

UNIP – UNIVERSIDADE PAULISTA

ALPHAVILLE

CST em Análise e Desenvolvimento de Sistemas

NOME DOS ALUNOS

Brenno Pinheiro Barbosa – N0954F5

Eric Aparecido Rodrigues Silva – G9115B6

Fernando Menezes dos Santos – G918JA9

João Paulo de Jesus Bispo – G940BD0

Lucas Gabriel de Oliveira Silva – R0411D5

Pablo Amorim de Gales – R077BF3

TÍTULO DO TRABALHO

Sistema Integrado para Gestão de Chamados e Suporte Técnico Baseado em IA

SANTANA DE PARNAÍBA – SP

2025

FOLHA DE ROSTO

NOME DOS ALUNOS

Brenno Pinheiro Barbosa – N0954F5

Eric Aparecido Rodrigues Silva – G9115B6

Fernando Menezes dos Santos – G918JA9

João Paulo de Jesus Bispo – G940BD0

Lucas Gabriel de Oliveira Silva – R0411D5

Pablo Amorim de Gales – R077BF3

TÍTULO DO TRABALHO

Sistema Integrado para Gestão de Chamados e Suporte Técnico Baseado em IA

Projeto Integrado Multidisciplinar IV apresentado como requisito parcial para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas pela Universidade Paulista – UNIP.

Orientador: [Nome do Orientador]

SANTANA DE PARNAÍBA – SP

2025

RESUMO

Este projeto detalha o desenvolvimento de um sistema integrado e multiplataforma para a gestão de chamados e suporte técnico, fundamentado no uso de Inteligência Artificial (IA). O objetivo central é otimizar o ciclo de vida do atendimento, desde a abertura do chamado até a sua resolução, por meio da automação da classificação, priorização e sugestão de soluções. A solução proposta abrange interfaces para desktop, web e mobile, atendendo às necessidades de diferentes perfis de usuários: clientes, técnicos e gestores. A arquitetura do sistema foi concebida para ser robusta, escalável e segura, com especial atenção à conformidade com a Lei Geral de Proteção de Dados (LGPD) em todas as etapas do tratamento de dados pessoais. O trabalho aborda desde a concepção da regra de negócio, levantamento de requisitos e

modelagem com diagramas UML, até a prototipação das interfaces e a estruturação do banco de dados. Adicionalmente, explora-se o potencial de ferramentas de IA generativa como apoio ao desenvolvimento, enriquecendo a base de conhecimento e a documentação do sistema. Espera-se que a implementação deste projeto resulte em uma redução significativa do tempo médio de atendimento, na otimização da alocação de recursos técnicos e na melhoria contínua da experiência do usuário final.

Palavras-chave: Gestão de Chamados, Suporte Técnico, Inteligência Artificial, Processamento de Linguagem Natural, Automação de TI.

ABSTRACT

This project details the development of an integrated, multi-platform system for managing technical support tickets, based on the use of Artificial Intelligence (AI). The main objective is to optimize the entire support lifecycle, from ticket creation to resolution, by automating classification, prioritization, and solution suggestions. The proposed solution includes desktop, web, and mobile interfaces, catering to the needs of different user profiles: clients, technicians, and managers. The system architecture is designed to be robust, scalable, and secure, with special attention to compliance with the General Data Protection Law (LGPD) at all stages of personal data processing. This work covers everything from the business rules conception, requirements gathering, and UML modeling, to interface prototyping and database structuring. Additionally, the potential of generative AI tools to support development is explored, enriching the system's knowledge base and documentation. The implementation of this project is expected to result in a significant reduction in the average response time, optimization of technical resource allocation, and continuous improvement of the end-user experience.

Keywords: Ticket Management, Technical Support, Artificial Intelligence, Natural Language Processing, IT Automation.

SUMÁRIO

1. INTRODUÇÃO
2. OBJETIVO GERAL
3. OBJETIVOS ESPECÍFICOS
4. JUSTIFICATIVA

- 5. FUNDAMENTAÇÃO TEÓRICA
- 6. METODOLOGIA
- 7. REGRA DE NEGÓCIO
- 8. LEVANTAMENTO DE REQUISITOS 8.1. Requisitos Funcionais 8.2. Requisitos Não Funcionais
- 9. ARQUITETURA DO SISTEMA
- 10. DIAGRAMAS UML 10.1. Diagrama de Casos de Uso 10.2. Diagrama de Classes 10.3. Diagrama de Sequência
- 11. ESTRUTURAÇÃO DO BANCO DE DADOS
- 12. PROTÓTIPOS DE INTERFACE 12.1. Interface Desktop (Administrador) 12.2. Interface Mobile (Técnico) 12.3. Interface Web (Cliente)
- 13. PLANO DE TESTES
- 14. PLANO DE IMPLANTAÇÃO
- 15. CONSIDERAÇÕES SOBRE LGPD E SEGURANÇA
- 16. CONCLUSÃO REFERÊNCIAS APÊNDICES ANEXOS

O texto a seguir é uma versão aprimorada e expandida da seção original, incorporando as melhores práticas de escrita acadêmica e os requisitos do manual da faculdade.

1. INTRODUÇÃO

A transformação digital redefiniu o cenário operacional das organizações modernas, tornando a tecnologia da informação um pilar central para a produtividade e a competitividade. Nesse contexto, a eficiência e a agilidade dos serviços de suporte técnico deixam de ser um mero apoio operacional e ascendem à condição de fator estratégico para a continuidade dos negócios. A interrupção de sistemas, falhas de hardware ou dificuldades no uso de softwares podem gerar impactos severos, que vão desde a queda na produtividade individual até perdas financeiras e danos à reputação da empresa.

Contudo, a gestão tradicional de chamados de suporte, frequentemente baseada em processos manuais como o recebimento de solicitações por e-mail ou telefone, revela-se cada vez mais inadequada para a complexidade e o volume das demandas atuais. A ausência de um sistema centralizado e inteligente resulta em uma série de desafios

crônicos: falta de visibilidade sobre o status dos atendimentos, dificuldade na priorização de demandas críticas, inconsistência na qualidade do serviço e sobrecarga das equipes de TI, que despendem tempo excessivo em tarefas repetitivas de triagem e classificação.

Diante deste cenário, o presente projeto propõe o desenvolvimento de um **Sistema Integrado para Gestão de Chamados e Suporte Técnico**, que se diferencia pela incorporação estratégica de **Inteligência Artificial (IA)**. O objetivo é transcender a simples organização de tíquetes, criando um ecossistema inteligente capaz de automatizar tarefas, acelerar a resolução de problemas e fornecer insights valiosos para a gestão. A solução será concebida como uma plataforma multiplataforma, com interfaces web, desktop e mobile, garantindo acessibilidade e usabilidade para todos os perfis de usuários envolvidos: os colaboradores que necessitam de suporte (clientes), os analistas responsáveis pela resolução (técnicos) e os gestores que monitoram a eficiência do serviço.

Este trabalho, desenvolvido como continuidade do Projeto Integrado Multidisciplinar (PIM) do semestre anterior, evolui do levantamento de requisitos para a fase de projeto e detalhamento da implementação, em conformidade com as diretrizes do PIM IV. Serão detalhados a arquitetura do sistema, o projeto de banco de dados, os diagramas UML que modelam o comportamento e a estrutura da aplicação, e os protótipos de interface, mantendo o alinhamento com as tecnologias especificadas: C#, ASP.NET, MS SQL Server e o desenvolvimento de uma aplicação nativa para Android. O projeto reforça, ainda, o compromisso com a segurança e a privacidade, adotando as melhores práticas e os princípios da Lei Geral de Proteção de Dados (LGPD) como requisitos fundamentais.

2. OBJETIVO GERAL

Desenvolver um sistema integrado e multiplataforma para a gestão eficiente de chamados de suporte técnico, incorporando um módulo de Inteligência Artificial (IA) para otimizar a classificação, priorização e sugestão de soluções para problemas recorrentes. O sistema deverá proporcionar uma plataforma intuitiva para que os usuários registrem suas solicitações, enquanto técnicos e gestores possam acompanhar, gerenciar e resolver os chamados de forma ágil, eficaz e inteligente. Este objetivo visa aprimorar a experiência do usuário, reduzir o tempo médio de resolução e otimizar a alocação de recursos da equipe de suporte.

3. OBJETIVOS ESPECÍFICOS

Para alcançar o objetivo geral de desenvolver um sistema de gestão de chamados com IA, foram definidos os seguintes objetivos específicos:

- **Desenvolver interfaces multiplataforma (web, desktop e mobile):** Criar interfaces de usuário intuitivas e responsivas para as plataformas web (ASP.NET), desktop (C# com Windows Forms, WPF ou Blazor) e mobile (Android), garantindo acessibilidade e usabilidade para clientes, técnicos e administradores em diversos dispositivos e sistemas operacionais.
- **Implementar módulo de IA para classificação e priorização automática de chamados:** Utilizar algoritmos de Processamento de Linguagem Natural (PLN) para analisar o conteúdo textual dos chamados, classificá-los automaticamente por categoria e atribuir níveis de prioridade com base em palavras-chave, contexto e criticidade, otimizando o fluxo de atendimento e reduzindo a triagem manual.
- **Integrar base de conhecimento para sugestão de soluções:** Desenvolver um módulo inteligente que conecte o sistema a uma base de conhecimento interna, contendo FAQs dinâmicas, tutoriais, manuais e histórico de resoluções, a fim de sugerir soluções automáticas aos usuários e técnicos, promovendo o autoatendimento e a agilidade na resolução.
- **Gerar relatórios gerenciais e Indicadores Chave de Desempenho (KPIs):** Incluir funcionalidades robustas de análise e visualização de dados para fornecer informações gerenciais em tempo real sobre o desempenho da equipe técnica, tempo médio de resolução, volume de chamados por categoria, taxas de reabertura e outros KPIs relevantes, permitindo tomadas de decisão estratégicas.
- **Garantir segurança e conformidade com a LGPD:** Implementar políticas e mecanismos de segurança da informação rigorosos, incluindo controle de acesso baseado em perfis, criptografia de dados sensíveis, anonimização de registros quando necessário e gestão de consentimento explícito, assegurando a proteção dos dados pessoais dos usuários em total conformidade com a Lei Geral de Proteção de Dados.
- **Otimizar a distribuição e escalonamento de chamados:** Desenvolver lógicas para a distribuição automática de chamados entre os técnicos, considerando especialidade e carga de trabalho, e implementar mecanismos de escalonamento automático para chamados críticos, garantindo que as

solicitações sejam direcionadas ao nível de suporte adequado de forma eficiente.

- **Permitir feedback e avaliação do atendimento:** Incorporar um sistema de feedback que permita aos usuários avaliar o atendimento recebido após o encerramento do chamado, coletando dados para a melhoria contínua dos serviços e o treinamento da equipe de suporte.

4. JUSTIFICATIVA

A ausência de um sistema de suporte técnico eficiente e inteligente representa um entrave significativo para a produtividade e a satisfação em qualquer organização contemporânea. Em um ambiente onde a dependência tecnológica é crescente, problemas técnicos são inevitáveis, e a capacidade de resolvê-los de forma ágil e eficaz torna-se um diferencial competitivo. A gestão ineficaz de chamados pode acarretar uma série de consequências negativas, que vão desde a perda de horas de trabalho devido a sistemas inoperantes até a insatisfação de usuários e clientes, culminando em prejuízos financeiros e danos à imagem da empresa.

Historicamente, muitas organizações ainda dependem de métodos rudimentares para o gerenciamento de suporte, como e-mails, telefonemas ou planilhas. Tais abordagens são inerentemente propensas a falhas, gerando atrasos no atendimento, dificuldade na rastreabilidade das solicitações, perda de informações cruciais e, conseqüentemente, retrabalho para as equipes de TI. A falta de visibilidade sobre o volume de chamados, os tempos de resposta e os gargalos operacionais impede que gestores tomem decisões baseadas em dados, comprometendo a melhoria contínua dos serviços.

Nesse contexto, a implementação de um sistema integrado de gestão de chamados, potencializado pela Inteligência Artificial, justifica-se pela sua capacidade de transformar o paradigma do suporte técnico. A IA, por meio de técnicas de Processamento de Linguagem Natural (PLN) e aprendizado de máquina, pode automatizar a triagem, classificação e priorização de chamados, liberando os técnicos para se concentrarem em problemas mais complexos e estratégicos. Além disso, a sugestão automática de soluções e a base de conhecimento dinâmica promovem o autoatendimento, empoderando os usuários e reduzindo a carga sobre a equipe de suporte.

Os benefícios estendem-se à gestão, que terá acesso a dashboards e relatórios em tempo real, permitindo uma análise aprofundada do desempenho, a identificação de tendências e a otimização proativa dos processos. A melhoria na experiência do usuário final, que terá seus problemas resolvidos mais rapidamente e com maior assertividade, reforça a importância deste projeto. Adicionalmente, a conformidade com a Lei Geral de Proteção de Dados (LGPD) é um pilar fundamental, garantindo que o tratamento de dados pessoais seja realizado com a máxima segurança e transparência, mitigando riscos legais e fortalecendo a confiança dos usuários. Portanto, este projeto não apenas endereça uma necessidade operacional crítica, mas também posiciona a organização na vanguarda da inovação tecnológica e da responsabilidade no tratamento de dados.

