

Implementação com MongoDB e FastAPI + Motor

QXD0099 - Desenvolvimento de Software para Persistência

Universidade Federal do Ceará - *Campus* Quixadá

Prof. Francisco Victor da Silva Pinheiro
victorpinheiro@ufc.br



Agenda

- Tecnologias utilizadas
- Entidades e Relacionamentos
- Funcionalidades Implementadas
- Estrutura do Projeto
- Projeto

Tecnologias Utilizadas

- ✓ **FastAPI** → Framework para APIs rápidas e performáticas
- ✓ **MongoDB** → Banco de dados NoSQL
- ✓ **Motor** → Driver assíncrono para MongoDB
- ✓ **Pydantic** → Validação e serialização de dados
- ✓ **Uvicorn** → Servidor ASGI para execução da API

Entidades e Relacionamentos

- **Professor**

- Atributos: id, nome, especialidade, email
- Relacionamento:
 - 1:1 com Curso (um professor pode estar vinculado a apenas um curso)

- **Curso**

- Atributos: id, nome, descricao, carga_horaria, professor_id, alunos
- Relacionamentos:
 - 1:1 com Professor (um curso tem um professor responsável)
 - 1:N com Alunos (um curso pode ter vários alunos)

Entidades e Relacionamentos

- **Aluno**
 - Atributos: id, nome, email, idade, cursos
 - Relacionamento:
 - N:N com Curso (um aluno pode estar matriculado em vários cursos)
- **Turma**
 - Atributos: id, nome, curso_id, alunos
 - Relacionamento:
 - 1:N com Curso (uma turma pertence a um curso)
 - N:N com Alunos (uma turma pode ter vários alunos e um aluno pode estar em várias turmas)

Entidades e Relacionamentos

- **Departamento**
 - Atributos: id, nome, chefe_id, cursos
 - Relacionamento:
 - 1:N com Cursos (um departamento pode oferecer vários cursos)
 - 1:1 com Professor (um professor pode ser chefe do departamento)

Funcionalidades Implementadas

- **CRUD Completo**
 - Criar, listar, atualizar e deletar para Professor, Curso, Aluno, Turma e Departamento.
- **Relacionamentos e Matrículas**
 - Matrícula de Aluno em Curso: permite associar alunos a cursos e vice-versa.
 - Listar Alunos de um Curso: retorna detalhes completos dos alunos matriculados.
- **Paginação e Listagem**
 - Listagem de cursos, alunos e professores com paginação.

Funcionalidades Implementadas






- **Consultas e Filtros**

- Busca de cursos por nome usando regex.
- Busca de alunos e cursos por ID com validação de ObjectId.
- Contagem de alunos por curso.

- **Consultas Complexas**

- Listagem de alunos por curso com lookup no MongoDB.
- Contagem total de alunos por curso utilizando agregação (\$unwind, \$group).

Estrutura do Projeto

-  database/ → Configuração da conexão com o **MongoDB**.
-  models/ → Definição das **entidades** usando **Pydantic**.
-  routes/ → **Rotas da API** para cada entidade e funcionalidades específicas.
-  main.py → Arquivo principal que **inicializa a API** com **FastAPI**.
-  .env → Configuração de **variáveis de ambiente** para conectar ao banco de dados.

Projeto

- O projeto implementa um sistema completo de gerenciamento acadêmico, garantindo:
 - Modelagem eficiente no MongoDB com relacionamentos 1:1, 1:N e N:N.
 - API otimizada com FastAPI e validações robustas.
 - Consultas avançadas com agregações e filtros personalizados.

Referências

- MongoDB – Site oficial
 - <http://www.mongodb.com>
- MongoDB Manual
 - <http://docs.mongodb.org/manual/>
 - <http://docs.mongodb.org/manual/MongoDBmanual.pdf>
- Slides: Building your first app: an introduction to MongoDB. Norman Graham. Consulting Engineer, 10gen.
- Slides: mongoDB.
 - Júlio Monteiro (julio@monteiro.eti.br).
- Slides Why MongoDB Is Awesome
 - John Nunemaker - Ordered List (john@orderedlist.com)



Obrigado!

Dúvidas?



Universidade Federal do Ceará - *Campus* Quixadá

Prof. Francisco Victor da Silva Pinheiro
victorpinheiro@ufc.br

