

**UNIVERSIDADE PAULISTA**

**ICET - INSTITUTO DE CIÊNCIAS EXATAS E TECNOLOGIA**

**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**PROJETO INTEGRADO MULTIDISCIPLINAR PIM II**

# Desenvolvimento de um Sistema Acadêmico Colaborativo com Apoio de IA

## Nome R.A

Gabriel Vinicius Rosa Pereira R3733D2

Isabella Santos Leal F3666E6

**SÃO JOSÉ DOS CAMPOS – SP**

## NOVEMBRO / 2025

|  |  |
| --- | --- |
| Gabriel Vinicius Rosa Pereira | R3733D2 |
| Isabella Santos Leal | F3666E6 |
| Aluno 3 |  |
| Aluno 4 |  |
| Aluno 5 |  |

**Desenvolvimento de um Sistema Acadêmico Colaborativo com Apoio de IA**

Projeto Integrado Multidisciplinar (PIM) desenvolvido como exigência parcial dos requisitosobrigatórios à aprovação semestral no Curso Superior de Tecnologia em Análise eDesenvolvimento de Sistemas da UNIP(Universidade Paulista), orientado pelo corpo docente do curso.

**São José dos Campos – SP**

**Novembro / 2025**

## RESUMO

O presente Projeto teve como objetivo desenvolver um Sistema Acadêmico, voltado ao gerenciamento de turmas, alunos, aulas e atividades em ambiente de rede local. O trabalho foi fundamentado nas disciplinas de C, Python e Engenharia de Software Ágil. A metodologia envolveu pesquisa aplicada e desenvolvimento experimental, utilizando práticas ágeis e integração entre linguagens para criar uma solução eficiente e sustentável. O programa contou com interface gráfica em Python (Tkinter) e módulos críticos desenvolvidos em C, tendo como objetivo alto desempenho e integração. A aplicação futura da Inteligência Artificial possibilitou o tratamento automático de dados e a detecção de inconsistências, otimizando o processo de avaliação acadêmica. Além disso, o projeto contribuiu para a redução do uso de papel, apoiando ações de educação ambiental. Como resultado, obteve-se um sistema funcional, colaborativo e de fácil utilização, capaz de aprimorar a comunicação entre professores e alunos, automatizar tarefas administrativas e promover a sustentabilidade institucional. Concluiu-se que o projeto atingiu seus objetivos, demonstrando a viabilidade técnica e pedagógica.

Palavras-chave: Sistema acadêmico; Inteligência Artificial; Sustentabilidade; Inovação tecnológica.

## SUMÁRIO XXXXXXXX

# INTRODUÇÃO

O avanço da tecnologia da informação tem transformado radicalmente os processos educacionais, promovendo maior integração e eficiência no gerenciamento acadêmico. Nesse contexto, este projeto tem como objetivo geral projetar e implementar um sistema acadêmico colaborativo integrado, que permita o gerenciamento de turmas, alunos e suas notas, com funcionalidades que promovam a colaboração entre os usuários e a aplicação de recursos de inteligência artificial para otimizar processos.

Além do objetivo geral, destacam-se objetivos específicos como a aplicação de práticas ágeis de engenharia de software para o desenvolvimento organizado do sistema, a implementação de algoritmos eficientes para busca e ordenação de dados, o uso da linguagem C para desenvolvimento de módulos críticos perto do hardware, e a utilização de diagramas UML para melhor análise e projeto do sistema. Também são considerados aspectos de redes de computadores para garantir o funcionamento em ambientes distribuídos e a inclusão de práticas sustentáveis, como a redução do uso de papel por meio do sistema digital.

A motivação deste projeto surge da necessidade de uma instituição de ensino que atualmente utiliza métodos descentralizados e manuais para o controle acadêmico, como planilhas e troca de e-mails, dificultando a colaboração e o acompanhamento eficiente. O sistema proposto visa suprir essas necessidades, proporcionando uma solução tecnológica inovadora que facilite o gerenciamento das atividades acadêmicas, ao mesmo tempo em que incentiva práticas sustentáveis e promove o uso de inteligência artificial para melhoria do desempenho.

Portanto, este projeto busca, por meio da integração de tecnologias emergentes e metodologias de desenvolvimento modernas, contribuir para a transformação digital no ambiente acadêmico, fortalecendo a colaboração, a eficiência e a inovação.

# 

## OBJETIVO GERAL

Desenvolver um sistema acadêmico que integre o gerenciamento de turmas, alunos, aulas e atividades em um ambiente digital, aplicando metodologias ágeis e tecnologias sustentáveis para otimizar processos educacionais e administrativos.

## OBJETIVOS ESPECÍFICOS

* Aplicar conceitos de Engenharia de SoftwareÁgil para planejar, organizar e acompanhar as etapas do desenvolvimento.
* Implementar estruturas de dados e algoritmos em Python para funcionalidades de busca, ordenação e geração de relatórios.
* Desenvolver módulos críticos em C estruturado, garantindo eficiência e melhor desempenho do sistema.
* Configurar o sistema em ambiente de rede local (LAN), aplicando o conceito de cliente-servidor e acessos simultâneos.
* Integrar recursos de Inteligência Artificial para análise automática de dados e identificação de inconsistências.
* Promover sustentabilidade e educação ambiental através da digitalização de processos e redução do uso de papel.
* Estimular a pesquisa e inovação tecnológica, aplicando novas ferramentas e técnicas para aprimorar o sistema.