5. FUNDAMENTAÇÃO TEÓRICA

O desenvolvimento de um sistema integrado para gestão de chamados e suporte técnico baseado em Inteligência Artificial (IA) exige uma sólida fundamentação teórica em diversas áreas do conhecimento. Esta seção aborda os principais conceitos que sustentam o projeto, desde a engenharia de software até as tecnologias de IA e as normativas de proteção de dados.

5.1. Engenharia de Software

A Engenharia de Software é a disciplina que aplica uma abordagem sistemática, disciplinada e quantificável ao desenvolvimento, operação e manutenção de software [1]. Ela engloba um conjunto de metodologias, ferramentas e técnicas para gerenciar a complexidade do desenvolvimento de sistemas, garantindo a qualidade, a manutenibilidade e a escalabilidade do produto final. No contexto deste projeto, os princípios da Engenharia de Software são aplicados no levantamento de requisitos, análise, projeto, implementação e testes do sistema.

5.1.1. Levantamento e Análise de Requisitos

O processo de levantamento e análise de requisitos é fundamental para o sucesso de qualquer projeto de software. Ele envolve a identificação, documentação e validação das necessidades dos usuários e das restrições do sistema [2]. Requisitos são classificados em funcionais (o que o sistema deve fazer) e não funcionais (como o sistema deve se comportar, como desempenho, segurança e usabilidade). Uma

compreensão clara e completa dos requisitos minimiza riscos e retrabalho nas fases posteriores do desenvolvimento.

5.1.2. Modelagem de Sistemas (UML)

A Unified Modeling Language (UML) é uma linguagem de modelagem gráfica padronizada utilizada para especificar, visualizar, construir e documentar os artefatos de um sistema de software [3]. Através de diversos tipos de diagramas, a UML permite representar diferentes perspectivas do sistema, como sua estrutura (diagramas de classe, de componentes) e seu comportamento (diagramas de caso de uso, de sequência, de atividades). No presente projeto, a UML é empregada para modelar as interações dos usuários, a estrutura de classes e o fluxo de eventos, facilitando a comunicação entre a equipe de desenvolvimento e os stakeholders.

5.2. Inteligência Artificial e Processamento de Linguagem Natural (PLN)

A Inteligência Artificial (IA) é um campo da ciência da computação dedicado à criação de sistemas capazes de realizar tarefas que normalmente exigiriam inteligência humana [4]. No contexto de suporte técnico, a IA é empregada para automatizar processos, otimizar a tomada de decisões e melhorar a experiência do usuário. O Processamento de Linguagem Natural (PLN) é uma subárea da IA que se concentra na interação entre computadores e a linguagem humana, permitindo que máquinas compreendam, interpretem e gerem texto de forma significativa [5].

5.2.1. Classificação e Priorização de Chamados

Utilizando técnicas de PLN, o sistema pode analisar o conteúdo textual dos chamados (descrições, mensagens) para identificar padrões e categorizá-los automaticamente. Algoritmos de aprendizado de máquina, como redes neurais ou máquinas de vetores de suporte (SVM), podem ser treinados com dados históricos de chamados para aprender a associar termos a categorias específicas e a atribuir níveis de prioridade com base na criticidade e no impacto do problema [6].

5.2.2. Sugestão de Soluções e Base de Conhecimento

A IA também pode ser utilizada para sugerir soluções automáticas, consultando uma base de conhecimento dinâmica. Ao analisar um novo chamado, o sistema pode buscar por problemas semelhantes já resolvidos e apresentar as soluções

correspondentes. Isso não apenas acelera o tempo de resposta, mas também promove o autoatendimento e a padronização das soluções [7]. Ferramentas de IA generativa podem auxiliar na criação e atualização dessa base de conhecimento, gerando FAQs e documentação a partir de históricos de chamados.

5.3. Desenvolvimento Multiplataforma

O desenvolvimento multiplataforma refere-se à criação de aplicações que podem ser executadas em diferentes sistemas operacionais e dispositivos com uma única base de código ou com adaptações mínimas [8]. No projeto, a escolha por interfaces web, desktop e mobile visa maximizar a acessibilidade e a conveniência para os usuários. As tecnologias empregadas, como ASP.NET para web e C# para desktop e mobile (via Xamarin/MAUI ou abordagens nativas), permitem a construção de uma solução coesa e integrada.

5.4. Banco de Dados Relacionais (MS SQL Server)

Bancos de dados relacionais são amplamente utilizados para armazenar e gerenciar dados de forma estruturada, organizando-os em tabelas com relações predefinidas [9]. O MS SQL Server é um sistema de gerenciamento de banco de dados relacional (SGBDR) robusto e escalável, desenvolvido pela Microsoft, que oferece alta performance, segurança e diversas ferramentas para administração e desenvolvimento. Sua escolha para este projeto garante a integridade, a persistência e a disponibilidade dos dados dos chamados, usuários e base de conhecimento.

5.5. Lei Geral de Proteção de Dados (LGPD)

A Lei Geral de Proteção de Dados (Lei nº 13.709/2018) é a legislação brasileira que regulamenta o tratamento de dados pessoais, tanto em meios físicos quanto digitais, por pessoa natural ou por pessoa jurídica de direito público ou privado [10]. A LGPD estabelece princípios, direitos dos titulares e deveres dos agentes de tratamento, visando proteger a privacidade e a liberdade dos indivíduos. No desenvolvimento do sistema, a conformidade com a LGPD é um requisito não funcional crítico, exigindo a implementação de medidas de segurança, controle de acesso, anonimização de dados e a obtenção de consentimento explícito para o tratamento de informações pessoais dos usuários.

6. METODOLOGIA

O desenvolvimento do Sistema Integrado para Gestão de Chamados e Suporte Técnico Baseado em Inteligência Artificial adota uma abordagem metodológica híbrida, combinando elementos de metodologias ágeis com a estrutura de um processo de desenvolvimento em cascata adaptado. Esta escolha visa equilibrar a flexibilidade para responder a mudanças com a necessidade de documentação e planejamento robustos, essenciais para um projeto acadêmico e para a garantia da qualidade do software.

6.1. Abordagem de Desenvolvimento

Para o ciclo de vida do projeto, foi empregada uma metodologia que integra as fases de concepção, elaboração, construção e transição, com ênfase na iteração e no feedback contínuo. As principais etapas incluem:

1. **Concepção:** Nesta fase inicial, foram definidos o escopo do projeto, os objetivos gerais e específicos, a justificativa e a identificação dos principais stakeholders. O levantamento preliminar de requisitos e a análise de viabilidade técnica e operacional foram realizados para estabelecer a base do projeto.
2. **Elaboração:** A fase de elaboração focou na análise aprofundada dos requisitos, na modelagem do sistema utilizando a Linguagem de Modelagem Unificada (UML) e no projeto da arquitetura. Foram criados os diagramas de caso de uso, classes e sequência, além da definição da estrutura do banco de dados e dos protótipos de interface. Esta fase também incluiu a pesquisa e seleção das tecnologias a serem empregadas.
3. **Construção:** A fase de construção envolve a implementação do sistema, seguindo o projeto detalhado na fase anterior. Esta etapa é iterativa, com o desenvolvimento de funcionalidades em ciclos curtos, permitindo a integração contínua e a realização de testes unitários e de integração. A codificação das interfaces (web, desktop, mobile), o desenvolvimento do módulo de IA e a implementação da lógica de negócio ocorrem aqui.
4. **Transição:** A fase final compreende a implantação do sistema em um ambiente de testes e, posteriormente, em um ambiente de produção simulado. Inclui a realização de testes de sistema e aceitação, a elaboração de manuais de usuário e planos de treinamento, e a preparação para a entrega do produto final. A coleta de feedback e a realização de ajustes finos são cruciais nesta etapa.

6.2. Ferramentas e Técnicas

- **Gerenciamento de Requisitos:** Utilização de técnicas de elicitação como entrevistas, análise de documentos e prototipagem para coletar e refinar os requisitos funcionais e não funcionais.
- **Modelagem:** Emprego da UML para a representação visual da estrutura e comportamento do sistema, facilitando a compreensão e a comunicação entre os membros da equipe.
- **Controle de Versão:** Uso de sistemas de controle de versão (e.g., Git) para gerenciar as alterações no código-fonte e nos documentos do projeto, garantindo a rastreabilidade e a colaboração eficiente.
- **Testes:** Aplicação de diferentes níveis de testes (unitários, de integração, de sistema e de aceitação) para garantir a qualidade, a robustez e a conformidade do sistema com os requisitos estabelecidos.
- **Documentação:** Elaboração de documentação técnica e de usuário em todas as fases do projeto, conforme as normas da ABNT, para garantir a manutenibilidade e a usabilidade do sistema.

Esta metodologia permite uma abordagem estruturada para o desenvolvimento, ao mesmo tempo em que incorpora a agilidade necessária para adaptar-se a novos conhecimentos e desafios que possam surgir ao longo do projeto, especialmente no que tange à integração de tecnologias de Inteligência Artificial.

7. REGRA DE NEGÓCIO

O Sistema Integrado para Gestão de Chamados e Suporte Técnico Baseado em Inteligência Artificial é regido por um conjunto de regras de negócio bem definidas, que asseguram seu funcionamento eficiente, inteligente e alinhado às necessidades operacionais. Estas regras são fundamentais para automatizar processos, otimizar a tomada de decisões e garantir a qualidade do serviço prestado. A seguir, detalhamos as principais regras de negócio que orientam o comportamento do sistema:

- **Identificação e Categorização Automática de Chamados:** Ao registrar um novo chamado, o sistema empregará técnicas avançadas de Processamento de Linguagem Natural (PLN) para analisar o conteúdo textual fornecido pelo usuário. Com base em palavras-chave, frases-chave e no histórico de chamados

previamente resolvidos, o sistema identificará e atribuirá automaticamente uma categoria ao problema. Esta automação visa reduzir o tempo de triagem manual e aumentar a precisão na classificação.

- **Priorização Dinâmica de Chamados:** A prioridade de um chamado será definida de forma dinâmica, considerando múltiplos fatores. O sistema aplicará regras de priorização baseadas no Acordo de Nível de Serviço (SLA) estabelecido, no impacto operacional do problema (ex: sistema crítico fora do ar), na urgência declarada pelo usuário e na criticidade inferida pela IA a partir do conteúdo do chamado. Chamados classificados como críticos serão automaticamente escalados para níveis superiores de suporte, conforme matriz de responsabilidades predefinida.
- **Sugestão Inteligente de Soluções:** Integrado a uma base de conhecimento robusta, o sistema utilizará algoritmos de IA para sugerir soluções automáticas aos usuários e técnicos. Ao abrir um chamado, a IA buscará por problemas semelhantes e apresentará artigos da base de conhecimento, FAQs ou resoluções anteriores que possam auxiliar na solução, promovendo o autoatendimento e acelerando o tempo de primeira resposta.
- **Distribuição Automática de Chamados:** O sistema será capaz de distribuir automaticamente os chamados para os técnicos mais adequados, levando em consideração a especialidade do técnico, sua carga de trabalho atual e a categoria do chamado. Isso garante uma alocação eficiente de recursos e otimiza o tempo de atendimento.
- **Notificações e Alertas Automatizados:** Para manter todos os envolvidos informados, o sistema enviará notificações automáticas por e-mail, push ou outros canais integrados a cada atualização significativa no status do chamado. Isso inclui a abertura, atribuição, atualização, resolução e encerramento do chamado, garantindo transparência e reduzindo a necessidade de acompanhamento manual.
- **Coleta de Feedback e Avaliação de Atendimento:** Após o encerramento de um chamado, o usuário será convidado a fornecer feedback sobre o atendimento recebido, por meio de uma pesquisa de satisfação com campos para avaliação e comentários. Esses dados serão armazenados e utilizados para análise de desempenho, identificação de pontos de melhoria e treinamento contínuo da equipe de suporte.

- **Gestão de Reabertura de Chamados:** O sistema permitirá a reabertura de chamados encerrados dentro de um período configurável (ex: 48 horas), caso o problema persista ou não tenha sido completamente resolvido. Após esse prazo, qualquer nova solicitação relacionada ao mesmo problema será registrada como um novo chamado, mantendo a organização do histórico.
- **Controle de Acesso Baseado em Perfis:** O acesso às funcionalidades do sistema será controlado por perfis de usuário (ex: Cliente, Técnico, Gestor), cada um com permissões específicas. Isso garante que cada usuário tenha acesso apenas às informações e funcionalidades necessárias para suas atribuições, reforçando a segurança e a integridade dos dados.
- **Geração de Relatórios e KPIs:** O sistema gerará relatórios gerenciais e indicadores chave de desempenho (KPIs) em tempo real, fornecendo uma visão abrangente sobre o volume de chamados, tempo médio de resolução, satisfação do cliente, desempenho da equipe, entre outros. Esses dados são cruciais para a gestão estratégica e a melhoria contínua dos processos de suporte.
- **Conformidade com a LGPD:** Todas as operações que envolvem o tratamento de dados pessoais serão realizadas em estrita conformidade com a Lei Geral de Proteção de Dados (LGPD), garantindo a privacidade e a segurança das informações dos usuários. Isso inclui a anonimização de dados sensíveis, o registro de consentimento e a implementação de medidas de segurança robustas.

