

**PROVA PRÁTICA**

Critérios	Indicadores avaliados	Conceito
Análise	1. Relaciona os conceitos e tecnologias back-end de aplicações	
Codificação	1. Codifica a camada de conexão da aplicação com os bancos de dados 2. Desenvolver as estruturas de integração da aplicação	
Critério de correção	<b>Cada item de cada questão é avaliado sob 4 aspectos</b> 1. Camada Controller/REST 2. Camada Service/BO 3. Camada Model: Entidades/VOs e tabelas 4. Camada Model: Repositórios/DAOs  <b>Total: 20 critérios (5 itens para desenvolver x 4 aspectos)</b>  <b>Conceito OT</b> - questões I, II e III possuírem 100% da resolução desenvolvida adequadamente em todos os critérios acima  <b>Conceito B</b> - questões I, II e III com no mínimo 80% de acertos (até 4 erros)  <b>Conceito S</b> - questões I, II e III com no mínimo 60% de acertos (até 6 erros)  <b>Conceito I</b> - critérios acima não atendidos ou projeto com erros de compilação	

**INSTRUÇÕES**

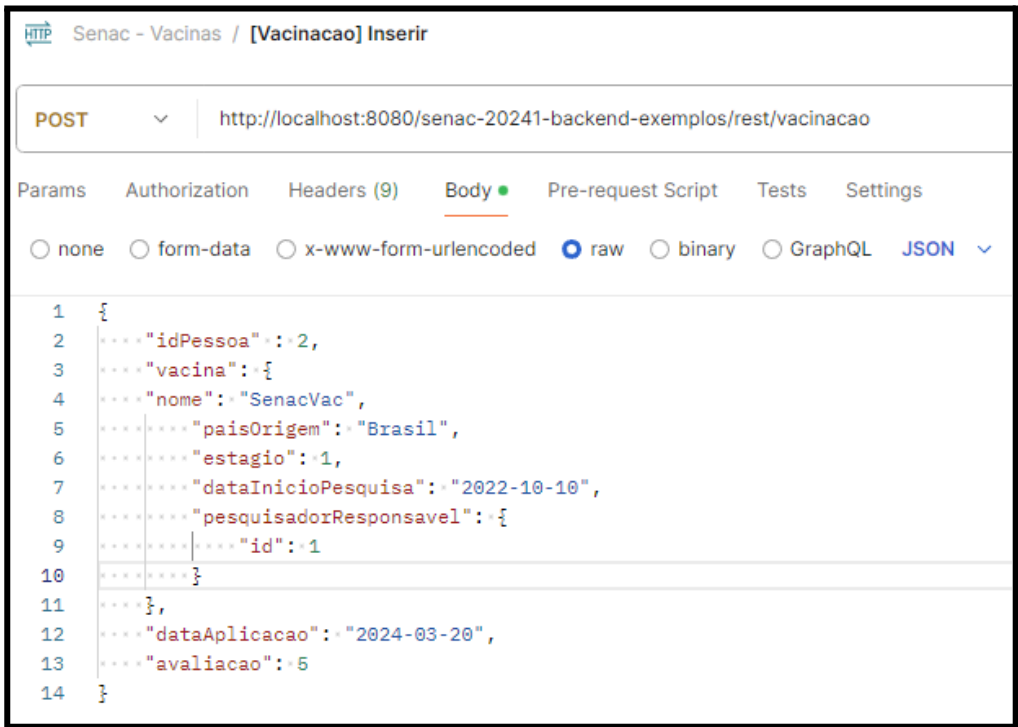
- Utilize o **seu projeto de backend** da disciplina no github, utilizando o que foi apresentado até o momento no curso
- **Caso não possua projeto pessoal no github (opção não recomendada):** clone o projeto da disciplina
  - <https://github.com/vilmar-junior/senac-20241-backend-exemplos/>
- **Entrega:** enviar o link do github do seu projeto na tarefa criada no teams (o horário dos commits e pushes será considerado, conforme o limite da prova)

Considere o estudo de caso do controle de vacinas, com os seguintes arquivos auxiliares:

- Coleção de testes do postman:
  - [https://github.com/vilmar-junior/senac-20241-backend-exemplos/blob/main/src/main/resources/Vacinas.postman\\_collection.json](https://github.com/vilmar-junior/senac-20241-backend-exemplos/blob/main/src/main/resources/Vacinas.postman_collection.json)
  - Realize o download do arquivo .json
  - **No Postman:** File > Import > Selecione o arquivo "Vacinas.postman\_collection.json"
- Scripts de banco:
  - [https://github.com/vilmar-junior/senac-20241-backend-exemplos/blob/main/src/main/resources/scripts/create\\_tables\\_vacina.sql](https://github.com/vilmar-junior/senac-20241-backend-exemplos/blob/main/src/main/resources/scripts/create_tables_vacina.sql)
  - [https://github.com/vilmar-junior/senac-20241-backend-exemplos/blob/main/src/main/resources/scripts/inserts\\_vacina.sql](https://github.com/vilmar-junior/senac-20241-backend-exemplos/blob/main/src/main/resources/scripts/inserts_vacina.sql)

Em todos as questões a seguir (itens I, II e III), o usuário deve receber uma exceção do tipo `VacinacaoException` caso a regra seja violada. Tal exceção, quando lançada deve contendo uma mensagem compreensível para o usuário final do sistema.

JSON contendo uma vacinação (considerando um usuário e vacina já cadastrados):



**I - Implemente as seguintes regras para exclusão de pessoas e vacinas:**

- 1. Pessoa não pode ser excluída caso já tenha recebido pelo menos uma dose vacina;
- 2. Vacina não pode ser excluída caso já tenha sido aplicada em pelo menos uma pessoa;

**II - Atualizar a média da vacina a cada nova aplicação**

- 1. Armazenar a **média** na vacina (valor decimal - *double*)
- 2. A cada nova aplicação de vacina (*insert/update de vacinação*), atualizar a média simples das aplicações da **vacina** (VacinaX) em questão, conforme a fórmula:

$$MediaVacinaX = \frac{SomatorioNotasVacinaX}{QuantidadeAplicacoesVacinaX}$$

**III - Ao aplicar uma dose de vacina (inserir/atualizar vacinação), o sistema deve verificar se a pessoa pode receber a vacina em questão, consoante a seguinte regra:**

Situação da vacina	Perfil de usuário que pode receber a dose da vacina
1 - Inicial	Pesquisador
2 - Testes	Pesquisador ou voluntário
3 - Aplicação em massa	Pesquisador, voluntário ou público em geral (todos)