## CONTEXTUALIZAÇÃO DO CASO

Atualmente, a instituição de ensino possui controles acadêmicos descentralizados, utilizando planilhas, e-mails e mensagens em aplicativos para gerenciamento de turmas, aulas e atividades. Essa situação gera dificuldades para colaboração eficiente entre professores e alunos, além de perdas de informações e uso excessivo de papel.

O sistema proposto tem como objetivo substituir esses controles manuais por uma plataforma colaborativa integrada, que permita o cadastro de turmas e alunos, registro de aulas e diário eletrônico, upload e consulta de atividades, e suporte a múltiplos usuários em rede local. Um dos objetivos é eliminar o uso de papel pelos professores, adotando práticas sustentáveis por meio da digitalização dos processos acadêmicos.

### 2. PROGRAMAÇÃO ESTRUTURADA EM C

A linguagem C é uma das linguagens de programação mais influentes e mais usadas desde que foi criada na década de 1970 por Dennis Ritchie nos Bell Labs. Desenvolvida originalmente para programação do sistema operacional UNIX, C tornou-se rapidamente uma linguagem padrão para o ensino da programação e para desenvolvimento de sistemas, devido à sua eficiência, portabilidade e capacidade de abstração (KERNIGHAN; RITCHIE, 1988).

**2.1 O Básico dos Fundamentos da Linguagem C**

A linguagem C é uma linguagem que possui múltiplos paradigmas; ela é imperativa, estruturada, orientada a objetos (via extensão, permite a abstração de dados de forma orientada a objetos) e funcional (permite a programação funcional), combinando esses paradigmas em diferentes formas para criar soluções eficientes. É também uma linguagem de baixo nível que permite acesso direto ao hardware e à memória, porém mantendo abstração suficiente que permite ser considerada uma linguagem de nível intermediário.

Em outras palavras, ela permite que programas sejam escritos em C sem sacrificar a portabilidade. Possui forte tipagem estática e dinâmica (determina o tipo dos dados sem explicitar que dados o código irá trabalhar) e forte tipagem (não converte dados de tipos diferentes entre eles automaticamente). Possui mecanismos para programação genérica e programação orientada a objetos (POO).

Sua filosofia consiste em oferecer um conjunto pequeno, porém poderoso, de construções que possam ser combinadas de uma forma muito flexível (DEITEL, 2016).

**2.2 Interfaces e Integrações**

A linguagem C tem a grande vantagem de ser integrável a outros ambientes e linguagens. Seu uso também é comum na programação de bibliotecas dinâmicas (DLLs), interfaces para bancos de dados, sistemas distribuídos e módulos de baixo nível em linguagens como Python, Java e Lua. Essa flexibilidade significa que C pode ser usada tanto em aplicações de sistema quanto no backend de alto desempenho em arquiteturas modernas.

**2.3 Virtudes de C**

* Eficiência e alto desempenho, por gerar código binário otimizado;
* Portabilidade, devido à sua padronização e compatibilidade entre compiladores (KERNIGHAN; RITCHIE, 1988).
* Controle de baixo nível, permitindo que desenvolvedores otimizem o uso de CPU e memória (SOUZA; LIMA, 2021).
* Base para outras linguagens, o que se torna um ponto ótimo para quem quer aprender a programar.

**2.4 Limitações da Linguagem C**

* Falta de características modernas orientadas a objetos;
* Gestão de memória manual, que pode resultar em erros como vazamentos de memória e falhas de segmentação (DEITEL, 2016).
* Baixo nível de abstração, que pode dificultar grandes projetos (SCHILDT, 2017).
* Ausência de mecanismos automáticos para tratar exceções (SOMMERVILLE, 2019).

### 3. Engenharia de Software Ágil

**3.1 Caracterização da Engenharia de Software**

A Engenharia de Software é a área da computação que é responsável e focada em aplicar boas práticas no desenvolvimento de sistemas. Voltada para a organização das etapas, definição de processos para garantir que o sistema seja construído de forma eficaz e eficiente e organizada.

De acordo com Pressman e Maxim (2016), a Engenharia de Software procura garantir a qualidade e eficiência do projeto utilizando ferramentas de modelagem e processos devidamente estruturados, facilitando o processo de documentação, pois fornece os meios necessários para representar graficamente a estrutura do projeto.

No projeto realizado, a Engenharia de Software foi utilizada para planejar e documentar o funcionamento do Sistema de Notas Acadêmico, assegurando clareza na comunicação com os membros da equipe e na organização do fluxo de informações.

**3.2 Metodologia Ágil - Scrum**

Para realizar o planejamento durante o desenvolvimento do Sistema Acadêmico, foi utilizada a Metodologia Ágil Scrum, que permite o controle iterativo e incremental das atividades. O projeto foi dividido em sprints quinzenais, e esse processo facilitou a entrega gradual das funcionalidades, além de permitir que o grupo revisasse e executasse as tarefas de forma organizada.

Para uma melhor estruturação do projeto, foi utilizada a ferramenta \_Trello,\_ que possibilitou a organização das sprints por meio de colunas, facilitando a visualização do andamento do projeto.

Os papéis foram definidos conforme a estrutura do Scrum:

Product Owner (PO) – Gabriel: responsável por definir e priorizar as funcionalidades do sistema, além de coordenar o levantamento de requisitos e organizar a distribuição das tarefas entre os membros da equipe.

Scrum Master (SM) – Isabella: responsável por acompanhar as etapas de desenvolvimento, garantir o cumprimento dos princípios ágeis e organizar o planejamento das sprints.

Equipe de Desenvolvimento (Dev Team) – Letícia, Thiago e Flávio: responsáveis por desenvolver as funcionalidades em C (backend) e Python (frontend), realizando a integração e os testes.