8. LEVANTAMENTO DE REQUISITOS

O levantamento de requisitos é uma etapa crucial no desenvolvimento de software, pois define as funcionalidades essenciais e as características técnicas que o sistema deve possuir para atender às necessidades dos usuários e aos objetivos do projeto. Neste trabalho, os requisitos foram categorizados em funcionais, que descrevem o comportamento e as funções que o sistema deve executar, e não funcionais, que especificam critérios de qualidade, desempenho, segurança e usabilidade. A análise detalhada desses requisitos garante que o sistema seja robusto, eficiente e alinhado às expectativas dos stakeholders.

8.1. Requisitos Funcionais

Os requisitos funcionais detalham as ações que o sistema deve ser capaz de realizar para o usuário. Eles são a base para o desenvolvimento das funcionalidades e interações do sistema.

- **RF001 – Gestão de Usuários e Perfis de Acesso:** O sistema deverá permitir o cadastro, edição e exclusão de usuários, incluindo clientes (solicitantes de suporte) e técnicos (prestadores de suporte). Cada usuário terá um perfil de acesso definido (ex: Cliente, Técnico, Gestor), que determinará as permissões e funcionalidades disponíveis. O cadastro incluirá dados pessoais essenciais, informações de contato, cargo/setor (para técnicos) e credenciais de login seguras. Técnicos terão permissões ampliadas para interagir com os chamados, enquanto gestores terão acesso a painéis de controle e relatórios.
- **RF002 – Abertura e Registro de Chamados:** Usuários (clientes) deverão ser capazes de registrar novos chamados de suporte técnico através das interfaces web, desktop e mobile. O processo de abertura incluirá campos para descrição detalhada do problema, anexos (imagens, documentos), e poderá sugerir automaticamente a categoria e a prioridade do chamado com base na descrição fornecida, utilizando o módulo de IA.
- **RF003 – Classificação e Priorização Automática por IA:** O módulo de Inteligência Artificial do sistema será responsável por analisar o conteúdo textual dos chamados (descrição, palavras-chave) e, com base em modelos de Processamento de Linguagem Natural (PLN) treinados, classificá-los automaticamente em categorias predefinidas (ex: hardware, software, rede, acesso). Adicionalmente, a IA atribuirá um nível de prioridade (ex: baixa, média, alta, crítica) considerando a urgência, o impacto potencial e o histórico de problemas semelhantes, otimizando a triagem e o direcionamento.
- **RF004 – Sugestão Inteligente de Soluções:** O sistema deverá integrar-se a uma base de conhecimento interna, utilizando a IA para sugerir soluções automáticas aos usuários no momento da abertura do chamado e aos técnicos durante o atendimento. Esta funcionalidade apresentará artigos de FAQ, tutoriais, manuais ou resoluções de chamados anteriores que correspondam ao problema descrito, promovendo o autoatendimento e acelerando a resolução.
- **RF005 – Acompanhamento do Status do Chamado:** Usuários (clientes e técnicos) deverão ter acesso a uma interface que permita consultar em tempo

real o status de seus chamados. Esta interface exibirá informações como data e hora das atualizações, técnico responsável, histórico de interações e o status atual (ex: aberto, em andamento, aguardando usuário, resolvido, fechado).

- **RF006 – Atualização e Resolução de Chamados por Técnicos:** Técnicos designados deverão ter acesso para atualizar os chamados atribuídos a eles. Isso inclui registrar diagnósticos, ações realizadas, soluções aplicadas, anexar evidências e alterar o status do chamado. A interface permitirá uma comunicação clara e documentada entre o técnico e o solicitante.
- **RF007 – Avaliação do Atendimento pelo Cliente:** Após o encerramento de um chamado, o sistema enviará uma solicitação de feedback ao usuário (cliente), permitindo que ele avalie o atendimento recebido por meio de uma nota (ex: escala de 1 a 5 estrelas) e um campo opcional para comentários. Esses dados serão coletados e utilizados para medir a satisfação do cliente e identificar oportunidades de melhoria.
- **RF008 – Geração de Relatórios Gerenciais e KPIs:** O sistema deverá gerar relatórios periódicos e em tempo real com métricas de desempenho e estatísticas dos chamados. Exemplos de relatórios incluem: tempo médio de resolução (TMR), número de chamados por categoria, volume de chamados por técnico, taxas de reabertura, nível de satisfação do cliente, entre outros. Os dados deverão ser exportáveis em formatos como PDF e CSV para análise externa.
- **RF009 – Notificações Automatizadas:** O sistema enviará notificações automáticas por e-mail, notificações push (para mobile) ou outros canais configuráveis para todos os envolvidos (cliente, técnico, gestor) a cada atualização significativa no status ou interação do chamado. Isso garante a transparência e mantém os usuários informados sem a necessidade de acompanhamento manual constante.
- **RF010 – Gestão de Reabertura de Chamados:** O sistema permitirá a reabertura de chamados encerrados dentro de um período configurável (ex: 48 horas após o fechamento), caso o problema persista ou não tenha sido completamente resolvido. Após esse prazo, qualquer nova solicitação relacionada ao mesmo problema será tratada como um novo chamado para manter a integridade do histórico.

- **RF011 – Distribuição Inteligente de Chamados:** Com base na categoria do chamado, na especialidade dos técnicos e na sua carga de trabalho atual, o sistema poderá sugerir ou realizar a distribuição automática de chamados, otimizando a alocação de recursos e garantindo que o chamado seja direcionado ao técnico mais qualificado para resolvê-lo.
- **RF012 – Escalonamento Automático de Chamados Críticos:** Chamados identificados como críticos pela IA ou por regras de negócio predefinidas deverão ser automaticamente escalados para níveis superiores de suporte ou para técnicos com maior senioridade, garantindo uma resposta rápida e eficaz a problemas de alto impacto.

8.2. Requisitos Não Funcionais

Os requisitos não funcionais descrevem as qualidades e restrições do sistema, impactando a experiência do usuário e a robustez da aplicação.

- **RNF001 – Responsividade e Acessibilidade Multiplataforma:** O sistema deverá ser totalmente responsivo, adaptando-se a diferentes tamanhos de tela e dispositivos (desktop, tablets e smartphones), garantindo uma experiência de usuário consistente e acessível em todas as plataformas (web, desktop, mobile).
- **RNF002 – Conformidade com a LGPD e Proteção de Dados:** O sistema deverá garantir a proteção dos dados pessoais dos usuários em estrita conformidade com a Lei Geral de Proteção de Dados (LGPD). Isso inclui a implementação de criptografia para dados sensíveis, controle de acesso rigoroso baseado em perfis, mecanismos para anonimização e exclusão de dados, e a obtenção de consentimento explícito para o tratamento de informações pessoais.
- **RNF003 – Desempenho e Tempo de Resposta:** As funcionalidades críticas do sistema (abertura, consulta, atualização de chamados) deverão apresentar um tempo de resposta inferior a 3 segundos, mesmo sob carga moderada, para garantir uma experiência de usuário fluida e eficiente.
- **RNF004 – Armazenamento de Dados em MS SQL Server:** Todos os dados do sistema, incluindo informações de usuários, chamados, base de conhecimento e logs, deverão ser armazenados em um banco de dados relacional MS SQL Server, garantindo integridade, segurança e escalabilidade.

- **RNF005 – Usabilidade e Interface Intuitiva:** A interface gráfica do usuário (GUI) deverá ser projetada com foco na usabilidade e intuitividade, seguindo princípios de design centrado no usuário. Isso implica em menus claros, navegação lógica, ícones representativos, feedback visual adequado e cores acessíveis, minimizando a curva de aprendizado e maximizando a eficiência do usuário.
- **RNF006 – Segurança da Informação:** Além da conformidade com a LGPD, o sistema deverá implementar medidas robustas de segurança da informação, como autenticação multifator, autorização baseada em papéis, proteção contra injeção de SQL e XSS, auditoria de acessos e logs de segurança, para proteger o sistema contra acessos não autorizados e vulnerabilidades.
- **RNF007 – Escalabilidade:** A arquitetura do sistema deverá ser projetada para permitir a escalabilidade horizontal e vertical, suportando um aumento no volume de usuários e chamados sem degradação significativa do desempenho.
- **RNF008 – Manutenibilidade:** O código-fonte deverá ser bem documentado, modular e seguir padrões de codificação, facilitando a manutenção, a correção de bugs e a implementação de futuras melhorias.
- **RNF009 – Confiabilidade:** O sistema deverá ser confiável, garantindo a integridade dos dados e a disponibilidade das funcionalidades críticas, com mecanismos de tratamento de erros e recuperação de falhas.

9. ARQUITETURA DO SISTEMA

A arquitetura do Sistema Integrado para Gestão de Chamados e Suporte Técnico Baseado em Inteligência Artificial foi concebida para ser robusta, escalável, segura e de fácil manutenção, suportando o desenvolvimento multiplataforma e a integração de módulos de IA. A abordagem adotada é baseada em uma arquitetura em camadas, que promove a separação de responsabilidades e facilita a evolução independente de cada componente. O sistema será composto por três camadas principais: Apresentação, Lógica de Negócio e Acesso a Dados, além de um módulo de Inteligência Artificial e uma base de conhecimento integrada.

9.1. Visão Geral da Arquitetura em Camadas

1. **Camada de Apresentação (User Interface - UI):** Responsável pela interação direta com o usuário. Esta camada será implementada em três diferentes plataformas para atender às diversas necessidades e cenários de uso:
 - **Interface Web:** Desenvolvida com **ASP.NET Core** e **C#**, utilizando frameworks front-end modernos (como Blazor, React ou Angular, dependendo da escolha específica de implementação) para garantir responsividade e uma experiência de usuário rica em navegadores web. Será o principal ponto de acesso para clientes e gestores.
 - **Interface Desktop:** Implementada em **C#** utilizando **Windows Forms** ou **WPF (Windows Presentation Foundation)**, ou até mesmo **Blazor Desktop** para uma abordagem mais unificada. Esta interface será destinada principalmente aos técnicos de suporte e administradores, oferecendo funcionalidades mais robustas e controle detalhado sobre os chamados.
 - **Interface Mobile:** Desenvolvida para a plataforma **Android** utilizando **C#** com **Xamarin** ou **.NET MAUI**, ou nativamente com Kotlin/Java, para permitir que usuários abram e acompanhem chamados de seus dispositivos móveis. Esta interface será otimizada para a experiência em smartphones e tablets.
2. **Camada de Lógica de Negócio (Business Logic Layer - BLL):** Contém as regras de negócio, validações e processos que governam o comportamento do sistema. Esta camada é independente da camada de apresentação e da camada de acesso a dados, garantindo que as regras de negócio sejam consistentes em todas as interfaces. Será implementada em **C#**, utilizando princípios de Programação Orientada a Objetos (POO) e padrões de projeto para promover a modularidade e a manutenibilidade.
3. **Camada de Acesso a Dados (Data Access Layer - DAL):** Responsável pela comunicação com o banco de dados. Esta camada abstrai os detalhes de implementação do banco de dados da camada de lógica de negócio, permitindo que alterações no esquema do banco de dados ou na tecnologia de persistência sejam realizadas com mínimo impacto nas outras camadas. Utilizará **ADO.NET** ou um ORM (Object-Relational Mapper) como **Entity Framework Core** para interagir com o **MS SQL Server**.

9.2. Módulo de Inteligência Artificial (IA)

O módulo de IA será um componente central da arquitetura, atuando de forma integrada com a camada de lógica de negócio. Ele será responsável por:

- **Processamento de Linguagem Natural (PLN):** Análise do texto dos chamados para extração de entidades, identificação de intenções e categorização automática.
- **Classificação e Priorização:** Utilização de modelos de Machine Learning (ML) treinados com dados históricos para classificar chamados por categoria e atribuir níveis de prioridade.
- **Sugestão de Soluções:** Consulta à base de conhecimento e aplicação de algoritmos de busca semântica para sugerir soluções relevantes aos problemas descritos.
- **Apoio à Geração de Conteúdo:** Ferramentas de IA generativa poderão ser utilizadas para auxiliar na criação de FAQs, documentação e respostas automáticas, enriquecendo a base de conhecimento.

Este módulo poderá ser implementado utilizando bibliotecas de ML em C# (como ML.NET) ou integrando-se a serviços de IA externos (como Azure AI Services ou Google Cloud AI Platform) via APIs, dependendo da complexidade e dos recursos disponíveis.

9.3. Base de Conhecimento

A base de conhecimento será um repositório centralizado de informações, incluindo FAQs, tutoriais, manuais, artigos técnicos e histórico de soluções de chamados. Ela será acessada pelo módulo de IA para a sugestão de soluções e pelos usuários para autoatendimento. A base será dinâmica, permitindo a atualização e expansão contínua com base nas interações do sistema e na contribuição dos técnicos.

