

**FACULDADE DE TECNOLOGIA DE SÃO JOSÉ DOS CAMPOS
FATEC PROFESSOR JESSEN VIDAL**

VINICIUS ASSIS PERETTA

**PORTFÓLIO DE APRENDIZAGEM DE PROJETOS
INTEGRADO (API) E RULE IA**

São José dos Campos
2026

VINICIUS ASSIS PERETTA

**PORTFÓLIO DE APRENDIZAGEM DE PROJETOS
INTEGRADO (API) E RULE IA**

Trabalho de Graduação apresentado à Faculdade de Tecnologia de São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Emanuel Mineda Carneiro

São José dos Campos
2026

VINICIUS ASSIS PERETTA

PORTFÓLIO DE APRENDIZAGEM DE PROJETOS INTEGRADO (API) E RULES IA

Trabalho de Graduação apresentado à Faculdade de Tecnologia de São José dos Campos, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Prof. Dr. Emanuel Mineda Carneiro- FATEC

Prof. Dr. Nilo Jeronimo Vieira - FATEC

Prof. Dr. Gerson da Penha Neto - FATEC

15/06/2026

DATA DA APROVAÇÃO

AGRADECIMENTOS

À Faculdade de Tecnologia de São José dos Campos (Fatec), pela infraestrutura e pela oportunidade de formação acadêmica e profissional de excelência ao longo destes anos. Aos professores, em especial, o orientador Emanuel Mineda Carneiro, pelos ensinamentos compartilhados, pela paciência e pelas orientações técnicas que foram fundamentais para a superação dos desafios propostos em cada semestre.

Aos colegas de turma pela parceria nas equipes de desenvolvimento, pelo auxílio mútuo nas madrugadas de código e pelas discussões que enriqueceram o aprendizado coletivo e fortaleceram os laços de amizade.

A todos que, direta ou indiretamente, contribuíram para que este ciclo se encerrasse com êxito, o autor expressa o seu mais profundo reconhecimento.

RESUMO

O presente portfólio acadêmico apresenta o desenvolvimento de soluções tecnológicas estruturadas ao longo de seis semestres da graduação em Análise e Desenvolvimento de Sistemas, com foco na aplicação prática de metodologias ágeis e engenharia de software. O objetivo central deste documento consiste em documentar a trajetória de evolução técnica e profissional, evidenciando a resolução de problemas reais propostos por empresas parceiras e a instituição de ensino FATEC SJC. A metodologia adotada fundamenta-se no framework Scrum, utilizando o modelo de Aprendizagem por Projetos Integradores (API) para a entrega de produtos mínimos viáveis (MVPs) em ciclos incrementais de desenvolvimento. Durante o percurso, desenvolveram-se sistemas variados, abrangendo desde plataformas de ensino e ferramentas de tradução de linguagem natural para SQL até aplicativos de geolocalização rural e sistemas de gestão com inteligência artificial. Os resultados obtidos demonstram a eficácia da integração entre tecnologias modernas, como Python, Java, Node.js, React e Large Language Models (LLMs), e a gestão eficiente de requisitos e equipes. Conclui-se que a participação ativa nesses projetos consolidou competências essenciais em desenvolvimento full-stack, automação de processos e liderança técnica, preparando o autor para os desafios do mercado de tecnologia. A experiência acumulada reforça a relevância do ensino aplicado para a formação de profissionais capazes de transformar necessidades organizacionais em ativos digitais de alto valor, promovendo inovação e eficiência em diversos domínios de atuação.

Palavras-Chave: Metodologias Ágeis; Inteligência Artificial; Desenvolvimento Web; Gestão de Projetos

ABSTRACT

This academic portfolio presents the development of structured technological solutions throughout six semesters of the Undergraduate Program in Analysis and Systems Development, focusing on the practical application of agile methodologies and software engineering. The main objective of this document is to record technical and professional evolution, highlighting the resolution of real-world problems proposed by partner companies and educational institutions. The adopted methodology is based on the Scrum framework, utilizing the Project-Based Learning (PBL) model to deliver Minimum Viable Products (MVPs) in incremental development cycles. Throughout the program, various systems were developed, ranging from educational platforms and natural language-to-SQL translation tools to rural geolocation applications and management systems powered by Artificial Intelligence. The results demonstrate the effectiveness of integrating modern technologies, such as Python, Java, Node.js, React, and Large Language Models (LLMs), with efficient requirement and team management. It is concluded that active participation in these projects consolidated essential skills in full-stack development, process automation, and technical leadership, preparing the author for the challenges of the technology market. The accumulated experience reinforces the relevance of applied learning in forming professionals capable of transforming organizational needs into high-value digital assets, promoting innovation and efficiency across various domains of activity

Keywords: Agile Methodologies; Artificial Intelligence; Web Development; Project Management

SUMÁRIO

1. INTRODUÇÃO	8
2. PROJETOS	9
3. CONSIDERAÇÕES FINAIS.....	16
REFERÊNCIAS BIBLIOGRÁFICAS	17
APÊNDICE A – PORTIFÓLIO.....	18

1. INTRODUÇÃO

O presente portfólio apresenta a trajetória de desenvolvimento acadêmico e profissional realizada por meio dos Projetos Integradores (API) entre os anos de 2023 e 2026. O modelo pedagógico adotado foca na resolução de problemas reais propostos por empresas parceiras, utilizando metodologias ágeis e tecnologias de ponta para a entrega de soluções de software de alto valor agregado.

Ao longo dos seis semestres, foram identificados e endereçados desafios críticos em diversos setores. As problemáticas abrangeram desde a falta de métodos estruturados para a aplicação da metodologia Scrum e dificuldades na extração de dados por usuários leigos (SQL), até lacunas em processos de RH, gestão de projetos institucionais e ausência de endereçamento formal em áreas rurais. Mais recentemente, o foco voltou-se para a complexidade do gerenciamento de regras de negócio dinâmicas e o cálculo automatizado de comissionamento. A relevância desses projetos reside na capacidade de transformar processos manuais, burocráticos e ineficientes em sistemas automatizados, transparentes e acessíveis, promovendo a autonomia dos usuários e a precisão organizacional.

Os resultados obtidos demonstram a entrega de produtos mínimos viáveis (MVPs) funcionais que sanaram as dores das empresas parceiras. Destacam-se a criação de uma plataforma didática para Scrum, um tradutor de linguagem natural para SQL, sistemas de gestão de pessoas e projetos, e um aplicativo de endereçamento rural via Plus Codes. O ciclo encerrou-se com o desenvolvimento de um agente de IA capaz de interpretar regras de negócio complexas, consolidando o domínio técnico em engenharia de software e inteligência artificial aplicado ao mercado corporativo.

Para a viabilização das soluções, utilizou-se um ecossistema tecnológico diversificado, adaptado às necessidades específicas de cada ciclo:

- Front-end: Desenvolvimento de interfaces web e mobile utilizando tecnologias como HTML5, CSS3, Bootstrap, React, Next.js, Tailwind CSS, Vue.js 3 e React Native.
- Back-end: Implementação de APIs robustas com Python (Flask e FastAPI), Java (Spring Boot), Node.js (Express) e TypeScript.
- Inteligência Artificial e Dados: Integração de Large Language Models (LLMs) via LangChain, Ollama e Google Vertex AI, além do gerenciamento de bancos de dados relacionais (PostgreSQL, MySQL) e não-relacionais (MongoDB, Firebase, LanceDB).
- Infraestrutura e Processos: Uso de serviços em nuvem (AWS), containerização, CI/CD com GitHub Actions e ferramentas de prototipagem como Figma.

2. PROJETOS

Os projetos API correspondem às atividades desenvolvidas ao longo da graduação, organizadas cronologicamente por ano e semestre. Cada projeto integrador foi elaborado a partir de demandas reais propostas por docentes e empresas parceiras, com o objetivo de aplicar, na prática, os conhecimentos adquiridos no curso. As descrições apresentam o contexto de desenvolvimento, os objetivos, as soluções implementadas e as tecnologias utilizadas, podendo ser consultadas integralmente no Apêndice A deste documento. Nas subseções a seguir, apresenta-se uma síntese de cada projeto.

Tabela 1 – Lista de Projetos API

Título do Projeto	Link do GitHub
Scrum Tutor	https://github.com/Phoenix-Team-Fatec/ScrumTutor
DataEase	https://github.com/Phoenix-Team-Fatec/DataEase
OAK-RH	https://github.com/Phoenix-Team-Fatec/OAK-RH
Lumen	https://github.com/Phoenix-Team-Fatec/API-4
GeoMaps	https://github.com/Phoenix-Team-Fatec/geo-maps
Rules AI	https://github.com/Phoenix-Team-Fatec/API_6

Fonte: Elaborado por <Vinicius Assis Peretta>, (2026).

