

# Exercício: Criação de uma API Simples e Cliente para Gestão de Cursos

Neste exercício, você criará uma API RESTful para gerenciar informações de cursos em uma plataforma educacional. Além disso, será necessário implementar um cliente simples para consumir essa API.

---

## Requisitos da API

Sua API deve ter os seguintes endpoints:

- Listar todos os cursos**  
Método: **GET**  
Rota: **/cursos**  
Descrição: Retorna uma lista de todos os cursos disponíveis.
  - Obter detalhes de um curso específico**  
Método: **GET**  
Rota: **/cursos/{id}**  
Descrição: Retorna os detalhes de um curso com base no ID fornecido.
  - Criar um novo curso**  
Método: **POST**  
Rota: **/cursos**  
Descrição: Adiciona um novo curso. O corpo da requisição deve conter informações como título, descrição e carga horária.
  - Atualizar um curso existente**  
Método: **PUT**  
Rota: **/cursos/{id}**  
Descrição: Atualiza as informações de um curso existente com base no ID.
  - Excluir um curso**  
Método: **DELETE**  
Rota: **/cursos/{id}**  
Descrição: Remove um curso com base no ID.
- 

## Requisitos do Cliente

Crie um cliente simples que permita interagir com a API. O cliente deve ser capaz de:

- Listar todos os cursos cadastrados.
- Consultar os detalhes de um curso específico pelo ID.
- Adicionar um novo curso fornecendo as informações necessárias.
- Atualizar os dados de um curso existente.

5. Excluir um curso.
- 

## Tecnologias Recomendadas

- **Backend (API):**
  - Linguagem: Python (com Flask ou FastAPI), Node.js (com Express), ou outra linguagem de sua escolha.
  - Banco de dados: SQLite (opcional) ou em memória para simplificação.
- **Cliente:**
  - Linguagem: Python (usando `requests`), JavaScript (usando `fetch` ou bibliotecas como `axios`), ou outra linguagem de sua escolha.

Aqui está um exercício completo de programação com o tema relacionado à educação.

---

## Exercício: Criação de uma API Simples e Cliente para Gestão de Cursos

Neste exercício, você criará uma API RESTful para gerenciar informações de cursos em uma plataforma educacional. Além disso, será necessário implementar um cliente simples para consumir essa API.

---

### Requisitos da API

Sua API deve ter os seguintes endpoints:

1. **Listar todos os cursos**  
Método: `GET`  
Rota: `/cursos`  
Descrição: Retorna uma lista de todos os cursos disponíveis.
2. **Obter detalhes de um curso específico**  
Método: `GET`  
Rota: `/cursos/{id}`  
Descrição: Retorna os detalhes de um curso com base no ID fornecido.
3. **Criar um novo curso**  
Método: `POST`  
Rota: `/cursos`  
Descrição: Adiciona um novo curso. O corpo da requisição deve conter informações como título, descrição e carga horária.
4. **Atualizar um curso existente**  
Método: `PUT`

Rota: `/cursos/{id}`

Descrição: Atualiza as informações de um curso existente com base no ID.

5. **Excluir um curso**

Método: `DELETE`

Rota: `/cursos/{id}`

Descrição: Remove um curso com base no ID.

---

## Requisitos do Cliente

Crie um cliente simples que permita interagir com a API. O cliente deve ser capaz de:

1. Listar todos os cursos cadastrados.
  2. Consultar os detalhes de um curso específico pelo ID.
  3. Adicionar um novo curso fornecendo as informações necessárias.
  4. Atualizar os dados de um curso existente.
  5. Excluir um curso.
- 

## Tecnologias Recomendadas

- **Backend (API):**
    - Linguagem: Python (com Flask ou FastAPI), Node.js (com Express), ou outra linguagem de sua escolha.
    - Banco de dados: SQLite (opcional) ou em memória para simplificação.
  - **Cliente:**
    - Linguagem: Python (usando `requests`), JavaScript (usando `fetch` ou bibliotecas como `axios`), ou outra linguagem de sua escolha.
- 

## CrITÉrios de Avaliação

- A API deve estar funcional e retornar respostas apropriadas para cada endpoint.
  - O cliente deve consumir corretamente os métodos da API.
  - O código deve ser organizado e legível.
  - É desejável adicionar tratamento de erros no cliente e na API.
- 

## Como enviar

- Envie o link para o seu código no GITHUB na tarefa correspondente do SIGAA.