**UNIVERSIDADE PAULISTA**

**SISTEMA INTEGRADO PARA GESTÃO DE CHAMADAS E SUPORTE TÉCNICO BASEADO EM IA**

Projeto integrado multidisciplinar – 4º Semestre

Bianca biazetto silveira

João ANTONIO do nascimento filho

JULIO CESAR TAVARES BERNARDO DA SILVA

VALDECYR GABRYEL BENTO

**assis**

**2025**

Bianca biazetto silveira

João ANTONIO do nascimento filho

JULIO CESAR TAVARES BERNARDO DA SILVA

VALDECYR GABRYEL BENTO

**SISTEMA INTEGRADO PARA GESTÃO DE CHAMADAS E SUPORTE TÉCNICO BASEADO EM IA**

Projeto integrado multidisciplinar

Projeto Integrador Multidisciplinar do 4º Semestre do curso de Tecnologia em Análise e Desenvolvimento de Sistemas apresentado à Universidade Paulista – UNIP.

Orientador Professor: Flavio Alves

**assis**

**2025**

**RESUMO**

Este trabalho tem o foco da implementação concreta de uma solução de suporte técnico baseada em Inteligência Artificial, voltada para a melhoria do atendimento de chamados em ambientes corporativos. O projeto propõe a entrega efetiva de uma plataforma funcional que utiliza IA para automatizar processos, agilizar diagnósticos e fornecer respostas mais precisas aos usuários. A solução contará com um sistema inteligente capaz de realizar triagens iniciais e encaminhar chamados de forma eficiente. A proposta também contempla o desenvolvimento de uma interface acessível em diferentes dispositivos, como computadores e celulares, garantindo flexibilidade e usabilidade. A aplicação será construída com base em metodologias ágeis, permitindo adaptações contínuas e alinhamento com as necessidades reais do setor de TI. Com isso, o suporte técnico deixa de ser apenas uma área reativa e passa a atuar estrategicamente, contribuindo diretamente para o desempenho das equipes e o crescimento das empresas. O uso da inteligência artificial será central para transformar o atendimento em uma experiência mais rápida, inteligente e integrada, elevando o papel do suporte dentro da organização.

**Palavras-chave:** Inteligência Artificial, Atendimento de chamados, Suporte técnico, TI, Automação de processos.

**ABSTRACT**

This work focuses on the concrete implementation of a technical support solution based on Artificial Intelligence, aimed at improving the handling of service requests in corporate environments. The project proposes the effective delivery of a functional platform that uses AI to automate processes, accelerate diagnostics, and provide more accurate responses to users. The solution will feature an intelligent system capable of performing initial triage and efficiently routing service requests. The proposal also includes the development of an interface accessible across multiple devices, such as computers and mobile phones, ensuring flexibility and usability. The application will be built using agile methodologies, allowing for continuous adaptations and alignment with the real needs of the IT sector. As a result, technical support evolves from being merely reactive to becoming a strategic function, directly contributing to team performance and business growth. The use of artificial intelligence will be central to transforming support into a faster, smarter, and more integrated experience, elevating its role within the organization.

**Keywords**: Artificial Intelligence, Service Request Management, Technical Support, IT, Process Automation.

**SUMÁRIO**

|  |  |  |
| --- | --- | --- |
| **1** | **INTRODUÇÃO.....................................................................................** | 6 |
| **2** | **Desenvolvimento de Software para Internet...................................** | 7 |
| **3** | **Gerenciamento de Projetos de Software.........................................** | 8 |
| **4** | **Programação Orientada a Objetos II................................................** | 9 |
| **5** | **Projeto de Sistemas Orientado a Objetos........................................** | 10 |
| **5.1** | **UML .......................................................................................................10** |  |
| **6** | **Tópicos Especiais de Programação Orientada a Objetos..............** | 11 |
| **7** | **Marketing Pessoal..............................................................................** | 12 |
| **8** | **Empreendedorismo............................................................................** | 13 |
| **9**  **10**  **11** | **Gestão de Qualidade..........................................................................**  **Relações Étnico-raciais e Afrodescendência..................................**  **CONSIDERAÇÕES FINAIS.................................................................** | 14  15  18 |
| **13** | **REFERÊNCIAS....................................................................................** | 19 |

1. **INTRODUÇÃO**

Este trabalho de final de curso tem com o objetivo o desenvolvimento de um sistema em três plataforma (Desktop, Mobile, Web) para uma empresa de pequeno porte da área de abertura de chamados. Utilizaremos a linguagem C# e o framework ASP.NET para o desenvolvimento do Desktop, para um projeto mais aceito no mercado utilizaremos o padrão de arquitetura MVC (Model, View, Controller) e a ferramenta utilizada será o Visual Studio 2022.

Para o aplicativo em Mobile iremos utilizar a linguagem em Java junto com Kotlin, o dispositivo que iremos utilizar será o Android Studio.

Já para a versão em Web iremos utilizar a linguagem HTML com a junção do SCC e um pouco de Java Script, como no desktop será utilizado a ferramenta Visual Studio (2022).