2.1 2024 – 1º semestre: DataEase

No primeiro semestre de 2024, desenvolveu-se o projeto DataEase (Peretta, 2024a). A proposta foi realizada em parceria com o CADI. Identificou-se que usuários de perfil não técnico enfrentavam severas barreiras para acessar e interpretar informações em bancos de dados, devido à falta de domínio da linguagem SQL. Tal cenário gerava uma dependência excessiva do setor de TI para a extração de relatórios simples. A contribuição pretendida pelo projeto foi democratizar o acesso aos dados, permitindo que consultas complexas fossem realizadas por meio de linguagem natural, conferindo autonomia aos usuários e agilidade na tomada de decisão.

Neste projeto, o autor atuou como membro do Developer Team, focando na implementação de funcionalidades críticas e na integração de inteligência artificial. Foi utilizado a biblioteca LangChain4J (LANGCHAIN4J, 2026), desenvolvida especificamente para facilitar a integração de modelos de linguagem de grande escala (LLMS) em ecossistemas java. Entre

as contribuições realizadas, destacam-se o desenvolvimento da função de inicialização do servidor local de modelos de linguagem, a criação de métodos para recuperação de metadados do banco de dados (como o nome e identificadores de usuário) e a otimização dos processos de cadastro de usuários. Além disso, houve uma participação ativa na construção e refinamento da interface de chat, garantindo que o fluxo de interação fosse intuitivo e eficiente.

A solução desenvolvida consiste em uma plataforma desktop que funciona como um tradutor inteligente de linguagem natural para SQL. O usuário interage com o sistema por meio de um campo de chat, onde insere perguntas em português (ex: "Quais foram as vendas do último mês?"); o sistema, utilizando modelos de linguagem (LLMs), processa a requisição, gera o comando SQL correspondente, executa-o no banco de dados configurado e retorna os resultados diretamente na tela. A aplicação permite ainda o cadastro de múltiplos bancos de dados e a alternância entre diferentes modelos de IA, oferecendo flexibilidade técnica e facilidade de uso.

As tecnologias e ferramentas utilizadas no projeto estão descritas a seguir:

- **Linguagens e Bibliotecas:** Utilizou-se a linguagem Java como base do sistema, integrada ao framework LangChain4j para a orquestração da inteligência artificial.
- **Inteligência Artificial e Dados:** Empregou-se o LM Studio para a execução local de LLMs e o Hugging Face para acesso a modelos abertos. O armazenamento de dados e configurações foi gerenciado pelo MySQL.
- **Repositório do Projeto:** O histórico de desenvolvimento, commits e documentação técnica estão registrados no GitHub.

2.2 2024 – 2º semestre: OAK: RH

No segundo semestre de 2024, desenvolveu-se o projeto intitulado OAK-RH (Peretta, 2024b). A proposta foi realizada em parceria com a Youtan, uma empresa de desenvolvimento de software. Identificou-se que o departamento de Recursos Humanos da organização enfrentava dificuldades críticas na coleta e organização de feedbacks sobre a cultura organizacional e projetos internos, além de realizar o controle de alocação de equipes de maneira manual. Tal cenário resultava em perda de informações e falhas de comunicação. A contribuição pretendida consistiu em centralizar esses processos em um ambiente digital, otimizando a análise do clima organizacional e a distribuição de profissionais por meio de uma abordagem tecnológica e estruturada.

Neste projeto, o autor atuou como integrante do Developer Team, sendo responsável pelo desenvolvimento de funcionalidades essenciais no ecossistema da aplicação. Entre as

principais contribuições, destaca-se a implementação de rotas e controladores para o gerenciamento de equipes, permitindo operações de criação, listagem, atualização e exclusão de dados. Adicionalmente, trabalhou-se na estruturação da lógica de serviços para garantir a integridade das informações e na integração entre o backend e a interface, assegurando que os feedbacks coletados fossem armazenados e recuperados de forma eficiente para a geração de relatórios.

A solução desenvolvida configura-se como um sistema web moderno e centralizado para a gestão de capital humano. A plataforma permite que o usuário interaja por meio de dashboards intuitivos, onde é possível criar formulários personalizados, responder a

pesquisas de clima e visualizar análises de desempenho individual e coletivo. O sistema conta com diferentes níveis de acesso e funcionalidades para exportação de relatórios em PDF, eliminando processos manuais e oferecendo uma visão estratégica sobre a alocação de colaboradores e a satisfação interna.

As tecnologias e ferramentas utilizadas para a execução do projeto estão sintetizadas abaixo:

- **Linguagens e Frameworks:** O desenvolvimento do backend foi realizado em Node.js com TypeScript e Express, enquanto o frontend utilizou React para a criação de interfaces dinâmicas.
- **Banco de Dados e Ferramentas:** Utilizou-se o PostgreSQL como banco de dados relacional e o Figma para a prototipagem da interface. O controle de versão foi gerenciado integralmente via Git.
- **Repositório do Projeto:** O código-fonte, a documentação das APIs e o histórico de sprints estão disponíveis para consulta no GitHub.

2.3 2025 – 1º semestre: Lumen

No primeiro semestre de 2025, desenvolveu-se o projeto intitulado Lumen (Peretta, 2025a). A proposta foi realizada em parceria com a FAPG – Fundação de Apoio à Pesquisa de Pós-Graduandos, uma instituição voltada ao fomento e suporte de atividades acadêmicas. Identificou-se que a organização não possuía um método estruturado para o acompanhamento de seus projetos, o que resultava em dificuldades para visualizar prazos, tarefas e responsáveis. A ausência de um sistema centralizado exigia que relatórios fossem gerados manualmente, comprometendo a transparência e a eficiência da gestão. A contribuição pretendida pelo projeto foi a implementação de uma plataforma de gestão integrada, capaz de acelerar a tomada de decisão e fortalecer o controle sobre as iniciativas da fundação.

Neste projeto, o autor atuou como integrante do Developer Team, participando ativamente na arquitetura e no desenvolvimento das funcionalidades do sistema. As contribuições focaram na criação de interfaces intuitivas para o gerenciamento de projetos e na implementação de lógicas para o acompanhamento de status de tarefas. Além disso, trabalhou-se na integração de um agente de inteligência artificial local, responsável por interpretar comandos do usuário para automatizar ações como a conclusão de tarefas e a listagem de itens atrasados, otimizando a experiência de uso e a produtividade da equipe gestora. Foi utilizado o Gemma 2b (OLLAMA, 2026), é um modelo de inteligência artificial aberto e leve, desenvolvido pelo google deepMInf, ele foi optado pela facilidade de ser utilizado localmente por conta da sua quantidade de parâmetros e a por ter sido treinado com códigos de programação e textos matemáticos.

A solução desenvolvida consiste em uma plataforma web centralizada para gestão de projetos. O usuário interage com o sistema através de um ambiente seguro que permite o

cadastro, edição e listagem de projetos, além da organização de etapas, tarefas e subtarefas. Um diferencial da plataforma é a inclusão de um assistente inteligente que processa mensagens em linguagem natural para realizar alterações rápidas no status das atividades. O sistema oferece ainda funcionalidades para a atribuição de responsáveis e recuperação de itens excluídos, eliminando a necessidade de planilhas manuais e fornecendo uma visão clara do andamento dos trabalhos.

As tecnologias e ferramentas utilizadas para a execução do projeto estão sintetizadas abaixo:

- **Linguagens e Frameworks:** O backend foi desenvolvido utilizando Node.js e TypeScript, enquanto o frontend foi construído com Vue.js para garantir uma interface reativa e moderna.
- **Inteligência Artificial e Ferramentas:** Utilizou-se o Ollama para a execução local do modelo de linguagem Gemma:2b, integrado via Axios. O banco de dados utilizado para a persistência das informações foi o PostgreSQL.
- **Repositório do Projeto:** Todo o código-fonte, documentação de requisitos e registros das sprints estão armazenados no GitHub.

2.4 2025 – 2º Semestre: GeoMaps

No primeiro semestre de 2023, desenvolveu-se o projeto API intitulado Scrum Tutor (Peretta, 2023a). Para esse projeto era profundamente importante a compreensão do framework Scrum que segundo Sutherland (2016), é descrito como um sistema autocorretivo, evolucionário e adaptativo desenhado para ser uma forma mais rápida e confiável de criar

produtos. A proposta desta API foi concebida em parceria com o CADI. O problema central identificado foi a dificuldade enfrentada por equipes no processo de organização do trabalho, especialmente na definição de prioridades e na compreensão dos papéis dentro de metodologias ágeis. A contribuição pretendida consistiu em facilitar o entendimento do framework Scrum por meio de uma abordagem prática, didática e interativa.

