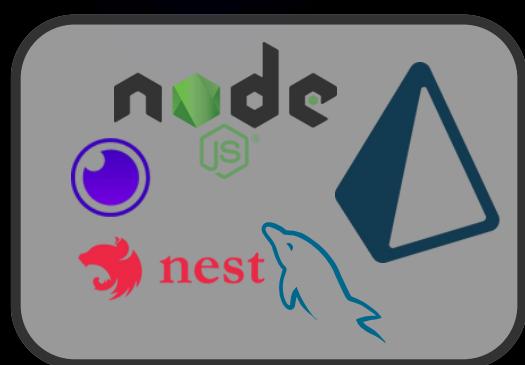




PORTFÓLIO - SISTEMA NEST.JS: API VAN ESCOLAR



POWERED BY:
JIVV SYSTEMS



História do Projeto

Durante a disciplina de Desenvolvimento Web (Backend), recebemos o desafio de implementar uma API REST completa para o mesmo projeto que havia sido prototipado em IHM. Este projeto representa um marco importante para a nossa jornada como desenvolvedores backend, combinando teoria e prática de forma integrada.

O empresário fictício **Sr. Lucas Matos**, dono da VanGo Carlos, precisava de uma solução rápida, prática e acessível para organizar o transporte escolar de seus alunos. Sua empresa havia crescido consideravelmente e a gestão manual dos alunos, rotas e pagamentos se tornara insustentável, gerando erros constantes e perda de eficiência operacional. Como o orçamento era limitado, ele contratou a **JJVV Systems**, uma pequena empresa americana especializada em prototipagem rápida e de baixo custo.

Essa decisão estratégica deixou sua marca no sistema: as rotas da API foram padronizadas em inglês para facilitar manutenção internacional, mas as mensagens e a documentação foram mantidas em português para melhor atender os usuários locais brasileiros.



□ Essa "dualidade" foi proposital, simulando a realidade de muitas empresas brasileiras que terceirizam desenvolvimento, mas precisam manter compatibilidade cultural e de linguagem com seus clientes finais.

Arquitetura do Sistema



O backend foi desenvolvido em [NestJS](#) com arquitetura modular, seguindo os princípios SOLID e padrões de design estabelecidos. Cada recurso possui seu próprio Controller, Service, Module e DTOs, garantindo isolamento de responsabilidades, testabilidade e manutenibilidade a longo prazo.

- **students**

Gestão completa dos alunos transportados

- **payments**

Controle de mensalidades e pagamentos

- **attendance**

Registro de chamadas de presença

- **routes**

Definição e gestão de rotas escolares

- **route-stops**

Paradas e pontos de embarque/desembarque

- **users**

Sistema de usuários administrativos

A persistência de dados foi implementada com **Prisma ORM**, conectado a um banco MySQL robusto, com migrations automáticas para versionamento seguro da estrutura de dados. O módulo database concentra toda a configuração do PrismaService, facilitando a injeção de dependência em toda a aplicação e garantindo conexões otimizadas.

Funcionalidades Implementadas



Alunos (/students)

CRUD completo com filtros avançados por nome, escola, status e poltrona. **Regra de negócio crítica:** não é permitido dois alunos ativos na mesma poltrona, garantindo controle de lotação e segurança no transporte.



Gestão de Rotas

Cadastro completo de rotas escolares
Associação inteligente a motoristas
Sincronização automática de paradas
Validação de orderIndex único por rota



Sistema de Autenticação

Login seguro com JWT
Perfil autenticado (/me)
Fluxo completo de recuperação de senha
Senhas criptografadas com bcrypt