Assim os três aplicativos serão ligados juntos em um banco de dados feito pelo SQL Server.

Algumas matérias nos ajudam a entender que para se realizar um projeto desse nível vai muito mais além doque apenas linhas de código, como várias estratégias para facilitar o entendimento do desenvolvedor assim como a matéria de gerenciamento de projeto de software, uma área que mostra como um líder organiza sua equipe e mostram ferramentas e métodos para isso. E também a área de Projeto de Sistemas Orientado a Objetos que auxiliam nas montagens de diagramas.

As etapas a seguir do Projeto Integrado Multidisciplinar apresentam definições e detalhes sobre os diversos conceitos utilizados ao longo do trabalho e das matérias estudadas como base. Com o objetivo de ajudar na compreensão do estudo desenvolvido, são explicados nos próximos tópicos os principais termos para entender o conceito de inovação e seus diferentes tipos e classificações. Além disso, são mostradas técnicas e métodos usados para fundamentar o projeto visando o aprimoramento do ambiente organizacional. Apresentando então a base teórica do trabalho, elaborada a partir de uma pesquisa aprofundada em artigos, estudos e publicações de autores da área.

1. **DESENVOLVIMENTO DE SOFTWARE PARA INTERNET**

A disciplina de Desenvolvimento de Software para Internet tem como objetivo capacitar os estudantes a projetarem, implementar e gerenciar sistemas web modernos, aplicando conceitos de programação, arquitetura de software e usabilidade. O avanço da tecnologia e a crescente demanda por soluções digitais interativas impulsionam a criação de aplicações cada vez mais inteligentes, capazes de atender às necessidades do usuário de forma rápida e eficiente.

O projeto TechSupport surge nesse contexto como uma plataforma inovadora voltada à abertura e gerenciamento de chamados técnicos, integrando Inteligência Artificial (IA) para otimizar o atendimento e a resolução de problemas. A utilização de IA no desenvolvimento web permite automatizar processos, compreender solicitações dos usuários e oferecer soluções personalizadas em tempo real, aumentando a produtividade e reduzindo falhas humanas.

Segundo Sommerville (2019), o desenvolvimento de software deve considerar não apenas os requisitos funcionais, mas também a qualidade, a confiabilidade e a experiência do usuário, aspectos fundamentais em sistemas baseados na web. Dessa forma, o TechSupport foi planejado com base em boas práticas de engenharia de software, utilizando tecnologias de front-end e back-end adequadas, além de aplicar princípios de design responsivo e segurança da informação. Assim, o projeto representa a aplicação prática dos conhecimentos adquiridos na disciplina, integrando teoria e prática para criar uma solução tecnológica moderna e eficiente, alinhada às demandas atuais do mercado e às tendências da transformação digital.

1. **GERENCIAMENTO DE PROJETOS DE SOFTWARE**

O gerenciamento de projetos de software tem se consolidado como um dos pilares fundamentais para o sucesso de empresas que atuam com tecnologia. Em um ambiente marcado por mudanças constantes e alta competitividade, saber conduzir projetos com eficiência deixou de ser um diferencial e passou a ser uma necessidade. A capacidade de planejar, adaptar e entregar soluções de forma ágil está diretamente ligada à sobrevivência e ao crescimento das organizações.

Cassol, Gonçalo e Ruas (2022) destacam que frameworks ágeis, como Scrum e Kanban, têm contribuído para tornar as empresas mais flexíveis e preparadas para lidar com demandas imprevisíveis. Essa abordagem é especialmente relevante em startups, onde o tempo de resposta pode ser decisivo. Já Stettiner e Rajão (2021) analisam como iniciativas como o MPS.BR ajudam empresas a estruturarem seus processos, oferecendo suporte técnico e metodológico que favorece a maturidade organizacional.

Além das ferramentas e metodologias, o papel do gestor de projetos também merece atenção. Lengler (2022) aponta que competências como liderança, comunicação clara, gestão de riscos e domínio técnico são essenciais para conduzir equipes e garantir entregas de qualidade. Em projetos de software, onde os desafios são multifacetados, essas habilidades se tornam ainda mais relevantes.

Um exemplo que ilustra bem essa realidade é o da DevTrack, uma startup voltada para soluções logísticas. Ao adotar o Scrum como metodologia de trabalho, a empresa conseguiu reduzir em 40% o tempo de entrega de novas funcionalidades, além de melhorar a integração entre os setores de desenvolvimento e produto. Nesse caso, o gerenciamento de projetos não foi apenas uma ferramenta de controle, mas sim um elemento estratégico para o crescimento do negócio.

Diante disso, é possível afirmar que o gerenciamento de projetos de software desempenha um papel central na transformação de ideias em soluções concretas. A combinação entre metodologias adequadas, políticas de apoio e competências gerenciais contribui diretamente para o sucesso dos projetos e para a sustentabilidade das empresas que atuam no setor tecnológico.

