**Documentação Técnica do Projeto ControleViagem  
Projeto realizado por Luis Henrique Silva de Lima e Luiz Augusto Santos Gumz**

**1. Visão Geral**

O ControleViagem é uma API RESTful desenvolvida em ASP .NET Core 8.0 para gerenciar registros de viagens, incluindo destino, data, meio de transporte e custo. A aplicação utiliza Entity Framework Core com SQLite para persistência e adota o padrão arquitetural MVC.

**2. Estrutura do Projeto**

O projeto segue a seguinte estrutura de diretórios:  
  
ControleViagem/  
├─ ControleViagem.sln  
├─ ControleViagem.csproj  
├─ Program.cs  
├─ appsettings.json  
├─ Controllers/  
│ └─ ViagensController.cs  
├─ Models/  
│ └─ Viagem.cs  
├─ Data/  
│ └─ ViagemContext.cs  
├─ Migrations/  
│ ├─ 20250529000534\_InitialCreate.cs  
│ └─ ViagemContextModelSnapshot.cs  
└─ viagem.db

**3. Configuração do Ambiente**

Pré-requisitos:  
- .NET 8.0 SDK  
- SQLite CLI   
  
Configuração de conexões em appsettings.json:  
```json  
{  
 "ConnectionStrings": {  
 "ViagemContext": "Data Source=viagem.db"  
 }  
}  
```

**4. Banco de Dados**

A aplicação utiliza SQLite para armazenamento. O arquivo de dados é `viagem.db`. As migrations geradas criando a tabela Viagens possuem colunas: Id (INTEGER, PK), Destino (TEXT), Data (DATETIME), Transporte (TEXT), Custo (REAL).

**5. Modelo de Dados**

```csharp  
public class Viagem  
{  
 public int Id { get; set; }  
 public string Destino { get; set; }  
 public DateTime Data { get; set; }  
 public string Transporte { get; set; }  
 public double Custo { get; set; }  
}  
```

**6. Endpoints REST**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Verbo | Rota | Descrição | Request Body | Resposta |
| GET | /api/viagens | Lista todas as viagens | – | 200 OK + JSON array de Viagem |
| GET | /api/viagens/{id} | Retorna viagem por ID | – | 200 OK + JSON de Viagem ou 404 |
| POST | /api/viagens | Cria nova viagem | JSON Viagem (sem Id) | 201 Created + JSON criada |
| PUT | /api/viagens/{id} | Atualiza viagem existente | JSON Viagem | 204 NoContent ou 404 |
| DELETE | /api/viagens/{id} | Remove viagem | – | 204 NoContent ou 404 |

**7. Build e Execução**

```bash  
dotnet restore  
dotnet build  
dotnet run --project ControleViagem/ControleViagem.csproj  
```

**8. Testes e Inspeção do Banco**

Use SQLite CLI ou GUI:  
```bash  
sqlite3 viagem.db  
.tables  
SELECT \* FROM Viagens;  
```

**9. Possíveis Melhorias**

- Autenticação/Autorização JWT  
- Validações com DataAnnotations  
- Documentação Swagger  
- Container Docker  
- Front-end em React/Vue

**10. Contribuição**

1. Fork do repositório  
2. git checkout -b feature/nome  
3. Commit e push  
4. Abrir Pull Request

**11. Licença**

Projeto sob MIT License.