



**Jonathan Weverton Rodrigues Batista**

**José Pedro Fernandes Pereira Abreu**

**Vinicius Soares Ferreira**

**Vitor Manoel Vidal Braz**

Van Escolar Backend — Documentação Oficial da API

Governador Valadares

2025

---

# Documento Oficial da API — *Van Escolar do Carlos Backend*

## 1. Introdução

A **API Van Escolar do Carlos** foi desenvolvida para atender às necessidades de gestão de transporte escolar do empresário fictício **Sr. Lucas Matos**.

Ela oferece funcionalidades de **cadastro de alunos, controle de rotas e paradas, chamadas de presença, gestão de pagamentos, autenticação de usuários e gerenciamento de motoristas**.

O sistema foi construído em **NestJS** com **Prisma ORM** e banco **MySQL**, seguindo arquitetura modular e boas práticas de validação e autenticação.

---

## 2. Arquitetura

A aplicação segue uma estrutura organizada em módulos independentes:

- **students** → Cadastro e gestão de alunos.
- **payments** → Controle financeiro de mensalidades.
- **attendance** → Registro de presença dos alunos.
- **routes** → Rotas de transporte.
- **route-stops** → Pontos/paradas de cada rota.
- **users** → Gestão de usuários do sistema.
- **auth** → Autenticação e autorização via JWT.

Módulo adicional:

- **database** → Concentra o PrismaService para conexão com MySQL.
- 

## 3. Funcionalidades Principais

### **Alunos (/students)**

- CRUD completo.
- Filtros por nome, escola, poltrona, status.
- Regra: apenas um aluno ativo por poltrona.

### **Pagamentos (/payments)**

- CRUD completo.
- Filtros por mês (YYYY-MM), status (aberto, pago, atrasado) e aluno.
- Relacionamento 1:N com alunos.

### **Chamadas (/attendance)**

- Registro de presença com chave composta (data + rota + aluno).
- Evita duplicidade de chamadas.

### **Rotas (/routes)**

- Cadastro de rotas.
- Associação a motoristas.
- Sincronização das paradas no PATCH.

### **Paradas de Rota (/route-stops)**

- CRUD de paradas.
- Validação de duplicidade de `orderIndex` por rota.

### **Usuários (/users)**

- Cadastro e gestão de usuários.
- Senhas criptografadas com **bcrypt**.

### **Autenticação (/auth)**

- Login e emissão de JWT.
- Consulta de perfil autenticado (/me).
- Recuperação de senha via token (/forgot-password + /reset-password).

---

## **4. Banco de Dados**

- **SGBD:** MySQL.
- **ORM:** Prisma.
- Relacionamentos:
  - 1:N → Aluno → Pagamentos
  - 1:N → Aluno → Chamadas
  - 1:N → Rota → Paradas
  - 1:1 → Rota → Motorista

Destaques técnicos:

- Índices únicos em poltrona ativa de aluno.
- Índices únicos em `orderIndex` de paradas.
- Chave composta em chamadas.

---

## 5. Guia de Execução

### Pré-requisitos

- Node.js 20+
- MySQL 8+
- NPM ou Yarn

### Passos

1. **Clonar o repositório**
  2. `git clone https://github.com/vitormanoelvb/backend-van-escolar-do-carlos.git`
  3. `cd backend-van-escolar-do-carlos`
  4. **Instalar dependências**
  5. `npm install`
  6. **Configurar variáveis de ambiente**  
Criar um arquivo `.env` na raiz com os parâmetros:
  7. `DATABASE_URL="mysql://usuario:senha@localhost:3306/van_escolar"`
  8. `JWT_SECRET="sua_chave_jwt"`
  9. `JWT_EXPIRES="7d"`
  10. **Rodar migrations**
  11. `npx prisma migrate dev`
  12. **Executar servidor em modo desenvolvimento**
  13. `npm run start:dev`
  14. **Acessar aplicação**  
A API estará disponível em:
  15. `http://localhost:3000`
- 

## 6. Testes e Documentação

- **Insomnia:** coleções exportadas para todas as entidades.
  - **Mensagens em português,** garantindo clareza para usuários finais.
  - **Rotas em inglês,** simulando outsourcing de empresa americana.
- 

## 7. Créditos

Desenvolvido por:

**Jonathan Weverton Rodrigues Batista**

**José Pedro Fernandes Pereira Abreu**

**Vinícius Soares Ferreira**

**Vitor Manoel Vidal Braz**

## **Repositório Oficial**

 <https://github.com/vitormanoelvb/backend-van-escolar-do-carlos/tree/main>

---