1. **PROGRAMAÇÃO ORIENTADA A OBJETOS II**

Dentro da matéria de Programação orientada a objetos II, ela nos desafia a montar um projeto com o objetivo de desenvolver um sistema de desktop robusto, utilizando a linguagem C# e no ambiente de desenvolvimento (Visual Studio 2022), com a persistência de dados gerenciada pelo banco de dados SQL Server. Para garantir a organização e a manutenibilidade do código, a arquitetura da aplicação seguirá o padrão Model-View-Controller (MVC), separando claramente as responsabilidades entre a interface do usuário, as regras de negócio e o acesso aos dados. A comunicação com o back-end será realizada através de uma API dedicada, cuja documentação e testes serão facilitados pela ferramenta Swagger, garantindo a escalabilidade para futuras integrações. Adicionalmente, o sistema contará com a integração de um componente de Inteligência Artificial (IA), que atuará como uma ajuda aos técnicos que a ferramenta auxiliara nas resoluções dos problemas.

Aplicações desktop, também chamadas aplicações de ambiente de trabalho, são software feito para instalar e rodar no sistema operativo do computador. Diferentemente das aplicações web, que rodam nos navegadores, as aplicações desktop usam os recursos locais do hardware processador, memória, e sistema de arquivos, diretamente por exemplo Fowler, 2003.

A ligação principal entre a aplicação desktop e o sistema operativo é crucial para o seu funcionamento. Sem dúvida, o sistema operativo fornece serviços usados pelas aplicações para interagir com o hardware, escondendo a complexidade dos componentes físicos Stallings, 2017. Ele age como uma camada de abstração e gestor de recursos, garantindo que os pedidos do software sejam atendidos de forma organizada e segura.

Como Tanenbaum e Bos destacaram lá em 2016, o sistema operativo age tipo a fundação para os programas rodarem, permitindo que o software funcione bem.

No mundo da engenharia de software, criar aplicativos para computadores exige muita atenção com a qualidade, já que eles têm acesso direto ao hardware do usuário. Pressman, em 2021, diz que o software é tipo um produto que precisa ser planejado direitinho para ficar bom. A Interface Gráfica do Usuário, tem um papel muito importante nesse caso. A facilidade de uso e a clareza na interação são o que fazem um aplicativo bom, por isso, um design focado no usuário é superimportante para a eficiência e intuição, como o Norman disse em 2013.

O sucesso de plataformas como o Windows já foi estudado por quem entende de interfaces complexas, mostrando o poder de uma benfeita, como o Petzold mostrou em 1998. Regras internacionais definem as características de qualidade, tipo funcionalidade, confiança, usabilidade e desempenho, ajudando na criação de softwares complicados. Á norma ISOIEC 25010, criada pela ABNT em 2011, por exemplo.

1. **PROJETO DE SISTEMAS ORIENTADO A OBJETOS**

Na matéria de programação orientada a objetos (POO), ela é um dos modelos mais populares no desenvolvimento de software. Em vez de apenas escrever códigos, ela organiza tudo em entidades chamadas objetos. Esses objetos são como pequenas "caixas" que guardam tanto atributos quanto métodos, tornando a modelagem do software mais intuitiva e próxima da nossa maneira de entender o mundo (SOMMERVILLE, 2011).

Um dos principais recursos que usamos para visualizar e documentar sistemas orientados a objetos é o Diagrama de Classes, que faz parte da Linguagem de Modelagem Unificada (UML). Esse diagrama é uma representação gráfica que mostra as classes de um sistema, suas características, métodos e como elas se relacionam entre si (LARMAN, 2010).

Usar diagramas de classes não só ajuda desenvolvedores e interessados a entenderem melhor o sistema, mas também é um diagrama fundamental para planejar, analisar e desenvolver um software. Com isso, conseguimos estruturar e modularizar o código de forma mais eficaz, facilitando o trabalho em equipe e a evolução do projeto, tornando o código mais profissional (FOWLER, 2004).

* 1. **UML**

A UML, ou (Linguagem de Modelagem Unificada), é uma linguagem, pra desenhar, entender, criar e documentar detalhes de um sistema de software. Ela é fundamental na Análise de Sistemas Orientada a Objetos, porque ajuda a mostrar de forma bem clara e fácil de entender como o sistema é organizado e como ele funciona, considerando diferentes pontos de vista. Assim, ela fica mais fácil de todo mundo que está no projeto se comunicar.

Segundo o Fowler (2014), a UML surgiu na tentativa de juntar as várias notações diferentes de modelagem orientada a objetos que existiam até os anos 90, como as idéias do Booch, Rumbaugh e Jacobson. Essa padronização foi um passo importante para tornar o processo de modelar sistemas mais acessível e menos propenso a confusões, especialmente em times que envolvem várias áreas. A UML não é uma metodologia de desenvolvimento em si; ela é uma ferramenta que apoia diferentes processos de engenharia de software.

Quando a gente fala de Análise de Sistemas Orientada a Objetos, ela ajuda bastante a identificar e descrever os principais elementos do problema como classes, objetos, relacionamentos, comportamentos e casos de uso usando diagramas visuais que seguem um padrão.

