

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

Samuel Alkmin Machado

PORTFÓLIO DE APRENDIZAGEM DE
PROJETOS INTEGRADO (API) E
Rules AI

São José dos Campos
2026

Samuel Alkmin Machado

**PORTFÓLIO DE APRENDIZAGEM DE
PROJETOS INTEGRADO (API) E
Rules AI**

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. Gerson da Penha Neto

São José dos Campos
2026

Samuel Alkmin Machado

**PORTFÓLIO DE APRENDIZAGEM DE
PROJETOS INTEGRADO (API) E
Rules AI**

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. Gerson da Penha Neto - FATEC

Prof. Dr. Emanuel Mineda Carneiro - FATEC

Prof. Me. Nilo Jeronimo Vieira - FATEC

____/____/____

DATA DA APROVAÇÃO

AGRADECIMENTOS

Agradeço primeiramente à FATEC São José dos Campos pela oportunidade de vivenciar uma formação que integra teoria e prática por meio da metodologia de Aprendizagem por Projetos Integrados, experiência que foi determinante para o meu desenvolvimento profissional e pessoal.

Ao professor orientador Gerson, expresso minha sincera gratidão pela orientação, pela disponibilidade e pelo compromisso em conduzir este trabalho com clareza e dedicação. Suas contribuições foram essenciais para a organização e a qualidade do documento apresentado.

Aos colegas de equipe Matheus, Vinicius e Gustavo, registro meu reconhecimento pela parceria construída ao longo dos semestres. O comprometimento, a colaboração e a troca de conhecimentos proporcionados pelo trabalho em conjunto foram fundamentais para o sucesso dos projetos desenvolvidos e para o aprendizado mútuo que cada entrega representou.

Por fim, agradeço a todos os professores e empresas parceiras que participaram dessa trajetória, proporcionando desafios reais que contribuíram de forma significativa para a minha formação como profissional de tecnologia.

RESUMO

O presente documento descreve a trajetória acadêmica e profissional desenvolvida ao longo de seis semestres na FATEC São José dos Campos, por meio da metodologia de Aprendizagem por Projetos Integrados (API), na qual o aluno participa de projetos reais em parceria com empresas do mercado. O primeiro projeto, denominado ScrumTutor, desenvolvido no segundo semestre de 2023, aborda a capacitação em metodologia ágil Scrum por meio de uma aplicação web educacional construída com HTML, CSS, Python e Flask, hospedada na AWS, que permite o estudo dos fundamentos do Scrum e a avaliação de integrantes de equipe pelo sistema PACER. O segundo projeto, DataEase, realizado no primeiro semestre de 2024, apresenta uma aplicação desktop integrada a modelos de linguagem que possibilita consultas a bases de dados corporativas por meio de linguagem natural, desenvolvida em Java com a biblioteca LangChain4j e o LM Studio para execução dos modelos. O terceiro projeto, OAK-RH, desenvolvido no segundo semestre de 2024 em parceria com a empresa Youtan, consiste em uma aplicação web com dashboards interativos para gestão do clima organizacional e acompanhamento de pesquisas de satisfação, construída com React, Node.js, TypeScript, PostgreSQL e Firebase. O quarto projeto, desenvolvido no primeiro semestre de 2025 em parceria com a FAPG, entrega uma aplicação web para gerenciamento de projetos com cadastro de etapas, tarefas e orçamento, incluindo funcionalidade de lixeira e um chatbot interativo viabilizado pelo Ollama. O quinto projeto, GeoMaps, desenvolvido no segundo semestre de 2025 em parceria com a Visiona, apresenta uma aplicação móvel para geração de endereços digitais em áreas rurais por meio de Plus Codes, construída com React Native, FastAPI, MongoDB e Docker. O sexto projeto, API_6, desenvolvido no primeiro semestre de 2026 em parceria com a DomRock, apresenta uma aplicação web para o gerenciamento dinâmico de regras de negócio com apoio de um agente de IA construído com LangGraph, integrado a múltiplos modelos de linguagem e observabilidade via MLflow. Ao longo dos projetos, o aluno exerceu diferentes papéis — desenvolvedor, Scrum Master e Product Owner —, o que lhe proporcionou uma formação abrangente tanto em competências técnicas quanto em habilidades interpessoais como comunicação, liderança, proatividade e pensamento analítico. A evolução tecnológica dos projetos evidencia uma progressão consistente, partindo de aplicações web simples com HTML e Flask até soluções complexas envolvendo agentes de IA com arquitetura em grafo, RAG e observabilidade. Os resultados obtidos demonstram a eficácia da metodologia API

como instrumento de formação profissional, ao articular teoria e prática em contextos reais de desenvolvimento de software.

Palavras-Chave: Aprendizagem por Projetos Integrados; metodologia ágil; desenvolvimento web; inteligência artificial; agentes de IA; modelos de linguagem.

ABSTRACT

The present document describes the academic and professional trajectory developed over six semesters at FATEC São José dos Campos, through the Integrative Project-Based Learning (API) methodology, in which the student participates in real projects in partnership with companies from the market. The first project, named ScrumTutor, developed in the second semester of 2023, addresses training in agile Scrum methodology through an educational web application built with HTML, CSS, Python, and Flask, hosted on AWS, which enables the study of Scrum fundamentals and the evaluation of team members through the PACER system. The second project, DataEase, carried out in the first semester of 2024, presents a desktop application integrated with language models that enables queries to corporate databases through natural language, developed in Java with the LangChain4j library and LM Studio for model execution. The third project, OAK-RH, developed in the second semester of 2024 in partnership with the company Youtan, consists of a web application with interactive dashboards for organizational climate management and satisfaction survey monitoring, built with React, Node.js, TypeScript, PostgreSQL, and Firebase. The fourth project, developed in the first semester of 2025 in partnership with FAPG, delivers a web application for project management with registration of stages, tasks, and budget, including a recycle bin feature and an interactive chatbot powered by Ollama. The fifth project, GeoMaps, developed in the second semester of 2025 in partnership with Visiona, presents a mobile application for generating digital addresses in rural areas through Plus Codes, built with React Native, FastAPI, MongoDB, and Docker. The sixth project, API_6, developed in the first semester of 2026 in partnership with DomRock, presents a web application for the dynamic management of business rules with the support of an AI agent built with LangGraph, integrated with multiple language models and observability via MLflow. Throughout the projects, the student performed different roles — developer, Scrum Master, and Product Owner —, which provided comprehensive training in both technical competencies and interpersonal skills such as communication, leadership, proactivity, and analytical thinking. The technological evolution across the projects demonstrates a consistent progression, starting from simple web applications with HTML and Flask to complex solutions involving AI agents with graph-based architecture, RAG, and observability. The results obtained demonstrate the effectiveness of the API methodology as

a professional training instrument, by articulating theory and practice in real software development contexts.

Keywords: Integrative Project-Based Learning; agile methodology; web development; artificial intelligence; AI agents; language models.

SUMÁRIO

1. INTRODUÇÃO	10
2. PROJETOS	12
5. CONSIDERAÇÕES FINAIS.....	20
REFERÊNCIAS BIBLIOGRÁFICAS	22
APÊNDICE A – PORTIFÓLIO.....	23

1. INTRODUÇÃO

A Aprendizagem por Projetos Integrados (API) constitui uma metodologia de ensino adotada pela FATEC São José dos Campos que articula teoria e prática por meio do desenvolvimento de projetos reais em parceria com empresas do mercado. Ao longo de cada semestre, os alunos são organizados em equipes e desafiados a resolver problemas concretos apresentados por empresas parceiras, aplicando práticas de metodologias ágeis, especialmente o framework Scrum, para o planejamento, desenvolvimento e entrega das soluções. Essa abordagem proporciona ao estudante uma experiência profissional progressiva, na qual diferentes papéis são exercidos — desenvolvedor, Scrum Master e Product Owner — e competências técnicas e interpessoais são desenvolvidas de forma integrada e contínua.

Os projetos desenvolvidos ao longo dos semestres abordaram problemáticas distintas, porém complementares, todas relacionadas a demandas reais de organizações que enfrentavam dificuldades em seus processos internos. O ScrumTutor, primeiro projeto realizado no segundo semestre de 2023, surgiu da necessidade de capacitar colaboradores na metodologia Scrum, cuja aplicação inadequada gerava ruídos de comunicação e atrasos nas entregas. O DataEase, desenvolvido no primeiro semestre de 2024, buscou solucionar as dificuldades de acesso e consulta a bases de dados corporativas, que comprometiam a confiabilidade das informações e a tomada de decisão. O OAK-RH, realizado no segundo semestre de 2024 em parceria com a empresa Youtan, endereçou a ausência de processos estruturados de feedback no setor de Recursos Humanos, que resultava em desmotivação e baixa colaboração entre equipes. O quarto projeto, desenvolvido no primeiro semestre de 2025 em parceria com a FAPG, respondeu à carência de um sistema padronizado de gerenciamento de projetos que provocava atrasos e dificuldades na alocação de recursos. O GeoMaps, desenvolvido no segundo semestre de 2025 em parceria com a Visiona, abordou a problemática do endereçamento em áreas rurais, cuja ausência compromete o acesso a serviços essenciais e a inclusão social dessas comunidades. Por fim, o API_6, desenvolvido no primeiro semestre de 2026 em parceria com a DomRock, enfrentou a problemática do gerenciamento dinâmico de regras de negócio em ambientes corporativos, cuja falta de registro adequado provoca inconsistências operacionais e comprometimento da rastreabilidade das decisões.