9.4. Banco de Dados (MS SQL Server)

O **MS SQL Server** será o sistema de gerenciamento de banco de dados relacional (SGBDR) utilizado para persistir todos os dados do sistema. Isso inclui:

- **Dados de Usuários:** Perfis, credenciais, permissões.
- **Dados de Chamados:** Descrições, status, histórico de interações, anexos.

- **Dados da Base de Conhecimento:** Artigos, FAQs, soluções.
- **Dados de Log e Auditoria:** Registros de atividades, logs de segurança.

A escolha do MS SQL Server garante alta performance, segurança, escalabilidade e integração com o ecossistema .NET, facilitando o desenvolvimento e a manutenção.

9.5. Diagrama de Componentes (Exemplo Conceitual)

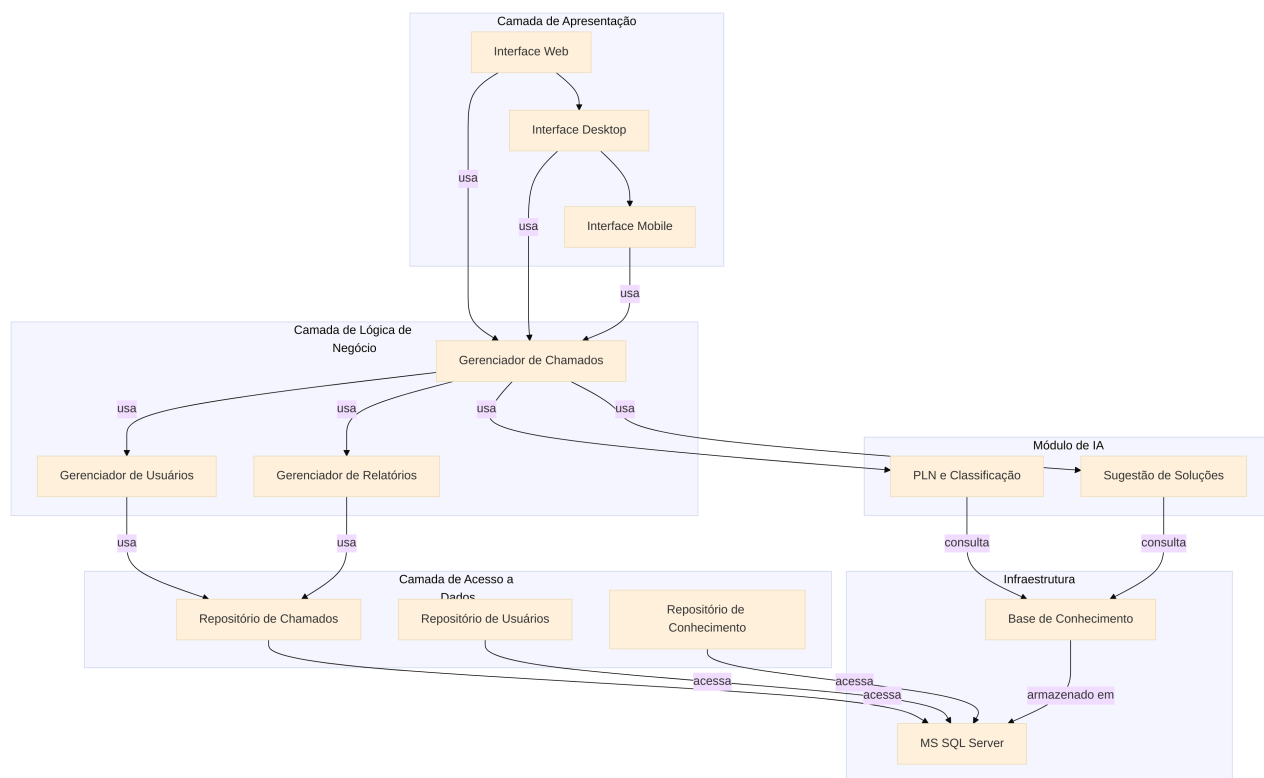


Figura 1: Diagrama de Componentes Conceitual da Arquitetura do Sistema.

Este diagrama ilustra a interconexão entre as principais camadas e módulos do sistema, destacando a separação de responsabilidades e o fluxo de comunicação entre eles. A camada de lógica de negócio atua como orquestradora, utilizando os serviços da camada de acesso a dados e do módulo de IA para processar as requisições das interfaces de apresentação.

10. DIAGRAMAS UML

A modelagem de sistemas é uma etapa fundamental no ciclo de vida do desenvolvimento de software, permitindo a representação visual da estrutura e do comportamento do sistema. A Unified Modeling Language (UML) é a linguagem padrão

para essa finalidade, oferecendo um conjunto de diagramas que auxiliam na compreensão, comunicação e documentação do projeto. Nesta seção, apresentamos os principais diagramas UML que modelam o Sistema Integrado para Gestão de Chamados e Suporte Técnico Baseado em IA, conforme as diretrizes do manual da faculdade.

10.1. Diagrama de Casos de Uso

O Diagrama de Casos de Uso descreve as funcionalidades do sistema do ponto de vista do usuário (ator), ilustrando como os diferentes atores interagem com o sistema para alcançar objetivos específicos. Ele é essencial para definir o escopo funcional do sistema.

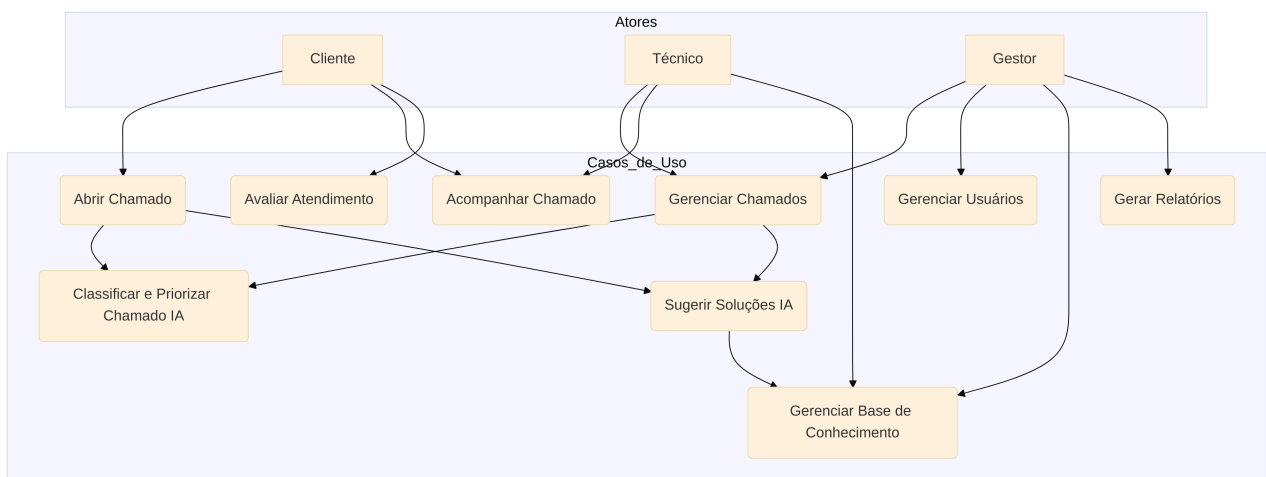


Figura 2: Diagrama de Casos de Uso do Sistema de Gestão de Chamados.

Descrição dos Casos de Uso:

- **Abertura de Chamados:** Permite que o **Cliente** registre uma nova solicitação de suporte, fornecendo detalhes do problema e anexos. A IA pode auxiliar na categorização inicial.
- **Acompanhar Chamados:** Permite que o **Cliente** visualize o status, histórico e interações de seus chamados abertos.
- **Avaliar Atendimento:** Após a resolução, o **Cliente** pode fornecer feedback sobre a qualidade do suporte recebido.
- **Gerenciar Chamados:** Permite que o **Técnico** e o **Gestor** visualizem, atribuam, atualizem o status, adicionem diagnósticos e soluções aos chamados. Inclui funcionalidades de escalonamento.

- **Gerenciar Usuários:** Permite que o **Gestor** cadastre, edite e remova usuários, além de gerenciar seus perfis e permissões de acesso.
- **Gerenciar Base de Conhecimento:** Permite que o **Técnico** e o **Gestor** criem, editem e publiquem artigos, FAQs e soluções na base de conhecimento, que será utilizada pela IA.
- **Gerar Relatórios:** Permite que o **Gestor** acesse e gere relatórios estatísticos e KPIs sobre o desempenho do suporte, volume de chamados, tempo de resolução, etc.
- **Interagir com IA:** Representa a interação interna do sistema com o módulo de Inteligência Artificial para classificação, priorização e sugestão de soluções, sendo um caso de uso que suporta os demais.

10.2. Diagrama de Classes

O Diagrama de Classes representa a estrutura estática do sistema, mostrando as classes, seus atributos, métodos e os relacionamentos entre elas. Ele é fundamental para o projeto do banco de dados e para a implementação orientada a objetos.

classDiagram

```
class Usuario {
    + id: int
    + nome: string
    + email: string
    + senha: string
    + perfil: string
    + dataCadastro: datetime
    + autenticar()
    + autorizarAcesso()
}

class Cliente {
    + telefone: string
    + endereco: string
}

class Tecnico {
    + especialidade: string
    + cargaTrabalho: int
    + atribuirChamado()
    + atualizarChamado()
}

class Gestor {
    + departamento: string
    + gerarRelatorio()
    + gerenciarUsuarios()
}

class Chamado {
    + id: int
    + titulo: string
    + descricao: string
    + dataAbertura: datetime
    + status: string
    + prioridade: string
    + categoria: string
    + dataFechamento: datetime
    + anexos: List<string>
    + abrirChamado()
    + atualizarStatus()
    + adicionarInteracao()
}

class Interacao {
    + id: int
    + dataHora: datetime
    + tipo: string
    + conteudo: string
}

class BaseConhecimento {
    + id: int
    + titulo: string
    + conteudo: string
    + palavrasChave: List<string>
    + dataPublicacao: datetime
    + buscarArtigo()
    + criarArtigo()
}
```



```

class Avaliacao {
    + id: int
    + nota: int
    + comentario: string
    + dataAvaliacao: datetime
}

class ModuloIA {
    + id: int
    + nome: string
    + classificarChamado(descricao: string): string
    + priorizarChamado(descricao: string): string
    + sugerirSolucao(descricao: string): List<BaseConhecimento>
}

Usuario <|-- Cliente
Usuario <|-- Tecnico
Usuario <|-- Gestor

Cliente "1" -- "*" Chamado : abre
Tecnico "1" -- "*" Chamado : atende
Chamado "1" -- "*" Interacao : possui
Chamado "1" -- "0..1" Avaliacao : recebe
ModuloIA "1" -- "*" Chamado : utiliza
ModuloIA "1" -- "*" BaseConhecimento : consulta
Tecnico "1" -- "*" BaseConhecimento : gerencia
Gestor "1" -- "*" Usuario : gerencia
Gestor "1" -- "*" Chamado : monitora

```

Figura 3: Diagrama de Classes do Sistema de Gestão de Chamados.

Descrição das Classes e Relacionamentos:

- **Usuario:** Classe base para todos os usuários do sistema, contendo atributos comuns como `id`, `nome`, `email`, `senha` e `perfil`. Métodos para autenticação e autorização.
- **Cliente, Tecnico, Gestor:** Classes que herdam de `Usuario`, adicionando atributos e métodos específicos para cada perfil (ex: `especialidade` para Técnico, `gerarRelatorio` para Gestor).
- **Chamado:** Representa uma solicitação de suporte, com atributos como `titulo`, `descricao`, `status`, `prioridade`, `categoria` e `dataAbertura`. Possui métodos para `abrirChamado`, `atualizarStatus` e `adicionarInteracao`.
- **Interacao:** Registra as comunicações e atualizações dentro de um chamado, como mensagens do cliente, diagnósticos do técnico, etc.
- **BaseConhecimento:** Armazena artigos, FAQs e soluções que podem ser consultados pelos usuários e pelo módulo de IA.

- **Avaliacao:** Registra o feedback do cliente sobre o atendimento de um chamado específico.
- **ModuloIA:** Representa o componente de Inteligência Artificial, com métodos para `classificarChamado`, `priorizarChamado` e `sugerirSolucao`.

Os relacionamentos indicam como as classes se conectam, por exemplo, um `Cliente` pode `abrir` muitos `Chamado`s, e um `Chamado` possui muitas `Interacao`s.

10.3. Diagrama de Sequência

O Diagrama de Sequência ilustra a ordem temporal das interações entre os objetos em um cenário específico, mostrando a sequência de mensagens trocadas para realizar uma funcionalidade. Ele é útil para entender o fluxo de controle e a colaboração entre os objetos.