Como o Guedes (2018) explica, "a UML traz uma variedade de diagramas que facilitam tanto a modelagem estrutural quanto a comportamental de sistemas complexos", tornando-se uma ferramenta bem versátil em várias etapas do desenvolvimento. Entre os diagramas mais usados nessa análise orientada a objetos, estão os diagramas de casos de uso, que mostram o que o sistema deve fazer do ponto de vista do usuário; os diagramas de classes, que ilustram a estrutura fixa do sistema; e os diagramas de sequência, que mostram como os objetos interagem ao longo do tempo. Esses diagramas ajudam a documentar, entender e validar o sistema enquanto ele é desenvolvido.

Larman (2007) diz que, se a UML for usada com atenção, junto com os princípios da orientação a objetos, ela ajuda bastante a criar sistemas mais coesos, mais fáceis de manter e de ampliar. Ele acredita que a UML não deve ser só uma ferramenta de documentação, mas uma forma de pensar o design e a arquitetura do sistema desde o começo do projeto. Além disso, a UML permite uma abstração que vai ficando mais detalhada aos poucos. Ou seja, o analista pode começar com uma visão geral conceitual e ir acrescentando detalhes técnicos e arquitetônicos aos poucos, o que torna o processo de passar da análise para o projeto mais tranquilo.

Essa estratégia ajuda a garantir que os requisitos fiquem bem ligados à implementação final.

1. **TÓPICOS ESPECIAIS DE PROGRAMAÇÃO ORIENTADA A OBJETOS**

A disciplina de Tópicos Especiais de Programação Orientada a Objetos (POO) desempenha papel estratégico no PIM, pois complementa os conhecimentos adquiridos em Programação Orientada a Objetos II ao introduzir recursos avançados de desenvolvimento, novas ferramentas, frameworks e técnicas modernas de implementação. No contexto do projeto, essa disciplina será aplicada principalmente na implementação do aplicativo mobile, integração entre diferentes plataformas e adoção de arquiteturas robustas, capazes de atender às demandas do sistema de suporte técnico inteligente. O PIM prevê que o sistema seja entregue em três frentes distintas: desktop, web e mobile. Assim, os Tópicos Especiais de POO fornecem base para a utilização de frameworks como .NET MAUI, Xamarin e Blazor, permitindo a construção de aplicações multiplataforma em C# e ASP.NET. Além disso, essa disciplina possibilita a integração com APIs de Inteligência Artificial, recurso essencial para a criação de FAQs dinâmicas, respostas automáticas e análise inteligente do histórico de chamados.

Ela apresenta pontos bastante positivos que enriquecem a experiência de aprendizado. Ela conta com a utilização de recursos avançados da linguagem C# e do ecossistema .NET, além da aplicação de padrões de arquitetura modernos como MVC e MVVM, que contribuem para uma melhor organização e manutenção do código. Também permite o desenvolvimento de aplicativos multiplataforma, possibilitando o registro de chamados por desktop, navegador web ou dispositivos Android. Outro destaque é a integração com APIs externas e soluções de inteligência artificial, que tornam o sistema mais inteligente e autônomo. Por utilizar frameworks amplamente adotados no mercado, ela oferece uma visão prática e alinhada com as exigências das empresas de tecnologia. Além disso, promove a segurança e a modularização do código por meio de princípios como encapsulamento e injeção de dependência, o que facilita os testes e aumenta a confiabilidade das APIs e serviços.

Apesar dos benefícios, a disciplina também apresenta alguns desafios que merecem atenção. Ela envolve uma complexidade significativa na integração entre diferentes ambientes — como Windows, Web e Android — o que pode gerar dificuldades técnicas durante o desenvolvimento. Além disso, a curva de aprendizado é elevada, especialmente para dominar frameworks mais recentes como .NET MAUI, Blazor e Xamarin, exigindo dedicação extra por parte dos alunos. Outro ponto crítico é a necessidade de aplicar corretamente os conceitos de orientação a objetos; falhas nesse aspecto podem resultar em um código pouco reutilizável e difícil de manter. Há também um risco maior de inconsistências entre as versões das aplicações para desktop, web e mobile, principalmente se o grupo não definir uma arquitetura unificada desde o início do projeto.

A disciplina garante que o sistema seja tecnologicamente moderno, escalável, multiplataforma e aderente às exigências do projeto. Além disso, possibilita que o grupo entregue uma solução inovadora, utilizando práticas que estão em evidência no mercado de TI, fortalecendo tanto a relevância acadêmica quanto a aplicabilidade profissional do trabalho.

1. **MARKETING PESSOAL**

No cenário atual, o marketing digital desempenha um papel essencial na promoção de serviços online, especialmente em nichos especializados como o suporte técnico assistido por inteligência artificial. Segundo Aguiar (2024), o marketing digital é fundamental para a divulgação de produtos e serviços online, utilizando estratégias como SEO e redes sociais para aumentar a visibilidade e atingir o público-alvo.