A relevância desses projetos reside não apenas na resolução de problemas reais, mas também na diversidade de soluções tecnológicas empregadas ao longo da formação. O percurso tecnológico evidencia uma evolução consistente, partindo de uma aplicação web construída

com HTML, CSS, Python e Flask hospedada na AWS, avançando para uma aplicação desktop em Java integrada a modelos de linguagem por meio do LangChain4j e do LM Studio, progredindo para aplicações web com React, Node.js, TypeScript, PostgreSQL e Firebase, passando por uma aplicação móvel desenvolvida com React Native, FastAPI, MongoDB e Docker, e culminando em uma solução com agentes de IA construídos em Python com LangGraph, integração a múltiplos modelos de linguagem (Qwen3-Coder, Gemini e Llama), banco vetorial para RAG e observabilidade via MLflow. Essa progressão demonstra a ampliação gradual do repertório técnico do aluno, abrangendo diferentes paradigmas de desenvolvimento, arquiteturas de software, bancos de dados relacionais e não relacionais, integração com modelos de linguagem, práticas de containerização e orquestração de agentes inteligentes.

Os resultados obtidos ao longo dos seis projetos confirmam a eficácia da metodologia API como instrumento de formação profissional. Cada projeto entregou uma solução funcional alinhada às necessidades da empresa parceira, ao mesmo tempo em que proporcionou o desenvolvimento de habilidades como comunicação, liderança, proatividade, organização, pensamento analítico e trabalho em equipe. A experiência acumulada evidencia que a integração entre ambiente acadêmico e demandas reais do mercado prepara o estudante de forma abrangente para os desafios do desenvolvimento de software contemporâneo, incluindo as fronteiras mais recentes da computação, como o desenvolvimento de agentes baseados em inteligência artificial generativa.

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
Título da API 1º semestre	https://github.com/Phoenix-Team-Fatec/ScrumTutor
Título da API 2º semestre	https://github.com/Phoenix-Team-Fatec/DataEase
Título da API 3º semestre	https://github.com/Phoenix-Team-Fatec/OAK-RH
Título da API 4º semestre	https://github.com/Phoenix-Team-Fatec/API-4
Título da API 5º semestre	https://github.com/Phoenix-Team-Fatec/geo-maps
Título da API 6º semestre	https://github.com/Phoenix-Team-Fatec/API_6

Fonte: Elaborado por Alkmin Machado, (2026).

2.1 2023 – 2º semestre: ScrumTutor

Desenvolvido no segundo semestre de 2023, o projeto **ScrumTutor** (Machado, 2023a) foi uma Aprendizagem por Projetos Integrados (API) realizada na FATEC São José dos Campos, sob orientação do Prof. Antônio Egydio São Tiago Graça. A proposta teve como tema a capacitação em metodologia ágil Scrum, que segundo Sutherland (2020) é definido como um método evolutivo, adaptável e que se corrige sozinho, planejado para acelerar e trazer mais segurança ao desenvolvimento de produtos. ,voltada a colaboradores de uma empresa parceira que enfrentavam dificuldades na compreensão e aplicação dos papéis, eventos e artefatos dessa metodologia. O objetivo do projeto foi resolver esse problema por meio de uma abordagem prática e interativa, contribuindo para a redução de ruídos de comunicação, atrasos nas entregas e baixa colaboração entre as equipes.

A solução desenvolvida consistiu em uma aplicação web de caráter educacional, que permitia ao usuário estudar os fundamentos do Scrum e avaliar os integrantes de sua própria equipe por meio do sistema PACER — um formulário estruturado com critérios de proatividade, autonomia, colaboração e entrega de resultados, pontuados de zero a quatro. O desenvolvimento seguiu práticas da metodologia ágil, incluindo a organização do backlog, a definição de requisitos junto ao cliente, a prototipação da interface no Figma e a implementação incremental das funcionalidades ao longo de sprints. Cerimônias como daily scrums foram utilizadas para alinhar o progresso e esclarecer regras de negócio com o cliente.

O projeto foi construído com HTML e CSS para a estruturação e estilização das páginas, Python como linguagem principal do back-end e Flask como framework para gerenciamento do servidor e renderização de templates dinâmicos. O controle de versão foi realizado com Git, e a aplicação foi hospedada na AWS utilizando o serviço EC2. O conteúdo sobre Scrum foi organizado em páginas temáticas com navegação por menus e submenus, buscando clareza na apresentação e facilidade de uso para o público-alvo.

2.2 2024 – 1º semestre: DataEase

Desenvolvido no primeiro semestre de 2024, o projeto **DataEase** (Machado, 2024a) foi uma Aprendizagem por Projetos Integrados (API) realizada na FATEC São José dos Campos, sob orientação do Prof. Giuliano Bertoti. A proposta abordou o tema de consulta inteligente a bases de dados corporativas, voltada a uma empresa parceira que enfrentava dificuldades no acesso e na extração de informações de forma manual e pouco intuitiva. O objetivo do projeto foi solucionar as inconsistências geradas por esse processo, que comprometiam a confiabilidade dos dados, provocavam retrabalho e impactavam negativamente a tomada de decisão e as atividades operacionais da equipe.

A solução desenvolvida consistiu em uma aplicação desktop integrada a um modelo de linguagem (LLM), que permitia ao usuário realizar consultas em suas bases de dados por meio de linguagem natural, de forma simples e acessível. O sistema foi projetado para suportar a integração com diferentes tipos de modelos de linguagem, ampliando sua flexibilidade de uso. O desenvolvimento seguiu práticas da metodologia ágil, com organização do backlog, definição de requisitos, modelagem de classes e implementação

incremental das funcionalidades ao longo das sprints, além de seguir as diretrizes conceituais de engenharia de software propostas por Sommerville (2019). . A atuação do Scrum Master incluiu não apenas a facilitação das cerimônias, mas também contribuições técnicas diretas, dada a alta complexidade da solução proposta.

O projeto foi construído inteiramente em Java, linguagem utilizada tanto para o desenvolvimento da interface com Java Swing quanto para a conexão com os modelos de linguagem por meio da biblioteca LangChain4j. O LM Studio foi empregado para a orquestração e execução local dos modelos, enquanto o MySQL foi utilizado para o armazenamento das informações de usuário. O controle de versão foi gerenciado com Git. A arquitetura da aplicação priorizou a usabilidade, permitindo que colaboradores sem conhecimento técnico em SQL pudessem extrair informações de bancos de dados utilizando perguntas em linguagem natural, traduzidas automaticamente em consultas estruturadas pelo modelo de linguagem integrado.

2.3 2024 – 2º semestre: OAK-RH

Desenvolvido no segundo semestre de 2024, o projeto **OAK-RH** (Machado, 2024b) foi uma Aprendizagem por Projetos Integrados (API) realizada na FATEC São José dos Campos, em parceria com a empresa Youtan, atuante no setor de software. A proposta abordou o tema de gestão do clima organizacional, voltada ao setor de Recursos Humanos da empresa parceira, que enfrentava dificuldades no monitoramento da satisfação dos colaboradores em relação ao ambiente de trabalho. O objetivo do projeto foi resolver a ausência de um processo estruturado de feedback e as falhas na comunicação interna, que resultavam em desmotivação e redução da colaboração entre as equipes.

A solução desenvolvida consistiu em uma aplicação web com dashboards interativos para o acompanhamento de pesquisas e feedbacks organizacionais, apresentando gráficos e indicadores que auxiliam na interpretação dos resultados. O sistema conta com dois modos de acesso: o modo administrador, voltado ao RH, permite criar equipes, cadastrar usuários, elaborar pesquisas e visualizar respostas nos dashboards; o modo usuário, destinado aos colaboradores, é subdividido entre líderes, que acessam as informações de seus liderados, e liderados, que visualizam apenas seus próprios dados e a evolução do seu desempenho ao longo do tempo. O desenvolvimento seguiu práticas da metodologia ágil, incluindo organização do backlog, definição de requisitos, estruturação da arquitetura backend no

padrão MVC e implementação incremental das funcionalidades ao longo das sprints. Durante o projeto, houve uma mudança significativa de escopo, migrando do desenvolvimento tradicional para a construção de uma API RESTful, o que exigiu rápida adaptação da equipe.

O frontend foi construído com React e a biblioteca Material UI, utilizando JavaScript e TypeScript, seguindo as instruções da literatura técnica propostas por Adriano (2021), para a lógica de interação e tipagem estática. O backend foi desenvolvido em Node.js com arquitetura MVC, utilizando Sequelize como ORM para a comunicação com o banco de dados PostgreSQL, responsável pelo armazenamento dos dados do sistema. O Firebase foi empregado para autenticação de usuários e armazenamento de credenciais. A integração entre frontend e backend foi realizada por meio do Axios, e o controle de versão foi gerenciado com Git. A aplicação priorizou a usabilidade e a organização do conteúdo, oferecendo interfaces tabuladas para o gerenciamento de equipes, usuários e pesquisas, além de gráficos de barras para o acompanhamento da evolução individual dos colaboradores.

