# API Escolar:
# Entrega 2: Implementação do padrão de arquitetura MVC (Model-View-Controller)

Faculdade IMPACTA

Faculdade IMPACTA

# Relatório 2

# 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: 10/04/2025

**Objetivo**

Implementação do modelo MVC (Model-View-Controller), com objetivo de separar as responsabilidades para melhorar a organização, facilitar manutenção e escalabilidade do código.

## 1. Descrição e Análise do Caso

Analisamos o código da Api referente a primeira entrega, a fim de identificar quais pontos seriam necessários para implementação do padrão de arquitetura MVC. A partir dos pontos levantados, entendemos que era necessário refatorar o código para atender aos requisitos do modelo MVC.

## 2. Implementação ou Procedimento

Após a fase de análise e estudos prévios, identificamos a necessidade de aprimorar a estrutura da aplicação com base nos princípios do padrão de

arquitetura MVC (Model-View-Controller). A principal ação foi a separação clara das responsabilidades entre as camadas, o que contribui diretamente para melhor organização do código, maior facilidade de manutenção, aumento da testabilidade e maior potencial de reutilização dos componentes.

Cada entidade (Alunos, Professores e Turmas) da aplicação passou a contar com seus respectivos arquivos de Model e Controller, alocados em diretórios específicos de acordo com a nova estrutura. Além disso, foi adotado o uso de Blueprints, recurso do Flask que permite dividir a aplicação em módulos menores e independentes. Essa abordagem favoreceu a modularização, facilitando o registro de rotas e a organização geral da API.

Os testes também foram reorganizados para acompanhar essa reformulação, garantindo compatibilidade com a nova arquitetura da aplicação. Essa refatoração representa um avanço significativo na escalabilidade do sistema, permitindo uma base sólida para a futura implementação de novas funcionalidades, como a integração com o banco de dados e o aprimoramento das regras de negócio.

## 3. Resultados

Com a aplicação da estrutura MVC, alcançamos um nível muito mais alto de organização e clareza no projeto. Essa mudança trouxe mais confiança no funcionamento da aplicação e abriu espaço para que ela cresça de forma mais estável no futuro.

Um dos principais ganhos foi a melhoria na legibilidade do código. Agora, com a separação das responsabilidades e a estrutura mais limpa, o projeto se tornou mais acessível para manutenção e colaboração. Isso facilita bastante a entrada de novos desenvolvedores na equipe, que poderão entender e contribuir com o projeto de forma mais rápida e eficiente.

## 4. Conclusão

A construção da API de cadastro escolar representou um passo importante no desenvolvimento de um sistema mais organizado, funcional e preparado para crescer. A decisão de adotar o padrão MVC trouxe ganhos reais para a estrutura do projeto.

Com a aplicação desse modelo, conseguimos separar as responsabilidades do código, tornando o sistema mais limpo e intuitivo. Essa melhoria não só beneficia quem já está envolvido no desenvolvimento, mas também deixa o projeto acessível para quem for trabalhar com ele no futuro.

De forma geral, os resultados foram muito positivos e reforçam a importância de investir em boas práticas de arquitetura desde as etapas iniciais. Agora, com uma base sólida, seguimos mais confiantes para os próximos desafios e evoluções da API.

## 5. 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.

## 6. Desafios Futuros e Melhorias

Desenvolver o Entregável 3, atendendo a implementação do Docker, Banco de dados e Swagger .