Para empresas de assistência técnica, a presença online sólida é crucial. Dicas como estabelecer uma presença online sólida, construir uma imagem de destaque e utilizar ferramentas de automação de marketing são recomendadas para aumentar a visibilidade e atrair mais clientes (Dextecnologia, 2021)

Além disso, o marketing digital permite uma comunicação direta e eficaz com os clientes, facilitando o atendimento e aumentando a satisfação. Ferramentas de automação e estratégias de marketing digital são essenciais para empresas de serviços, ajudando a criar uma conexão eficaz com os clientes e a aumentar a visibilidade online (Ideal Marketing, 2025).

Portanto, para o TechSupport, investir em estratégias de marketing digital é fundamental para estabelecer uma presença online forte, atrair clientes e oferecer um atendimento de qualidade, alinhado às necessidades do público-alvo.

1. **EMPREENDEDORISMO**

Empreender na área de tecnologia hoje em dia vai muito além de saber programar ou entender de sistemas. Precisa também ter visão de negócio, saber lidar com mudanças rápidas e entender como se posicionar num mercado cada vez mais competitivo. A tecnologia está em tudo, e quem decide abrir uma empresa nesse setor precisa estar preparado para inovar constantemente.

Segundo Cassol et. al. (2022), empresas incubadas que conseguem absorver conhecimento externo e transformar isso em inovação têm mais chances de se destacar. Isso mostra que não basta ter uma boa ideia — é preciso estar atento ao que acontece ao redor e saber aplicar isso no negócio. Já Stettiner(2021) analisam como políticas públicas, como o PIPE-FAPESP, ajudam startups a crescer, oferecendo apoio financeiro e técnico. Esse tipo de incentivo é essencial, principalmente para quem está começando e ainda não tem estrutura para competir com grandes empresas.

Além disso, Lengler et. al. (2024) falam sobre as competências empreendedoras que fazem diferença nesse meio. Liderança, criatividade, resiliência e capacidade de gestão são algumas das habilidades que ajudam o empreendedor a enfrentar os desafios do dia a dia. E isso vale tanto para quem está em uma startup quanto para quem já tem uma empresa consolidada.

Um exemplo prático é o TechSupport, que trabalhará com suporte técnico usando inteligência artificial. A empresa irá investir em presença digital, automação e atendimento personalizado, pois tudo isso ajuda a atrair clientes e se manter relevante. Nesse caso, o marketing digital não é só uma ferramenta de divulgação, mas parte da estratégia de crescimento.

No fim das contas, empreender em tecnologia é um processo que exige preparo, estudo e adaptação. Quem entende isso e busca se desenvolver tem mais chances de transformar uma ideia em um negócio de sucesso.

1. **GESTÃO DE QUALIDADE**

A disciplina de Gestão da Qualidade é um dos pilares fundamentais para o sucesso do Projeto Integrado Multidisciplinar (PIM), pois tem como objetivo assegurar que o sistema desenvolvido seja entregue com padrões elevados de confiabilidade, eficiência e aderência aos requisitos estabelecidos. Essa disciplina estuda conceitos, técnicas e ferramentas voltados à melhoria contínua dos processos de desenvolvimento de software, permitindo que o grupo tenha maior controle sobre as atividades e os resultados esperados. No contexto do PIM, a Gestão da Qualidade será aplicada principalmente na definição de planos de teste, critérios de homologação, roteiros de verificação e documentação de evidências de conformidade. Esses elementos são cruciais para evitar erros no sistema, reduzir retrabalho e garantir que as funcionalidades desenvolvidas estejam em total alinhamento com o escopo definido pelo grupo e com a Lei Geral de Proteção de Dados (LGPD).

Entre os pontos positivos, destaca-se a estruturação de processos formais de teste, abrangendo testes unitários, de integração, funcionais e de aceitação. Também é relevante a aplicação de normas e modelos de qualidade de software, como a ISO/IEC 25010, que estabelece atributos essenciais como usabilidade, segurança, manutenibilidade, confiabilidade e desempenho. Há um incentivo ao uso de métricas de qualidade, como a taxa de falhas, o tempo médio para correção e a cobertura de testes, o que contribui para uma avaliação mais precisa do sistema. A geração de relatórios de conformidade garante transparência e confiabilidade na entrega, enquanto a aplicação de ciclos de melhoria contínua, como o PDCA, permite revisões e otimizações constantes do projeto. Além disso, o uso de ferramentas de automação, como Selenium e Postman, ajuda a mitigar o alto custo de tempo, tornando a execução de testes repetitivos mais rápida e eficiente.

Entre os pontos negativos, observa-se que o processo de qualidade demanda um alto custo de tempo, especialmente porque os testes precisam ser realizados em diversas plataformas, como desktop, web e mobile. Além disso, pode haver resistência por parte dos membros da equipe em aplicar todos os processos de qualidade, seja pela pressão dos prazos ou pelo desconhecimento técnico. Outro desafio é a necessidade de utilizar ferramentas específicas, como Selenium, Postman, NUnit, JUnit e TestRail, que exigem aprendizado adicional e disponibilidade de recursos tecnológicos. Há ainda o risco de que a equipe se concentre apenas no funcionamento aparente do sistema, negligenciando métricas mais profundas e relevantes de qualidade.

