

## Relatório

# 2

## Projeto Flask: Construindo Padrão MVC.

Grupo 4: Alan Araújo da Silveira, Ana Cláudia Monteiro Misquita, Fernanda Aparecida Figueiredo da Silva, João Pedro Silva Antunes, Matheus Barros Ferreira

**Data da atividade: 10/04/2025**

### Objetivo

Desenvolvimento de uma API (Application Programming Interface) com utilização do framework Flask, para um sistema escolar, com funcionalidades para o gerenciamento de informações de alunos, turmas e professores. O sistema tem como objetivo fornecer uma interface eficiente para a criação, leitura, atualização e exclusão de dados (CRUD), permitindo uma administração mais ágil e precisa das informações escolares. Além disso, a API foi construída com a prática de Test-Driven

Development (TDD), garantindo a qualidade e confiabilidade do código desde as primeiras fases de desenvolvimento. Na segunda entrega, foi estabelecido e implementado o padrão MVC (Model - View - Controller), que organiza o código em camadas, facilitando a manutenção, a reutilização e o crescimento do projeto. Ele separa responsabilidades, deixando o desenvolvimento mais claro, eficiente e colaborativo.

## **2. Introdução**

A API será desenvolvida com foco na integração de sistemas no contexto educacional, para organizar e acessar informações fundamentais a fim de melhorar a gestão e a comunicação entre alunos, professores e administradores. No segundo momento, o grupo realizou a implementação do MVC. Para a criação das camadas, foi necessário refatorar o código para que se encaixasse ao novo padrão de arquitetura de software, camadas essas que foram, Model - responsável por: Acessar o banco de dados, validar regras, salvar e manipular dados, e, Controller - responsável por: Receber ações do usuário (como cliques ou requisições), chamar o Model para processar dados e passar o resultado para View (a camada View ainda não foi implementada para esta entrega).

## **3. Descrição e Análise do Caso**

Inicialmente, quando nos foi apresentado detalhes de como seriam construídas as camadas, imaginamos que não haveriam tantas dificuldades, visto que, a construção não exigia muitas alterações no ato de refatorar, no entanto, houveram alguns obstáculos enfrentados que tornaram o processo um tanto desafiador, mas que superados após muita análise, estudo, e algumas tentativas.

## **4. Implementação ou Procedimento**

A integrante Ana iniciou o processo de criação das camadas de Model alunos, professores e turmas, que resumidamente, somente foi necessário adequar os códigos feitos na primeira entrega, para um outro formato. O processo foi bem simples ao passar os arquivos para Model e realizar as devidas modificações, a integrante Ana explicou que a parte mais dificultosa, foi dar o git push para a branch correta. O integrante Alan auxiliou a Ana no desenvolvimento e procedimento de envio de arquivos para o Github, ao ajudar no desenvolvimento, notou-se alguns problemas para passar nos testes, pois os dicionários utilizados como Banco de Dados não estavam devidamente organizados. Para contornar o problema, foi criado exclusivamente um arquivo de Banco de Dados, para organizar um dicionário com todos os dados do sistema, assim, passamos a primeira camada: Models. Após isso, partimos para a criação da segunda fase: Controllers. Nesse momento todos os integrantes do grupo tiveram problemas sérios de comunicação (que após algumas conversas, foi declarado ter sido a maior adversidade durante o projeto como um todo), e por isso, a integrante Fernanda fez a criação da camada Controllers e após finalizado e enviado, foi informada que o integrante João Pedro, também havia realizado a criação, gerando redundância no código. O dia seguinte foi marcado por decisões de qual Controller seria usado para compor o entregável 2, assim, foi decidido utilizar o criado pelo integrante João Pedro, pois o seu estava com menos erros ao passar pelos testes, gerando uma facilidade maior ao fazer a correção dos mesmos. Ao comentar sobre suas realizações e dificuldades no entregável 2, João Pedro, explica que corrigiu problemas que estavam acontecendo no método de PUT e DELETE, para que todo o ecossistema funcionasse da maneira correta. Também informou que em seus commits mais recentes, realizou ajustes no formato dos ID's, estruturando de maneira mais uniformizada, garantindo assim, identificadores mais funcionais para diferentes cenários do projeto.

## **5. Resultados**

Apesar dos conflitos e dos obstáculos enfrentados, os objetivos finais do grupo foram alcançados, e as camadas foram estruturadas no Padrão MVC de Arquitetura de Software de acordo com o que foi solicitado, garantindo a efetivação dos princípios que geram fácil manutenção, escalabilidade, e organização.

## **6. Conclusão**

A refatoração do projeto não foi tão simples como eram as nossas expectativas, afinal, estamos criando um projeto com estruturas de arquitetura, que contribuirá para um projeto sofisticado e duradouro ao final. No entanto, é importante reconhecer que acertar faz nós nos sentirmos bem, mas os erros que nos fazem ter um crescimento mais consistente e coerente, que irá acrescentar à vida profissional e pessoal, reforçando a nós mesmos que o conhecimento que nos leva cada vez mais perto do sucesso de cada um.

## **7. Impacto e Conexão com o Mundo Real**

Esse padrão de arquitetura tem tido um impacto enorme nos últimos tempos, por ser uma maneira de criar aplicações mais eficientes, modularizadas e mais fáceis de se manter ativos. No mundo real o MVC vai poder permitir reutilizar códigos entre aplicações, podendo economizar tempo e recursos, o MVC irá ajudar a reduzir a complexidade dos códigos usados e torná-los mais modularizados ajudando na sua manutenção. O MVC também tem a facilidade de realizar testes e em desenvolvimento, simplifica o teste de objetos-não-visuais, ou seja, não relacionados com a implementação das interfaces gráficas e com o desenvolvimento.

## **8. Desafios Futuros e Melhorias**

Acredita-se que o código é uma estrutura viva, e assim como tudo que é vivo, precisa ser constantemente transformada e melhorada. Por isso, recorrentemente, são feitas buscas para a otimização do desempenho dos testes, funções, Models, Controllers, mais para frente também será implementado a camada de Views, e assim por diante. Mas além disso, é de suma importância que a comunicação entre todos os integrantes dos grupos seja mais clara, e objetiva, visando o crescimento e bem estar de todos, logo, haverão discussões de alternativas para que as futuras entregas sejam feitas da forma mais transparente possível.