**🚗 CP2 - Locadora de Carros com API REST em ASP.NET Core**

Bem-vindo à sua **CP2** de ASP.NET Core Web API! O objetivo desta avaliação é testar seus conhecimentos em ASP.NET Core, Entity Framework Core com Oracle e lógica de programação orientada a objetos, com foco na construção de uma API RESTful funcional e bem estruturada.

📌 Leia atentamente as instruções antes de começar.

**🎯 Objetivo**

Você deve desenvolver uma API REST em ASP.NET Core que permita o **cadastro de carros** e o **registro de uma locação com cálculo do valor total**. O sistema **não precisa salvar a locação no banco**, apenas realizar o cálculo com base nos dados informados e no carro escolhido.

**📜 Regras do Projeto**

**🚘 Entidade Carro**

* Modelo (ex: "Civic")
* Marca (ex: "Honda")
* Ano (ex: 2020)
* Valor da Diária (ex: 150.00)

**🔢 Endpoint de Locação**

* Criar um **endpoint POST**:  
  POST /api/locacoes/calcular
* Esse endpoint deve receber um JSON com:

{

"carroId": 1,

"dataInicio": "2025-04-25",

"dataFim": "2025-04-30"

}

* O cálculo do valor total da locação deve considerar:

int dias = (dataFim - dataInicio).Days;

double subtotal = dias \* valorDiaria;

if (dias >= 7)

aplicar 10% de desconto;

else if (dias >= 3)

aplicar 5% de desconto;

* A resposta deve ser um relatório com:
  + Modelo e marca do carro
  + Período da locação
  + Valor da diária
  + Subtotal
  + Desconto aplicado (%)
  + Valor final

**📍 Endpoints obrigatórios**

* CRUD completo para **Carros**
* POST /api/locacoes/calcular (não precisa salvar, só calcular)

**🗃️ Banco de Dados**

* Banco **Oracle** com **EF Core**
* Configure corretamente no appsettings.json e Program.cs
* Não esquecer dos dois comandos que atualizam o seu banco de dados corretamente > **Add-Migration InitialCreate** e **Update-Database**

**🔍 Critérios de Avaliação**

| **Critério** | **Pontos** |
| --- | --- |
| Estrutura do projeto e organização | 1.0 |
| Implementação da entidade Carro com EF Core + Oracle | 2.0 |
| Endpoint /calcular com lógica correta do cálculo e retorno formatado | 4.0 |
| Lógica de desconto aplicada conforme os dias da locação | 2.0 |
| Uso correto de boas práticas (nomes, organização, comentários) | 1.0 |
| **Total** | **10.0** |

**⚠️ Observações**

* 🔹 O projeto deve ser desenvolvido **individualmente**
* 🔹O Projeto deve ser compactado em ZIP e colocado na tarefa (igual a CP1)
* ❌ **Proibido o uso de ferramentas de IA** (ChatGPT, Copilot, Bard, etc)
* ❌ Projetos idênticos entre alunos serão zerados
* ❌ Projeto que não compilar ou não rodar = **nota 0**

**📤 Exemplo de Entrada**

POST /api/locacoes/calcular

{

"carroId": 1,

"dataInicio": "2025-04-25",

"dataFim": "2025-04-30"

}

**📥 Exemplo de Saída**

{

"carro": "Civic",

"marca": "Honda",

"dataInicio": "2025-04-25",

"dataFim": "2025-04-30",

"valorDiaria": 150.00,

"subtotal": 900.00,

"desconto": "10%",

"valorFinal": 810.00

}

**Boa CP2! 🚀 capriche no seu código e demonstre seu domínio com APIs, EF Core e lógica aplicada.**