Essa metodologia proporcionou maior produtividade e transparência no processo, evitando entregas fora do prazo, além de favorecer o trabalho colaborativo e o foco em resultados de forma contínua.

**3.3 Processo de desenvolvimento dos artefatos de Engenharia de Software**

Os artefatos de Engenharia de Software foram criados com o propósito de representar, de forma estruturada e organizada, o funcionamento do sistema e suas interações com o usuário, sendo incrementados a cada sprint para facilitar a visualização do projeto de maneira gráfica.

Para compreender todas as informações que envolvem o sistema, como o sistema processa esses dados e como eles se conectam, foram elaborados o Diagrama de Fluxo de Dados (DFD) e o Diagrama de Contexto.

O diagrama de fluxo de dados (DFD) foi utilizado para detalhar o fluxo de dados entre as entidades externas, mostrando como as informações entram, são processadas e saem do sistema.

O diagrama de contexto foi criado inicialmente para delimitar o escopo do sistema e as interações com os atores externos.

Esses diagramas foram elaborados com base no levantamento de requisitos funcionais e não funcionais, contribuindo para uma visão clara e objetiva do funcionamento do Sistema Acadêmico.

Para a construção visual, foi utilizada a ferramenta digital Draw.io, que permitiu criar representações claras, padronizadas e de fácil compreensão pelos integrantes do grupo.

**3.4 Levantamento de Requisitos**

A especificação de requisitos é uma das etapas mais importantes no desenvolvimento de software, pois descreve as funcionalidades e restrições do sistema. Nesta fase, são definidos os requisitos funcionais, que descrevem o comportamento esperado, e os requisitos não funcionais, que tratam de aspectos como segurança, desempenho e confiabilidade.

**Requisitos Funcionais (RF)**

RF001 – Gerenciar Usuários e Turmas O sistema deve permitir cadastrar, editar e excluir administradores, professores, alunos e turmas, relacionando-os conforme suas funções.

RF002 – Autenticar Usuários O sistema deve permitir o login de administradores, professores e alunos, validando as credenciais de acesso.

RF003 – Registrar Notas dos Alunos O sistema deve permitir registrar, atualizar e armazenar as notas dos alunos em suas respectivas disciplinas.

RF004 – Consultar Boletim e Desempenho O sistema deve permitir consultar as notas, médias e desempenho geral de cada aluno, apresentando essas informações de forma clara.

RF005 – Gerar Relatórios e Gráficos O sistema deve gerar gráficos e relatórios de desempenho das turmas, permitindo a visualização das médias de cada aluno.

**Requisitos Não Funcionais (RNF)**

RNF001 – Garantir Segurança dos Dados O sistema deve garantir a segurança e confidencialidade dos dados armazenados, protegendo logins, senhas e notas.

RNF002 – Oferecer Interface Intuitiva O sistema deve fornecer uma interface gráfica simples e intuitiva, desenvolvida em Tkinter, permitindo fácil navegação.

RNF003 – Assegurar Disponibilidade Local O sistema deve operar localmente em ambiente Windows, sem necessidade de conexão à internet.

RNF004 – Otimizar Desempenho de Execução O sistema deve responder rapidamente às ações do usuário, com tempo de resposta inferior a 5 segundos.

RNF005 – Operar em Configuração de Hardware Mínima O sistema deve funcionar corretamente em computadores com processador dual-core 2.0 GHz, 4 GB de RAM e sistema operacional Windows.

**3.5 Elementos fundamentais da UML adotados**

A Linguagem de Modelagem Unificada (UML) é um padrão internacional utilizado para representar graficamente o funcionamento de um sistema, ajudando a compreender o seu comportamento.

De acordo com Sommerville (2019), a UML fornece um conjunto de diagramas que auxiliam na visualização e o entendimento das diferentes partes de um sistema, permitindo que os desenvolvedores compreendam de maneira clara como os componentes interagem entre si e com os usuários.

No Sistema Acadêmico, a UML foi usada como ferramenta principal para o planejamento e a documentação, garantindo que cada etapa do desenvolvimento fosse conduzida de forma padronizada e organizada. Essa abordagem acabou permitindo uma comunicação eficiente entre os integrantes do grupo e maior facilidade no acompanhamento do projeto proposto.

**3.6 Descrição dos Processos, Entidades Externas, Bancos e Fluxos**

O Sistema Acadêmico foi planejado com três entidades externas principais: o Administrador, o Professor e o Aluno.

Cada um possui funções específicas dentro do fluxo de dados:

O Administrador é responsável pelo cadastro de professores, alunos e turmas; o Professor realiza o registro de notas e a geração de gráficos de desempenho; e o Aluno consulta suas notas e médias no sistema.

O fluxo de informações ocorre de forma estruturada: os dados são inseridos por meio da interface em Python/Tkinter, processados pelo backend em C e armazenados temporariamente em estruturas de memória (arrays).

Essas estruturas garantem o controle dos registros e o retorno correto das informações.

Os fluxos de dados representam as ações como: login, cadastro, registro e consulta de notas. Permitindo compreender como cada ator interage com o sistema e quais informações são passadas em cada processo.

### 4. ESTRUTURA DE DADOS EM PYTHON

**4.1 PYTHON DATA STRUCTURES**

A linguagem Python é bastante utilizada no desenvolvimento de sistemas devido à sua simples sintaxe, alta legibilidade e grande compatibilidade com outras linguagens. De acordo com Lutz (2013), Python é uma linguagem multiparadigma interpretada, que suporta programação de alto nível, procedural, orientada a objetos e funcional, tornando-se ideal para o desenvolvimento de sistemas complexos.