Neste projeto, o autor autou no papel de Product Owner (PO), além de atuar diretamente no desenvolvimento técnico da solução. Como PO, realizaram-se o entendimento e o refinamento de requisitos junto ao cliente, definindo-se a implementação de duas abordagens avaliativas: o método PACER para desempenho de equipe e questionários teóricos ao final de cada artigo explicativo. Adicionalmente, aplicaram-se práticas de metodologias ágeis por meio da criação e manutenção do backlog do produto, organização das sprints e atualização contínua da documentação no repositório GitHub.

A solução desenvolvida configura-se como uma plataforma web projetada para guiar usuários através dos elementos essenciais do Scrum, como eventos, papéis e artefatos. O usuário interage com o sistema navegando por páginas informativas e utilizando uma área de avaliação de equipes, cujos resultados são processados e enviados automaticamente por e-mail. No âmbito do desenvolvimento, implementaram-se ajustes de front-end para garantir a usabilidade e a coerência visual, além da estruturação da lógica de envio de mensagens via servidor SMTP.

As tecnologias e ferramentas utilizadas para a execução do projeto estão sintetizadas abaixo:

- **Linguagens e Frameworks:** Utilizaram-se Python e o framework Flask para o desenvolvimento do backend, enquanto HTML5, CSS3 e Bootstrap foram empregados na construção da interface web.
- **Ferramentas e Infraestrutura:** O design da interface foi elaborado no Figma, a hospedagem ocorreu na plataforma AWS e o controle de versão foi gerenciado via Git.
- **Repositório do Projeto:** Todo o código-fonte e a documentação das sprints encontram-se disponíveis no GitHub.

2.5 ANO 3 – 1º semestre

No segundo semestre de 2025, desenvolveu-se o projeto GeoMaps (Peretta, 2025b). A proposta foi realizada em parceria com a Visiona Tecnologia Espacial S.A., empresa que atua no setor de sensoriamento remoto e soluções espaciais. Identificou-se que uma parcela significativa das propriedades rurais no Brasil carece de um endereço formal ou CEP definido, o que compromete a logística, o acesso a serviços públicos e a precisão em registros comerciais

e agrícolas. A contribuição pretendida pelo projeto foi solucionar a ausência de identificação territorial clara por meio de um sistema de endereçamento digital, facilitando a localização e a roteirização em áreas rurais.

Neste projeto, o autor desempenhou o papel de Scrum Master, sendo responsável por zelar pela aplicação da metodologia ágil e garantir a fluidez dos processos da equipe. Como Scrum Master, atuou-se na facilitação das cerimônias (Daily, Planning e Review), na remoção de impedimentos que pudessem comprometer a velocidade do time e na organização do fluxo de trabalho por meio do refinamento constante das metas de cada sprint. Paralelamente, contribuiu-se tecnicamente no desenvolvimento do sistema, focando na implementação de rotas de integração, manipulação de dados geoespaciais e na estruturação da lógica de exibição das propriedades no mapa.

Foi implementado o GitFlow (VALENTE, 2020) com aplicações ao contexto da API, principalmente a ideia de branches develop, main, feature e hotfix, pull requests, proteção de branches, revisões do código e testes manuais.

A solução desenvolvida consiste em um aplicativo móvel voltado para o endereçamento digital de imóveis rurais. O usuário interage com a plataforma por meio de um mapa interativo, onde é possível visualizar propriedades cadastradas no CAR (Cadastro Ambiental Rural), identificar localizações exatas via códigos Plus Codes e realizar registros colaborativos sobre as condições das vias. O sistema oferece ainda funcionalidades de roteirização rural, alertas geolocalizados e um painel de administração multinível, permitindo uma gestão eficiente dos dados territoriais e facilitando a navegação em regiões sem cobertura de CEP tradicional.

As tecnologias e ferramentas utilizadas no projeto estão detalhadas a seguir:

- **Linguagens e Frameworks:** Utilizou-se Python com o framework FastAPI para a construção da API backend, enquanto o frontend mobile foi desenvolvido em React Native (TypeScript).
- **Banco de Dados e Mapas:** O armazenamento de dados não relacionais e geoespaciais foi gerenciado pelo MongoDB, com a integração da API do Google Maps para visualização e cálculos de rota.
- **Repositório do Projeto:** Todo o código-fonte, gerenciamento de tags e documentação técnica estão centralizados no GitHub.

2.6 2026 – 1º semestre: RulesAI

No primeiro semestre de 2026, desenvolveu-se o projeto intitulado RulesAI (Peretta, 2026a). A proposta foi realizada em parceria com a Dom Rock, uma empresa especializada em

soluções de inteligência de dados e análise preditiva. Identificou-se que organizações enfrentam desafios significativos na gestão de regras de negócio dinâmicas, especialmente no contexto de cálculo de comissionamento mensal, onde alterações frequentes em políticas de vendas e legislações geram inconsistências operacionais e perda de rastreabilidade. A contribuição pretendida consistiu em criar um ecossistema inteligente que centralizasse essas regras, garantindo transparência, precisão nos cálculos e facilidade na atualização de diretrizes complexas.

Neste projeto, o autor atuou como integrante do Developer Team, com foco especial na camada de inteligência artificial e integração de dados. As contribuições envolveram o desenvolvimento de um agente de IA capaz de interpretar documentos de regras em formato PDF, utilizando técnicas de RAG (Retrieval-Augmented Generation) para responder a consultas sobre comissionamento. Além disso, trabalhou-se na implementação de rotas no backend para o processamento de regras e na construção de interfaces reativas que permitem aos gestores visualizar e validar os cálculos de maneira intuitiva, garantindo que o conhecimento organizacional fosse preservado e aplicado corretamente.

A solução desenvolvida, denominada RulesAI, configura-se como um sistema híbrido composto por uma API robusta, um painel administrativo web e um agente de inteligência artificial. O usuário interage com a plataforma por meio de um dashboard onde pode gerenciar perfis de funcionários, configurar exceções de cargo e disparar o processamento de comissões. Paralelamente, um chat inteligente permite que o usuário realize perguntas sobre as regras vigentes, recebendo respostas fundamentadas na base de conhecimento da empresa. O sistema automatiza cálculos complexos, reduz conflitos entre colaboradores e oferece uma trilha de auditoria completa para cada decisão tomada pelo motor de regras.

As tecnologias e ferramentas utilizadas para a execução do projeto estão sintetizadas abaixo:

- **Linguagens e Frameworks:** O backend foi estruturado com Java (Spring Boot) para o motor de regras e Python (FastAPI) para a orquestração da IA. O frontend foi desenvolvido em Vue.js.
- **Inteligência Artificial e Dados:** Utilizaram-se os frameworks LangChain e LangGraph para a lógica do agente, com groq (GROQ, 2026) e hugging face (HUGGING FACE, 2026). O armazenamento foi gerenciado por MongoDB (NoSQL) e LanceDB (vetorial).
- **Repositório do Projeto:** O código-fonte, os experimentos de machine learning rastreados via MLflow e a documentação técnica encontram-se registrados no GitHub.

3. CONSIDERAÇÕES FINAIS

A trajetória acadêmica percorrida ao longo dos seis semestres do curso de Análise e Desenvolvimento de Sistemas, por meio dos Projetos Integradores, proporcionou uma evolução consistente e estruturada das competências técnicas e comportamentais. A transição entre o primeiro projeto, focado nos fundamentos da agilidade, até o último ciclo, centrado em arquiteturas complexas de Inteligência Artificial, evidencia o amadurecimento profissional diante de demandas reais do mercado de trabalho. O contato direto com empresas parceiras permitiu que o conhecimento teórico fosse validado na prática, transformando conceitos de sala de aula em soluções tecnológicas tangíveis.

As atividades integradoras foram fundamentais para o desenvolvimento de habilidades essenciais, como a capacidade de trabalho em equipe e a organização de fluxos produtivos sob pressão de prazos. A aplicação rigorosa de metodologias ágeis, especialmente o framework Scrum, permitiu o exercício de diferentes papéis, desde a gestão de requisitos como Product Owner e Scrum Master até a implementação técnica como membro do time de desenvolvimento. O enfrentamento de desafios complexos — como a integração de modelos de linguagem locais, a manipulação de dados geoespaciais e a automação de regras de negócio — exigiu a busca constante por novas tecnologias e a adoção de soluções criativas para problemas de interoperabilidade e desempenho.

Os projetos desenvolvidos consolidaram o domínio sobre diversas linguagens e ferramentas, incluindo Python, Java, JavaScript, TypeScript e frameworks como React, Vue e Spring Boot. Além do ganho técnico, houve uma evolução significativa na compreensão da importância da experiência do usuário (UX) e da qualidade de software em ambientes corporativos. A vivência em projetos com diferentes propósitos — educacional, social e comercial — ampliou a visão estratégica sobre como a tecnologia pode atuar como facilitadora de processos e propulsora de inovação.