A disciplina de Gestão da Qualidade é essencial no desenvolvimento de sistemas, pois promove práticas que asseguram a confiabilidade, segurança e conformidade dos projetos. Ela garante que o sistema seja seguro, funcional, validado e confiável, fortalecendo a credibilidade do grupo no momento da apresentação e reduzindo a probabilidade de erros críticos. Além disso, assegura que os relatórios, manuais e evidências de teste estejam bem documentados, aumentando a robustez acadêmica e técnica do trabalho. A Gestão da Qualidade é fundamental para assegurar a conformidade legal, especialmente no que se refere à Lei Geral de Proteção de Dados (LGPD). Por meio de testes de segurança e auditorias, a equipe pode verificar se o sistema lida com dados pessoais de forma segura, garantindo a confidencialidade e integridade das informações. Isso inclui a verificação de mecanismos de criptografia, a gestão de acessos e a documentação dos processos de tratamento de dados, o que é um requisito central da LGPD. Essa abordagem não só protege os usuários, mas também minimiza os riscos legais para o projeto.

1. **RELAÇÕES ÉTNICO-RACIAIS E AFRODESCENDÊNCIA**

Debater as intricadas relações étnico-raciais, em especial aquelas relativas à população negra, é fundamental para entender o Brasil de hoje. Essa conversa não é só sobre o passado, pelo ao contrário, revela como as estruturas de desigualdade e poder funcionam no presente. É crucial entender que o racismo vai além de preconceito pessoal, formando um sistema que organiza e hierarquiza a nossa sociedade. Logo, uma análise crítica dessas dinâmicas é urgentemente precisa para construir uma democracia de verdade. Conforme defende o filósofo (Silvio Almeida em 2019), atos individuais ou institucionais em sociedades racistas frequentemente replicam a discriminação racial, às vezes sem intenção.

A importância desse assunto fica ainda mais clara ao desconstruir a ideia de democracia racial, que por muito tempo escondeu as desigualdades gritantes, comprovadas por dados sociais e econômicos. Nessa situação, a luta por reconhecimento se manifesta através do direito de cada um contar sua própria história e enfrentar as estruturas de opressão. Como diz ( Djamila Ribeiro em 2017).  
 No contexto apresentado, a ideia de "lugar de fala" ganha destaque, incitando a reflexão sobre quais vozes sempre foram ouvidas, e quais foram silenciadas, desafiando as relações de poder na fala. Estudar a fundo este tema ajuda a entender e valorizar os conhecimentos e vivências que foram, por muito tempo, ignoradas.  
 Portanto, a matéria de Relações Étnico-Raciais e Afrodescendência ultrapassa o campo acadêmico, é um conhecimento essencial, fundamental para a formação profissional. É crucial examinar a importância das mulheres negras na área da computação sob a visão da desigualdade, conceito desenvolvido (Kimberlé Crenshaw em 1989.)

Segundo ela, racismo e sexismo se combinam, gerando um obstáculo duplo para essas mulheres. Um exemplo histórico é o livro "Estrelas Além do Tempo", de Margot Lee hetterly, lançado em 2016. As contribuições de matemáticas como Katherine Johnson e de programadoras pioneiras como Dorothy Vaughan foram cruciais para o sucesso da NASA. Elas superaram a segregação e se tornaram figuras importantes na corrida espacial.

Esse apagamento histórico resulta em problemas atuais na tecnologia, originando sistemas enviesados e discriminatórios. O estudo "Gender Shades", conduzido por (Joy Buolamwini) e (Timnit Gebru em 2018), revelou que algoritmos de reconhecimento facial cometem erros em quase 35% dos casos envolvendo mulheres negras, uma disparidade significativa em relação à precisão com homens brancos. Isso não é mero erro, mas o que a socióloga (Ruha Benjamin) define como "design discriminatório", em 2019, onde o preconceito social é incorporado e automatizado nos sistemas.

Portanto, a inclusão de mulheres negras na tecnologia é muito mais do que uma simples medida corretiva; é algo essencial para o futuro da inovação. Na área de Análise e Desenvolvimento de Sistemas, a diversidade de equipes é fundamental para criar soluções mais equitativas, seguras e eficazes para toda a sociedade, evitando que a desigualdade se perpetue com a automação.

Fica nítido que a área de Análise e Desenvolvimento de Sistemas transcende a mera tecnicidade, configurando-se como uma atividade social com fortes implicações éticas. Ao incorporarmos o conhecimento sobre Relações Étnico-Raciais, constatamos que noções como racismo estrutural e interseccionalidade ganham forma concreta em códigos, algoritmos e interfaces, e não são apenas ideias abstratas. A trajetória de mulheres negras na informática, a exemplo das precursoras da NASA, nos transmite um ensinamento essencial: a omissão de certos grupos não só prejudica a inovação, mas também origina tecnologias problemáticas e injustas. Sendo assim, o objetivo do profissional de tecnologia do século XXI é ultrapassar a simples execução do código e encarar a missão de criar soluções que incentivem a igualdade, assegurando que a tecnologia seja um instrumento de inclusão, e não um reflexo das disparidades da nossa época.