Neste trabalho, Python foi a linguagem principal utilizada para integrar a interface GUI ao núcleo de processamento em C, permitindo uma comunicação eficaz do usuário com a lógica de negócios do software. Conforme descrevem Van Rossum e Drake (2022), Python possui funcionalidades nativas para integração com linguagens de mais baixo nível, o que a torna ideal para o desenvolvimento de sistemas híbridos eficientes.

**4.2 Estrutura Condicional e de Repetição**

As estruturas de decisão e repetição são essenciais em qualquer sistema para o controle lógico do mesmo. No código desenvolvido, foram utilizadas as expressões if, elif e else, que são estruturas de decisão e serviram para definir como o programa deve se comportar de acordo com as ações do usuário.

Um exemplo de utilização dessas estruturas é o login, pois, após o acesso do usuário, o sistema verifica se o perfil é de administrador, professor ou aluno. Caso seja um dos três, o sistema executa a função atribuída, redirecionando-o para a respectiva aba.

Por outro lado, as estruturas de repetição percorrem a lista de artigos ou dados retornados da biblioteca em C, principalmente para o cálculo de médias e notas. Em outras palavras, essas estruturas permitem que as operações sejam executadas de forma organizada, única e rápida.

**4.3 Interação entre Python e C**

No projeto, isso possibilita que funções de alto nível, como add\_student, register\_grade ou get\_class\_grades, executem quando necessário sua lógica implementada em C diretamente a partir do Python. Assim, a lógica de negócios permanece em C (tornando o mesmo mais eficiente e rápido), Python atua como uma camada intermediária, lidando com a interface e o controle da aplicação.

De acordo com Van Rossum e Drake (2022), a capacidade de interoperar com outras linguagens é um dos principais motivos pelos quais Python é considerada uma ferramenta poderosa para o desenvolvimento de sistemas embarcados, pois combina a velocidade de C com a notação simples de Python.

**4.4 A Interface e a Visualização de Recursos**

A biblioteca Tkinter, própria do Python, foi utilizada para programar a interface do usuário, e a maior parte do programa é escrita em Python, pois o Tkinter é rápido e fácil de usar.

**4.5 Principais Componentes do Sistema**

O sistema foi construído em torno de quatro elementos principais:

* Interface Gráfica do Usuário (Tkinter): pacote GUI utilizado para a interação com o usuário por meio de janelas, formulários e abas funcionais.
* Fluxo de Controle e Estruturas de Dados Integradas: utilização da lógica condicional e da repetição para aprimorar o controle do sistema e o processamento de dados.

**4.6 Importância no Desenvolvimento de Software**

O uso das estruturas de dados e dos recursos da linguagem Python foi fundamental para a conceituação e implementação do sistema desenvolvido. A combinação de Python e C proporcionou modularidade, eficiência e facilidade de manutenção, características essenciais para o desenvolvimento de softwares acadêmicos e corporativos.

### 5. ANÁLISE E PROJETO DE SISTEMAS

**5.1 Caracterização da Análise e do Projeto de Sistemas**

O processo de análise e projeto de sistemas desempenha um papel estratégico no desenvolvimento de softwares, uma vez que fornece mecanismos para compreender, organizar e planejar as funcionalidades antes da implementação. Segundo Booch, Rumbaugh e Jacobson (2005), a análise de sistemas tem como objetivo identificar e documentar os requisitos, os fluxos de informação e as interações entre os usuários e o sistema. Já o projeto estabelece a estrutura técnica necessária para implementar essas funcionalidades. Além disso, busca compreender detalhadamente os requisitos funcionais e não funcionais.

No Sistema Acadêmico, essa fase foi determinante, pois possibilitou mapear todas as funcionalidades essenciais do sistema, como o gerenciamento de usuários, o registro de notas e a geração de relatórios, além de criar representações gráficas por meio de diagramas UML, contemplando casos de uso, diagrama de classes e diagramas de sequência. Esses recursos auxiliam na visualização das interações entre os usuários e os componentes do sistema, garantindo que as decisões tomadas durante o projeto refletissem corretamente os requisitos levantados.

**5.2 Lógica Adotada para o Sistema**

A lógica utilizada no Sistema Acadêmico foi desenvolvida com base em uma arquitetura modular, estruturada em três camadas principais, sendo elas: apresentação, processamento e dados. A camada de apresentação implementada em Python, utilizando a biblioteca Tkinter é responsável por fornecer uma interface gráfica intuitiva, possibilitando a interação do usuário com o sistema, permitindo que administradores, professores e alunos realizem suas operações de forma eficiente. A camada de processamento, desenvolvida em linguagem C, centraliza toda a lógica de negócio, incluindo controle de usuários, turmas e registro de notas, garantindo desempenho elevado e confiabilidade. Já a camada de dados utiliza arquivos no formato JSON para armazenar e recuperar informações de forma estruturada, simulando o funcionamento de um banco de dados local.

Essa abordagem modular facilita o entendimento do código, simplifica manutenções futuras e garante que alterações em uma camada não afetem diretamente as demais, seguindo boas práticas de engenharia de software e promovendo a escalabilidade do sistema.