Em suma, as experiências obtidas por meio das APIs foram determinantes para a formação de um perfil profissional resiliente e adaptável. A reflexão sobre esse percurso acadêmico demonstra que a integração entre ensino e prática de mercado é o pilar que sustenta a preparação para os desafios da carreira em tecnologia. Conclui-se que o aprendizado contínuo, exercitado ao longo desses seis ciclos, constitui a base necessária para a atuação em um setor em constante transformação, reafirmando o valor da graduação para o desenvolvimento técnico, acadêmico e pessoal.

REFERÊNCIAS BIBLIOGRÁFICAS

GROQ. Groq Documentation. 2026. Disponível em: <https://console.groq.com/docs/>. Acesso em: 23 maio 2026.

HUGGING FACE. Hugging Face: the AI community building the future. 2026. Disponível em: <https://huggingface.co/>. Acesso em: 23 maio 2026.

LANGCHAIN4J. Intro. 2026. Disponível em: <https://docs.langchain4j.dev/intro/>. Acesso em: 23 maio. 2026.

OLLAMA. Gemma:2b. 2026. Disponível em: <https://ollama.com/library/gemma:2b>. Acesso em: 23 maio. 2026.

PERETTA, Vinicius Assis. Projeto de API – 1º Semestre. GitHub, 2023. Disponível em: <https://github.com/Phoenix-Team-Fatec/ScrumTutor>. Acesso em: 10 maio. 2026.

PERETTA, Vinicius Assis. Projeto de API – 2º Semestre. GitHub, 2023. Disponível em: <https://github.com/Phoenix-Team-Fatec/DataEase>. Acesso em: 10 maio. 2026.

PERETTA, Vinicius Assis. Projeto de API – 3º Semestre. GitHub, 2023. Disponível em: <https://github.com/Phoenix-Team-Fatec/OAK-RH>. Acesso em: 10 maio. 2026.

PERETTA, Vinicius Assis. Projeto de API – 4º Semestre. GitHub, 2023. Disponível em: <https://github.com/Phoenix-Team-Fatec/API-4>. Acesso em: 10 maio. 2026.

PERETTA, Vinicius Assis. Projeto de API – 5º Semestre. GitHub, 2023. Disponível em: <https://github.com/Phoenix-Team-Fatec/geo-maps>. Acesso em: 10 maio. 2026.

PERETTA, Vinicius Assis. Projeto de API – 6º Semestre. GitHub, 2023. Disponível em: https://github.com/Phoenix-Team-Fatec/API_6. Acesso em: 10 maio. 2026.

SUTHERLAND, Jeff. Scrum: a arte de fazer o dobro do trabalho na metade do tempo. 2. ed. Rio de Janeiro: Sextante, 2016.

VALENTE, Marco Tulio. Gerenciando Branches com Git-flow, GitHubFlow. Engenharia de Software Moderna, 2020. Disponível em: <https://engsoftmoderna.info/artigos/gitflow.html>. Acesso em: 23 maio 2026.

APÊNDICE A – PORTIFÓLIO

1.1 Apresentação do Aluno



Nome: Vinicius Assis Peretta

Idade: 20 anos

Curso: Análise e Desenvolvimento de Sistemas – FATEC

Histórico Acadêmico

- *Ensino Médio:* Escola Objetivo – Caçapava/SP
- *Ensino Técnica:* SENAI – Curso Técnico em Análise e Desenvolvimento de Sistemas
- *Ensino Superior:* FATEC – Curso Tecnólogo em Análise e Desenvolvimento de Sistemas

Motivação para Ingresso na FATEC

Minha maior motivação foi o interesse em aprofundar conhecimentos na área de Tecnologia da Informação e a aplicação deles. AS API e projetos onde poderia colocar em prática as tecnologias foi algo que chamou muita minha atenção.

Histórico Profissional

- 2024 - 2025: Estagiário em Análise e Desenvolvimento de Sistemas
Foco em Qualidade de Software (QA) e DevOps.
 - Conversão de testes java para Kotlin
 - Otimização de testes
 - Testes manuais
- 2025 - ATUAL: Analista de Sistemas Júnior
Foco em Qualidade de Software (QA) e DevOps.
 - Criar testes automatizados para mobile E2E utilizando Java e Espresso.

- Estudo e aplicação de novas tecnologias para teste (robolectric e cypress).
- Otimizar testes automatizados já criados.
- Scripts para automatizar processos internos.

Informações de Contato

- E-mail: viniasisperetta@gmail.com
 - GitHub: <https://github.com/peretta>
 - LinkedIn: <https://www.linkedin.com/in/vinicius-assis-peretta-5a2436227/>
-

Principais Conhecimentos Técnicos

- Git e versionamento de código
 - Controle de versão em ambientes colaborativos
 - Gerenciamento de branches, pull requests e resolução de conflitos
 - Trabalho utilizando GitFlow em equipes de desenvolvimento
- Java / Kotlin
 - Automação de testes mobile End-to-End com Espresso
 - Criação de testes unitários e de interface com Robolectric
 - API simples com Spring
- Python
 - Desenvolvimento web com Django e Flask, e integrações com IA
 - Pandas
- React
 - Desenvolvimento de interfaces web baseadas em componentes
 - Construção de interfaces interativas para sistemas e dashboards
 - Integração com APIs e serviços backend
- React Native
 - Desenvolvimento de aplicações mobile multiplataforma
- TypeScript / Node.js
 - Desenvolvimento de aplicações web e APIs

- Implementação de automação de testes E2E com Cypress
- SQL
 - Modelagem e manipulação de banco de dados
 - Escrita de queries para análise e validação de dados
 - Manipulação de bases SQLite utilizadas em cenários de testes automatizados

Índice de Projetos

Projeto	Semestre	Empresa Parceira	Link
Scrum Tutor	2023-1 (1º Sem.)	FATEC (Prof. Egideo)	Ir para a
DataEase	2024-1 (2º Sem.)	FATEC (Prof. Bertotti)	seção Ir para
OAK-RH	2024-2 (3º Sem.)	Youtan (Prof. Massanori)	a seção Ir para a
Lumen	2025-1 (4º Sem.)	FAPG (Prof. Juliana)	seção Ir para
GeoMaps	2025-2 (5º Sem.)	Visiona (Prof. Gérson)	a seção Ir
RulesAI	2026-1 (6º Sem.)	Dom Rock (Prof. Walmir)	para a seção Ir para a seção

Projeto – *Scrum Tutor*

Semestre: Primeiro Semestre – 2023-1

Empresa Parceira: Fatec – Professor Responsável: *Egideo*

Problema

As equipes da empresa possuem dificuldades no processo de organização do trabalho, principalmente na definição de prioridades, prazos e responsabilidades. A falta de conhecimento sobre papéis e funções dentro de uma metodologia ágil gera retrabalho, comunicação ineficiente e baixa previsibilidade das entregas. Sem um método estruturado, os projetos acabam atrasando e o fluxo de desenvolvimento se torna pouco transparente para gestores e colaboradores.

✓ Solução Desenvolvida

A solução desenvolvida consiste em um sistema web projetado para auxiliar equipes no entendimento e na aplicação correta da metodologia Scrum, oferecendo uma experiência clara, didática e prática. O website reúne em páginas distintas todos os elementos essenciais do framework — eventos, papéis, artefatos, apêndices e bibliografia — permitindo que qualquer equipe compreenda rapidamente suas responsabilidades, fluxos e prioridades. Além disso, a plataforma disponibiliza uma área completa de avaliação de equipes, baseada no método PACER, onde líderes podem avaliar proatividade, autonomia, colaboração e entregas, recebendo os resultados diretamente por e-mail

🔗 Link do Repositório Git

- GitHub: [Acessar](#)

🧰 Tecnologias Utilizadas

Tecnologia	Funcionalidade
HTML	Define a estrutura e o conteúdo das páginas web.
CSS	Responsável pela estilização e layout das páginas.
Flask	Framework web em Python utilizado para criar e gerenciar o servidor da aplicação.
Python	Linguagem de programação utilizada no desenvolvimento do backend com Flask .
AWS	Plataforma de nuvem utilizada para hospedagem e deploy da aplicação web.
Figma	Ferramenta de prototipação para o design da interface e fluxo de navegação do site.
Git	Sistema de controle de versão utilizado para gerenciar e versionar o código-fonte.

👤 Contribuições Pessoais

Atuei como Product Owner (PO) e também contribuí no desenvolvimento da solução.