2.4 2025 – 1º semestre: Lumen

Desenvolvido no primeiro semestre de 2025, o projeto **Lumen** (Machado, 2025a) foi uma Aprendizagem por Projetos Integrados (API) realizada na FATEC São José dos Campos, em parceria com a FAPG (Fundação de Apoio à Pesquisa e Gestão), instituição atuante na área de pesquisa de pós-graduação. A proposta abordou o tema de gerenciamento de projetos, voltada a uma empresa parceira que enfrentava dificuldades no planejamento, controle e acompanhamento dos projetos em desenvolvimento. O objetivo do projeto foi resolver a ausência de um processo padronizado de gestão, que resultava em falhas no levantamento de requisitos, atrasos nas entregas, dificuldades na alocação de recursos financeiros e falta de visibilidade sobre o andamento das atividades.

A solução desenvolvida consistiu em uma aplicação web para gerenciamento de projetos, permitindo o cadastro de informações como nome, descrição, datas, orçamento e participantes. Cada projeto possui etapas e tarefas que podem ser atribuídas a membros específicos, possibilitando o acompanhamento detalhado do progresso por meio de tabelas com informações de prazo, status e projeto associado. O sistema conta ainda com uma funcionalidade de lixeira, na qual projetos excluídos permanecem disponíveis por 30 dias para possível restauração, e com um chatbot interativo que permite atualizar o status das

tarefas de forma dinâmica e intuitiva. O desenvolvimento seguiu práticas da metodologia ágil, com a atuação do Product Owner na documentação do Product Backlog, detalhamento de histórias de usuário com critérios de aceitação e cenários de teste, priorização de tarefas e registro do andamento das Sprint Backlogs, garantindo alinhamento contínuo com o cliente por meio de reuniões de validação de requisitos e regras de negócio.

O frontend foi construído com React, utilizando JavaScript e TypeScript para a criação de interfaces componentizadas, tipagem estática e integração com o backend por meio do consumo de APIs RESTful. O backend foi desenvolvido em Node.js, com a implementação de CRUDs, rotas e schemas para as entidades do sistema, incluindo definição de relações e regras de validação. O PostgreSQL foi utilizado como banco de dados relacional para o armazenamento dos dados, o banco de dados foi modelado seguindo os ensinamentos da literatura técnica de Date (2005) , e o Firebase foi empregado para autenticação de usuários. O chatbot foi viabilizado com o Ollama, ferramenta utilizada para executar e gerenciar modelos de linguagem localmente. O controle de versão foi gerenciado com Git, e a documentação do projeto foi organizada de forma estruturada no repositório do GitHub, com artefatos separados por arquivos para facilitar a navegação e o acompanhamento do progresso pela equipe.

2.5 2025 – 2º semestre: GeoMaps

Desenvolvido no segundo semestre de 2025, o projeto **GeoMaps** (Machado, 2025b) foi uma Aprendizagem por Projetos Integrados (API) realizada na FATEC São José dos Campos, em parceria com a Visiona, empresa atuante no setor geoespacial. A proposta abordou o tema de endereçamento digital para áreas rurais, voltada à problemática enfrentada por populações residentes em regiões sem endereços formais. O objetivo do projeto foi solucionar as dificuldades de identificação e localização dessas residências, que comprometem a atuação de órgãos governamentais, a realização de entregas de produtos e o acesso de familiares e visitantes, impactando diretamente a integração dessas comunidades a serviços essenciais e sua inclusão social e logística.

A solução desenvolvida consistiu em uma aplicação móvel que permite o cadastro de usuários com informações pessoais e a identificação do código CAR do imóvel associado ao CPF informado. Após a autenticação, a aplicação disponibiliza um mapa interativo com a visualização da propriedade por meio de área demarcada e, caso o imóvel não possua

endereço formal, possibilita a geração de um endereço digital (Plus Code) a partir do centróide da propriedade ou da seleção de um ponto dentro de seus limites, além de permitir a atualização do endereço e a consulta ao histórico de alterações. O sistema também oferece funcionalidades de rotas e alertas, permitindo ao usuário definir trajetos, visualizar condições climáticas e registrar alertas viários classificados por gravidade e tipo. Complementarmente, foi desenvolvida uma aplicação web para o gerenciamento da solução móvel, com controle de usuários e alertas cadastrados. O desenvolvimento seguiu práticas da metodologia ágil, com organização do backlog, definição de requisitos, estruturação da arquitetura backend em camadas com princípios de Clean Code e implementação incremental das funcionalidades ao longo das sprints.

O frontend móvel foi construído com React Native, para o desenvolvimento mobile foi utilizado as instruções presentes no livro técnico GRIFFITHS (2019), utilizando JavaScript e TypeScript para a criação de interfaces dinâmicas e componentizadas. O backend foi desenvolvido em Python com o framework FastAPI, responsável pela criação da API RESTful e pela implementação das regras de negócio, incluindo a geração de Plus Codes por meio da biblioteca geoespacial Shapely. O MongoDB foi utilizado como banco de dados não relacional, com conexão realizada pelo driver PyMongo, e a exploração e padronização dos dados fornecidos pela empresa parceira foi conduzida em Jupyter Notebooks com a biblioteca Pandas. O portal administrativo web foi desenvolvido em React com integração ao backend. O Node.js foi empregado como ambiente de execução do servidor web, o Docker foi utilizado para a containerização e o deploy da aplicação, e o controle de versão foi gerenciado com Git, com o projeto organizado em submódulos separados para frontend e backend. A documentação completa do backend foi elaborada no repositório, detalhando a estrutura do projeto e as instruções para execução em ambiente local.

2.6 2026 – 1º semestre: rules AI

Desenvolvido no primeiro semestre de 2026, o projeto **rules AI** (Machado, 2026a) foi uma Aprendizagem por Projetos Integrados (API) realizada na FATEC São José dos Campos, em parceria com a DomRock, empresa atuante na área de Inteligência Artificial. A proposta abordou o tema de gerenciamento dinâmico de regras de negócio com o apoio de agentes de IA, voltada à problemática enfrentada por organizações que precisam ajustar continuamente suas regras comerciais em resposta às demandas do mercado. O objetivo do

projeto foi resolver as fragilidades operacionais decorrentes da ausência de um registro adequado dessas regras, que provocavam inconsistências na execução de processos, divergências entre critérios adotados por diferentes colaboradores, dependência de conhecimento não documentado e comprometimento da rastreabilidade das decisões tomadas.

A solução desenvolvida consistiu em uma aplicação web para o gerenciamento de regras de negócio, capaz de centralizar informações de lojas, marcas e funcionários, e de permitir que o usuário crie novas regras por meio de linguagem natural. A solicitação do usuário é enviada para uma API em Python construída com LangGraph, na qual um agente de IA processa o conteúdo e retorna a regra de negócio estruturada. Cada regra criada é vinculada a uma loja, marca ou funcionário, passando a ser utilizada no cálculo de comissão, que ao final exibe as regras aplicadas, garantindo transparência na apuração dos valores. O agente conta ainda com observabilidade implementada via MLflow, permitindo o monitoramento de tokens consumidos, a avaliação da qualidade das respostas, o versionamento de prompts e o controle de erros. O desenvolvimento seguiu práticas da metodologia ágil, com organização do backlog, definição de requisitos junto à empresa parceira, prototipação da interface e implementação incremental das funcionalidades ao longo das sprints. Durante o processo, foi realizada uma migração arquitetural do núcleo do agente, que passou de uma estrutura linear baseada em LangChain para um grafo iterativo construído com LangGraph, composto pelos nós Agent Node, Code Editor e Review Node, viabilizando fluxos condicionais com maior capacidade de raciocínio e tomada de decisão dinâmica.

O frontend foi construído com Vue.js para a criação de interfaces dinâmicas e componentizadas. O backend foi desenvolvido em Java com o framework Spring Boot, responsável pela implementação das regras de negócio e pela integração entre os módulos do sistema. O agente de IA foi construído em Python com FastAPI para a exposição da API RESTful, utilizando LangChain e LangGraph para a orquestração do fluxo, com integração aos modelos Qwen3-Coder-30B-A3B-Instruct (Hugging Face, 2026), Gemini 2.0 Flash Preview (Google) e Meta-Llama/Llama-4-Scout-17B-16E-Instruct (Groq). O módulo de RAG (Retrieval-Augmented Generation) foi implementado por meio de um banco de dados vetorial estruturado em duas coleções, com embeddings gerados pelo modelo nomic-embed-text:latest executado via Ollama. O MongoDB foi utilizado como banco de dados não relacional para o armazenamento de funcionários, lojas, marcas e regras de negócio,

enquanto o Node.js serviu como ambiente de execução do servidor web. O MLflow foi empregado para o versionamento de modelos e a observabilidade do agente, o Docker para a containerização e o deploy da aplicação, e o controle de versão foi gerenciado com Git, com a utilização de submódulos para isolar o repositório do agente de IA do restante do projeto.