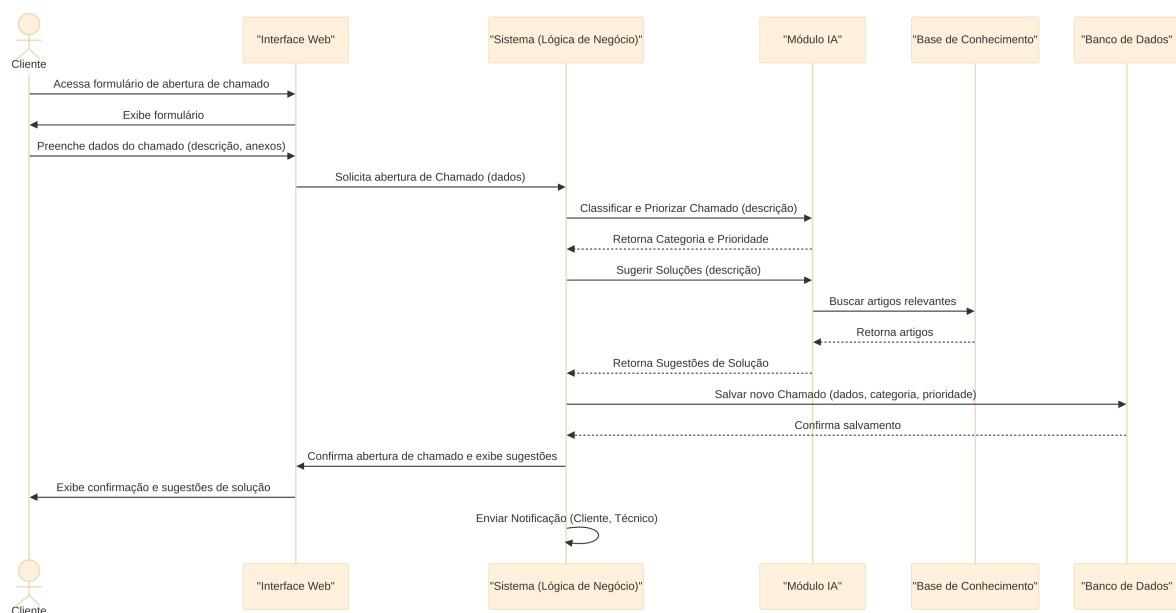


Figura 4: Diagrama de Sequência para Abertura de Chamado com IA.

Descrição do Fluxo:

Este diagrama detalha o processo de abertura de um chamado pelo cliente, destacando a interação com o módulo de IA:

1. O **Cliente** acessa o formulário de abertura de chamado na **Interface Web**.
2. A **Interface Web** exibe o formulário.
3. O **Cliente** preenche os dados do chamado, incluindo a descrição do problema.

4. A **Interface Web** envia a solicitação de abertura de chamado para o **Sistema** (Camada de Lógica de Negócio).
5. O **Sistema** invoca o **Módulo IA** para classificar e priorizar o chamado com base na descrição.
6. O **Módulo IA** retorna a categoria e a prioridade inferidas para o **Sistema**.
7. Simultaneamente, o **Sistema** solicita ao **Módulo IA** sugestões de solução para o problema.
8. O **Módulo IA** consulta a **Base de Conhecimento** para encontrar artigos relevantes.
9. A **Base de Conhecimento** retorna os artigos ao **Módulo IA**.
10. O **Módulo IA** retorna as sugestões de solução para o **Sistema**.
11. O **Sistema** salva o novo chamado, incluindo a categoria e prioridade definidas pela IA, no **Banco de Dados**.
12. O **Banco de Dados** confirma o salvamento.
13. O **Sistema** envia a confirmação de abertura e as sugestões de solução para a **Interface Web**.
14. A **Interface Web** exibe a confirmação e as sugestões ao **Cliente**.
15. O **Sistema** também envia notificações automáticas ao **Cliente** e ao **Técnico** responsável.

Este fluxo demonstra como a IA é integrada ao processo de abertura de chamados para automatizar e otimizar a triagem e o fornecimento de informações relevantes desde o início do atendimento.

11. ESTRUTURAÇÃO DO BANCO DE DADOS

A persistência dos dados é um pilar fundamental para qualquer sistema de informação, garantindo a integridade, a segurança e a disponibilidade das informações. Para o Sistema Integrado para Gestão de Chamados e Suporte Técnico, optou-se pela utilização do **MS SQL Server** como Sistema Gerenciador de Banco de Dados Relacional (SGBDR), em conformidade com as diretrizes do manual da faculdade. A estruturação do banco de dados foi projetada para suportar todas as funcionalidades do sistema, desde o registro de usuários e chamados até a gestão da base de conhecimento e a coleta de feedback.

11.1. Modelo Entidade-Relacionamento (MER) Conceitual

O Modelo Entidade-Relacionamento (MER) é uma ferramenta conceitual que permite representar as entidades de interesse do sistema e os relacionamentos entre elas. A seguir, apresentamos um MER simplificado que ilustra as principais entidades e suas conexões:

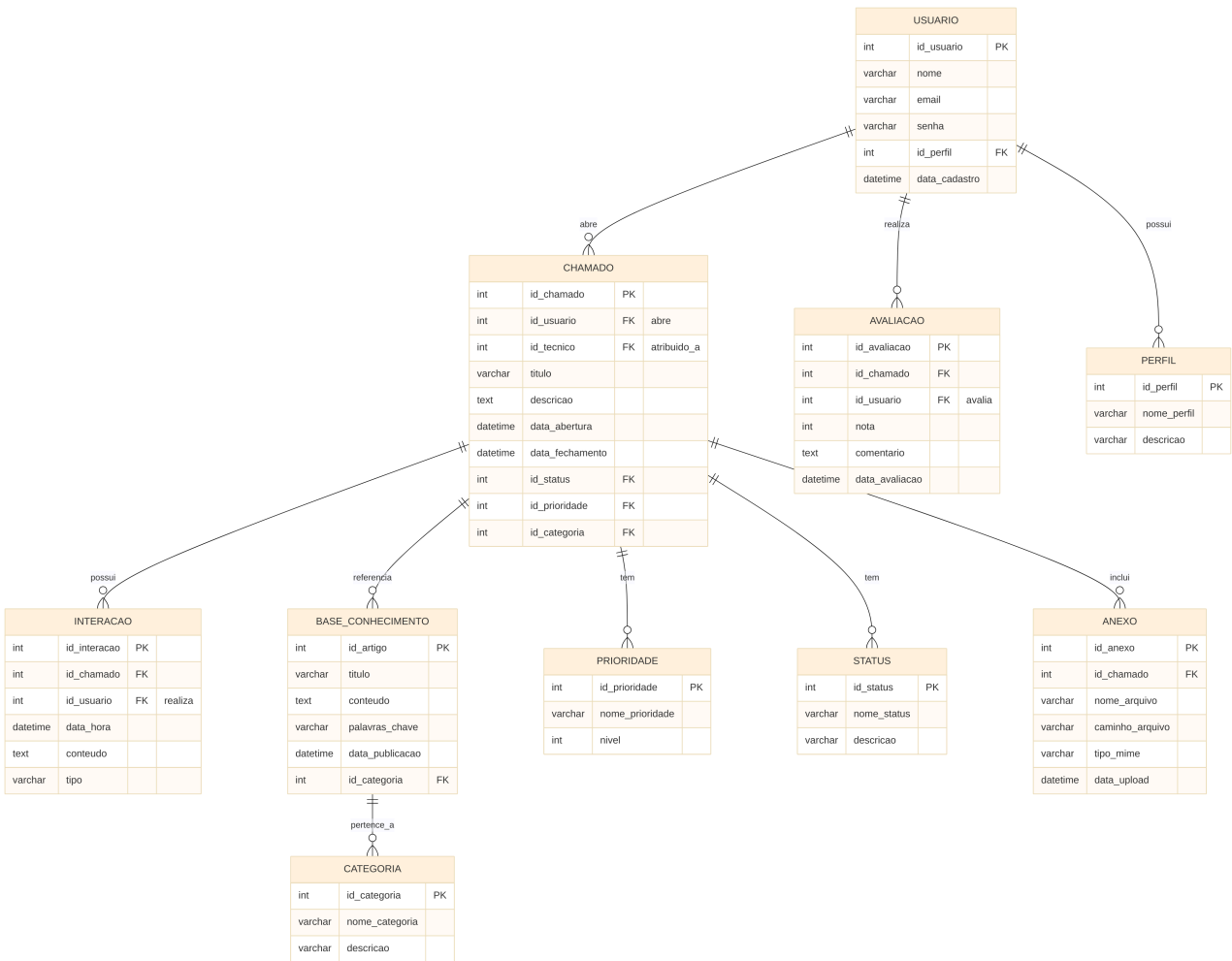


Figura 5: Modelo Entidade-Relacionamento Conceitual do Banco de Dados.

11.2. Dicionário de Dados (Tabelas Principais)

Para detalhar a estrutura do banco de dados, apresentamos um dicionário de dados para as tabelas principais, descrevendo cada campo, seu tipo de dado, restrições e propósito. Esta documentação é crucial para a implementação e manutenção do banco de dados.

Tabela: USUARIO

Campo	Tipo de Dado	Restrições	Descrição	Exemplo
id_usuario	INT	PK , IDENTITY	Identificador único do usuário.	1
nome	VARCHAR(255)	NOT NULL	Nome completo do usuário.	João Silva
email	VARCHAR(255)	NOT NULL , UNIQUE	Endereço de e-mail do usuário, usado para login.	joao.silva@email.com
senha	VARCHAR(255)	NOT NULL	Senha do usuário (armazenada de forma segura, ex: hash).	2a10\$abc...
id_perfil	INT	FK	Chave estrangeira para a tabela PERFIL .	1 (Cliente)
data_cadastro	DATETIME	NOT NULL	Data e hora do cadastro do usuário.	2025-09-23 10:00:00
telefone	VARCHAR(20)	NULL	Telefone de contato do usuário (opcional).	(11) 98765-4321
endereco	VARCHAR(500)	NULL	Endereço completo do usuário (opcional).	Rua A, 123, SP
especialidade	VARCHAR(100)	NULL	Especialidade do técnico (apenas para perfil 'Técnico').	Redes

Campo	Tipo de Dado	Restrições	Descrição	Exemplo
departamento	VARCHAR(100)	NULL	Departamento do gestor (apenas para perfil 'Gestor').	TI

Tabela: PERFIL

Campo	Tipo de Dado	Restrições	Descrição	Exemplo
id_perfil	INT	PK , IDENTITY	Identificador único do perfil.	1
nome_perfil	VARCHAR(50)	NOT NULL , UNIQUE	Nome do perfil de acesso (ex: Cliente, Técnico, Gestor).	Cliente
descricao	VARCHAR(255)	NULL	Descrição detalhada do perfil.	Usuário comum

Tabela: CHAMADO

Campo	Tipo de Dado	Restrições	Descrição	Exemplo
id_chamado	INT	PK , IDENTITY	Identificador único do chamado.	1001
id_usuario	INT	FK	Chave estrangeira para o usuário que abriu o chamado.	1
id_tecnico	INT	FK , NULL	Chave estrangeira para o técnico atribuído ao chamado.	2
titulo	VARCHAR(255)	NOT NULL	Título resumido do problema.	Problema de acesso
descricao	TEXT	NOT NULL	Descrição detalhada do problema.	Não consigo logar...
data_abertura	DATETIME	NOT NULL	Data e hora da abertura do chamado.	2025-09-23 10:15:00
data_fechamento	DATETIME	NULL	Data e hora do fechamento do chamado.	2025-09-23 15:30:00
id_status	INT	FK	Chave estrangeira para o status atual do chamado.	1 (Aberto)
id_prioridade	INT	FK	Chave estrangeira para a prioridade do chamado.	3 (Alta)
id_categoria	INT	FK	Chave estrangeira para a categoria do chamado.	2 (Software)

Tabela: INTERACAO

Campo	Tipo de Dado	Restrições	Descrição	Exemplo
id_interacao	INT	PK , IDENTITY	Identificador único da interação.	1
id_chamado	INT	FK	Chave estrangeira para o chamado relacionado.	1001
id_usuario	INT	FK	Chave estrangeira para o usuário que realizou a interação.	1
data_hora	DATETIME	NOT NULL	Data e hora da interação.	2025-09-23 10:30:00
conteudo	TEXT	NOT NULL	Conteúdo da interação (mensagem, diagnóstico).	Verifiquei o log...
tipo	VARCHAR(50)	NOT NULL	Tipo de interação (ex: Mensagem, Diagnóstico, Solução).	Mensagem

Tabela: BASE_CONHECIMENTO

Campo	Tipo de Dado	Restrições	Descrição	Exemplo
id_artigo	INT	PK , IDENTITY	Identificador único do artigo.	1
titulo	VARCHAR(255)	NOT NULL	Título do artigo ou FAQ.	Como resetar senha
conteudo	TEXT	NOT NULL	Conteúdo completo do artigo.	Para resetar sua...
palavras_chave	VARCHAR(500)	NULL	Palavras-chave para busca (separadas por vírgula).	senha, reset, acesso
data_publicacao	DATETIME	NOT NULL	Data de publicação do artigo.	2025-01-15 09:00:00
id_categoria	INT	FK	Chave estrangeira para a categoria do artigo.	2 (Software)

Observação: Outras tabelas como `AVALIACAO` , `CATEGORIA` , `PRIORIDADE` , `STATUS` e `ANEXO` seguirão um padrão similar de definição, com campos que representam suas características e relacionamentos com as tabelas principais.