- Fui responsável por entender e refinar os requisitos com o cliente (professor e orientador), em especial a dúvida sobre a avaliação solicitada: se seria apenas baseada no modelo PACER ou se incluiria também avaliações ao final de cada texto explicativo.
- Conduzi o alinhamento com o cliente, definindo que seriam implementadas as duas abordagens:
 - a avaliação PACER para medir o desempenho da equipe;
 - e perguntas ao final dos artigos para reforçar o entendimento do conteúdo teórico.
- Participei da definição das páginas e estrutura de navegação (menu, página de boas-vindas, páginas de bibliografia, eventos, papéis, artefatos etc.), garantindo coerência entre o conteúdo e a jornada do usuário.
- Atuei na implementação e ajustes de front-end (HTML5 + CSS3 + Bootstrap), incluindo melhorias de design, centralização de cards, atualização de layout e remoção/reorganização de páginas desnecessárias.
- Contribuí diretamente na implementação do formulário PACER, bem como no seu funcionamento dentro da aplicação (envio dos dados para o back-end e processamento).
- Implementei, em conjunto com o time, a funcionalidade de envio de e-mail usando Flask-Mail, configurando o servidor SMTP e a lógica de envio da avaliação para o e-mail da equipe.
- Fui responsável por grande parte da organização do repositório no GitHub:
 - atualização contínua do README.md (MVP, backlog, tecnologias, links, vídeo de apresentação, LinkedIn, prioridades);
 - criação e manutenção do backlog do produto e documentação das Sprints;
 - limpeza de arquivos obsoletos e organização de pastas.

Hard skills:

Tecnologia	O que consigo fazer
Python (Flask)	Criar rotas, processar dados de formulários, montar classes de avaliação e configurar envio de e-mail automático.
HTML5/CSS3	Montar páginas estáticas, estruturar menus e aplicar estilos para melhorar legibilidade e layout.

Bootstrap	Acelerar o front-end com componentes prontos (cards, grids, formulários) e padronizar a interface.
Figma	Desenhar protótipos de telas e validar fluxos de navegação com o time antes de codar.
Git/GitHub	Manter histórico de commits, organizar branches, documentar MVP/backlog e estruturar o repositório.

Soft Skills:

Cenário — refinando o produto com o cliente

No início do projeto, o professor Egídio pediu uma avaliação de equipes, mas ficou em aberto se seria só pelo modelo PACER ou se também haveria perguntas ao final de cada texto explicativo. Como Product Owner, marquei uma conversa para entender a expectativa real: o cliente queria medir desempenho da equipe e reforçar o aprendizado teórico. Negocie e propus implementar as duas abordagens, o que evitou retrabalho depois. Com o escopo alinhado, organizei o backlog e as Sprints, priorizando páginas essenciais (eventos, papéis, artefatos) e a funcionalidade PACER com envio por e-mail. Quando surgiram dúvidas de design no time, usei a visão de produto para manter coerência entre conteúdo didático e jornada do usuário. Também cuidei da organização do repositório — README, documentação de sprints e limpeza de arquivos — para que qualquer membro soubesse o que estava pronto e o que vinha a seguir.

Alguns dos commits no projeto:

- Base do HTML e início do Flask
 - *"Base do HTML (temporário)"*
 - *"SPRINT 2 - Início do Flask"*
- Backlog, MVP e organização do README
 - *"ADD BACKLOG README.md"*
 - *"Criacao MVP, VP e OBJ README.md"*
 - *"Sprints, Time e Tecnologias README.md"*
- Design e conteúdo
 - *"Boas-Vindas"*
 - *"Bibliografia UPDATE"*

- “Correção do Design”
 - “Centralizando cards”
 - Funcionalidade PACER e e-mail
 - “Implementação do funcionamento do form da PACER”
 - “Implementação do EMAIL”
 - “Conteúdo do email”
 - “Correção do bug”
 - Organização do repositório
 - “ORGANIZAÇÃO DAS PASTAS DO GIT”
 - Diversos “Update README.md” , incluindo backlog, prioridades e links.
-

► Detalhes Técnicos (código – envio de e-mail PACER)

[\[Voltar\]](#)

✓ Projeto – *DataEase*

Semestre: Segundo Semestre – 2024-1

Empresa Parceira: Fatec – Professor Responsável: *Bertotti*

Problema

Os usuários da empresa enfrentam dificuldades para acessar e interpretar informações armazenadas no banco de dados, pois grande parte deles não possui conhecimento adequado em SQL. Como resultado, dependem constantemente do setor técnico para extrair relatórios e consultas simples, tornando o processo lento e burocrático. Essa limitação prejudica a tomada de decisões e impede que usuários tenham autonomia para explorar os dados que já estão disponíveis.

✓ Solução Desenvolvida

A solução criada consiste em um aplicativo Desktop chamado DataEase, projetado para permitir que usuários sem conhecimento técnico em SQL consultem bancos de dados de

forma simples e intuitiva. O sistema oferece uma interface simples, composta por um campo de entrada para linguagem natural, um botão de envio e uma área destinada à visualização dos resultados retornados. Sua principal funcionalidade é a tradução automática de linguagem natural em comandos SQL, executando as consultas diretamente no banco escolhido pelo usuário retornando os dados desejados, ou seja você pergunta e ele responde - o sql é feito por baixo dos panos. Além disso, o DataEase possibilita o cadastro de usuários, a configuração de múltiplos bancos de dados e a troca do modelo de linguagem utilizado. Com isso, elimina a dependência do setor técnico, acelera o acesso à informação e garante autonomia aos usuários na exploração dos dados corporativos.

 Link do Repositório Git

- GitHub: [Acessar](#)

 Tecnologias Utilizadas

Tecnologias Utilizadas

Tecnologia	Funcionalidade
Java	Linguagem responsável pelo desenvolvimento de todo o sistema, desde a interface a conexão com o modelo de linguagem.
MySQL	Armazenamento de informações do usuário
LangChain4j	Biblioteca do java utilizada para conexão dos modelos.
LM Studio	Ferramenta utilizada para execução dos modelos.
Git	Sistema de controle de versão utilizado para gerenciar e versionar o código-fonte.

 Contribuições Pessoais

Atuei como Desenvolvedor (Dev Team) no projeto. Minhas principais contribuições foram:

- Implementação do módulo de comunicação com o modelo de linguagem, utilizando *LangChain4j* e *LM Studio*, incluindo a criação da função para estabelecer conexão com o servidor LLM (ligar servidor LM).

- Desenvolvimento de funcionalidades-chave do CRUD de usuários, incluindo otimizações no método `cadastrarUser`, implementação do método `getNameDB` e criação do `getIdUsuario`.
- Criação e gerenciamento do banco de dados MySQL, com tabelas para usuários e conexões, além de consultas e selects de validação.
- Desenvolvimento das interações entre a interface e o back-end, incluindo atualizações na tela de chat para a Sprint 3, ajustes visuais e funcionamento da área de entrada de comandos.
- Implementação da conexão em tempo real com o banco, permitindo que o texto gerado pelo modelo fosse executado diretamente no banco pelo usuário.
- Comentários, limpeza e organização do código durante as Sprints, garantindo clareza e documentação interna adequada.
- Participação no versionamento Git, com commits contínuos envolvendo criação de funcionalidades, correções e refinamento do fluxo interno da aplicação.

Hard skills:

Tecnologia	O que consigo fazer
Java	Implementar métodos de cadastro (<code>cadastrarUser</code>), buscar IDs/nomes de banco (<code>getIdUsuario</code> , <code>getNameDB</code>) e integrar a tela de chat ao back-end.
MySQL	Criar schema, tabelas de usuários/conexões e consultas de validação usadas pelo CRUD.
LangChain4j	Montar a ponte entre a aplicação Java e o modelo de linguagem para gerar SQL a partir de texto.
LM Studio	Configurar e validar a função de conexão com o servidor LLM local (ligar servidor LM).
Hugging Face	Explorar e testar modelos de linguagem compatíveis com o fluxo de tradução NL → SQL.

Soft Skills:

Cenário — entregando o coração do DataEase

Na Sprint 3, o time precisava que o usuário digitasse uma pergunta em linguagem natural e recebesse o resultado da consulta na hora — mas a integração com o modelo ainda não estava estável. Assumi a tarefa de conectar o LM Studio via LangChain4j, testando localmente até a função `ligar servidor LM` funcionar de ponta a ponta.

Enquanto isso, participei das dailies e alinhamentos de backlog, avisando cedo quando a conexão com o banco ou o CRUD de usuários bloqueava outras entregas. Com proatividade, fui além do mínimo: otimizei o `cadastrarUser` em duas etapas, comentei o código para facilitar revisão e modeleei o MySQL para suportar múltiplos bancos por usuário. Quando a tela de chat precisou de ajustes visuais e funcionais, trabalhei de forma autônoma na integração front-back, garantindo que o texto gerado pelo modelo fosse executado diretamente no banco escolhido — cumprindo a promessa do produto: *perguntar e receber a resposta, sem saber SQL*.