5. CONSIDERAÇÕES FINAIS

A trajetória acadêmica desenvolvida ao longo da graduação na FATEC São José dos Campos, por meio da metodologia de Aprendizagem por Projetos Integrados, proporcionou uma formação progressiva e consistente, na qual cada semestre representou uma etapa de amadurecimento técnico e profissional. O contato direto com demandas reais de empresas parceiras permitiu que os conhecimentos teóricos adquiridos em sala de aula fossem aplicados em contextos práticos, consolidando competências que dificilmente seriam desenvolvidas em abordagens exclusivamente tradicionais de ensino. A evolução dos projetos — de uma aplicação web educacional com HTML e Flask até uma solução com agentes de IA orquestrados por grafos, integração a múltiplos modelos de linguagem e observabilidade via MLflow — evidencia a ampliação contínua do repertório técnico e a capacidade de adaptação a diferentes cenários e tecnologias, inclusive àquelas que representam a vanguarda do desenvolvimento de software.

As principais contribuições dos projetos residem no desenvolvimento de habilidades que transcendem o domínio técnico. O trabalho em equipe, exercido de forma constante ao longo dos semestres, exigiu comunicação clara, negociação de conflitos e alinhamento de expectativas entre os integrantes. A organização das atividades por meio de metodologias ágeis, especialmente o framework Scrum, proporcionou disciplina no planejamento, na priorização de entregas e no acompanhamento do progresso. O levantamento de requisitos junto a clientes reais ensinou a importância de formular perguntas objetivas, validar regras de negócio e manter o alinhamento contínuo entre a equipe de desenvolvimento e o solicitante. A vivência em diferentes papéis — desenvolvedor, Scrum Master e Product Owner — ampliou a compreensão sobre as responsabilidades envolvidas em cada função e sobre a dinâmica de um time ágil. Entre os desafios enfrentados, destacam-se a mudança de escopo durante o desenvolvimento do OAK-RH, que exigiu rápida adaptação à arquitetura RESTful, a complexidade técnica da integração com modelos de linguagem no DataEase, a necessidade de explorar e padronizar dados geoespaciais no GeoMaps, e a identificação da limitação estrutural do agente de IA no API_6, que motivou a migração do núcleo do LangChain para o LangGraph e resultou em um sistema mais robusto, capaz de tomar decisões dinâmicas ao longo de sua execução. Em cada situação, a busca por soluções envolveu estudo autônomo, experimentação prática, pensamento analítico e colaboração entre os membros da equipe.

Os projetos desenvolvidos contribuíram de forma significativa para a formação profissional, ao proporcionar experiências que simulam as condições reais do mercado de trabalho. O contato com tecnologias diversas, a necessidade de cumprir prazos e entregar soluções funcionais, e a interação com clientes e stakeholders reais prepararam o aluno para enfrentar os desafios do desenvolvimento de software contemporâneo com maior segurança e autonomia, incluindo a construção de sistemas baseados em inteligência artificial generativa, área que se consolida como uma das mais relevantes da computação atual. Além do crescimento técnico, a trajetória fortaleceu competências interpessoais como liderança, proatividade, resiliência, organização e pensamento analítico, qualidades indispensáveis para a atuação profissional em equipes multidisciplinares. A metodologia API demonstrou ser um instrumento eficaz de formação, ao articular de forma contínua o ambiente acadêmico e as necessidades do mercado, e as experiências acumuladas ao longo dos semestres constituem uma base sólida tanto para a inserção profissional quanto para a continuidade do aprendizado ao longo da carreira.

REFERÊNCIAS BIBLIOGRÁFICAS

ADRIANO, Thiago da Silva. **Guia prático de TypeScript: melhore suas aplicações JavaScript**. São Paulo: Casa do Código, 2021.

DATE, Chris J. **Database in depth: relational theory for practitioners**. [S.L.]: O'Reilly Media, 2005.

GRIFFITHS, D. **Use a Cabeça! Desenvolvendo Para Android**. 2. ed. Rio de Janeiro: Alta Books, 2019.

HUGGING FACE. **Sentence Transformers**. Disponível em: <https://huggingface.co/sentence-transformers>. Acesso em: 01 mar. 2026.

Machado, Samuel Alkmin. **Projeto de API – 1º Semestre**. GitHub, 2023. Disponível em: <https://github.com/Phoenix-Team-Fatec/ScrumTutor>. Acesso em: 10 maio. 2026. \

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

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

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

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

Machado, Samuel Alkmin. **Projeto de API – 6º Semestre**. GitHub, 2023. Disponível em: https://github.com/Phoenix-Team-Fatec/API_6. Acesso em: 10 maio. 2026.

SCHWABER, Ken; SUTHERLAND, Jeff. **The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game**. [S.l.]: Scrum.org & Scrum Inc., 2020.

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

APÊNDICE A – PORTIFÓLIO

Introdução

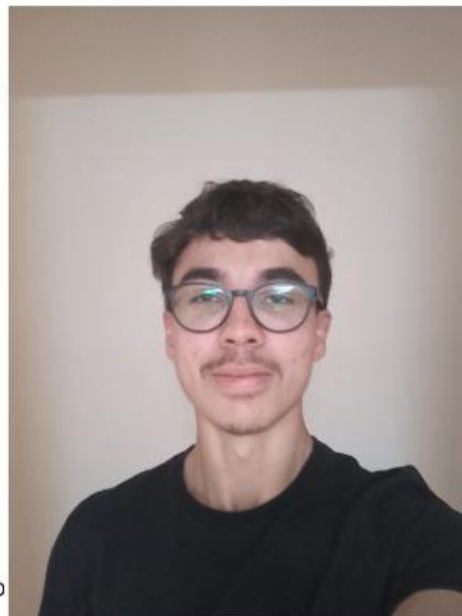
Olá meu nome é Samuel, tenho 20 anos e atualmente estou cursando o 5º semestre do curso de Análise e Desenvolvimento de Sistemas na Fatec São José dos Campos - Prof. Jessen Vidal.

Histórico Acadêmico

Entre 2021 e 2022, realizei um curso técnico em Mecânica no Cephas (Colégio de Educação Profissional Hélio Augusto de Souza), com duração de um ano e meio. Apesar de ter concluído a formação, não cheguei a atuar na área.

Motivação

Meu primeiro contato com Tecnologia da Informação ocorreu durante a graduação, momento que despertei grande interesse, principalmente o quanto podemos utilizar a tecnologia para resolver problemas reais. Desde então venho constantemente aprimorando os meus conhecimentos, com foco em Ciência de Dados, área que pretendo consolidar minha carreira profissional, além de demonstrar interesse em redes de computadores e sistemas operacionais, especialmente no ambiente Linux.



Histórico Profissional

Atualmente sou trainee na DataSide, uma empresa de consultoria focada em dados, inteligência artificial e negócios, cuja missão é acompanhar organizações em sua jornada de evolução com IA aplicada ao ambiente corporativo. Minhas atribuições concentram-se no meu desenvolvimento profissional em ciência de dados, estudando tópicos como Machine Learning, Análise Exploratória de Dados, MLOps e Engenharia de Software com Python. No decorrer do meu avanço técnico, serei integrado a projetos da empresa, atuando com clientes reais do mercado.

clientes reais do mercado.

Contatos

- [Linkedin](#)

-
- [GitHub](#)

Meus Principais Conhecimentos



Python •



Linux •



React •



FastAPI •



JavaScript •



TypeScript •



Node.js •



PostgreSQL •



scikit-learn •



pandas •



Docker •



Git

Meus Projetos

(2023-2) | Fatec-Sjc | Web Site para capacitação do Scrum

- [API - 1º Semestre](#)

(2024-1) | Fatec-Sjc | Aplicação Desktop para consulta com linguagem natural em dados

- [API - 2º Semestre](#)

(2024-2) | Fatec-Sjc | Aplicação web com dashboards organizacionais

- [API - 3º Semestre](#)

(2025-1) | Fatec-Sjc | Aplicação web para gerenciamento inteligente de projetos

- [API - 4º Semestre](#)

(2025-2) | Fatec-Sjc | Aplicação mobile para criação de endereços digitais

- [API - 5º Semestre](#)

(2026-1) | Fatec-Sjc | Aplicação web para gerenciamento de regras de negócio

- [API - 6º Semestre](#)

Em 2023-2

Empresa: FATEC São José dos Campos - SP

Profº Antônio Egydio São Tiago Graça

Problema:

A empresa parceira identificou dificuldades na aplicação da metodologia ágil Scrum por parte de seus colaboradores. A falta de entendimento sobre os papéis, eventos e artefatos da metodologia tem gerado ruídos de comunicação entre as equipes, atrasos nas entregas e baixa colaboração entre os membros. Essa defasagem de conhecimento impacta diretamente a produtividade e a qualidade das entregas, dificultando a adoção eficiente de práticas ágeis no ambiente organizacional.

Solução:

Desenvolvimento de uma aplicação web voltado à formação de pessoas na metodologia **Scrum**, possibilitando também a **avaliação de integrantes da própria equipe**. O objetivo é capacitar o usuário a aplicar os princípios do Scrum em situações reais do seu cotidiano profissional.

