TRABALHO PRÁTICO A1
ADS28 – CONSTRUÇÃO DE BACKEND
Profº Gustavo Clay

Objetivo

Desenvolver uma API REST utilizando Node.js e Express, aplicando os conceitos de API Rest, rotas, métodos HTTP, middlewares, versionamento com Git e documentação básica. O trabalho visa consolidar os conhecimentos sobre construção de backend, colaboração em equipe e uso de ferramentas de desenvolvimento.

Formação dos Grupos

- Grupos de 3 a 5 alunos.
- Cada grupo deve criar um repositório único no GitHub.
- Todos os integrantes devem contribuir ativamente para o projeto.

Requisitos Técnicos

1. Estrutura da API

- Criar 5 CRUDs completos, cada um representando um recurso distinto, seguindo rigorosamente as convenções REST:
    - Métodos HTTP corretos para cada operação (GET, POST, PUT, DELETE)
    - URLs semânticas (ex: /produtos, /produtos/:id)
    - Status codes adequados (200, 201, 404, 400, etc.)
    - Requisições e Respostas estruturadas em JSON
- Não é necessário usar banco de dados. Os dados podem ser armazenados em arrays/objetos em memória.
- Validação básica de campos obrigatórios (ex: retornar 400 Bad Request se campo obrigatório estiver faltando).
- Cada entidade deve ter campos diferentes para não caracterizar repetição.

2. Tema Livre

- Cada grupo deverá escolher um domínio de aplicação para modelar seus recursos, desde que permita a criação de pelo menos 5 entidades relacionadas logicamente.

3. Collection no Postman

- Criar uma collection organizada com todas as rotas implementadas.
- Cada requisição deve conter exemplos de corpo (body) e headers quando necessário.
- Exportar a collection e incluir o arquivo .json no repositório.
    - Clique nos três pontos (⋮) ao lado do nome da collection.
    - Selecione More > Export.
    - Selecione o formato Collection v2.1.
    - Clique em Export JSON e salve o arquivo .json.

4. Documentação no README.md

O README deve conter:

- Nome do projeto e descrição breve
- Instruções para instalação e execução
- Lista de endpoints com exemplos de requisição/resposta
- Nome dos integrantes, nome dos usuários do GITHUB e breve descrição das contribuições de cada membro

5. Uso do GitHub

Criar Issues detalhadas para cada etapa:

- Configuração inicial do projeto
- Implementação de cada CRUD
- Criação da collection no Postman
- Documentação
  Atribuir issues aos integrantes e descrever o que foi feito em cada uma

Entrega

Data limite: **03/10**
Forma de entrega: **Enviar o link do repositório GitHub no Classroom e informar nos comentários o nome dos membros e matrícula.**

Critérios de Avaliação

- **Nota: 3,0 pontos**
- Peso: 300 pontos

A pontuação de cada item será baseada nos critérios descritos na rubrica de avaliação.

Critério	**Pontuação Máxima (Peso)**	**Valor na Nota**
Implementação dos CRUDs	120 pontos	1,2 pontos
Uso do GitHub (issues e colaboração)	100 pontos	1,0 ponto
Collection no Postman	60 pontos	0,6 pontos
Documentação no README.md	20 pontos	0,2 pontos
Pontuação Total	**300 pontos**	**3,0 pontos**

RUBRICA DE AVALIAÇÃO - TRABALHO PRÁTICO A1 - ADS28 – CONSTRUÇÃO DE BACKEND - Profº Gustavo Clay						
Critério	**Excelente (100%)**	**Bom (75%)**	**Regular (50%)**	**Fraco (25%)**	**Nulo (0%)**	**Peso (pts)**
1. Implementação CRUD	5 CRUDs completos funcionando perfeitamente. Todos os métodos HTTP corretos, status codes adequados, URLs semânticas, validação básica implementada. Estrutura JSON consistente.	4-5 CRUDs funcionando com pequenos erros em status codes ou validações. URLs semânticas aplicadas na maioria dos casos.	3 CRUDs implementados ou vários métodos faltando. Erros frequentes em status codes ou estrutura de respostas.	1-2 CRUDs implementados com erros graves. URLs não semânticas, status codes inadequados.	Nenhum CRUD funcional ou projeto não entregue.	**120**
2. Uso do GitHub (issues e colaboração)	Issues detalhadas para todas as etapas (projeto + 5 CRUDs + Postman + docs), atribuição clara aos integrantes, descrição completa das contribuições. Evidência de colaboração equilibrada.	Issues criadas para maioria das etapas, mas com descrição superficial. Atribuição presente mas não totalmente clara.	Issues mínimas (apenas 2-3), descrição breve, pouca evidência de divisão de trabalho.	1 issue geral ou issues sem atribuição. Quase nenhum registro de colaboração.	GitHub não usado ou sem issues.	**100**
3. Collection no Postman	Collection completa com todas as 25+ rotas (5 CRUDs), exemplos de body/headers, organização clara por recurso, exportação correta.	Collection com 20+ rotas, mas falta alguns exemplos ou organização.	Collection com 10-15 rotas, exemplos incompletos ou com erros.	Collection com menos de 10 rotas, muito básica ou desorganizada.	Collection não entregue ou vazia.	**60**
4.Documentação no README.md	README completo: descrição clara, instruções de instalação, lista de endpoints com exemplos, integrantes com GitHub e contribuições detalhadas.	README com todas seções, mas falta clareza ou exemplos completos. Integrantes listados mas contribuições genéricas.	README muito resumido, falta informações importantes ou exemplos.	README confuso ou incompleto (apenas título e integrantes).	Sem documentação ou README em branco.	**20**