Alguns dos commits no desenvolvimento:

- *Função ligar servidor LM funciona*
-Finalização das modificações da tela Chat para Sprint 3
- *Método getNameDB feito*
- *Código comentado e adição do método getIdUsuario*
- *Otimização do método cadastrarUser (1/2)*
- *Otimização do método cadastrarUser (2/2)*
- *Atualização telaChat*
- *Criação de banco de dados e select*
- *Conexão realizada, dados sendo enviados*

► Detalhes Técnicos (exemplo de código utilizado no projeto)

[\[Voltar\]](#)

Projeto – OAK-RH

Semestre: Terceiro Semestre – 2024-2

Empresa Parceira: Youtan - Empresa de software | Professor Responsável: ~~Massanori~~

Problema

O departamento de Recursos Humanos da Youtan relatou dificuldades para coletar e organizar os feedbacks referentes à cultura da empresa e aos projetos internos. Além disso, o controle sobre a alocação de equipes acontece de maneira manual, o que aumenta a chance de erros e falhas de comunicação. Esse cenário gera perda de informações relevantes, dificulta a análise de clima organizacional e compromete a distribuição correta de profissionais entre os projetos.

Solução Desenvolvida

A solução proposta consiste em um sistema web moderno e centralizado para gestão de feedbacks, pesquisas de clima e organização de equipes, desenvolvido especialmente para atender às necessidades do RH da Youtan. A plataforma “OAKRH” reúne em um único ambiente o cadastro de usuários com diferentes níveis de acesso, a criação de formulários personalizados e o gerenciamento de equipes. O sistema permite coletar, armazenar e visualizar respostas de avaliações técnicas, comportamentais e de satisfação, oferecendo dashboards completos para análise de desempenho individual e coletivo. Além disso, possibilita comparar avaliações ao longo do tempo, aplicar filtros estratégicos e exportar relatórios em PDF. Dessa forma, a solução elimina processos manuais, reduz erros de comunicação e oferece uma visão estruturada que fortalece a cultura organizacional e melhora a tomada de decisão do RH.

Link do Repositório Git

- GitHub: [Acessar](#)

Tecnologias Utilizadas

Tecnologia	Utilização
Next.js	Framework React com SSR e otimizações
React	Biblioteca para construção da UI
TypeScript	Tipagem estática para maior segurança

Tailwind CSS	Framework de CSS utilitário para styling
Axios	Cliente HTTP para requisições à API
next-themes	Gerenciamento de temas (dark/light mode)
Node. js	Runtime JavaScript no servidor
TypeScript	Tipagem estática para maior segurança
Express.js	Framework HTTP para construção da API
PostgreSQL	Banco de dados relacional principal
Firebase	Autenticação e armazenamento em nuvem
Multer	Upload e processamento de arquivos
Axios	Cliente HTTP para requisições
CORS	Controle de acesso entre domínios
dotenv	Gerenciamento de variáveis de ambiente

Contribuições Pessoais

Atuei como Desenvolvedor (Dev Team), contribuindo diretamente tanto no back-end quanto no front-end. Minhas principais entregas incluem:

- Criação do modelo, controller, service e rotas de Equipes, permitindo cadastrar, listar, editar e excluir equipes dentro da plataforma.
- Implementação de componentes do front-end, como listar usuários pertencentes a uma equipe, pop-ups de edição, pop-up de membros e telas do portal do usuário.
- Estruturação completa de logout e Protected Routes, garantindo segurança de acesso e bloqueio de páginas para usuários não autenticados.
- Implementação da integração dos formulários no front-end, conectando as rotas às interfaces e garantindo fluxo entre categorias, formulários e respostas.
- Configuração do back-end para criação, edição e gerenciamento de categorias, incluindo ajustes em CategoryModel, CategoryController e CategoryService.

- Desenvolvimento dos dashboards de gráficos de usuários, equipes e formulários, permitindo visualização clara e analítica para o RH.
- Refatoração da estrutura de pastas do projeto, correções de merge, resolução de conflitos e ajustes gerais no código.
- Implementação inicial do servidor React, configuração do .gitignore, remoção do node_modules do repositório e padronização da base do projeto.
- Participação ativa na construção das telas iniciais, fluxo do portal do usuário e no desenvolvimento contínuo da interface.

Hard skills:

Tecnologia	O que consigo fazer
TypeScript	Escrever controllers tipados, services reutilizáveis e componentes React com contratos claros de dados.
Next.js / React	Montar dashboards com gráficos, portal do usuário, pop-ups de membros e fluxo completo de formulários.
Node.js (Express)	Criar CRUD de equipes e categorias (model → controller → service → rotas) e expor endpoints REST.
PostgreSQL	Persistir relações entre equipes, usuários e categorias; consultar dados para gráficos e relatórios.
Firebase (Auth)	Implementar login/logout, Protected Routes e bloqueio de páginas para usuários não autenticados.
Tailwind CSS	Construir interfaces responsivas para telas administrativas e do portal do colaborador.

Soft Skills:

Cenário — construindo o módulo de equipes do zero

O RH da Youtan precisava alocar colaboradores em projetos e a plataforma ainda não tinha gestão de equipes. Na planning, peguei o épico de Equipes e quebrei em entregas: model, controller, service, rotas e, depois, a interface com listagem de membros e pop-ups de edição.

Durante a sprint, mantive colaboração constante com quem cuidava de formulários e dashboards — avisei cedo quando mudanças no back-end impactavam o front. Com proatividade, resolvi conflitos de merge, padronizei a estrutura de pastas e entreguei Protected Routes para que apenas usuários logados acessassem o portal. Quando o PO pediu visibilidade analítica, implementei os gráficos de usuários, equipes e formulários, traduzindo a necessidade de negócio em telas que o RH conseguia usar no dia a dia sem depender de planilhas.

Alguns commits:

- Gráficos de usuários e equipes implementados
- Gráficos dos formulários concluídos
- Telas iniciais do portal do usuário desenvolvidas
- Integração da tela de formulários com o front-end
- Atualização dos models e controller de categorias
- Rotas, controllers e services de categorias criados
- Criar e excluir equipes implementado
- Pop-up de membros e lista de usuários da equipe
- Logout e Protected Route implementados
- EquipeModel, EquipeController e rotas principais entregues
- Classe líder criada
- Servidor React criado e estrutura inicial montada
- Projeto inicial criado e configurado

► Detalhes Técnicos

[\[Voltar\]](#)

Projeto – *Lumen*

Semestre: Quarto Semestre – 2025-1

Empresa Parceira: FAPG - FUNDAÇÃO DE APOIO À PESQUISA DE PÓS-GRADUANDOS |

Professor Responsável: *Juliana*

Problema

A FAPG não possui uma forma estruturada de acompanhar o andamento dos projetos em execução, o que dificulta a visualização de prazos, tarefas e responsáveis. A ausência de um acompanhamento centralizado faz com que relatórios sejam feitos manualmente e muitas vezes de forma tardia. Essa falta de transparência afeta a tomada de decisão, impede a análise de desempenho e compromete a eficiência da gestão dos projetos desenvolvidos pela instituição.

Solução Desenvolvida

A solução proposta consiste em uma plataforma web centralizada para gestão completa dos projetos da FAPG, oferecendo uma visão integrada de prazos, tarefas, responsáveis e áreas de atuação. O sistema permite login seguro, garantindo proteção aos dados sensíveis da instituição. Entre as principais funcionalidades, destacam-se o cadastro, edição e listagem de projetos, além do registro de etapas, tarefas e subtarefas, sua atribuição a membros específicos e acompanhamento de status. A plataforma também possibilita a organização por áreas de atuação e a recuperação de projetos excluídos. Essa abordagem elimina planilhas manuais, aumenta a transparência, acelera a tomada de decisão e fortalece o controle sobre o andamento das iniciativas da FAPG.

Link do Repositório Git

- GitHub: [Acessar](#)

Repositório

Tecnologia	Funcionalidade
React	Biblioteca JavaScript utilizada para o desenvolvimento do frontend da aplicação, permitindo a criação de interfaces dinâmicas e componentizadas.
Node. js	Ambiente de execução JavaScript multiplataforma, utilizado para rodar o código no servidor.
JavaScript	Linguagem de programação utilizada no desenvolvimento do projeto, responsável pela lógica de interação e integração entre os componentes.

TypeScript	Superset do JavaScript que adiciona tipagem estática e recursos de orientação a objetos, aumentando a robustez e a manutenção do código.
PostgreSQL	Banco de dados relacional, utilizado para armazenamento de dados do sistema
Firebase	Serviço em nuvem com banco de dados não relacional (NoSQL), utilizado para o armazenamento e autenticação de usuários (e-mails e senhas).
Ollama	Ferramenta utilizada para executar e gerenciar modelos
Git	Sistema de controle de versão distribuído, utilizado para o gerenciamento do código-fonte e colaboração em equipe.

 Contribuições Pessoais

Atuei como Desenvolvedor no time. Trabalhei diretamente na implementação de funcionalidades essenciais relacionadas ao ciclo de vida dos projetos, especialmente nos módulos de etapas, tarefas e projetos excluídos. Desenvolvi o CRUD completo de Etapas, incluindo a relação entre etapas e usuários, garantindo que a atribuição e acompanhamento ocorressem de forma consistente no banco de dados. Implementei a integração da tela “Minhas Tarefas”, permitindo que cada membro visualizasse suas atividades pendentes de maneira centralizada.