**5.3 Desenvolvimento dos Diagramas UML**

A modelagem UML do Sistema Acadêmico teve como objetivo fornecer uma representação estruturada e compreensível das funcionalidades e do comportamento do sistema. Durante a fase de análise e projeto, foram desenvolvidos três tipos principais de diagramas UML:

* O diagrama de casos de uso, que identificou os atores Administrador, Professor e Aluno e suas interações com o sistema;
* O diagrama de classes apresentou as entidades do sistema e os relacionamentos entre elas, servindo como base para a estruturação em C;
* O diagrama de sequência descreveu o fluxo de mensagens entre os objetos, demonstrando a ordem das operações durante os processos essenciais, como o registro de notas, autenticação de usuários e a geração de boletins.

A modelagem UML foi realizada com a ferramenta Astah Community que garantiu padronização e clareza na documentação, permitindo que a equipe de desenvolvimento visualizasse de forma completa a lógica do sistema e implementasse o software de maneira consistente com os requisitos (Sommerville, 2019).

**5.4 Proposta de Arquitetura e Padrões Utilizados**

O Sistema Acadêmico foi estruturado utilizando arquitetura em camadas, que separa de forma clara as responsabilidades de cada componente, permitindo modularidade e uma manutenção simplificada. A camada de apresentação, implementada em Python/Tkinter, gerencia a interface com o usuário e captura suas interações. A camada de lógica de negócio, desenvolvida em C, centraliza o processamento de dados, incluindo o cálculo de médias, controle de cadastros e validação de informações. A camada de dados passou a utilizar arquivos JSON para armazenar e recuperar registros de usuários, turmas e notas. Essa solução garante persistência das informações, organização hierárquica dos dados e fácil integração futura com APIs ou bancos de dados reais. Essa organização promove baixo acoplamento entre as camadas e alta coesão, que facilitam futuras expansões, integração com banco de dados e manutenção da solução, garantindo eficiência, confiabilidade e escalabilidade. (Sommerville, 2019).

**5.5 Justificativa das Escolhas Técnicas**

As escolhas técnicas adotadas no desenvolvimento do Sistema Acadêmico foram fundamentadas na busca por eficiência, clareza e facilidade de manutenção.

A utilização da linguagem C no backend permitiu o controle detalhado sobre o processamento de dados, garantindo desempenho elevado na manipulação de cadastros, notas e turmas. Já a escolha do Python com Tkinter para a camada de apresentação se justifica pela facilidade de criação de interfaces gráficas intuitivas, que facilitam a interação do usuário com o sistema. O uso de arquivos JSON foi adotado para garantir a persistência e a portabilidade dos dados, permitindo que o sistema mantenha informações entre execuções de forma leve e compatível com futuras integrações de Inteligência Artificial.

A UML foi escolhida como ferramenta de modelagem padrão, pois permite representar de forma padronizada a estrutura do sistema, suas interações e fluxos de dados.

Ferramentas como Astah Community e Lucidchart possibilitaram a geração de diagramas de casos de uso, classes e sequências, que serviram de referência para a implementação e manutenção do sistema, entregando eficiência técnica com organização e clareza na documentação. (Booch; Rumbaugh; Jacobson, 2005; Sommerville, 2019).

**5.6 Importância da Análise e Projeto de Sistemas no Desenvolvimento**

A análise e o projeto de sistemas representam etapas fundamentais no desenvolvimento de software, pois garantem a organização, documentação, e validação das funcionalidades antes da implementação. Essas fases possibilitam a compreensão detalhada dos requisitos do usuário, a estruturação lógica do sistema e o estabelecimento da base necessária para uma implementação eficiente.

Além disso, o uso de diagramas UML facilitou a visualização das interações entre usuários e componentes, permitindo à equipe de desenvolvimento implementar o sistema de maneira consistente e eficiente. De acordo com Pressman (2020), a análise e o projeto de sistemas aumentam a previsibilidade do desenvolvimento, reduzem falhas e promovem maior qualidade e manutenção do software.

### 6. REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS

Para o funcionamento do sistema acadêmico proposto, será utilizada uma rede local (LAN), responsável pela comunicação entre os computadores administrativos, o servidor principal e os demais dispositivos conectados.

O layout da rede prevê um servidor central instalado na sala administrativa da instituição, conectado a um switch gigabit, que distribui a conexão para os computadores dos setores de secretaria, coordenação e professores. Além disso, haverá pontos de acesso wireless (Wi-Fi) para permitir o acesso de notebooks e dispositivos móveis dos colaboradores autorizados.

Os principais componentes de rede utilizados serão:

* Servidor dedicado, responsável por hospedar o sistema acadêmico e o banco de dados.
* Switch gigabit (24 portas), para interligar os computadores da rede local.
* Roteador e access point, responsáveis pelo controle de tráfego e pelo acesso sem fio.
* Cabeamento estruturado categoria 6 (Cat6), garantindo maior velocidade e estabilidade.
* Serviços de rede como DHCP (atribuição automática de endereços IP), DNS (resolução de nomes de domínio) e firewall (proteção de rede).

A configuração das máquinas prevê que o servidor utilize um endereço IP fixo, enquanto os computadores clientes e dispositivos móveis receberão endereços dinâmicos por DHCP. Todos os equipamentos terão acesso controlado à internet, com autenticação de usuário e antivírus corporativo instalado.

Essa estrutura de rede garante segurança, desempenho e escalabilidade, permitindo o acesso simultâneo ao sistema acadêmico por vários usuários e assegurando a integridade dos dados armazenados.

### 7. EDUCAÇÃO AMBIENTAL

A Educação Ambiental tem um papel essencial na formação de cidadãos conscientes e na promoção de práticas sustentáveis dentro das instituições de ensino. Nesse sentido, os sistemas computacionais exercem grande importância, pois possibilitam a automatização de processos e a disseminação de informações de maneira sustentável, reduzindo o consumo de recursos naturais.