11.3. Scripts SQL (Exemplo de Criação de Tabela)

A implementação do banco de dados no MS SQL Server envolverá a criação de scripts SQL para definir o esquema das tabelas, chaves primárias, chaves estrangeiras, índices e restrições de integridade. Abaixo, um exemplo de script para a criação da tabela `USUARIO` :

```

CREATE TABLE USUARIO (
    id_usuario INT PRIMARY KEY IDENTITY(1,1),
    nome VARCHAR(255) NOT NULL,
    email VARCHAR(255) NOT NULL UNIQUE,
    senha VARCHAR(255) NOT NULL,
    id_perfil INT NOT NULL,
    data_cadastro DATETIME NOT NULL DEFAULT GETDATE(),
    telefone VARCHAR(20),
    endereco VARCHAR(500),
    especialidade VARCHAR(100),
    departamento VARCHAR(100),
    FOREIGN KEY (id_perfil) REFERENCES PERFIL(id_perfil)
);

CREATE TABLE PERFIL (
    id_perfil INT PRIMARY KEY IDENTITY(1,1),
    nome_perfil VARCHAR(50) NOT NULL UNIQUE,
    descricao VARCHAR(255)
);

-- Exemplo de inserção de perfis iniciais
INSERT INTO PERFIL (nome_perfil, descricao) VALUES ('Cliente', 'Usuário solicitante de suporte');
INSERT INTO PERFIL (nome_perfil, descricao) VALUES ('Técnico', 'Usuário responsável pelo atendimento de chamados');
INSERT INTO PERFIL (nome_perfil, descricao) VALUES ('Gestor', 'Usuário com acesso a relatórios e gestão do sistema');

```

Este exemplo demonstra a criação de duas tabelas (`USUARIO` e `PERFIL`) e a definição de suas colunas, tipos de dados, chaves primárias e estrangeiras, além de restrições como `NOT NULL` e `UNIQUE`. O uso de `IDENTITY(1,1)` garante que o `id_usuario` e `id_perfil` sejam auto-incrementais. A cláusula `FOREIGN KEY` estabelece o relacionamento entre as tabelas, garantindo a integridade referencial. Scripts similares serão desenvolvidos para todas as tabelas do sistema, garantindo uma base de dados robusta e bem estruturada para o projeto.

12. PROTÓTIPOS DE INTERFACE

Os protótipos de interface são representações visuais das telas do sistema, essenciais para validar o design, a usabilidade e a experiência do usuário antes da fase de implementação. Eles permitem visualizar como os usuários interagirão com o sistema em diferentes plataformas (desktop, mobile e web), garantindo que as funcionalidades sejam acessíveis e intuitivas. Nesta seção, apresentamos os protótipos das principais interfaces do Sistema Integrado para Gestão de Chamados e Suporte Técnico, destacando a consistência visual e a adaptabilidade para cada ambiente.

12.1. Interface Desktop (Administrador)

A interface desktop é projetada para administradores e técnicos, oferecendo um painel de controle completo para gerenciar usuários, chamados, base de conhecimento e configurações do sistema. O foco é na eficiência e na apresentação de informações detalhadas.

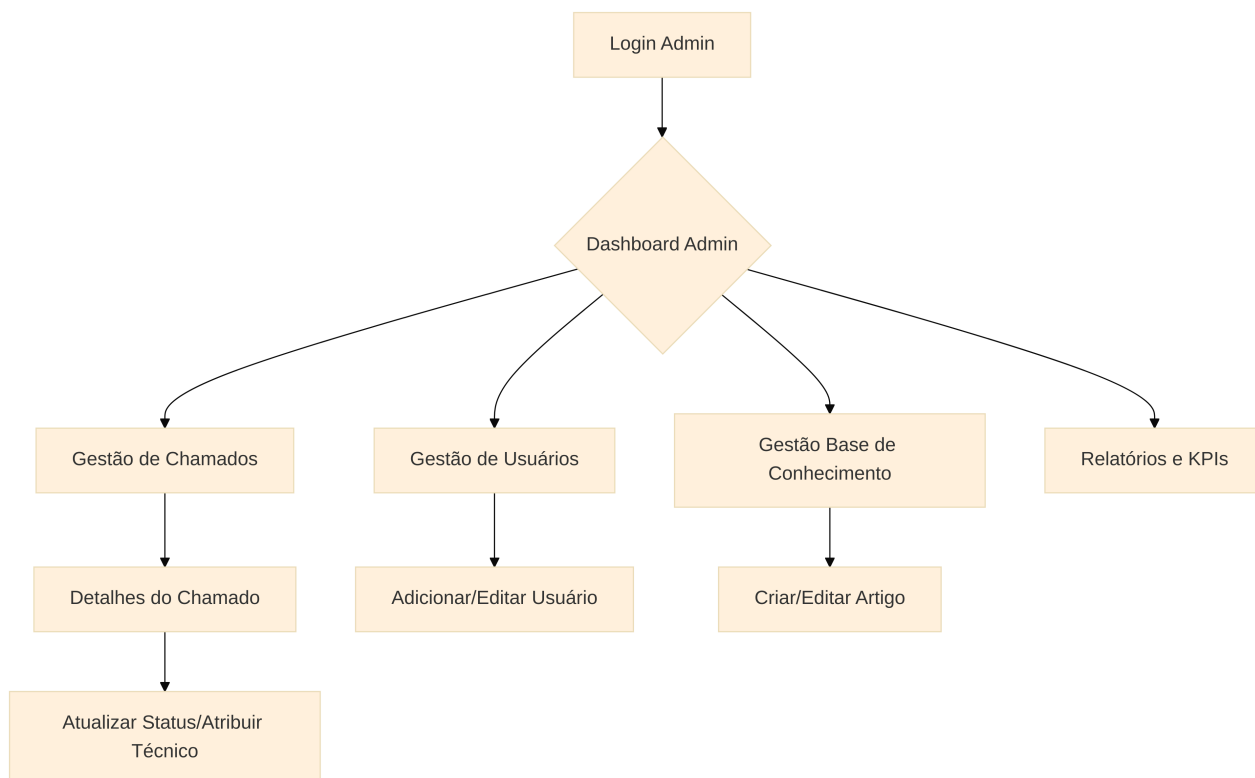


Figura 6: Fluxo de Telas da Interface Desktop (Administrador).

Descrição da Interface Desktop:

- **Dashboard Administrativo:** Visão geral com gráficos e métricas em tempo real sobre o volume de chamados, status, prioridades e desempenho da equipe. Acesso rápido às principais funcionalidades.
- **Gestão de Usuários:** Tela para cadastro, edição, exclusão e gerenciamento de perfis e permissões de acesso de clientes, técnicos e gestores.
- **Gestão de Chamados:** Lista detalhada de todos os chamados, com filtros avançados por status, categoria, prioridade, técnico responsável e cliente. Permite visualizar detalhes do chamado, histórico de interações, anexos e realizar ações como atualização de status, atribuição de técnico e registro de soluções.

- **Gestão da Base de Conhecimento:** Interface para criar, editar e organizar artigos, FAQs e tutoriais que compõem a base de conhecimento, com funcionalidades de busca e categorização.
- **Relatórios e KPIs:** Seção dedicada à geração de relatórios personalizados e visualização de Indicadores Chave de Desempenho, com opções de exportação para PDF e CSV.

12.2. Interface Mobile (Técnico)

A interface mobile é otimizada para técnicos em campo, permitindo o acesso e a atualização de chamados de forma ágil e prática através de smartphones. O design é focado na usabilidade em telas menores e na mobilidade.

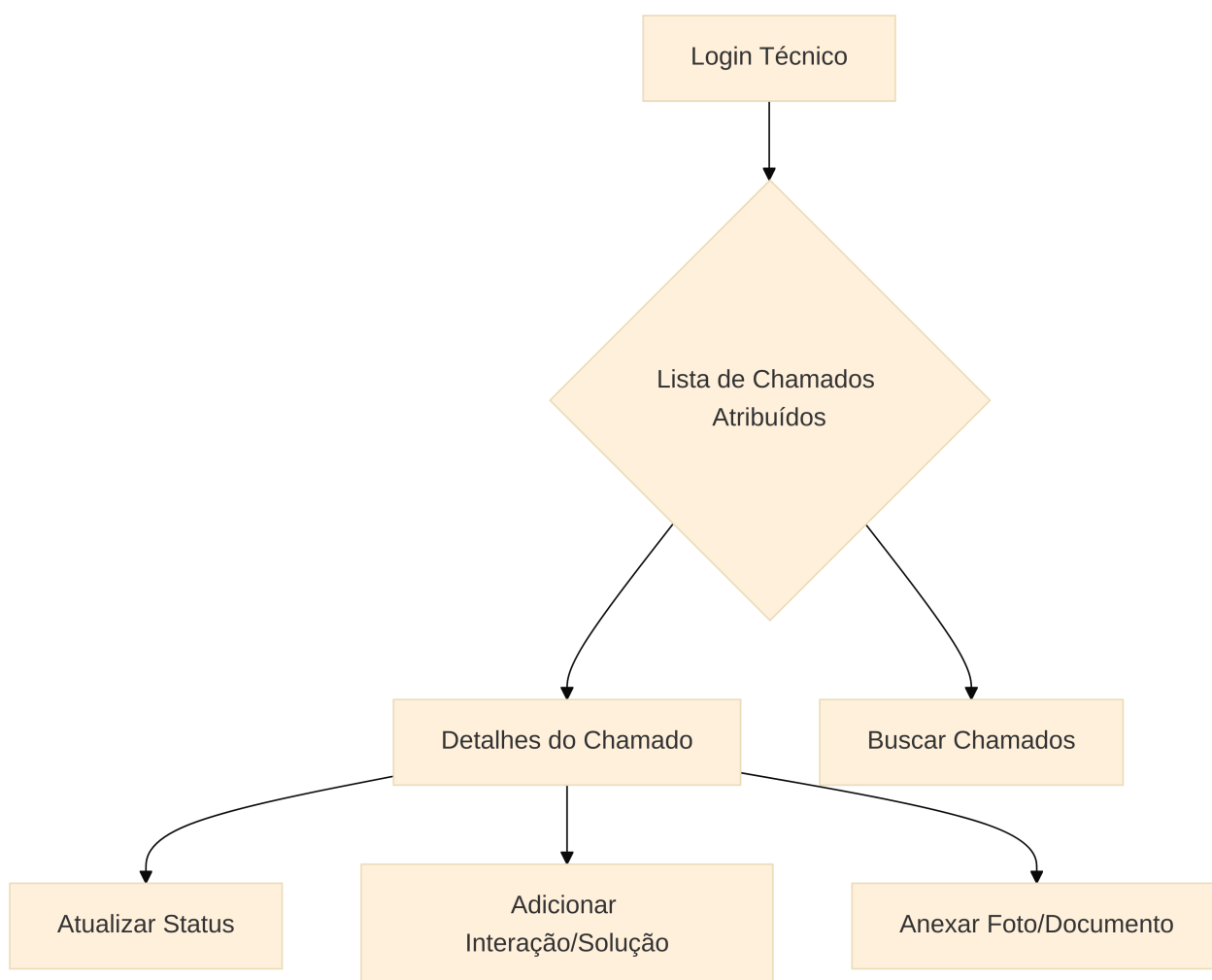


Figura 7: Fluxo de Telas da Interface Mobile (Técnico).

Descrição da Interface Mobile:

- **Lista de Chamados Atribuídos:** Exibe os chamados pendentes e em andamento atribuídos ao técnico, com informações essenciais como título, cliente, prioridade e tempo restante para SLA.
- **Detalhes do Chamado:** Tela com todas as informações do chamado, histórico de interações, anexos e sugestões de solução da IA. Permite ao técnico atualizar o status, adicionar comentários, registrar a solução e anexar fotos ou documentos diretamente do dispositivo móvel.
- **Buscar Chamados:** Funcionalidade de busca rápida para localizar chamados específicos por ID, cliente ou palavra-chave.

12.3. Interface Web (Cliente)

A interface web é destinada aos clientes, proporcionando um portal de autoatendimento para abertura, acompanhamento e avaliação de chamados. O design é limpo, intuitivo e responsivo, garantindo uma boa experiência em qualquer navegador.

