef-ti/](https://www.estrategiaconcursos.com.br/blog/requisitos-funcionais-nao-funcionais-cef-ti/). Citado na página 13.

MODELO DER. [Https://miro.com/pt/diagrama/o-que-e-diagrama-entidade-relacionamento/](https://miro.com/pt/diagrama/o-que-e-diagrama-entidade-relacionamento/). Citado na página 36.