[Repositório](#)

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

Atuando como desenvolvedor no projeto, fui responsável pela implementação e estilização das páginas Papéis, Bibliografia e Apêndice, além dos componentes Menu, Submenu e dos formulários de avaliação PACER. O PACER consiste em uma avaliação de zero a quatro nos seguintes tópicos: proatividade, autonomia, colaboração e entrega de resultados. Todos os componentes foram desenvolvidos utilizando templates do framework Flask.

Também contribuí no desenvolvimento da lógica de envio dos formulários e na resolução de conflitos de versionamento utilizando Git. Em um dos casos, resolvi conflitos nos arquivos style.css e artefatos.html, ocasionados pela edição simultânea do mesmo trecho de código por dois integrantes do grupo. Para solucionar o problema, removi duplicidades e mantive a versão correta de cada trecho.

Por fim, realizei a hospedagem da aplicação na AWS, configurando a instância, executando o build do projeto e instalando todas as dependências necessárias para o funcionamento do sistema.

Hard Skills

Tecnologia	Proficiência	Descrição
HTML	faço com autonomia	Estruturação semântica e acessível de páginas web.
CSS	faço com autonomia	Estilização e criação de layouts responsivos e modernos com Flexbox/Grid.
Flask	faço com ajuda	Desenvolvimento de aplicações web, renderização de templates dinâmicos.
Python	faço com autonomia	Desenvolvimento da lógica de back-end, manipulação de dados e automação de scripts.
AWS	faço com ajuda	Implantação e hospedagem de aplicações em serviços cloud (EC2).
Git	faço com autonomia	Controle de versão, trabalho em equipe com branches e gestão de repositórios.

Figma	faço com ajuda	Prototipagem de interfaces de usuário (UI)
--------------	----------------	--

Soft Skills

- Comunicação:** *Durante uma daily scrum, o grupo apresentava dificuldades em definir a pergunta adequada para esclarecer uma regra de negócio com o cliente. Essa regra estava relacionada à forma como o conteúdo sobre o framework Scrum seria apresentado aos usuários, envolvendo tanto a organização das páginas quanto o mapa de navegação da aplicação. Ao perceber a falta de alinhamento entre os integrantes, sugeri que estruturássemos previamente as telas do sistema e o conteúdo previsto em cada uma delas. Essa organização facilitou a compreensão do contexto geral e permitiu formular uma pergunta clara e objetiva ao cliente, que, por sua vez, conseguiu explicar de maneira precisa a regra necessária para o desenvolvimento da funcionalidade.*
- Trabalho em Equipe:** *Um dos requisitos do projeto era desenvolver o sistema de formulários que permitia ao usuário avaliar os membros de sua equipe com base no PACER. Durante essa etapa, um integrante do grupo encontrou dificuldades para implementar a lógica de recebimento e tratamento dos dados enviados pelo elemento `<form>` do HTML. Colaborei diretamente com ele na construção dessa lógica utilizando Python e o framework Flask, discutindo alternativas, dividindo tarefas e revisando o fluxo de dados.*

O principal desafio era estruturar a captura das notas preenchidas no formulário e organizá-las em objetos. Para resolver esse problema, desenvolvemos juntos uma rotina que recebia os valores enviados via request.form, organizando-os em listas separadas para cada critério do PACER (proatividade, autonomia, colaboração e entrega). Além disso, adicionamos dinamicamente as avaliações referentes aos desenvolvedores de teste (D.T), utilizando um laço for para percorrer a quantidade de membros definida no sistema.

Após coletar todos os valores, implementamos a criação de objetos da classe avaliacao, nos quais cada instância representava a avaliação individual de um papel da equipe (P.O, S.M ou D.T). Esses objetos eram armazenados no vetor notas, garantindo que todas as informações estivessem estruturadas corretamente para posterior exibição ou processamento.

Ao trabalhar em conjunto nessa implementação, conseguimos finalizar a funcionalidade de forma eficiente, fortalecendo a cooperação, o aprendizado mútuo e o ritmo de desenvolvimento do grupo.

Em 2024-1

Empresa: *FATEC São José dos Campos - SP*
Profº Giuliano Bertoti

Problema:

A empresa parceira enfrenta dificuldades relacionadas ao acesso e consulta de informações em sua base de dados corporativa. Atualmente, o processo de extração de dados é realizado de forma manual e pouco intuitiva, o que gera inconsistências nas informações utilizadas pelos colaboradores e impacta a tomada de decisão. Essas divergências comprometem a confiabilidade dos dados e resultam em retrabalho e atrasos nas atividades operacionais da equipe.

Solução:

Foi desenvolvida uma aplicação desktop integrada a um modelo de linguagem (LLM), que possibilita um usuário realizar consultas em suas bases de dados por meio de linguagem natural, de forma simples e intuitiva.
O sistema permite a integração com diferentes tipos de modelos de linguagem.

[Repositório](#)

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

Atuando como Scrum Master, e considerando a grande complexidade técnica envolvida na solução proposta, precisei contribuir diretamente na parte técnica do projeto. Fui

responsável pela integração dos modelos de linguagem à aplicação utilizando a biblioteca LangChain4j, pela modelagem das classes e pelo gerenciamento dos modelos no LM Studio. Também desenvolvi a interface da aplicação e realizei a integração dela com o código dos modelos, além de modelar e implementar todo o banco de dados.

► Gerando SQL com linguagem natural

Hard Skills

Tecnologia	Proficiência	Descrição
Java	faço com ajuda	Desenvolvimento da interface e modelagem das classes.
MySQL	faço com ajuda	Modelagem e desenvolvimento do banco de dados
LangChain4j	faço com ajuda	Conexão com os modelos
LM Studio	faço com ajuda	Orquestração e execução dos modelos
Git	faço com autonomia	Controle de versão, trabalho em equipe com branches e gestão de repositórios.

Soft Skills

- **Liderença:** Atuando como Scrum Master, identifiquei que parte da equipe estava com dificuldades na utilização da linguagem Java. Para compreender melhor as necessidades individuais, conversei separadamente com cada membro, buscando entender suas principais barreiras técnicas. A partir desse diagnóstico, propus a realização de cursos online gratuitos para reforçar o aprendizado e ofereci suporte direto nas tarefas de maior complexidade. Essas ações ajudaram a reduzir impedimentos e contribuíram para o avanço contínuo do time.
- **Proatividade:** Após a definição do design da aplicação, tomei a iniciativa de iniciar o desenvolvimento da interface utilizando Java Swing, estruturando os primeiros componentes visuais e estabelecendo o padrão de organização do frontend. Ao avançar de forma antecipada, facilitei o trabalho dos demais integrantes e acelerei a evolução inicial da aplicação.

[Voltar](#)

Em 2024-2

Empresa: Youtan

Área de atuação: Empresa de Software

Problema:

O setor de Recursos Humanos da empresa parceira enfrenta dificuldades na gestão do clima organizacional e na compreensão da satisfação dos colaboradores em relação ao ambiente de trabalho. A ausência de um processo estruturado de feedback e as falhas na comunicação interna têm dificultado o monitoramento das percepções dos funcionários, resultando em desmotivação e redução da colaboração entre as equipes.

Solução:

Foi desenvolvida uma aplicação web que oferece dashboards interativos para o acompanhamento das pesquisas e feedbacks organizacionais, com gráficos e indicadores que auxiliam na interpretação dos resultados.

O sistema permite a criação manual de pesquisas e o gerenciamento tabulado de equipes, usuários e pesquisas.

► Details
[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).
Git	Sistema de controle de versão distribuído, utilizado para o gerenciamento do código-fonte e colaboração em equipe.

Contribuições Pessoais

Atuando como desenvolvedor, fui responsável pela estruturação do backend seguindo o padrão de projeto MVC, organizando as pastas e arquivos de forma modular. O padrão MVC é composto por três camadas — Models, Views e Controllers — e, com base nele, a arquitetura do projeto foi definida da seguinte forma:

```

├─ database/
│   └─ config/
│   └─ controllers/
│   └─ entities/
│   └─ services/
│       └─ routes.ts
├─ middlewares/
└─ server.ts

```

Realizei a conexão com o banco de dados e desenvolvi a modelagem das classes **Usuário** (administrador, líder e liderado), **Equipe**, **Perguntas** e **Formulários**, implementando um CRUD completo para cada uma delas, incluindo **models**, **services**, **controllers**, relações e rotas correspondentes.

No frontend, contribui para o desenvolvimento da **sidebar**, integração com a biblioteca **Material UI** e comunicação com o backend por meio do **Axios**, implementando o cadastro de usuários, criação de formulários e uma tabela para controle das respostas dos participantes.

► Modelagem da relação entre as classes equipe e usuário

Hard Skills

Tecnologia	Proficiência	Descrição
------------	--------------	-----------

React	faço com ajuda	Desenvolvimento de página e componentes, além da integração com backend
Node.js	faço com autonomia	Utilizado como ambiente de execução do JavaScript no servidor
JavaScript	faço com autonomia	Desenvolvimento e integração de componentes
TypeScript	faço com autonomia	Criação de classes, tipagem estática e recursos de programação orientada a objetos
PostgreSQL	faço com ajuda	Modelagem e implementação do banco de dados, definição de regras de negócios, criação de tabelas e relações entre elas.
Git	faço com autonomia	Controle de versão, trabalho em equipe com branches e gestão de repositórios.