O sistema acadêmico proposto pelo grupo contribui diretamente para uma sociedade mais sustentável ao substituir processos tradicionais que envolvem o uso excessivo de papel por soluções digitais. Entre as ações voltadas à Educação Ambiental, destacam-se:

1. **Boletim online:** permite que acessem os boletins de notas e frequência de forma digital, eliminando a necessidade de impressão e reduzindo o consumo de papel e tinta. Essa prática estimula a conscientização sobre o uso racional dos recursos naturais.
2. **Envio eletrônico de comunicados e documentos escolares:** todas as comunicações entre a escola, alunos e responsáveis serão realizadas por meio do sistema acadêmico, evitando o uso de impressos e contribuindo para a preservação do meio ambiente.

Essas ações reforçam o compromisso da instituição com a sustentabilidade e demonstram como a tecnologia pode ser uma aliada da Educação Ambiental, promovendo a conscientização e o desenvolvimento de práticas mais responsáveis dentro do ambiente escolar.

### 8. INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial é essencial para o desenvolvimento de sistemas modernos, pois permite maior precisão e eficiência nas operações. No sistema acadêmico, a IA se torna importante ao possibilitar o processamento inteligente das informações lançadas pelos professores e acessadas pelos alunos.

O processo de definição e utilização da IA neste projeto foi estruturado para aprimorar o desempenho do software e garantir a confiabilidade dos dados inseridos e corrigir erros no código.

A IA será aplicada ao sistema, analisando automaticamente as notas e detectando inconsistências que possam ocorrer durante os lançamentos, como valores fora do padrão. Além disso, o sistema poderá identificar alunos com baixo rendimento e, futuramente, ter um chat de perguntas e respostas com IA integrada para ajudar alunos a sanar suas dúvidas. Dessa forma, a Inteligência Artificial não apenas facilita o trabalho dos professores, mas também proporciona uma experiência melhor aos alunos, tornando o sistema mais inteligente.

As estratégias de utilização da IA no PIM envolvem a coleta e o tratamento contínuo dos dados acadêmicos, permitindo que o sistema aprenda com as informações e melhore seu desempenho ao longo do tempo, em conjunto com novas implementações no código. Essa integração possibilita agilidade ao realizar tarefas repetitivas e a redução de erros humanos.

Assim, a aplicação da Inteligência Artificial cumpre todos os requisitos da disciplina, contribuindo para um sistema acadêmico de notas mais inteligente e dinâmico.

### 9. PESQUISA, TECNOLOGIA E INOVAÇÃO

A pesquisa, tecnologia e inovação são elementos essenciais para o avanço e a competitividade no desenvolvimento de sistemas de informação. A pesquisa aplicada representa o processo sistemático de investigação para a geração de conhecimento novo ou para a resolução de problemas específicos, com foco na aplicação prática dos resultados no desenvolvimento tecnológico (OCDE, 2018).

A inovação tecnológica é compreendida como a implementação de novos ou significativamente melhorados produtos, processos, métodos organizacionais ou de marketing que trazem benefícios competitivos. Segundo Schumpeter (1942), a inovação é o motor do desenvolvimento econômico e tecnológico, capaz de transformar estruturas, mercados e práticas organizacionais normais. No desenvolvimento de sistemas, isso se traduz na constante busca por tecnologias que aumentem a eficiência, segurança, usabilidade e adaptabilidade dos softwares criados.

Nos últimos anos, avanços em áreas como inteligência artificial, aprendizado de máquina, big data e computação em nuvem têm revolucionado a forma como os sistemas são concebidos e utilizados. A inteligência artificial, por exemplo, possibilita a automação inteligente de processos, análise preditiva e suporte à decisão, capacidades que agregam alto valor ao sistema acadêmico colaborativo proposto, ampliando a eficiência do gerenciamento de dados e a interação entre usuários (Tidd e Bessant, 2018).

Além disso, a inovação tecnológica deve ser acompanhada por uma pesquisa constante voltada não apenas ao desenvolvimento de novas ferramentas, mas também à análise de seu impacto social, econômico e ambiental. A sustentabilidade tecnológica passa a ser um requisito indispensável, garantindo que os sistemas produzidos não apenas atendam às necessidades atuais, mas também sejam capazes de evoluir e se adaptar a futuras demandas, minimizando impactos negativos.

Outro aspecto relevante é a inovação aberta, que envolve a colaboração entre diferentes agentes como universidades, empresas e governo para a criação e disseminação de conhecimento tecnológico. Esse conceito é especialmente valioso em ambientes acadêmicos, onde o compartilhamento de informações e recursos pode acelerar o desenvolvimento de soluções inovadoras (Chesbrough, 2003).

Integradas ao desenvolvimento do sistema deste projeto, proporcionam um diferencial competitivo e contribuem para a transformação digital no âmbito acadêmico, promovendo a colaboração eficiente, segurança dos dados e melhor experiência para os usuários. Esse alinhamento permite a criação de um ambiente digital dinâmico e resiliente, capaz de apoiar as necessidades institucionais e acadêmicas de forma integrada e inovadora.

### 10. DESENVOLVIMENTO DO PROJETO

O desenvolvimento do projeto baseou-se na integração entre os conhecimentos adquiridos nas disciplinas do curso de Analise e Desenvolvimento de Sistemas, aplicando técnicas de programação estruturada, engenharia de software ágil e princípios de redes de computadores. O sistema acadêmico foi construído utilizando Python para a interface gráfica (com a biblioteca Tkinter) e C para os módulos de backend.