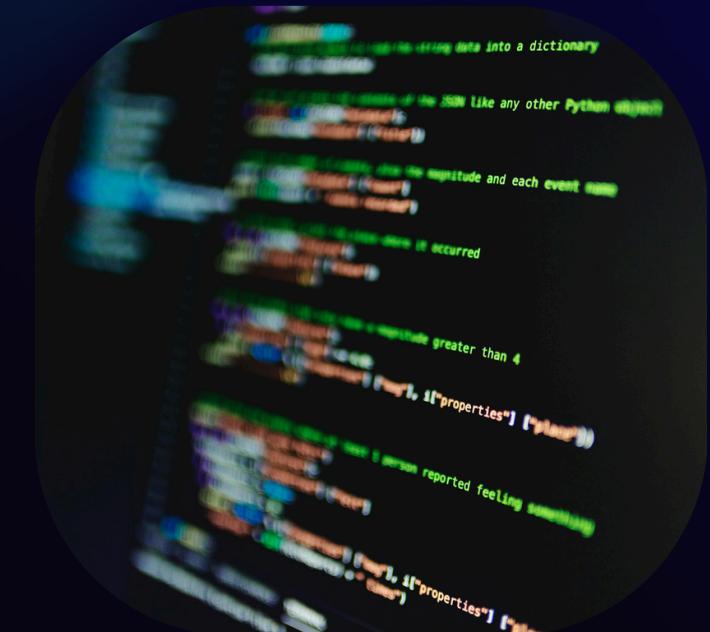
Pagamentos (/payments)

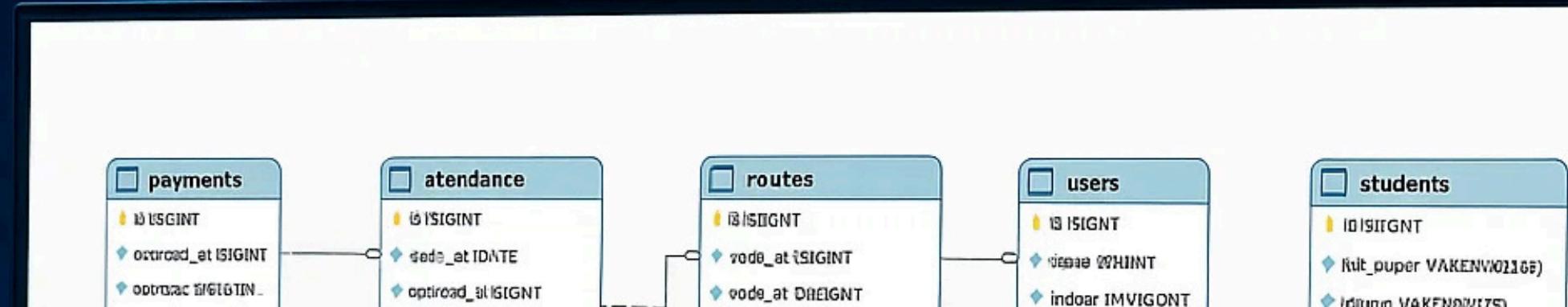
Sistema completo de gestão financeira com filtros por mês, status e aluno, além de Sistema completo de gestão financeira com filtros por mês, status e aluno, além de paginação otimizada. **Relacionamento direto** com os alunos para controle preciso de mensalidades e histórico financeiro.



Chamadas (/attendance)

Registro eficiente de presença por data + rota + aluno, permitindo controle operacional completo. **Chave composta** evita duplicidade de chamada no mesmo dia, garantindo integridade dos dados.





Banco de Dados e Relacionamentos

Tecnologias Utilizadas

SGBD: MySQL para robustez e performance

ORM: Prisma com migrations automáticas

Versionamento: Controle completo da estrutura

Destaque Técnico: Índices únicos estratégicos, chaves compostas e relacionamentos otimizados para máxima performance.



A modelagem do banco priorizou integridade referencial, performance de consultas e escalabilidade futura, com índices estratégicos em poltronas de alunos ativos, orderIndex de paradas por rota, e chave composta em chamadas (data, rota, aluno).



Documentação e Acessibilidade

Coleções Insomnia

Documentação interativa completa para todas as entidades: Alunos, Pagamentos, Chamadas, Rotas, Paradas, Usuários e Autenticação. Cada endpoint possui exemplos práticos e casos de uso reais.