Soft Skills

- **Flexibilidade e Resiliência:** *Durante o projeto, enfrentamos uma mudança significativa no escopo, passando do desenvolvimento tradicional para a construção de uma API RESTful. Esse novo cenário exigiu rápida adaptação ao uso de métodos HTTP, criação de rotas, consumo de dados em JSON e compreensão dos princípios de arquitetura REST. Busquei me aprofundar nesses tópicos por meio de estudos adicionais e experimentação prática, o que me permitiu continuar contribuindo com o desenvolvimento sem ser impactado negativamente pelas mudanças.*
- **Proatividade:** *Tomei a iniciativa de iniciar o desenvolvimento da interface do sistema, pesquisando previamente padrões de projeto e boas práticas para aplicações modernas. Com base nisso, dei início à estruturação do frontend utilizando React e a biblioteca Material UI, definindo estilos, componentes e a organização base da aplicação. Essa ação antecipada facilitou o alinhamento visual do projeto e acelerou o avanço da equipe.*

[Voltar](#)

Em 2025-1

Empresa: FAPG (Fundação de Apoio à Pesquisa e Gestão)

Área de atuação: Pesquisa de pós graduação

Problema:

A empresa parceira enfrenta dificuldades no gerenciamento e acompanhamento dos projetos em desenvolvimento. A ausência de um processo padronizado de planejamento e controle tem resultado em falhas no levantamento de requisitos, atrasos nas entregas e dificuldades na alocação de recursos financeiros. Essa falta de visibilidade sobre o andamento das atividades compromete o cumprimento de prazos e afeta a eficiência geral da gestão de projetos.

Solução:

Foi desenvolvida uma aplicação web para o gerenciamento de projetos, permitindo o cadastro de nome, descrição, datas, orçamento e participantes.

Cada projeto possui etapas e tarefas que podem ser atribuídas a membros específicos, possibilitando o acompanhamento detalhado do progresso.

O sistema conta com uma funcionalidade de lixeira, onde projetos excluídos permanecem disponíveis por 30 dias para possível restauração.

As tarefas são exibidas em uma tabela com informações como prazo, status e projeto associado.

Além disso, foi implementado um chatbot interativo que permite atualizar o status das tarefas para “concluída” ou “em andamento” de forma dinâmica e intuitiva.

[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

Atuando como Product Owner, fui responsável pela documentação do projeto no GitHub, onde organizei e mantive o Product Backlog, detalhei histórias de usuário e realizei a priorização das tarefas. Também registrei e acompanhei o andamento das Sprint Backlogs, garantindo visibilidade do progresso, alinhamento entre a equipe e aderência aos requisitos definidos.

Exemplo de User Storie

Rank	Prioridade	US	Estimativa	Sprint	Requisitos do parceiro
#01	Alta	Como usuário do sistema, quero fazer login com credenciais seguras, para proteger as informações dos projetos.	15 Horas	1	RF5, RNF1

Além das atribuições de PO, contribuí tecnicamente no backend, desenvolvendo o CRUD, as rotas e os schemas das entidades Tarefas, Subtarefas e Áreas de Atuação, incluindo a definição de suas relações e regras de validação.

No frontend, atuei diretamente na integração com o backend das entidades que desenvolvi, criando funções para interpretar os JSONs retornados pela API e refatorando trechos do código React em páginas e componentes, como:

TaskDetails

Modal responsável por exibir informações detalhadas sobre uma tarefa.

- Criação de *useEffects* para carregar subtarefas armazenadas no banco de dados.
- Implementação de *useStates* para armazenar e manipular dados das subtarefas.
- Definição de interfaces *TypeScript* para a tipagem de subtarefas.

Página de tarefas

Página para visualização e gerenciamento de tarefas.

- Criação de *useEffects* para carregar as tarefas existentes no banco de dados.
- Implementação de *useStates* para manipular os dados de cada tarefa.
- Definição de interfaces *TypeScript* para garantir tipagem adequada das tarefas.

Hard Skills

Tecnologia	Proficiência	Descrição
React	faço com ajuda	Desenvolvimento de página e componentes, além da integração com backend
Node.js	faço com autonomia	Utilizado como ambiente de execução do JavaScript no servidor
JavaScript	faço com autonomia	Desenvolvimento e integração de componentes
TypeScript	faço com autonomia	Criação de classes, tipagem estática e recursos de programação orientada a objetos
PostgreSQL	faço com ajuda	Modelagem e implementação do banco de dados, definição de regras de negócios, criação de tabelas e relações entre elas.
Git	faço com autonomia	Controle de versão, trabalho em equipe com branches e gestão de repositórios.

Soft Skills

- **Comunicação:** *Como Product Owner, mantive contato direto com o cliente para compreender suas necessidades e garantir o alinhamento dos requisitos ao longo do desenvolvimento. Em uma das reuniões, busquei validar a regra de negócio relacionada aos filtros da aplicação, elaborando uma pergunta clara e objetiva para evitar ambiguidades e assegurar um entendimento comum entre ambas as partes. Essa abordagem facilitou a comunicação e permitiu que o cliente esclarecesse exatamente o comportamento esperado. A seguir, a mensagem enviada ao cliente para validação:*

Mensagem

"Bom dia Walmir!

Gostaria de saber se você está de acordo com os filtros que vamos entregar nessa sprint: • Filtro por Área de atuação

- Status
- Filtro por projetos em que o usuário é coordenador
- Filtro por projetos em que o usuário é membro

Além disso, na questão de exclusão de um projeto, pensamos em quando o usuário deletar um projeto, enviaremos ele para uma lixeira, e em um período de 30 dias o projeto estará disponível para recuperação, após o período ele será excluído permanentemente, você está de acordo com esta metodologia ?

Segue também os cenários de testes que fiz com base no backlog apresentado anteriormente, lembrando que irei substituir os critérios de aceitação pelos cenários de testes."

- **Organização:** *Exercendo o papel de Product Owner, fui responsável pela documentação estruturada do projeto no GitHub, organizando o Product Backlog, as histórias de usuário e os Sprint Backlogs. Estruturei os artefatos do projeto em arquivos separados dentro do repositório, garantindo clareza na navegação e no versionamento. Também registrei os critérios de aceitação em formato tabular e em um arquivo dedicado. Além disso, por solicitação do cliente, foram elaborados cenários de teste para cada User Story presente no backlog. Essa organização contribuiu para uma leitura mais objetiva dos artefatos, além de facilitar o acompanhamento do progresso pela equipe ao longo das sprints.*

Exemplo de critério de aceitação

US	Cenário de Teste 1	Cenário de Teste 2
#01	Ação: Usuário acessa a página de cadastro e preenche os campos obrigatórios (nome, sobrenome, email, senha). Resultado esperado: Sistema cria a conta e exibe mensagem de sucesso.	Ação: Usuário insere email e senha cadastrados na página de login. Resultado esperado: Sistema redireciona para a dashboard e exibe boas-vindas.

[Voltar](#)

Em 2025-2

Empresa: Visiona

Área de atuação: Geoespacial

Problema:

A população residente em áreas rurais enfrenta dificuldades relacionadas à ausência de endereços formais. Essa limitação compromete a identificação e a localização dessas residências por órgãos governamentais, além de dificultar a realização de entregas de produtos e o acesso de familiares e visitantes. A falta de um sistema padronizado de endereçamento impacta diretamente a integração dessas comunidades a serviços essenciais, afetando sua inclusão social e logística.

Solução:

Foi desenvolvida uma aplicação mobile que permite o cadastro de usuários por meio de informações pessoais, como nome, CPF, e-mail, data de nascimento e fotografia. A partir do CPF informado, o sistema realiza a identificação do código CAR do imóvel associado ao usuário.

Após o cadastro e autenticação, a aplicação disponibiliza um mapa interativo que permite a visualização da propriedade por meio de uma área demarcada. Caso o imóvel não possua endereço formal, o sistema possibilita a geração de um endereço digital (Plus Code) a partir do centróide da propriedade ou da seleção de um ponto dentro de seus limites, além de permitir a atualização do endereço e a consulta ao histórico de alterações.

A solução também oferece funcionalidades de rotas e alertas, possibilitando ao usuário definir trajetos e visualizar condições climáticas e alertas ao longo do percurso. Usuários cadastrados podem registrar alertas viários, classificados por grau de gravidade (leve, moderado ou grave) e por tipo, como trânsito, acidente, veículo no acostamento ou presença da polícia rodoviária.

Por fim, foi desenvolvida uma aplicação web para o gerenciamento da solução mobile, permitindo o controle de usuários registrados e dos alertas cadastrados no sistema

[Repositório](#)

Tecnologia	Funcionalidade
------------	----------------

React Native	Biblioteca JavaScript utilizada no desenvolvimento do frontend da aplicação móvel, permitindo a criação de interfaces dinâmicas e componentizadas.
Python	Linguagem de programação utilizada no desenvolvimento do backend e na implementação das regras de negócio do sistema.
Node.js	Ambiente de execução JavaScript multiplataforma, utilizado para executar o código do servidor e suportar a aplicação web.
MongoDB	Banco de dados não relacional (NoSQL), utilizado para o armazenamento de dados relacionados a usuários, imóveis e alertas.
FastAPI	Framework utilizado para a criação da API RESTful, responsável pela comunicação entre frontend e backend.
Docker	Ferramenta utilizada para a containerização e o deploy da aplicação, garantindo padronização do ambiente de execução.
JavaScript	Linguagem de programação utilizada no desenvolvimento da aplicação web (painel administrativo), responsável pela lógica de interação e integração entre componentes.
TypeScript	Superset do JavaScript que adiciona tipagem estática e recursos de orientação a objetos, aumentando a robustez e a manutenção do código.
Git	Sistema de controle de versão distribuído, utilizado para o gerenciamento do código-fonte e colaboração em equipe.