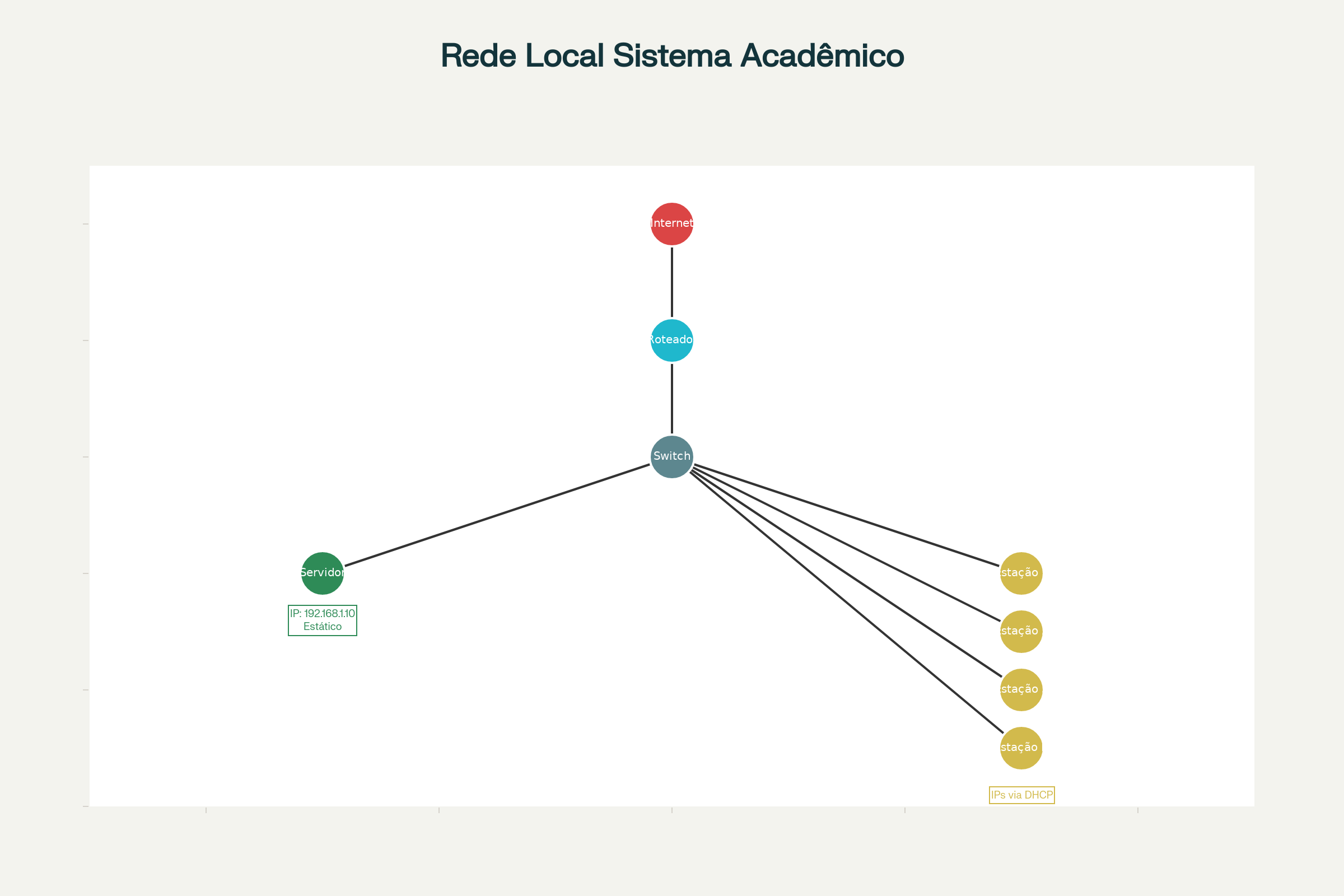
Durante o processo de desenvolvimento, foram implementadas funções para cadastro de alunos e professores, registro e consulta de notas, geração de gráficos estatísticos e armazenamento de dados em JSON. A comunicação entre Python e C foi realizada por meio de bibliotecas dinâmicas (DLL), utilizando o módulo ctypes, possibilitando que as rotinas em C fossem executadas a partir do código Python aonde se encontrava o frontend (Tkinter).

A aplicação foi projetada para operar em uma rede local (LAN), permitindo o acesso simultâneo de múltiplos usuários. Cada perfil de acesso (administrador, professor e aluno) possui funções específicas, o que garante segurança no uso do sistema. O desenvolvimento também incluiu práticas de sustentabilidade, como a eliminação do uso de papel, e considerou aspectos de educação ambiental, como apresentada no documento.

**10.1 Inteligência Artificial No Futuro do Projeto**

No futuro, a Inteligência Artificial poderia ser aplicada a esse sistema para automatizar diversas tarefas, por exemplo: algoritmos de aprendizado de máquina poderiam analisar o histórico de notas e identificar alunos com risco de baixo desempenho, sugerindo intervenções pedagógicas personalizadas aos professores. O sistema também poderia prever tendências de rendimento por turma ou matéria, auxiliando a gestão escolar em decisões estratégicas. Poderiam ser implementados chatbots integrados que ofereceriam suporte instantâneo a alunos e professores, respondendo dúvidas sobre notas, prazos e cadastros. Além disso, técnicas de processamento de linguagem natural poderiam corrigir automaticamente erros de digitação nos cadastros. A IA também ajudaria na detecção de inconsistências nos dados, garantindo maior precisão e segurança acadêmica.