Acesso ao Código

O repositório completo está disponível no GitHub com código organizado, comentado e seguindo padrões de mercado. A estrutura modular facilita contribuições e futuras expansões do sistema.

Repositório no Git Hub:

 Link

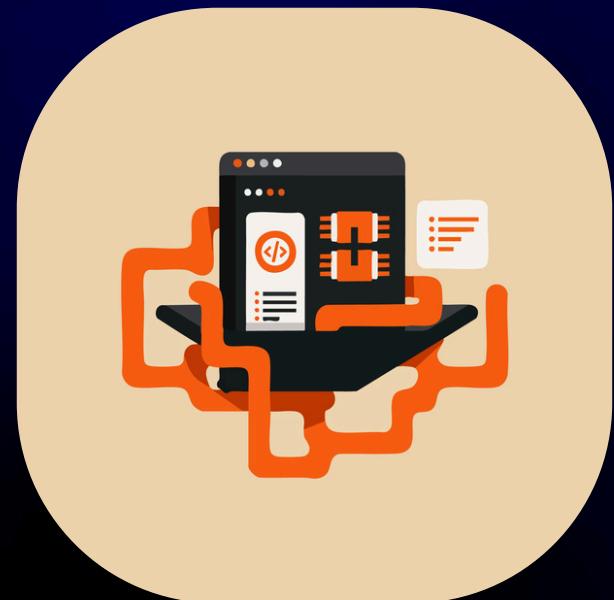
README Detalhado

Instruções passo-a-passo para instalação, configuração do .env, execução local e testes automatizados. Inclui troubleshooting comum e boas práticas de desenvolvimento.



Localização Brasileira

Todas as mensagens de resposta, validações e documentação mantidas em português, garantindo acessibilidade completa para o público final e facilitando manutenção por equipes locais.



Diferenciais Técnicos



Segurança Robusta

Autenticação JWT completa implementada com guards personalizados, middleware de validação e fluxo seguro de recuperação de senha. Senhas protegidas com bcrypt e tokens com expiração controlada.



Validações Rigorosas

Todos os DTOs implementados com class-validator, garantindo integridade de dados de entrada. Validações customizadas para regras de negócio específicas e mensagens de erro informativas.



Tratamento de Erros

Sistema padronizado de tratamento de exceções com mensagens claras em português. Logs estruturados e códigos HTTP apropriados para cada situação de erro.

Arquitetura Limpa

Organização modular seguindo princípios SOLID e boas práticas do NestJS. Separação clara de responsabilidades entre camadas.

Escalabilidade

Estrutura preparada para crescimento, com padrões que facilitam adição de novas funcionalidades e módulos sem impacto no código existente.

1



2

Storytelling Único

Narrativa diferenciada simulando outsourcing americano para justificar rotas em inglês e mensagens em português - refletindo realidade do mercado.