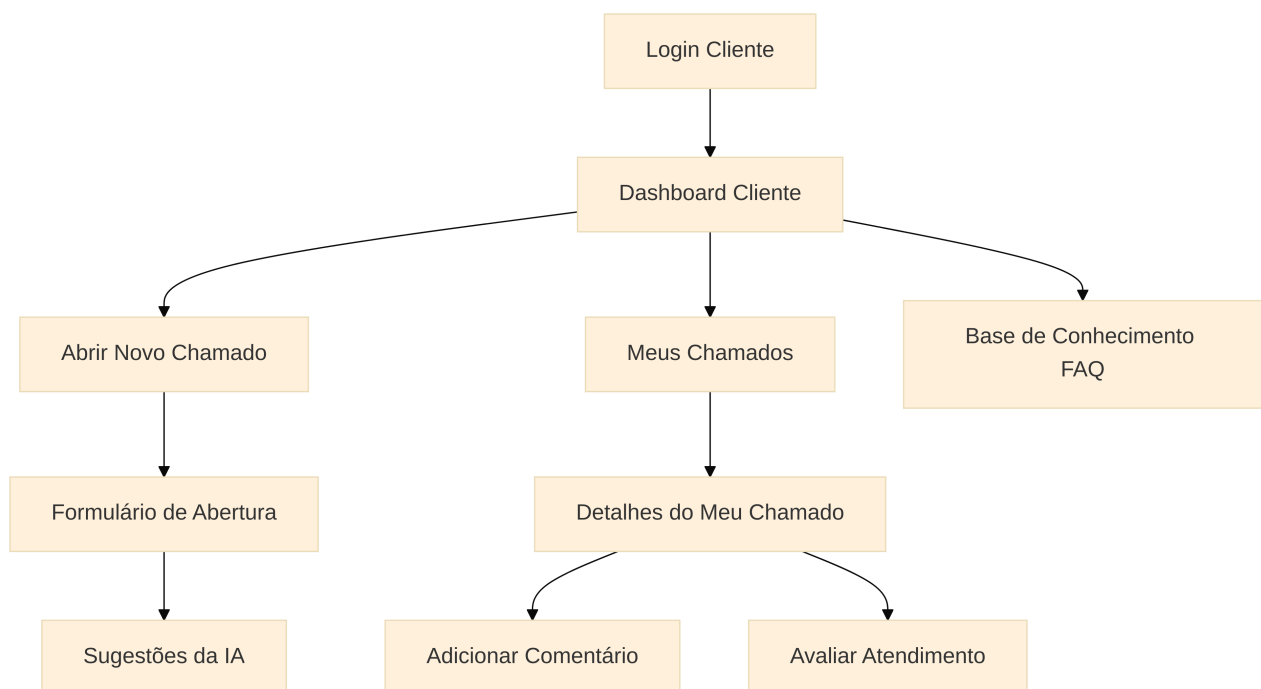


Figura 8: Fluxo de Telas da Interface Web (Cliente).

Descrição da Interface Web:

- **Dashboard do Cliente:** Visão personalizada com o resumo dos chamados abertos, em andamento e resolvidos. Acesso rápido às funcionalidades de

abertura e acompanhamento.

- **Abrir Novo Chamado:** Formulário intuitivo para registrar novas solicitações de suporte. Durante o preenchimento da descrição, o sistema pode apresentar sugestões de solução da base de conhecimento, impulsionadas pela IA, para promover o autoatendimento.
- **Meus Chamados:** Lista de todos os chamados abertos pelo cliente, com detalhes do status, técnico responsável e histórico de interações. Permite adicionar novos comentários e, após a resolução, avaliar o atendimento.
- E[Base de Conhecimento]* Seção de autoatendimento onde o cliente pode pesquisar artigos, tutoriais e perguntas frequentes para tentar resolver seu problema antes de abrir um chamado, reduzindo o volume de solicitações para a equipe de suporte.

13. PLANO DE TESTES

O plano de testes é um documento essencial que descreve a estratégia, o escopo, os recursos e o cronograma das atividades de teste para garantir a qualidade do software. Ele assegura que o Sistema Integrado para Gestão de Chamados e Suporte Técnico atenda aos requisitos funcionais e não funcionais, seja robusto, seguro e ofereça uma boa experiência ao usuário. A estratégia de testes será dividida em diferentes níveis, cada um com objetivos específicos.

13.1. Níveis de Teste

1. Testes de Unidade:

- **Objetivo:** Verificar se cada componente individual do software (classes, métodos) funciona conforme o esperado.
- **Escopo:** Serão testadas as classes da camada de lógica de negócio e da camada de acesso a dados. Por exemplo, testar se o método `salvarChamado()` realmente persiste os dados no banco de dados, ou se o método `calcularPrioridade()` retorna o valor correto.
- **Ferramentas:** Utilização de frameworks de teste de unidade como **xUnit** ou **NUnit** para C#.

2. Testes de Integração:

- **Objetivo:** Verificar a interação entre os diferentes componentes do sistema, garantindo que eles funcionem corretamente em conjunto.
- **Escopo:** Testar a integração entre a camada de lógica de negócio e a camada de acesso a dados, a integração com o módulo de IA e a comunicação entre as interfaces e o backend. Por exemplo, testar se um chamado aberto na interface web é corretamente processado pela lógica de negócio, classificado pela IA e salvo no banco de dados.

3. Testes de Sistema:

- **Objetivo:** Validar o sistema como um todo, verificando se ele atende a todos os requisitos funcionais e não funcionais especificados.
- **Escopo:** Realização de testes end-to-end, simulando cenários de uso reais. Isso inclui testar todas as funcionalidades em todas as plataformas (web, desktop, mobile), bem como os requisitos não funcionais como desempenho, segurança e usabilidade.

4. Testes de Aceitação do Usuário (UAT):

- **Objetivo:** Validar se o sistema atende às necessidades e expectativas dos usuários finais.
- **Escopo:** Um grupo de usuários selecionados (clientes, técnicos, gestores) utilizará o sistema em um ambiente de homologação para realizar suas tarefas diárias e fornecer feedback. O UAT é a última etapa antes da implantação em produção.

13.2. Tipos de Teste

- **Testes Funcionais:** Verificam se as funcionalidades do sistema se comportam conforme o esperado. Ex: testar se um usuário consegue abrir um chamado, se um técnico consegue atualizá-lo, se um gestor consegue gerar um relatório.
- **Testes de Usabilidade:** Avaliam a facilidade de uso e a intuitividade das interfaces. Serão realizados por meio de observação de usuários e coleta de feedback.
- **Testes de Desempenho:** Medem o tempo de resposta, a escalabilidade e a estabilidade do sistema sob diferentes cargas de trabalho. Serão utilizados ferramentas de teste de carga para simular múltiplos usuários acessando o sistema simultaneamente.

- **Testes de Segurança:** Identificam vulnerabilidades e garantem que o sistema esteja protegido contra ameaças. Serão realizados testes de penetração, análise de vulnerabilidades e verificação da conformidade com a LGPD.

13.3. Roteiro de Testes (Exemplo)

A seguir, um exemplo de roteiro de teste para a funcionalidade de "Abertura de Chamado":

ID do Teste	Descrição do Teste	Passos para Execução	Resultado Esperado	Status (Passou/Falhou)
TC-001	Abertura de chamado com sucesso (todos os campos preenchidos).	1. Fazer login como cliente. 2. Acessar a tela "Abrir Novo Chamado". 3. Preencher todos os campos obrigatórios. 4. Clicar em "Salvar".	O chamado é criado com sucesso. O cliente recebe uma notificação. O chamado aparece na lista "Meus Chamados" com o status "Aberto".	
TC-002	Tentativa de abertura de chamado sem preencher campos obrigatórios.	1. Fazer login como cliente. 2. Acessar a tela "Abrir Novo Chamado". 3. Deixar um campo obrigatório em branco. 4. Clicar em "Salvar".	O sistema exibe uma mensagem de erro indicando qual campo obrigatório não foi preenchido. O chamado não é criado.	
TC-003	Abertura de chamado com anexo.	1. Fazer login como cliente. 2. Acessar a tela "Abrir Novo Chamado". 3. Preencher todos os campos. 4. Anexar um arquivo (imagem ou PDF). 5. Clicar em "Salvar".	O chamado é criado com sucesso. O anexo está disponível para visualização nos detalhes do chamado.	
TC-004	Verificação da classificação automática por IA.	1. Fazer login como cliente. 2. Abrir um chamado com a descrição "Minha impressora não está funcionando". 3. Verificar os detalhes do chamado criado.	O chamado é automaticamente classificado na categoria "Hardware" e com a prioridade adequada (ex: "Média").	

Este plano de testes abrangente garante que o sistema seja entregue com alta qualidade, atendendo a todas as especificações e proporcionando uma experiência

positiva para o usuário.

14. PLANO DE IMPLANTAÇÃO

O plano de implantação descreve os passos necessários para instalar, configurar e colocar o Sistema Integrado para Gestão de Chamados e Suporte Técnico em um ambiente de produção, garantindo uma transição suave do ambiente de desenvolvimento para o ambiente de uso real. Este plano visa minimizar o tempo de inatividade e os riscos associados à implantação, assegurando que o sistema esteja totalmente operacional para os usuários finais.

14.1. Requisitos de Ambiente

Antes da implantação, é necessário garantir que o ambiente de produção atenda aos seguintes requisitos de hardware e software:

- **Servidor de Aplicação Web:**
 - Sistema Operacional: Windows Server 2019 ou superior.
 - Servidor Web: IIS (Internet Information Services) 10 ou superior.
 - Framework: .NET 8 (ou a versão mais recente utilizada no desenvolvimento).
 - Recursos: Mínimo de 4 GB de RAM, 2 vCPUs, 50 GB de espaço em disco.
- **Servidor de Banco de Dados:**
 - Sistema Operacional: Windows Server 2019 ou superior.
 - SGBD: MS SQL Server 2019 ou superior.
 - Recursos: Mínimo de 8 GB de RAM, 4 vCPUs, 100 GB de espaço em disco (com plano de crescimento).
- **Estações de Trabalho (Desktop):**
 - Sistema Operacional: Windows 10 ou superior.
 - Framework: .NET 8 (ou a versão correspondente).
- **Dispositivos Móveis:**

- Sistema Operacional: Android 8.0 (Oreo) ou superior.

14.2. Estratégia de Implantação

A estratégia de implantação adotada será a **implantação em fases (phased deployment)**. Nesta abordagem, o sistema será liberado inicialmente para um grupo piloto de usuários (um departamento ou uma equipe específica) antes de ser disponibilizado para toda a organização. Isso permite coletar feedback, identificar e corrigir problemas em um ambiente controlado, e ajustar o plano de treinamento antes da liberação em larga escala.

14.3. Passos da Implantação

1. Preparação do Ambiente:

- Configurar os servidores de aplicação web e de banco de dados conforme os requisitos especificados.
- Instalar e configurar o MS SQL Server, IIS e o .NET Runtime.
- Configurar as regras de firewall e as políticas de segurança necessárias.

2. Implantação do Banco de Dados:

- Executar os scripts SQL para criar o banco de dados, as tabelas, os relacionamentos e os índices.
- Executar scripts de carga inicial para popular tabelas de dados básicos (ex: perfis, categorias, status, prioridades).

3. Implantação da Aplicação (Backend e Frontend Web):

- Publicar a aplicação web (ASP.NET Core) no servidor IIS.
- Configurar a string de conexão com o banco de dados no arquivo de configuração da aplicação.
- Testar a conectividade entre a aplicação web e o banco de dados.

4. Distribuição das Aplicações Cliente (Desktop e Mobile):

- Empacotar a aplicação desktop em um instalador (ex: MSIX) e disponibilizá-la em um compartilhamento de rede ou via ferramenta de distribuição de software.

- Publicar a aplicação mobile (APK para Android) em uma loja de aplicativos interna ou disponibilizá-la para download em um portal seguro.

5. Fase Piloto:

- Selecionar um grupo de usuários para participar da fase piloto.
- Realizar o treinamento deste grupo e fornecer suporte intensivo.
- Coletar feedback sobre a usabilidade, desempenho e funcionalidades do sistema.
- Monitorar o sistema em busca de erros e problemas de desempenho.

6. Ajustes e Correções:

- Analisar o feedback da fase piloto e corrigir os problemas identificados.
- Realizar os ajustes necessários na configuração do sistema ou na aplicação.

7. Implantação Completa:

- Comunicar a todos os usuários sobre a data e o cronograma da implantação completa.
- Realizar o treinamento de todos os usuários em ondas.
- Disponibilizar o acesso ao sistema para toda a organização.

8. Pós-Implantação:

- Monitorar continuamente o desempenho e a estabilidade do sistema.
- Fornecer suporte contínuo aos usuários.
- Planejar futuras atualizações e melhorias com base no feedback contínuo e nas novas necessidades do negócio.

14.4. Plano de Rollback (Contingência)

Em caso de falha crítica durante a implantação, um plano de rollback será acionado para reverter o sistema ao estado anterior. Os passos incluem:

- Desativar o acesso ao novo sistema.
- Restaurar o backup do banco de dados realizado antes da implantação.
- Reverter a aplicação para a versão anterior (se aplicável).

- Analisar a causa da falha em um ambiente de testes antes de tentar uma nova implantação.

Este plano de implantação estruturado visa garantir uma transição bem-sucedida e a adoção eficaz do novo sistema de gestão de chamados por toda a organização.

15. CONSIDERAÇÕES SOBRE LGPD E SEGURANÇA

A proteção de dados pessoais e a segurança da informação são aspectos cruciais no desenvolvimento de qualquer sistema, especialmente aqueles que tratam informações sensíveis de usuários, como é o caso do Sistema Integrado para Gestão de Chamados e Suporte Técnico. A conformidade com a Lei Geral de Proteção de Dados (LGPD) e a implementação de práticas robustas de segurança são requisitos não funcionais de alta prioridade, visando garantir a privacidade dos indivíduos e a integridade dos dados.