Contribuições Pessoais

Atuando como Desenvolvedor, fui responsável pela criação e configuração dos submódulos Git, separando o projeto em dois repositórios distintos, um destinado ao frontend e outro ao backend. Em seguida, realizei a configuração inicial da API, instalando as bibliotecas necessárias ao desenvolvimento e registrando-as no arquivo `requirements.txt`. Também estruturei as pastas do projeto seguindo um padrão de arquitetura em camadas, adotando boas práticas de desenvolvimento, como os princípios de Clean Code.

Desenvolvi o código base para o funcionamento do FastAPI, incluindo a criação do arquivo `main.py` para inicialização do servidor e a configuração da conexão com o banco de dados. Para isso, utilizei o driver oficial do MongoDB para Python (PyMongo). Além disso, fui

responsável pela implementação da funcionalidade de geração e atualização do Plus Code (endereço digital) associado a uma propriedade, utilizando a biblioteca geoespacial Shapely. Também desenvolvi a rota HTTP responsável pela comunicação dessa funcionalidade com o frontend..

Com base nos dados fornecidos pela empresa parceira, criei um Jupyter Notebook no Google Colab para realizar a exploração inicial dos dados, analisando o tamanho da base e identificando possíveis duplicidades por meio da biblioteca Pandas. Após essa etapa, realizei a conversão dos arquivos para o formato JSON, viabilizando a construção da base de dados no MongoDB.

Também desenvolvi scripts para manipulação e padronização dos dados, criando schemas para a estrutura dos arquivos JSON. Além disso, implementei funções e rotas responsáveis pela seleção de propriedades com base no CPF do usuário ou no Plus Code associado ao imóvel.

Por fim, fui responsável pela documentação completa do backend, detalhando a estrutura do projeto, o fluxo do código e as instruções necessárias para execução da aplicação em ambiente local, permitindo que outros desenvolvedores possam clonar e utilizar o repositório corretamente.

Estrutura da api

```
├─ app/  
  │├─ core/  
  │├─ main.py  
  │├─ models/  
  │├─ repositories/  
  │├─ routes/  
  │├─ schemas/  
  │├─ services/  
  │├─ utils/  
  │├─ main.py  
├─ requirements.txt
```

Conexão com o banco de dados

```

from pymongo import AsyncMongoClient
import os
from dotenv import load_dotenv

load_dotenv()

MONGO_URL = os.environ.get('MONGO_URL')
DB_NAME = os.environ.get('DB_NAME')

client = AsyncMongoClient(MONGO_URL)

db = client[DB_NAME]

collection = db['area_imovel_projeto']
users_collection = db["users"]

async def ensure_indexes():
    await users_collection.create_index(
        "email",
        name="uniq_email",
        unique=True,
        partialFilterExpression={"email": {"$exists": True, "$type":
"string"}}
    )
    await users_collection.create_index(
        "cpf",
        name="uniq_cpf",
        unique=True,
        partialFilterExpression={"cpf": {"$exists": True, "$type":
"string"}}
    )

if __name__ == "__main__":
    print("Conexão com o Mongo feita com sucesso")

```

Função e rota do pluscode

```
def generate_plus_code(lat: float, long: float) -> str:
    try:
        propertie_pluscode = pluscodes.encode(lat, long)
        return propertie_pluscode
    except Exception as e:
        raise Exception(f"Erro ao gerar o pluscode: {e}")

@area_imovel_router.post('/properties/{cod_imovel}/pluscode')
async def add_pluscode(cod_imovel: str, request: CreatePlusCode):
    try:
        property_pluscode = await
add_properties_plus_code_service(cod_imovel, request)
        return {
            "message": "PlusCode adicionado com sucesso",
            "cod_imovel": cod_imovel,
            "result": property_pluscode
        }
    except Exception as e:
        raise HTTPException(status_code=400, detail=str(e))
```

Por fim, desenvolvi o portal do administrador, utilizando React para a criação das páginas de login, gerenciamento de usuários e administração de ocorrências, garantindo a integração com o backend desenvolvido.

Hard Skills

Tecnologia	Proficiência	Descrição
React	faço com ajuda	Desenvolvimento de página e componentes, além da integração com backend
Python	faço com autonomia	Desenvolvimento da api RESTful e scripts para comunicação com o banco de dados
Node.js	faço com autonomia	Utilizado como ambiente de execução do JavaScript no servidor

JavaScript	faço com autonomia	Desenvolvimento e integração de componentes
TypeScript	faço com autonomia	Criação de classes, tipagem estática e recursos de programação orientada a objetos
MongoDB	faço com ajuda	Modelagem e implementação do banco de dados não relacional, definição de regras de negócios ,criação de collections, documents e relações entre eles
Git	faço com autonomia	Controle de versão, trabalho em equipe com branches e gestão de repositórios.

Soft Skills

- **Comunicação:** *Durante o planejamento do projeto, identifiquei limitações técnicas do Django em relação à integração com o banco de dados não relacional MongoDB, o que poderia comprometer o andamento do desenvolvimento. Diante dessa situação, preparei uma argumentação técnica e a apresentei à equipe durante uma das reuniões, propondo a migração do framework para o FastAPI. Busquei expor os pontos de forma clara e objetiva, destacando os benefícios em termos de organização, escalabilidade e compatibilidade com o banco de dados, de modo a facilitar a compreensão de todos os membros e promover uma tomada de decisão coletiva e fundamentada.*
- **Proatividade:** *Diante da demanda de criação de um portal administrativo para a aplicação, tomei a iniciativa de antecipar o desenvolvimento do frontend antes mesmo da definição formal das tarefas pela equipe. Realizei pesquisas sobre estruturas adequadas para painéis administrativos e, com base nisso, dei início à construção da interface utilizando React, definindo a organização dos componentes, as rotas e a estrutura base do projeto. Essa ação antecipada permitiu que a equipe tivesse um ponto de partida consolidado, agilizando as entregas subsequentes da sprint.*

[Voltar](#)

Em 2026-1

Empresa: DomRock

Área de atuação: Inteligência Artificial

Problema:

No contexto corporativo, regras de negócio raramente permanecem estáticas. Elas são continuamente ajustadas em resposta a novas demandas do mercado, como alterações de precificação, lançamento ou retirada de produtos, reestruturação de acordos comerciais e revisões em políticas de vendas e parcerias.

Essa natureza dinâmica, quando não acompanhada de um registro adequado e organizado, resulta em fragilidades operacionais: inconsistências na execução dos processos, divergências entre os critérios adotados por diferentes colaboradores, dependência de conhecimento não documentado e comprometimento da rastreabilidade das decisões tomadas.

Solução:

Foi desenvolvida uma aplicação web voltada ao gerenciamento de regras de negócio. O sistema centraliza informações de lojas, marcas e funcionários, permitindo que o usuário crie novas regras por meio de linguagem natural. Esse processo consiste no envio da solicitação do usuário para uma API desenvolvida em Python com LangGraph, onde o agente de IA realiza o processamento e retorna a regra de negócio estruturada.

Cada regra criada é vinculada a uma loja, marca ou funcionário, e o cálculo de comissão passa a ser realizado com base nessa nova regra. Ao final do cálculo, são exibidas as regras utilizadas no processo, garantindo transparência na apuração dos valores.

Além disso, foi implementada a observabilidade do agente de IA por meio do MLflow, possibilitando o monitoramento da quantidade de tokens consumidos a cada interação com os modelos de linguagem, a avaliação da qualidade das respostas geradas, o versionamento de prompts e o controle de erros.

[Repositório](#)

Tecnologia	Funcionalidade
Vue.js	Framework JavaScript utilizado no desenvolvimento do frontend da aplicação, permitindo a criação de interfaces dinâmicas e componentizadas.

Python	Linguagem de programação utilizada no desenvolvimento do agente de IA, escolhida pela ampla compatibilidade com bibliotecas de inteligência artificial e processamento de linguagem natural.
Node.js	Ambiente de execução JavaScript multiplataforma, utilizado para executar o código do servidor e suportar a aplicação web.
MongoDB	Banco de dados não relacional (NoSQL), utilizado para o armazenamento de dados relacionados a funcionários, lojas, marcas e regras de negócio.
FastAPI	Framework Python utilizado para a criação da API RESTful, responsável pela comunicação entre o backend e o agente de IA.
Docker	Ferramenta utilizada para a containerização e o deploy da aplicação, garantindo a padronização do ambiente de execução.
Java	Linguagem de programação utilizada no desenvolvimento do backend, responsável pela implementação das regras de negócio e pela lógica de integração entre os módulos do sistema.
Spring Boot	Framework Java utilizado para a construção da API REST, facilitando a configuração e a estruturação dos serviços do backend.
MLflow	Plataforma utilizada para o versionamento de modelos e a observabilidade do agente de IA, permitindo o rastreamento de experimentos e métricas de desempenho.
LangGraph	Biblioteca utilizada para a orquestração do fluxo de execução do agente de IA, estruturando suas etapas em formato de grafo.
LangChain	Biblioteca utilizada para a integração do agente de IA com modelos de linguagem e ferramentas externas, viabilizando o processamento de linguagem natural.
Git	Sistema de controle de versão distribuído, utilizado para o gerenciamento do código-fonte e a colaboração entre os membros da equipe.