3



Créditos

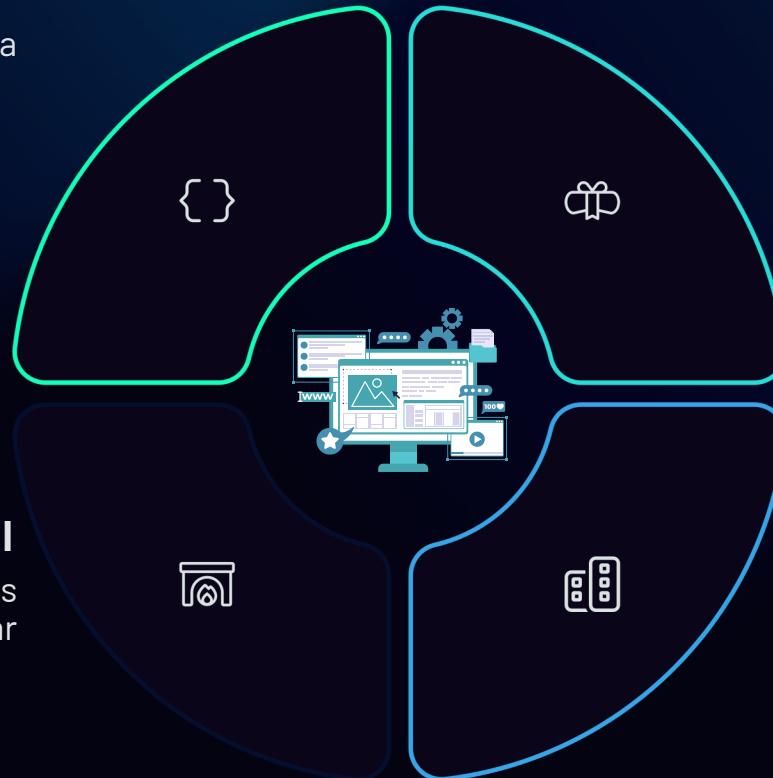
Desenvolvedores

Jonathan Weverton Rodrigues Batista

José Pedro Fernandes Pereira Abreu

Vinícius Soares Ferreira

Vitor Manoel Vidal Braz



Este projeto representa não apenas conhecimento técnico, mas também capacidade de entender requisitos de negócio, criar soluções práticas e entregar software de qualidade profissional.



Instituição: Univale

Disciplina: Desenvolvimento Web
(Backend)

Foco em NestJS e APIs REST

JJVV Systems

Empresa Americana fictícia especializada em prototipagem rápida. E soluções de baixo custo

Conclusão

O projeto [Van Escolar - API](#) demonstra minha evolução como desenvolvedor backend e capacidade de entregar soluções completas e robustas. Este sistema vai além de um simples exercício acadêmico - representa uma aplicação real que poderia ser implementada em produção.

Análise de Requisitos

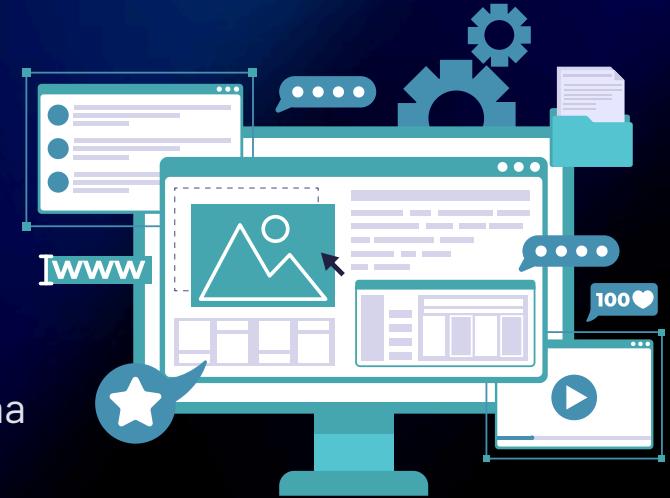
Compreensão profunda das necessidades do negócio, identificação de stakeholders e mapeamento de processos críticos para o transporte escolar.

Regras de Negócio

Implementação de lógicas complexas como controle de poltronas, validação de chamadas e gestão de pagamentos com precisão.

Este projeto consolidou minha expertise em NestJS e demonstrou que posso criar sistemas backend completos, desde a concepção até a entrega, sempre priorizando qualidade, segurança e experiência do usuário.

A combinação de tecnologias modernas (NestJS, Prisma, MySQL, JWT), arquitetura bem estruturada e storytelling envolvente fazem deste projeto um diferencial importante no meu portfólio como desenvolvedor backend.



Modelagem de Dados

Design inteligente do banco de dados com relacionamentos otimizados, índices estratégicos e integridade referencial garantida.

API REST Robusta

Entrega de uma interface bem documentada, segura e eficiente, seguindo padrões de mercado e boas práticas de desenvolvimento.

JJV SYSTEMS