

Trabalho de Férias – Desenvolvimento de API REST para Gerenciamento de Cursos Online

Objetivo O objetivo deste trabalho é desenvolver uma API RESTful utilizando Django e Django REST Framework. Você irá criar um sistema capaz de gerenciar cursos online, módulos de estudo, alunos e avaliações, exercitando conceitos de backend, autenticação, permissões, relacionamento entre modelos e endpoints customizados.

Descrição do Sistema O sistema deverá conter:

1. Usuários - Perfis: - Aluno: pode se inscrever em cursos, acessar módulos e avaliar cursos. - Professor: pode criar cursos e módulos, além de visualizar alunos inscritos. - Informações básicas: nome completo, e-mail, senha e uma bio opcional.

2. Cursos - Cada curso deve ter: - Título - Descrição - Professor responsável - Data de criação - Apenas professores podem criar cursos. - Cada curso pode conter vários módulos.

3. Módulos - Cada curso é dividido em módulos. - Cada módulo deve ter: - Título - Conteúdo (texto, vídeo ou link) - Ordem (posição do módulo dentro do curso) - Apenas professores podem criar ou editar módulos. - Alunos inscritos podem visualizar os módulos do curso.

4. Inscrição de Alunos - Um aluno pode se inscrever em vários cursos. - Um curso pode ter vários alunos inscritos. - Cada inscrição deve registrar a data em que ocorreu. - Professores podem visualizar a lista de alunos de seus cursos.

5. Avaliações - Alunos podem avaliar cursos nos quais estão inscritos. - Cada avaliação deve conter: - Nota (por exemplo, de 1 a 5) - Comentário (opcional) - Data da avaliação - Apenas alunos inscritos podem avaliar. - É possível calcular a média de avaliações de cada curso.

6. Endpoints REST Recomendados | Endpoint | Função | |-----|-----| | /users/me/ | Retorna informações do usuário logado. | | /courses/ | Listar todos os cursos. Professores podem criar novos cursos. | | /courses/<id>/modules/ | Listar módulos de um curso específico. | | /courses/<id>/enroll/ | Permite que um aluno se inscreva em um curso. | | /courses/<id>/reviews/ | Permite que um aluno avalie um curso. | | /courses/popular/ | Retorna os cursos mais bem avaliados. | | /courses/<id>/progress/ | Retorna o progresso do aluno em um curso. |

7. Regras e Permissões - Somente professores podem criar ou editar cursos e módulos. - Somente alunos inscritos podem acessar módulos e criar avaliações. - Usuários não autenticados não podem acessar nenhuma funcionalidade. - Evite duplicação de inscrições e avaliações.

8. Funcionalidades Extras (Opcional) - Buscar cursos por palavras-chave (título ou descrição). - Dashboard para professores mostrando número de alunos por curso. - Ordenação automática de módulos ao criar novos. - Paginação e filtros para cursos, módulos e avaliações.

Instruções para Desenvolvimento 1. Crie a aplicação somente em Django e Django REST Framework, sem front-end. 2. Organize o projeto com apps separados para usuários, cursos e módulos. 3. Garanta que a API seja testável via Postman ou Insomnia. 4. Documente todos os endpoints, incluindo: - Método HTTP - URL - Parâmetros esperados - Exemplo de resposta 5. Faça commits regulares usando Git para acompanhar a evolução do projeto.

Critérios de Avaliação | Item | Peso | ----- | ----- | Estrutura de models e relacionamentos | 25% | | Endpoints e lógica REST | 25% | | Autenticação e permissões | 20% | | Funcionalidades extras e custom endpoints | 20% | | Organização do projeto e documentação | 10% |

Observações Finais - Este projeto é uma oportunidade para aplicar conceitos de backend de forma prática. - É recomendado planejar a estrutura dos modelos antes de começar a criar endpoints. - A documentação dos endpoints é tão importante quanto a implementação.