Contribuições Pessoais

Atuando como desenvolvedor, fui responsável pela criação de um submódulo Git destinado a isolar o repositório do agente de IA, seguido da configuração inicial do ambiente de desenvolvimento, incluindo a criação de um ambiente virtual com o uv e a instalação das

bibliotecas necessárias para a construção do agente.

Também realizei a criação dos embeddings e do banco de dados vetorial, estruturado em duas coleções: uma contendo os dados e as regras de negócio fornecidas pela empresa parceira, e outra com o código-base desenvolvido para o cálculo de comissão. Para a geração dos embeddings, utilizei o modelo nomic-embed-text:latest, executado por meio do Ollama. Além disso, desenvolvi o retriever, componente responsável por recuperar informações do banco vetorial com base na consulta do usuário, constituindo assim o módulo de RAG (Retrieval-Augmented Generation) do agente.

Realizei a integração com as APIs do Google, da Hugging Face e da Groq para utilização dos seguintes modelos de linguagem: Qwen3-Coder-30B-A3B-Instruct (Hugging Face), Gemini 2.0 Flash Preview (Google) e Meta-Llama/Llama-4-Scout-17B-16E-Instruct (Groq). Com esses modelos, utilizando o LangGraph, construí um grafo composto por três nós: Agent Node, Code Editor e Review Node. O fluxo tem como objetivo gerar um objeto JSON que será utilizado para calcular as novas comissões.

Agent Node

Responsável por orquestrar o fluxo do agente. Com base na consulta do usuário, aciona ferramentas (tools) para buscar informações no banco de dados vetorial. Para evitar chamadas excessivas, foi implementado um guardrail que limita o número de requisições a no máximo dez. Quando as informações necessárias para contextualizar a criação do JSON são obtidas, o fluxo avança para o nó Code Editor. Os modelos utilizados neste nó são o Gemini 2.0 Flash Preview e o Meta-Llama/Llama-4-Scout.

Code Editor

Utiliza o Qwen3-Coder-30B-A3B-Instruct como modelo de linguagem e é responsável por gerar o objeto JSON para o cálculo da comissão, seguindo um esquema pré-estabelecido que orienta o modelo na construção correta do objeto. Dois tipos de objeto podem ser gerados: Override, quando a solicitação altera percentuais de comissão por marca ou cargo (por exemplo, "Aumente a comissão da loja 22 em 10%"), e Intercorrência, quando a solicitação concede bônus ou ajustes a matrículas individuais (por exemplo, "Funcionário 10 recebe bônus de R\$500"). Após a criação do objeto, o fluxo avança para o Review Node.

Exemplo de esquema pré estabelecido

```

class IntercorrenciaSazonal(BaseModel):
    """Bônus ou ajuste sazonal vinculado a matrículas individuais."""

    matricula: str = Field(description="Matrícula do funcionário")
    tipo: Literal["bonus_fixo", "bonus_venda", "admissao_bonus",
"perc_bonus"] = Field(
        description="Tipo da intercorrência conforme
calcular_comissionamento()"
    )
    valor: float = Field(description="Valor em reais ou percentual
conforme o tipo")
    vigencia_inicio: date = Field(description="Início da vigência")
    vigencia_fim: date = Field(description="Fim da vigência
(inclusive)")

    @model_validator(mode="after")
    def validar_periodo(self) -> IntercorrenciaSazonal:
        if self.vigencia_fim < self.vigencia_inicio:
            raise ValueError(
                f"vigencia_fim ({self.vigencia_fim}) não pode ser
anterior "
                f"a vigencia_inicio ({self.vigencia_inicio})"
            )
        return self

    def esta_vigente(self, ano: int, mes: int) -> bool:
        primeiro_dia_mes = date(ano, mes, 1)
        ultimo_dia_mes = date(ano, mes, calendar.monthrange(ano, mes)
[1])

        return self.vigencia_inicio <= ultimo_dia_mes and
self.vigencia_fim >= primeiro_dia_mes

```

Exemplo de corpo de resposta do objeto

Override

```
{
  "tipo": "override",
  "justificativa": "<qual regra foi aplicada e por quê>",
  "override": {
    "descricao": "<texto legível>",
    "data_inicio": "YYYY-MM-DD",
    "data_fim": "YYYY-MM-DD",
    "perc_override": {"cod_marca,cod_cargo": percentual},
    "marca_override": {"cod_marca_origem": cod_marca_referencia},
    "perc_adicional": {"cod_marca,cod_cargo": valor_decimal}
  },
  "intercorrencias": null
}
```

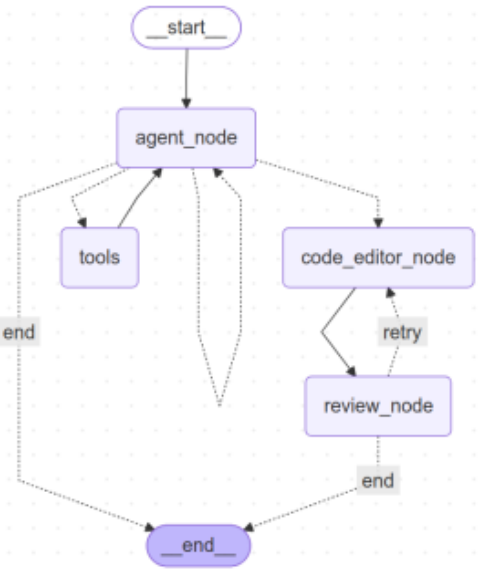
Intercorrência

```
{
  "tipo": "intercorrencia",
  "justificativa": "<qual regra foi aplicada e por quê>",
  "override": null,
  "intercorrencias": [
    {
      "matricula": "MATRIC-XXX",
      "tipo": "bonus_fixo",
      "valor": 500.0,
      "vigencia_inicio": "YYYY-MM-DD",
      "vigencia_fim": "YYYY-MM-DD"
    }
  ]
}
```

Review Node

Tem como objetivo verificar se o objeto JSON foi gerado corretamente conforme o esquema esperado, além de validar a regra de negócio quanto a erros e possíveis inconsistências. Caso o objeto esteja fora do padrão, o fluxo retorna ao Code Editor, repetindo o processo por no máximo dez iterações até que o JSON seja gerado adequadamente.

Fluxo Construído



Por fim, utilizando o MLflow, implementei a observabilidade do agente de IA, monitorando a quantidade de tokens consumidos em cada iteração do fluxo, a ocorrência de erros, o versionamento de prompts e a qualidade das respostas geradas pelo agente.

Hard Skills

Tecnologia	Proficiência	Descrição
LangChain	faço com autonomia	Desenvolvimento de página ecomponentes, além da integração com backend
LangGraph	faço com autonomia	Desenvolvimento de página ecomponentes, além da integração com backend
Python	faço com autonomia	Desenvolvimento do agente de IA
MLFlow	faço com autonomia	Integração do agente de IA com modelos de linguagem e ferramentas externas, viabilizando o processamento de linguagem natural.

Git	faço com autonomia	Controle de versão, trabalho em equipe com branches e gestão de repositórios.
------------	--------------------	---

Soft Skills

- **Pensamento analítico** *Pensamento analítico: Após a entrega da primeira sprint, analisei o comportamento do agente de IA e identifiquei uma limitação estrutural: até aquele momento, o agente operava de forma linear, executando etapas fixas e sequenciais, sem capacidade de adaptar seu fluxo conforme o contexto da solicitação. A partir dessa avaliação, concluí que a arquitetura baseada no LangChain não atenderia às necessidades de um agente mais iterativo e com maior capacidade de raciocínio. Com essa análise fundamentada, propus durante a sprint planning a migração do núcleo do agente para o LangGraph, que possibilita a construção de grafos com fluxos condicionais e iterativos. Essa mudança resultou em um agente mais robusto, capaz de tomar decisões dinâmicas ao longo de sua execução.*
- **Proatividade:** *Diante da necessidade de monitoramento do agente de IA, tomei a iniciativa de pesquisar e implementar a observabilidade do sistema antes mesmo da definição formal dessa tarefa pela equipe. Realizei estudos sobre ferramentas de rastreamento de modelos de linguagem e, com base nisso, dei início à integração do MLflow ao projeto, configurando o monitoramento de tokens consumidos, o versionamento de prompts e o registro de métricas de qualidade das respostas. Essa ação antecipada permitiu que a equipe dispusesse de uma estrutura de observabilidade funcional desde as primeiras iterações do agente, facilitando a identificação de falhas e a otimização contínua do fluxo.*

[Voltar](#)