**10.2 Diagrama de Rede**



**10.3 Diagrama UML - Yuki**

**10.4 YUKI CASO PRECISE**

**10.5 YUKI CASO PRECISE**

**10.6 YUKI CASO PRECISE**

21

### 11. CONCLUSÃO

O desenvolvimento do Sistema Acadêmico com o futuro apoio de Inteligência Artificial permitiu compreender, na prática, a integração entre teoria e aplicação tecnológica. A hipótese de que seria possível criar um protótipo de uma ferramenta eficiente, colaborativa e sustentável para gestão acadêmica foi confirmada. O sistema alcançou seus objetivos ao oferecer funcionalidades que facilitam o trabalho dos professores e profissionais da área de ensino. O uso de práticas ágeis e linguagens complementares, como Python e C, proporcionou um desenvolvimento modular, escalável e de alto desempenho. O grupo constatou que a aplicação integrada dos conhecimentos das disciplinas envolvidas resultou em um protótipo tecnológico relevante e alinhado às necessidades atuais da educação. Assim, o projeto consolidou-se como uma experiência enriquecedora de aprendizado, inovação e responsabilidade social.

Para futuras melhorias no projeto esperamos uma migração de linguagem para linguagens mais adaptáveis como React, NodeJS e até mesmo visamos o uso de Kotlin para desenvolvimento de uma boa interface front-end para o projeto assim proporcionando a todos os nossos educadores e alunos uma ótima interface para uso do sistema em formato mobile, assim proporcionando uma versatilidade de uso visando que os profissionais e alunos possam tanto lançar notas como verificar notas de qualquer lugar com seus celulares.

### 12. REFERÊNCIAS

BEZERRA, E. Princípios de Análise e Projeto de Sistemas com UML. 3. ed. Rio de Janeiro: Elsevier, 2018.

CHESBROUGH, H. Open Innovation: The New Imperative for Creating and Profiting from Technology. Boston: Harvard Business School Press, 2003.

DEITEL, P. J.; DEITEL, H. M. C: Como Programar. 8. ed. São Paulo: Pearson, 2016.

FERREIRA, F. Redes de Computadores: Fundamentos, Tecnologias e Aplicações. 2. ed. São Paulo: Érica, 2020.

KERNIGHAN, B. W.; RITCHIE, D. M. The C Programming Language. 2. ed. Englewood Cliffs: Prentice-Hall, 1988.

LUTZ, M. Learning Python. 5. ed. Sebastopol: O’Reilly Media, 2013.

MARTINS, J. R.; SILVA, T. C. Educação Ambiental e Sustentabilidade: Práticas e Desafios. São Paulo: Atlas, 2021.

OCDE – ORGANIZAÇÃO PARA COOPERAÇÃO E DESENVOLVIMENTO ECONÔMICO. Manual de Oslo: Diretrizes para a Coleta e Interpretação de Dados sobre Inovação. 4. ed. Paris: OCDE/Eurostat, 2018.

PRESSMAN, R. S.; MAXIM, B. R. Engenharia de Software: Uma Abordagem Profissional. 8. ed. Porto Alegre: AMGH, 2016.

SCHILDT, H. C Completo e Total. 3. ed. São Paulo: Pearson, 2017.

SCHWABER, K.; SUTHERLAND, J. Guia do Scrum: O Guia Definitivo para o Scrum: As Regras do Jogo. Scrum.org, 2020.

SCHUMPETER, J. A. Capitalism, Socialism and Democracy. New York: Harper & Brothers, 1942.

TIDD, J.; BESSANT, J. Managing Innovation: Integrating Technological, Market and Organizational Change. 6. ed. Hoboken: John Wiley & Sons, 2018.

VAN ROSSUM, G.; DRAKE, F. L. Python Tutorial. 4.0.3. Python Software Foundation, 2022.

VASCONCELOS, A.; MOURA, R. Scrum: Gestão Ágil para Projetos de Sucesso. 2. ed. Rio de Janeiro: Brasport, 2020.

WAZLAWICK, R. S. Análise e Projeto de Sistemas de Informação Orientados a Objetos. 4. ed. Rio de Janeiro: Elsevier, 2021.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. The Unified Modeling Language User Guide. 2. ed. Boston: Addison-Wesley, 2005. ISBN 9780321112069.

PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de Software: uma abordagem profissional. 9. ed. Porto Alegre: AMGH, 2016. ISBN 9788580554795. Disponível em: https://integrada.minhabiblioteca.com.br/#/books/9786558040118/. Acesso em: 01 nov. 2025.

PRESSMAN, Roger S. Engenharia de Software: uma abordagem profissional. 9. ed. Porto Alegre: AMGH, 2020.

ASTAH Community. Astah Community – Modeling Tool. Disponível em: https://astah.net/. Acesso em: 01 nov. 2025.

LUCIDCHART. Lucidchart – Diagram Software. Disponível em: https://www.lucidchart.com/. Acesso em: 01 nov. 2025.

TRELLO. Trello – Organize Anything, Together. Disponível em: https://trello.com/. Acesso em: 31 out. 2025.

# FICHA DE CONTROLE DO PIM

Grupo Nº \_\_\_\_\_ Ano\_\_\_\_\_\_ Período: \_\_\_\_\_ Orientador \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Tema:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Alunos:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| RA | Nome | E-mail | Curso | Visto do aluno |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

Registros:

|  |  |
| --- | --- |
| Data do encontro | Observações |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

