



# **API Escolar: Desenvolvimento do CRUD e TDD**



## Relatório

# 1

## API School System - Grupo 10

Ana Beatriz Silva Santos - 2401228

Luiz Otávio RA: 2401300

Murillo Rodrigues Santos Pereira - 2400338

Pablo Vavrik RA: 2400125

Uatila dos Santos Silva - 2400250

Data da atividade: 19/03/2025

### Objetivo

Desenvolvimento de uma API que visa integrar o acesso e gerenciamento do sistema escolar. Visando atender as necessidades essenciais de uma instituição de ensino. Com foco nas entidades alunos, professores e turmas.

### 1. Introdução

Essa primeira entrega implementa os primeiros processos do sistema (CRUD) e seus respectivos testes (TDD), a fim de garantir a total funcionalidade e fluxo dos métodos e funções implementadas na API. A API oferece total capacidade e robustez para crescimento futuro, visando o crescimento e implementação de novas funcionalidades futuras.

## **2. Descrição e Análise do Caso**

A implementação de uma API escolar com operações CRUD e desenvolvimento orientado por TDD transformou a gestão educacional. Com o CRUD, o sistema permite a criação, leitura, atualização e exclusão eficiente de dados como alunos, professores e turmas, centralizando a informação e agilizando processos. O uso do TDD assegura a confiabilidade do sistema, minimizando erros e falhas, enquanto a escalabilidade facilita a adaptação do sistema conforme o crescimento da escola. Além disso, a integração com outras plataformas, como pagamentos e ensino a distância, proporcionam um ambiente digital mais conectado e eficiente. Em resumo, a API escolar melhora a gestão administrativa, reduz custos e melhora a experiência de todos os envolvidos no processo educacional.

## **3. Implementação ou Procedimento**

A API se baseia no modelo CRUD (Create, Read, Update e Delete), onde foram desenvolvidas rotas e funções específicas para cada entidade (Aluno, Professor e Turma) visando atender o total gerenciamento dos mesmos pela instituição. Os métodos foram desenvolvidos baseando-se em uma série de 19 testes pré-definidos, desenvolvidos utilizando uma rigorosa metodologia, onde no início do projeto foi feito o estudo de caso e levantamento das necessidades do cliente. Os testes foram criados servindo como parâmetro para a criação do código principal e funcionamento da API. Após a implementação do código em Python, foram testados via Postman todos os endpoints e também foram testados com a biblioteca Unittest, com a execução do código, para passar nos testes. Os testes passaram e foi feita uma otimização e limpeza do código.

## **4. Resultados**

Todas as funcionalidades desenvolvidas apresentam 100% de êxito. O sistema atende, em sua plenitude, a todos os testes propostos, onde possíveis falhas e erros foram mitigados.

## **5. Conclusão**

A criação da API com Flask e Python, utilizando TDD, foi uma experiência valiosa que nos permitiu aplicar de forma prática os conceitos de testes e desenvolvimento orientado à testes. Ao longo do projeto, percebemos a importância de escrever código de forma estruturada e de garantir que cada funcionalidade atendessem aos requisitos do sistema escolar. A abordagem de TDD e a manipulação com a biblioteca Unittest, não só ajudou a identificar e corrigir falhas de maneira antecipada, mas também proporcionou mais confiança na robustez da API ao passar em todos os testes. Esse processo fortaleceu meu entendimento sobre boas práticas de desenvolvimento e a importância da qualidade no código.

## **6. Impacto e Conexão com o Mundo real**

Uma API para gerenciamento de sistemas escolares melhora a eficiência administrativa e a confiabilidade dos dados. O CRUD permite criar, ler, atualizar e excluir informações rapidamente, centralizando o controle sobre alunos, professores, turmas e notas. A metodologia de testes TDD proposta garante que o sistema seja bem testado e confiável, minimizando falhas.

A API facilita a escalabilidade e integração com outras plataformas, como sistemas de pagamento e ensino a distância. Além disso, reduz custos operacionais ao automatizar processos e eliminar erros manuais. Assim, as escolas ganham em agilidade, precisão e sustentabilidade, proporcionando uma melhor experiência para alunos, pais e administradores.

## **7. Desafios Futuros e Melhorias**

Entrega do projeto Flask com MVC, Banco de Dados e Microserviço completo. Além de poder abranger mais entidades e criar um sistema escolar mais completo.