Também implementei o Soft Delete, a funcionalidade de restauração de projetos, e o processo automático de remoção definitiva após 30 dias, garantindo conformidade, rastreabilidade e organização. Modelei e ajustei migrations ao longo do projeto, além de estruturar rotas do back-end e integrações necessárias com o React. Contribuí ainda para melhorias de interface e experiência do usuário, incluindo pop-ups e ajustes de layout.

Apliquei uma IA capaz de chamar funções do backend, como por exemplo "concluir tarefas"

Hard skills:

Tecnologia	O que consigo fazer
React	Integrar a tela "Minhas Tarefas", exibir atividades por usuário e melhorar UX com pop-ups e ajustes de layout.

Node. js	Implementar CRUD de etapas, rotas de soft delete/restauração e endpoints chamados pelo assistente de IA.
TypeScript	Escrever services tipados (ex.: OllamaService) e manter contratos claros entre front, back e IA.
PostgreSQL	Criar e evoluir migrations, modelar vínculos etapa-usuário-tarefa e aplicar regra de exclusão após 30 dias.
Ollama	Montar prompts estruturados em JSON para ações como <code>complete_task</code> , <code>complete_etapa</code> e <code>list_overdue</code> .
Migrations	Ajustar schema ao longo do projeto sem perder dados, incluindo campos de exclusão lógica e restauração.

Soft Skills:

Cenário — quando a FAPG apagou um projeto por engano

A cliente pediu que projetos excluídos pudessem ser recuperados — excluir definitivamente só depois de 30 dias. Implementei o soft delete com restauração e a regra automática de remoção permanente, explicando na review como isso protegia contra erros humanos sem perder rastreabilidade.

Na mesma sprint, outro membro travou na tela "Minhas Tarefas". Assumi a integração front-back para que cada pesquisador visse só o que lhe cabia. Depois, propus usar Ollama para comandos do tipo "finalize a tarefa 12" — em vez de navegar menus, o usuário fala e o sistema executa. Isso exigiu colaboração com quem mantinha as rotas de tarefas e visão de produto para não complicar o fluxo de quem prefere a interface tradicional. O resultado foi uma plataforma mais transparente para a gestão de projetos da FAPG, com menos planilhas e mais controle sobre prazos e responsáveis.

Alguns commits

- CRUD Etapa e relação com usuário
- Atualização de migrations
- Soft Delete de projetos e etapas
- Integração da tela "Minhas Tarefas"
- Exclusão de etapa/tarefa com vínculos
- Regra de exclusão definitiva após 30 dias

- Restauração de projetos e ajuste de rotas
 - IA para concluir tarefas e melhorar ações de etapa
- Detalhes (opcional – códigos, diagramas, prints)

[\[Voltar\]](#)

✓ Projeto – *GeoMaps*

Semestre: Quinto Semestre – 2025-2

Empresa Parceira: Visiona Tecnologia Espacial S.A | Professor Responsável: *Gérson*

Problema

Grande parte das propriedades rurais no Brasil não possui um endereço formal ou CEP definido, o que dificulta processos logísticos, entrega de serviços e registro de localizações exatas. Embora o Cadastro Ambiental Rural tenha avançado na regularização ambiental, ele não resolve a ausência de identificação territorial clara. Essa falta de endereçamento impacta o acesso a serviços públicos, limita ações governamentais e prejudica atividades comerciais e agrícolas que dependem da localização precisa de propriedades.

✓ Solução Desenvolvida

Um aplicativo móvel (Android/React Native) para endereçamento digital de imóveis do CAR, roteirização rural, registro colaborativo de condições de vias e exibição de alertas, visualização das propriedades, multinível de usuários e painel do administrador.

Link do Repositório Git

- GitHub: [Acessar](#)
-

Tecnologias Utilizadas

Tabela de Tecnologias e Sua Aplicação

Tecnologia	Onde foi usada
Python	Linguagem de programação utilizada no backend
FastAPI	Framework web moderno para construção de APIs REST em Python
MongoDB	Banco de dados NoSQL orientado a documentos com suporte a geoespacial
PyMongo	Driver oficial MongoDB para Python com suporte assíncrono
Pydantic	Validação e serialização de dados com tipagem
Google Maps API	Integração com serviços de mapas e geolocalização
Plus Codes	Sistema de endereçamento digital para localidades
JWT	Autenticação e autorização com tokens
Bcrypt	Hash seguro de senhas
React	Biblioteca JavaScript para construção de interfaces
React Native	Framework para desenvolvimento de aplicações mobile nativas
Expo	Plataforma de desenvolvimento para React Native (iOS/Android/Web)
TypeScript	Linguagem com tipagem estática baseada em JavaScript
React Router	Roteamento para aplicações web em React
Vite	Build tool moderno e rápido para aplicações web
Tailwind CSS	Framework CSS utilitário para estilização
React Native Maps	Componente de mapas interativos para mobile
Axios	Cliente HTTP para requisições de API
ESLint	Linter para padronização e qualidade de código

Contribuições Pessoais

Atuei como Scrum Master, e no desenvolvimento técnico (backend e mobile). Minhas contribuições podem ser divididas em papel de processo e entregas técnicas.

Como Scrum Master

- Condução de cerimônias Scrum (daily, planning, review e retrospectiva), mantendo foco em impedimentos e acompanhamento de Sprint Goals.
- Apoio ao Product Owner na organização e priorização do backlog, decompondo épicos como: endereçamento digital (Plus Code), propriedades no mapa, ocorrências, rotas, alertas e modo offline.
- Facilitação da comunicação entre squads de backend e frontend, reduzindo retrabalho em integrações.
- Monitoramento da velocidade do time, ajuste de capacidade e acompanhamento de entregas entre Sprints.
- Gestão de Pull Requests e merges principais, garantindo integração contínua das branches de feature com develop (protegi a branch master e develop, para que alterações fossem feitas nela era necessário ser revisadas e aceitas).

Como Desenvolvedor (Backend e Mobile)

- Autenticação e rotas de usuário
 - Implementação e/ou integração das rotas de registro e autenticação (UserRegisterandAuthRoutes), com JWT.
 - Ajustes de fluxo de login e telas estilizadas de autenticação no app.
- Funcionalidades administrativas e certificados
 - Desenvolvimento e merge da *feature/admin-functions* no backend.
 - Implementação da geração de certificados em PDF (*feature/certificado_pdf*), incluindo merges e correções.
- Integração com Weather API e filtros de Plus Code
 - Implementação da *feature/weather-api* para obter dados meteorológicos e integrar aos cenários de alerta/ocorrência.
 - Desenvolvimento da *feature/Filter_Pluscode_Router* para filtrar e roteirizar propriedades/endereços por Plus Code.
- Correções de ocorrências e estabilidade da API

- Ajustes na *feature/fix-ocorrencia*, corrigindo problemas de registro/consulta de ocorrências.
- Desenvolvimento do app GeoMaps (frontend mobile)
 - Criação e ajustes nas telas de login, cadastro e perfil (*feature/register-screens*, *feature/profile-tab*, ajustes de login e atualização de informações).
 - Implementação da lógica para gerar Plus Code no mapa e por coordenadas, bem como correções de add Plus Code.
 - Funcionalidade para listar propriedades do usuário e exibi-las no mapa, incluindo cores e estados:
 - *propriedade aparece no mapa*
 - *Correções de add pluscode*
 - Desenvolvimento de alertas e rotas:
 - *feature/integrando_botao_alertas* (botão de alertas);
 - *feature/Trancing_Routes* (traçado de rotas).
 - Implementação e correções de Plus Code no mapa e histórico (*plus code no mapa*, *removendo plus code do mapa*, *Atualizando plus code e vendo histórico*).
 - Entrega da funcionalidade de mapa offline (*feature/offline-map*) e correções de UI do mapa (*fix/map-ui*).
- Integração contínua e qualidade do código
 - Merge frequente de branches de *feature* para *develop* nos dois repositórios (backend e frontend), garantindo integração estável:
 - PRs: #1, #2, #5, #6, #7, #8, #9, #10, #11, #12, #13, #16, #19, #21, #22, #23, #24, #25, #27, #28 etc.
 - Resolução de conflitos, ajuste de estrutura de pastas e refinamento de componentes conforme o projeto evoluía.