1. **CONSIDERAÇÕES FINAIS**

Cada disciplina foi fundamental na definição dos princípios e diretrizes que

nortearam o desenvolvimento do programa, visando promover eficiência

operacional, inovação tecnológica e qualidade no atendimento aos usuários de

serviços de TI. A colaboração entre diferentes áreas do conhecimento enriqueceu o

processo de planejamento, evidenciando a relevância da troca de experiências

desde a fase inicial do projeto. Durante o desenvolvimento, diversas lições foram

aprendidas, fornecendo uma base sólida para a implementação prática da solução e

reforçando a importância do trabalho multidisciplinar na gestão de chamados

técnicos com suporte de inteligência artificial.

**REFERÊNCIAS**

AGUIAR, Camilly Gabriely de. **O papel do marketing digital na divulgação de produtos e serviços online.** Revista do Centro Paula Souza, v. 6, n. 1, p. 75, 2024. Disponível em: https://ric.cps.sp.gov.br/bitstream/123456789/28161/1/75%20O%20PAPEL%20DO%20MARKETING%20DIGITAL%20NA%20NA%20DIVULGA%C3%87%C3%83O%2DE%20PRODUTOS%20E%20SERVI%C3%87OS%20ONLINE.pdf. Acesso em: 7 de outubro de 2025.

ALBAHARI, Joseph; ALBAHARI, Ben. **C# 10 in a Nutshell: The Definitive Reference.** Sebastopol: O’Reilly Media, 2022.

ALMEIDA, Silvio Luiz de. **Racismo estrutural.** São Paulo: Sueli Carneiro; Pólen, 2019.

ANICHE, Mauricio. **Test-Driven Development: Teste e Design no Mundo Real com .NET.** São Paulo: Casa do Código, 2014.

ARIAS, Ángel. **Aprende a Programar ASP.NET y C#: 2ª Edición.** Madrid: IT Campus Academy, 2015.

BENEDETTO, Hellen; SOUZA, Rafaela K.; LENGLER, Fernando Ramos; INACIO, Eliane Cristina; AZEVEDO, Ana Carolina de. **Competências empreendedoras e inovação no ambiente acadêmico: a integração transformadora do programa Impulsiona SENAI nas unidades curriculares de projeto integrador.** Revista Contemporânea, v. 4, n. 7, p. e5028, 2024.

BENJAMIN, Ruha. **Race After Technology: Abolitionist Tools for the New Jim Code.** Medford: Polity Press, 2019.

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.

BUOLAMWINI, Joy; GEBRU, Timnit. **Gender shades: Intersectional accuracy disparities in commercial gender classification.** In: Conference on Fairness, Accountability and Transparency. PMLR, 2018. p. 77–91.

CASSOL, Alessandra; GONÇALO, Carlos Roberto; RUAS, Roberto Lima. **Redefining the relationship between intellectual capital and innovation: the mediating role of absorptive capacity.** BAR - Brazilian Administration Review, v. 13, n. 1, p. 1–25, 2016.

CASSOL, Alessandra; SILVA, André Luiz da; FERREIRA, João Carlos; SANTOS, Daniel de Oliveira. **Determinants of entrepreneurial intentions and the moderation of entrepreneurial education: A study of the Brazilian context.** The International Journal of Management Education, v. 20, n. 3, p. 100716, 2022. Disponível em: https://www.researchgate.net/publication/364151927. Acesso em: 6 de outubro de 2025.

DE DADOS, **Componentes do Modelo. Projeto de Banco de Dados.** [S.l.]: 2000.

DEITEL, Harvey; DEITEL, Paul; DEITEL, Abbey. **Android: como programar.** Porto Alegre: Bookman Editora, 2015.

DEXTECNOLOGIA. **8 dicas de marketing para empresas de assistência técnica.** 2021. Disponível em: <https://dextecnologia.com/8-dicas-de-marketing-para-empresas-de-assistencia-tecnica/>. Acesso em: 7 de outubro de 2025.

DIMES, Troy. **Programação em C# para iniciantes.** [S.l.]: Babelcube Inc., 2016.

EVANS, Eric. **Domain-driven design: atacando as complexidades no coração do software.** Rio de Janeiro: Alta Books, 2009.

FACELI, Katti et al. **Inteligência artificial: uma abordagem de aprendizado de máquina.** Rio de Janeiro: LTC, 2021

FOWLER, Martin. **Padrões de arquitetura de aplicações corporativas.** Porto Alegre: Bookman, 2009.

FOWLER, Susan J. **Micros serviços prontos para a produção: Construindo sistemas padronizados em uma organização de engenharia de software.** São Paulo: Novatec Editora, 2019.

FREEMAN, Adam; SANDERSON, Andrew. **Pro ASP.NET Core MVC 2.** Nova York: Apress, 2017.

FREITAS, Thayanne Tavares. **O que é lugar de fala?** Horizontes Antropológicos, n. 54, p. 361–366, 2019.