15.1. Conformidade com a Lei Geral de Proteção de Dados (LGPD)

A LGPD (Lei nº 13.709/2018) estabelece diretrizes claras para a coleta, armazenamento, tratamento e compartilhamento de dados pessoais, impondo obrigações a todas as organizações que operam no Brasil. Para o sistema proposto, a conformidade com a LGPD será assegurada através das seguintes medidas:

- **Princípios da LGPD:** O desenvolvimento e a operação do sistema estarão alinhados aos princípios da LGPD, como finalidade, adequação, necessidade, livre acesso, qualidade dos dados, transparência, segurança, prevenção, não discriminação e responsabilização e prestação de contas.
- **Consentimento Explícito:** Para a coleta e o tratamento de dados pessoais que não se enquadrem em outras bases legais (como cumprimento de obrigação legal ou execução de contrato), será obtido o consentimento explícito e inequívoco do titular dos dados. Este consentimento será registrado e armazenado de forma segura.
- **Anonimização e Pseudonimização:** Sempre que possível e aplicável, dados pessoais serão anonimizados ou pseudonimizados para reduzir o risco de identificação dos titulares, especialmente em cenários de análise estatística ou geração de relatórios.

- **Direitos dos Titulares:** O sistema será projetado para facilitar o exercício dos direitos dos titulares, como acesso aos dados, correção, exclusão, portabilidade e revogação do consentimento. Mecanismos claros serão implementados para que os usuários possam exercer esses direitos de forma simples e eficaz.
- **Registro de Atividades de Tratamento:** Todas as operações de tratamento de dados pessoais (coleta, acesso, modificação, exclusão) serão registradas em logs de auditoria, permitindo a rastreabilidade e a prestação de contas em caso de incidentes.
- **Relatório de Impacto à Proteção de Dados (RIPD):** Será elaborada uma análise de risco e um RIPD para identificar e mitigar potenciais riscos à privacidade e aos direitos dos titulares de dados, especialmente em relação ao módulo de IA que processa informações textuais dos chamados.

15.2. Medidas de Segurança da Informação

Além da conformidade legal, o sistema implementará um conjunto abrangente de medidas técnicas e organizacionais para garantir a segurança da informação, protegendo os dados contra acessos não autorizados, vazamentos, alterações indevidas ou destruição. As principais medidas incluem:

- **Controle de Acesso Baseado em Papéis (RBAC):** O acesso às funcionalidades e aos dados do sistema será rigorosamente controlado com base nos perfis de usuário (Cliente, Técnico, Gestor), garantindo que cada indivíduo tenha acesso apenas às informações e operações estritamente necessárias para suas atribuições (princípio do menor privilégio).
- **Autenticação Segura:** Implementação de mecanismos de autenticação robustos, como senhas fortes (com requisitos de complexidade, expiração e histórico), e, idealmente, suporte a autenticação multifator (MFA) para perfis de maior privilégio (técnicos e gestores).
- **Criptografia de Dados:**
 - **Dados em Trânsito:** Toda a comunicação entre as interfaces do sistema (web, desktop, mobile) e o servidor será protegida por criptografia TLS/SSL (HTTPS), impedindo a interceptação e a leitura dos dados por terceiros.
 - **Dados em Repouso:** Dados sensíveis armazenados no banco de dados (como senhas, informações de contato) serão criptografados ou

armazenados de forma hash (para senhas), protegendo-os mesmo em caso de acesso não autorizado ao banco de dados.

- **Validação e Sanitização de Entradas:** Todas as entradas de dados fornecidas pelos usuários serão rigorosamente validadas e sanitizadas para prevenir ataques comuns, como Injeção de SQL, Cross-Site Scripting (XSS) e outros tipos de vulnerabilidades de segurança.
- **Logs de Auditoria e Monitoramento:** O sistema registrará detalhadamente todas as ações relevantes dos usuários e eventos do sistema, incluindo tentativas de login, acessos a dados sensíveis, alterações em chamados e configurações. Esses logs serão monitorados para detectar atividades suspeitas e auxiliar na investigação de incidentes.
- **Backup e Recuperação de Dados:** Serão implementadas políticas de backup regulares e testadas para o banco de dados e arquivos do sistema, garantindo a capacidade de recuperação em caso de perda de dados ou desastres.
- **Atualizações de Segurança:** O sistema e suas dependências (frameworks, bibliotecas, sistema operacional do servidor) serão mantidos atualizados com os patches de segurança mais recentes para mitigar vulnerabilidades conhecidas.
- **Treinamento e Conscientização:** Os usuários do sistema, especialmente técnicos e gestores, receberão treinamento sobre as políticas de segurança da informação e as melhores práticas para a proteção de dados, reforçando a cultura de segurança na organização.

Ao integrar a conformidade com a LGPD e um conjunto abrangente de medidas de segurança desde a concepção do projeto, o Sistema Integrado para Gestão de Chamados e Suporte Técnico será construído sobre uma base sólida de confiança e proteção, essencial para o tratamento de informações sensíveis e para a credibilidade da solução.

16. CONCLUSÃO

O presente Projeto Integrado Multidisciplinar (PIM) demonstrou a viabilidade e a relevância do desenvolvimento de um **Sistema Integrado para Gestão de Chamados e Suporte Técnico Baseado em Inteligência Artificial**. Ao longo deste trabalho, foram detalhados os aspectos fundamentais para a concepção de uma solução robusta, escalável e eficiente, capaz de transformar a maneira como as organizações gerenciam suas demandas de suporte técnico.

A análise aprofundada dos requisitos funcionais e não funcionais, aliada à modelagem detalhada por meio de diagramas UML (Casos de Uso, Classes e Sequência), permitiu a criação de uma arquitetura de sistema em camadas bem definida. Esta arquitetura, que integra interfaces multiplataforma (web, desktop e mobile) com um módulo de Inteligência Artificial e um banco de dados MS SQL Server, reflete o compromisso com as tecnologias e diretrizes estabelecidas pelo manual da faculdade.

A incorporação da Inteligência Artificial, com foco em Processamento de Linguagem Natural (PLN), é o diferencial estratégico do sistema. A capacidade de classificar e priorizar chamados automaticamente, bem como de sugerir soluções com base em uma base de conhecimento dinâmica, promete uma redução significativa no tempo médio de resolução, otimizando a alocação de recursos humanos e promovendo o autoatendimento. Os protótipos de interface apresentados ilustram a preocupação com a usabilidade e a experiência do usuário em cada plataforma.

Adicionalmente, o projeto enfatizou a importância da conformidade com a **Lei Geral de Proteção de Dados (LGPD)** e a implementação de rigorosas medidas de segurança da informação. Isso garante que o tratamento de dados pessoais seja realizado com a máxima responsabilidade e transparência, construindo a confiança dos usuários e mitigando riscos legais.

Em suma, este PIM não apenas propõe uma solução tecnológica inovadora para um problema real, mas também serve como um arcabouço teórico e prático para o desenvolvimento de sistemas complexos. A metodologia adotada, o detalhamento técnico e a preocupação com a qualidade e a segurança posicionam o projeto como um modelo para futuras implementações no campo da gestão de serviços de TI com IA. A expectativa é que a materialização deste sistema traga benefícios tangíveis, como o aumento da produtividade, a melhoria da satisfação do usuário e a otimização contínua dos processos de suporte técnico.

REFERÊNCIAS

[1] PRESSMAN, Roger S. **Engenharia de Software: Uma Abordagem Profissional**. 8. ed. Porto Alegre: AMGH, 2016.

[2] SOMMERVILLE, Ian. **Engenharia de Software**. 10. ed. São Paulo: Pearson Education do Brasil, 2019.

- [3] BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML Guia do Usuário**. 2. ed. Rio de Janeiro: Campus, 2005.
- [4] RUSSELL, Stuart J.; NORVIG, Peter. **Inteligência Artificial: Uma Abordagem Moderna**. 3. ed. Rio de Janeiro: Elsevier, 2013.
- [5] JURAFSKY, Daniel; MARTIN, James H. **Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition**. 3. ed. Upper Saddle River, NJ: Prentice Hall, 2023. Disponível em: <https://web.stanford.edu/~jurafsky/slp3/>. Acesso em: 23 set. 2025.
- [6] ALPAYDIN, Ethem. **Introduction to Machine Learning**. 4. ed. Cambridge, MA: MIT Press, 2020.
- [7] CHEN, Hsinchun. **Information Retrieval: Still at the Crossroads**. In: Annual Review of Information Science and Technology, v. 34, p. 3-57, 1999.
- [8] DEITEL, Paul J.; DEITEL, Harvey M. **C# 2012 e Programação com .NET 4.5**. 5. ed. São Paulo: Pearson Education do Brasil, 2014.
- [9] ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de Banco de Dados**. 7. ed. São Paulo: Pearson Education do Brasil, 2018.
- [10] BRASIL. Lei nº 13.709, de 14 de agosto de 2018. **Lei Geral de Proteção de Dados Pessoais (LGPD)**. Diário Oficial da União, Brasília, DF, 14 ago. 2018. Disponível em: http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/L13709.htm. Acesso em: 23 set. 2025.

ATIVIDADE DE EXTENSÃO UNIVERSITÁRIA

A extensão universitária representa um pilar fundamental na formação acadêmica, promovendo a interação entre a instituição de ensino e a comunidade externa. Ela permite que os conhecimentos adquiridos em sala de aula sejam aplicados na resolução de problemas reais, gerando impacto social e enriquecendo a experiência dos estudantes. Conforme as diretrizes da Universidade Paulista (UNIP), a atividade de extensão universitária deve ser uma ponte permanente entre a universidade e a sociedade, contribuindo para o desenvolvimento mútuo.

Proposta de Atividade de Extensão

Considerando o escopo do Sistema Integrado para Gestão de Chamados e Suporte Técnico Baseado em Inteligência Artificial, propõe-se a realização de uma atividade de extensão com foco na **capacitação e demonstração tecnológica para pequenas empresas, escolas públicas ou Organizações Não Governamentais (ONGs)**. Esta iniciativa visa compartilhar o conhecimento desenvolvido no projeto, oferecendo um benefício direto à comunidade e reforçando o papel social da universidade.

Objetivos da Atividade:

- **Demonstrar o Sistema:** Apresentar o funcionamento do sistema de gestão de chamados com IA, destacando suas funcionalidades e os benefícios que pode trazer para a otimização do suporte técnico em pequenas organizações.
- **Capacitar em Boas Práticas:** Oferecer uma oficina prática sobre boas práticas na gestão de chamados, segurança da informação e a importância da conformidade com a LGPD no tratamento de dados.
- **Promover a Conscientização:** Sensibilizar os participantes sobre o potencial da Inteligência Artificial para otimizar processos e a importância da proteção de dados no ambiente digital.
- **Coletar Feedback:** Obter feedback dos participantes sobre as necessidades de suporte técnico em suas organizações, o que pode gerar insights para futuras melhorias no sistema.

Público-Alvo:

- Gestores e colaboradores de pequenas empresas locais.
- Equipes administrativas e de TI de escolas públicas.
- Membros de ONGs que necessitam otimizar seus processos de suporte interno.

Formato da Atividade:

A atividade será estruturada como uma **oficina interativa**, com duração de aproximadamente 2 a 4 horas, podendo ser realizada presencialmente (em espaços da universidade ou nas próprias instituições parceiras) ou online, dependendo da logística e do alcance desejado. A oficina incluirá:

1. **Apresentação Teórica:** Breve introdução sobre gestão de chamados, suporte técnico, conceitos básicos de IA e LGPD.

2. **Demonstração Prática:** Apresentação ao vivo do sistema desenvolvido, mostrando as interfaces web, desktop e mobile, e como a IA atua na classificação, priorização e sugestão de soluções.
3. **Sessão de Perguntas e Respostas:** Espaço aberto para dúvidas e discussões com os participantes.
4. **Material de Apoio:** Distribuição de material informativo (digital ou impresso) com os principais conceitos abordados e um guia rápido de uso do sistema (versão demonstrativa, se aplicável).

Evidências da Atividade

Cada membro do grupo deverá registrar sua participação individual na atividade de extensão, documentando as ações realizadas e a contribuição para a comunidade. As evidências podem incluir:

- **Relatório Individual:** Descrição detalhada da participação, responsabilidades e aprendizados.
- **Registro Fotográfico:** Fotos da oficina, com consentimento dos participantes.
- **Infográficos:** Criação de infográficos que resumam os conceitos apresentados ou o fluxo de uso do sistema.
- **Vídeos:** Gravação de trechos da demonstração ou depoimentos dos participantes (com autorização).
- **Publicações:** Possibilidade de criação de pequenos artigos ou posts em blogs/redes sociais sobre a experiência.

Esta atividade de extensão não só valida a aplicação prática dos conhecimentos adquiridos no PIM, mas também fortalece o vínculo entre a academia e a sociedade, promovendo a inovação e o desenvolvimento tecnológico em benefício da comunidade.