Hard skills:

Tecnologia	O que consigo fazer
------------	---------------------

React Native (Mobile)	Criar telas de login/cadastro/perfil, botão de alertas, traçado de rotas e mapa offline.
Python (FastAPI)	Desenvolver endpoints de auth, admin, certificados PDF, ocorrências e filtros por Plus Code.
MongoDB (Geoespacial)	Armazenar e consultar propriedades com coordenadas; integrar dados de mapa e histórico de Plus Code.
Google Maps API	Exibir propriedades no mapa, posicionar marcadores customizados e apoiar roteirização rural.
Plus Codes	Gerar, adicionar, remover e consultar histórico de códigos por coordenadas ou seleção no mapa.
JWT	Proteger rotas de usuário e administrador com autenticação baseada em token.
Mapas Offline	Entregar funcionalidade de mapa utilizável sem conexão contínua, com correções de UI.

Soft Skills:

Cenário — duas squads, um mapa, muitos PRs

Como Scrum Master do GeoMaps, o time estava dividido entre backend (FastAPI/MongoDB) e mobile (React Native). Na daily de uma sprint crítica, o mobile esperava o endpoint de Plus Code e o back-end ainda ajustava o formato das coordenadas (`cordinates` vs `coordinates`). Facilitei uma conversa rápida entre as squads, alinhamos o contrato da API e evitamos dois dias de retrabalho.

Paralelamente, protegi as branches `master` e `develop` exigindo revisão de PRs antes do merge — o que gerou atrito inicial, mas reduziu bugs em produção. Quando surgiram conflitos entre `feature/offline-map` e `feature/Trancing_Routes`, medi a gestão de conflitos: priorizamos o que desbloqueava a demo com a Visiona. Também atuei como dev, implementando auth JWT, alertas no mapa e listagem de propriedades do usuário. Essa combinação de liderança, facilitação e entrega técnica manteve o fluxo da sprint previsível mesmo com integrações complexas entre mapa, clima e roteirização.

Alguns Commits

- *Merge pull request #11 – develop*

- Merge pull request #10 – feature/admin-functions
- Merge pull request #9 – develop
- Merge pull request #8 – feature/weather-api
- Merge pull request #7 – feature/Filter_Pluscode_Router
- Merge pull request #6 – feature/fix-ocorrencia
- Merge pull request #5 – feature/certificado_pdf
- Merge pull request #2 – feature/certificado_pdf
- Merge pull request #1 – feature/UserRegisterandAuthRoutes
- Merge pull request #28 – develop
- Merge pull request #27 – fix/map-ui
- Merge pull request #25 – feature/offline-map
- Merge pull request #24 – develop
- Merge pull request #23 – feature/profile-tab
- plus code no mapa / removendo plus code do mapa / Atualizando plus code e vendo histórico
- Merge pull request #22 – feature/Trancing_Routes
- Merge pull request #21 – feature/integrando_botao_alertas
- Correções de add pluscode / gerar plus code no mapa e por coordenadas
- propriedade aparece no mapa
- Merge pull request #19, #16, #13, #12, #11, #9, #7, #2, #1 – feature/register-screens, feature/tela_login_estilizada, feature/pluscode-screen, feature/list_user_properties

Esses commits mostram tanto minha coordenação e integração de PRs quanto minha participação direta no desenvolvimento técnico do GeoMaps.

► Details

[\[Voltar\]](#)

Projeto – Gerenciamento de Regras de Negócio

Semestre: Sexto Semestre – 2025-2

Empresa Parceira: Dom Rock | Professor Responsável: *Professor Walmir*

Problema

Empresas enfrentam desafios significativos no gerenciamento de regras de negócio, especialmente quando estas são dinâmicas e sofrem alterações frequentes. As regras são constantemente atualizadas devido ao lançamento de novos produtos, descontinuidade de produtos, mudanças em precificação, acordos comerciais, políticas de vendas e parcerias.

Decorrente dessa atividade, a base de conhecimento de regras geralmente não está registrada e organizada de forma adequada, ocasionando inconsistência operacional, conflito entre regras aplicadas por múltiplos colaboradores, conhecimento tácito e perda de rastreabilidade de processos.

Especificamente, no contexto de cálculo de comissionamento mensal de funcionários, as dificuldades incluem:

- Aplicação incorreta de percentuais de comissionamento conforme cargo e legislação trabalhista
- Gerenciamento de exceções e particularidades por cargo
- Falta de transparência nas decisões de cálculo
- Dificuldade em manter regras consistentes em diferentes cenários de venda

Solução Desenvolvida

Um sistema integrado de gerenciamento de regras de negócio e cálculo de comissionamento composto por três camadas:

1. Backend (Spring Boot): API REST para cálculo e auditoria de comissões, gerenciamento de regras de negócio, armazenamento em MongoDB e importação de dados via Excel.
 2. Frontend Web (Vue.js 3): Aplicação SPA (Single Page Application) para criação, visualização e gerenciamento de regras de negócio com interface intuitiva e autenticação.
 3. Agente IA (LLMs): Sistema inteligente baseado em modelos de linguagem generativos (LLM) para:
 - Interpretação de regras de negócio em linguagem natural
 - Geração automática de código com base em requisitos
 - Indexação e recuperação de regras via RAG (Retrieval-Augmented Generation)
 - Processamento de PDFs e código-fonte para construção de base de conhecimento
-

 Links dos Repositórios Git

- Backend (Spring Boot): [Acessar](#)
- Frontend (Vue.js): [Acessar](#)
- ML Agent (Python/LLM): [Acessar](#)

 Tecnologias Utilizadas

 Tabela Consolidada de Tecnologias e Sua Aplicação

Tecnologia	Onde foi usada
Java 17	Linguagem de programação para backend
Spring Boot	Framework web para construção de API REST com features avançadas
MongoDB	Banco de dados NoSQL para armazenamento de regras, comissões e dados de funcionários
Maven	Gerenciador de dependências e build
Python 3.10+	Linguagem para desenvolvimento do agente IA
LangChain	Framework para orquestração de agentes IA e gerenciamento de prompts
FastAPI	Framework web moderno para APIs Python
LanceDB	Banco de dados vetorial para RAG (Retrieval-Augmented Generation)
MLflow	Rastreamento de experimentos e modelos de ML
Ollama LangChain	Execução local de modelos LLM
Groq Google	Integração com modelos Groq
Vertex AI / Gemini	Integração com modelos Google

HuggingFace	Acesso a modelos de IA abertos
PyMUPDF	Processamento de PDFs para indexação de regras
Vue	Biblioteca JavaScript para construção de interfaces reativas
Vue Router	Roteamento para aplicação web
Pandas	Manipulação e análise de dados

Hard skills:

Tecnologia	O que consigo fazer
Spring Boot	Desenvolver motor de comissionamento com auditoria etapa a etapa, CRUD de usuários/lojas/marcas e APIs REST.
Python (LangChain/FastAPI)	Montar agente IA que interpreta regras, gera código e consulta base indexada via RAG.
HuggingFace	Acessar e testar modelos abertos compatíveis com o pipeline de geração e recuperação de regras.
Vue.js 3	Criar interfaces para listagem, criação e edição de regras de negócio com fluxo autenticado.
Pandas	Importar e validar dados de funcionários e vendas a partir de planilhas Excel.
MongoDB	Armazenar regras dinâmicas, resultados de comissão e trilhas de auditoria por funcionário.

Soft Skills:

Cenário — quando a regra de comissão mudou na véspera do fechamento

A Dom Rock precisava calcular comissionamento mensal com dezenas de exceções — férias, licença, bônus por faixa, override de marca — e ainda rastrear *por que* cada valor foi calculado. Na sprint do motor de comissionamento, implementei a função `calcular_comissionamento` com auditoria etapa a etapa, para que o RH pudesse explicar qualquer valor ao funcionário.

Quando o cliente pediu suporte a novas regras temporárias (overrides de percentual por mês), negocieei com o PO o escopo mínimo viável para não atrasar o fechamento: overrides via dicionário, sem refatorar todo o engine. Revisei PRs de colegas (feature/commission-engine, feature/rules-listing) aplicando gestão de qualidade — garantindo testes de unidade e interfaces consistentes entre service e implementação. Com proatividade, documentei o tutorial de importação de dados e corriji duplicidade de funcionários na base. O resultado foi um sistema em que regras deixam de ser conhecimento tácito e passam a ser rastreáveis, auditáveis e editáveis pela equipe de negócio.

Commits e Participações Relevantes

- Merge pull request #15 from Phoenix-Team-Fatec/fix/docs-dom-rock
- Merge pull request #14 from Phoenix-Team-Fatec/feature/mvc-lojas
- Funcionários sem repetição
- CRUD de usuário,
- Alteração no comando que exibe versões
- Interfaces e implementações em services
- Merge branch 'develop' into refactor/servico-interface-impl
- Merge pull request #4 from Phoenix-Team-Fatec/feature/rules-listing
- Tutorial importar dados
- Devcontainer alterado
- Merge pull request #3 from Phoenix-Team-Fatec/feature/commission-engine
- Testes de unidade
- marcas, lojas e funcionarios
- auditoria

► Details

[\[Voltar\]](#)

