

## PROVA DE SUFICIÊNCIA PROGRAMAÇÃO WEB II - 2025 / 2

### OBSERVAÇÕES GERAIS: CONFORME O QUESTIONÁRIO QUE VOCÊ CONCORDOU

- (i) Você pode optar por não entregar a prova: a prova não será avaliada e o aluno deve continuar a cursar a disciplina normalmente;
- (ii) Você pode realizar a prova e não obter a nota para aprovação: nesse caso deve continuar a cursar a disciplina normalmente. A nota desta prova não será utilizada no cálculo da média do aluno;
- (iii) Você pode entregar a prova e obter a nota para aprovação: você não pode cursar a disciplina e a nota desta prova será utilizada como média final do aluno na disciplina.
- (iv) Questões que apresentarem cópias serão anuladas.

Observações no caso de obter nota de aprovação (6,0):

- (v) mesmo que seja uma nota muito próxima da média (ex. 6,1) será utilizada esta nota. Não será permitido refazer a prova ou cursar a disciplina para aumentar a média final do aluno.
- (vi) Você deixa de frequentar a disciplina, mas continua pagando os créditos da mesma.
- (vii) Você pode optar por se inscrever em outra disciplina oferecida nos horários desta disciplina. Mas vai pagar a mais os créditos desta disciplina.

### **IMPORTANTE!!!**

O código-fonte produzido neste trabalho deverá ser **postado em um repositório do GitHub** até **07/08/2025** e o link do GitHub deverá ser enviado para o **E-mail do professor**. Para validar a nota da prova de suficiência será realizada uma **arguição oral** entre o aluno e o professor na data **07/08/2025** sem possibilidade de reagendamento.

### **ENUNCIADO:**

1. Faça um Web Service REST com comunicação JSON conforme os serviços definidos a seguir. Os serviços devem retornar os códigos de erros conforme os códigos de resposta REST (<https://docs.microsoft.com/pt-br/partner/develop/errorcodes>), caso um serviço seja invocado de forma incorreta. **(25%)**
2. A partir dos serviços, os dados devem ser persistidos em um banco de dados relacional através de um framework ORM (JPA, Entity Manager, Sequelize, etc) com nomenclatura padrão de BD, isto é, nome da tabela no plural e nome da classe no singular, por exemplo. As tabelas e atributos devem ser gerados por este framework **(25%)**
3. Algum serviço deve ser configurado para ser acessado somente caso o usuário estiver autenticado por meio de um token. Para isso, deve-se utilizar algum framework de segurança como o OAuth **(15%)**
4. Deve-se disponibilizar uma documentação utilizando Swagger **(15%)**
5. Uso de MVC + DAO **(15%)**
6. Todos os atributos nas classes modelo devem ser validados **(5%)**
7. **Fique à vontade para incluir novos serviços em caso de necessidade.**



URL BASE: <http://localhost:8080/RestAPIFurb/>

Tipo	Rota	JSON (body)	Resultado
GET	/comandas	Não há	[ { "idUsuario":1, "nomeUsuario":"joao",  "telefoneUsuario":"478888888" }, { "idUsuario":2, "nomeUsuario":"Pedro",  "telefoneUsuario":"479999999" } ]
GET	/comandas/{id}	Não há	{ "idUsuario":1, "nomeUsuario":"joao",  "telefoneUsuario":"478888888", "produtos":[ { "id":1, "nome":"X-Salada", "preco":30 }, { "id":2, "nome":"X-Bacon", "preco":35 } ] }

POST	/comandas	<pre>{   "idUsuario":1,   "nomeUsuario":"joao",    "telefoneUsuario":"478888888",   "produtos":[     {       "id":1,       "nome":"X-Salada",       "preco":30     },     {       "id":2,       "nome":"X-Bacon",       "preco":35     }   ] }</pre>	<pre>{   "id":1,   "idUsuario":1,   "nomeUsuario":"joao",    "telefoneUsuario":"478888888",   "produtos":[     {       "id":1,       "nome":"X-Salada",       "preco":30     },     {       "id":2,       "nome":"X-Bacon",       "preco":35     }   ] }</pre>
PUT	/comandas/{id}	<pre>{   "produtos":[     {       "id":1,       "nome":"X-Galinha",       "preco":25     }   ] }</pre> <p><b>** Possibilitar passar SOMENTE as informações que foram alteradas daquela comanda. Os demais itens devem permanecer idênticos. Não é possível remover item específico da comanda.</b></p>	OPCIONAL
DELETE	/comandas/{id}	Não há	{"success":{"text":"comanda removida"}}