GAMMA, Erich et al. **Design patterns: Abstraction and reuse of object-oriented design. In: European Conference on Object-Oriented Programming.** Berlin, Heidelberg: Springer Berlin Heidelberg, 1993. p. 406–431.

GAMMA, Erich. **Padrões de projetos: soluções reutilizáveis.** Porto Alegre: Bookman Editora, 2009.

GERON, Aurelien. **Mãos à obra: aprendizado de máquina com Scikit-Learn, Keras & Tensorflow.** Rio de Janeiro: Alta Books, 2021.

GOODFELLOW, Ian et al. **Deep learning.** Cambridge: MIT Press, 2016.

HEJLSBERG, Anders; WILTAMUTH, Scott; GOLDE, Peter. **C# language specification.** Boston: Addison-Wesley Longman Publishing Co., Inc., 2003.

IDEAL MARKETING. **Estratégias de marketing digital para o setor de serviços.** 2025. Disponível em: <https://www.idealmarketing.com.br/blog/estrategias-de-marketing-digital-para-o-setor-de-servicos/>. Acesso em: 7 de outubro de 2025.

ISO/IEC 25010:2011. **Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE).** Disponível em: https://iso25000.com/index.php/en/iso-25000-standards/iso-25010. Acesso em: 6 out. 2025.

JUNIOR, Edemilton Alcides Galindo; ROCHA, Romeu Dias; DE SOUZA MACIEL, Ronierison. **Desenvolvimento de API REST com Spring Boot.** Paulo Afonso: UniRios, 2021

JURAN, J. M. **Juran’s Quality Handbook: The Complete Guide to Performance Excellence. 7. ed.** Nova York: McGraw-Hill Education, 2016.

LENGLER, Leandro dos Santos Ferreira. **Competências gerenciais no setor de tecnologia da informação: um estudo com gestores de TI.** Dissertação (Mestrado em Administração) – Centro Universitário Unihorizontes, Belo Horizonte, 2022.

LONGWORTH, Jackson. Resenha de: BENJAMIN, Ruha. **Race After Technology: Abolitionist Tools for the New Jim Code. Science & Technology Studies,** v. 34, n. 2, p. 92–94, 2021.

MARTIN, Robert C. **Arquitetura Limpa: o guia do artesão para estrutura e design de software.** Rio de Janeiro: Alta Books Editora, 2019.

MARTIN, Robert C. **Clean Code – Refactoring, Patterns, Testen und Techniken für sauberen Code: Deutsche Ausgabe.** Frechen: MITP-Verlags GmbH & Co. KG, 2013.

MICHAELIS, Mark**. Essential C# 12.0.** Boston: Addison-Wesley Professional, 2023.

NASCIMENTO, Abdias. **O quilombismo.** Petrópolis: Vozes, 1980.

NEWMAN, Sam. **Criando Microsserviços – 2ª Edição: Projetando sistemas com componentes menores e mais especializados.** São Paulo: Novatec Editora, 2022.

NOBLE, Safiya Umoja. **Algorithms of oppression: How search engines reinforce racism.** Nova York: New York University Press, 2018.

PRESSMAN, Roger S. **Engenharia de Software: uma abordagem profissional. 8. ed.** Porto Alegre: AMGH, 2016.

RIBEIRO, Djamila. **O que é lugar de fala?** Belo Horizonte: Letramento, 2017. (Feminismos Plurais).

RICHARDSON, Leonard; AMUNDSON, Mike. **RESTful Web APIs.** Sebastopol: O'Reilly Media, 2013.

RUSSELL, Stuart; NORVIG, Peter. **Inteligência Artificial: uma abordagem moderna.** Rio de Janeiro: Campus, 2004.

SILVA, Tarcízio; BIRHANE, Abeba (Ed.). **Comunidades, algoritmos e ativismos digitais: Olhares afrodiaspóricos.** São Paulo: LiteraRUA, 2020.

SOMMERVILLE, Ian. **Engenharia de Software.** 10. ed. São Paulo: Pearson, 2019.

STETTINER, Caio Flavio; LIMA, Rafael de Oliveira; MORAES, Juliana Gomes de. **O PIPE-FAPESP como política pública de apoio ao empreendedorismo e seu impacto em uma startup do mercado imobiliário.** Revista Humanidades & Inovação, v. 8, n. 1, 2021. Disponível em: https://revista.unitins.br/index.php/humanidadeseinovacao/article/download/974/762. Acesso em: 6 de outubro de 2025.

STETTINER, Raoni Guerra; RAJÃO, Lucas. **Tradição e modernidade: des(entendimentos) no desenvolvimento de um software durante o processo de implantação da MPS.BR.** Dissertação (Mestrado em Administração Pública) – Universidade Federal de Minas Gerais, Belo Horizonte, 2021.

TAULLI, Tom. **Introdução à inteligência artificial: uma abordagem não técnica.** São Paulo: Novatec Editora, 2020.

TROELSEN, Andrew; JAPIKSE, Philip. **Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming.** Nova York: Apress, 2022.