**Passo a Passo para a Adaptação**

1. **Modelo de Dados (Model):**
   * No arquivo do modelo (userModel.js), vamos criar um novo arquivo chamado alunoModel.js.
   * Alteramos as consultas SQL e os parâmetros para refletir os campos da nova tabela (ALUNOS).
2. **Controlador (Controller):**
   * Copiaremos userController.js e criaremos um novo controlador, alunoController.js.
   * Neste arquivo, ajustaremos as funções para operar sobre a tabela ALUNOS, considerando as colunas RM, NOME, IDADE e TURMA.
3. **Rotas (Routes):**
   * No arquivo userRoutes.js, criaremos um novo arquivo chamado alunoRoutes.js.
   * Alteraremos o endpoint para /alunos e as chamadas para o novo controlador (alunoController).
4. **Configuração do Servidor:**
   * No index.js, vamos adicionar a rota alunoRoutes para incluir as novas rotas da API de ALUNOS.

**Explicação das Mudanças**

Para adaptar a API de USERS para ALUNOS:

1. **Modelos:** Criamos um novo modelo (alunoModel.js) ajustando as queries SQL para refletir a estrutura da tabela ALUNOS.
2. **Controladores:** Em alunoController.js, importamos alunoModel e utilizamos métodos específicos para lidar com a tabela ALUNOS.
3. **Rotas:** O novo arquivo alunoRoutes.js contém rotas dedicadas ao recurso alunos.
4. **Servidor:** Em index.js, incluímos alunoRoutes para expor as novas rotas.