Desenvolvimento Web com .NET e Bases de Dados [25E2_4]

Daniel Vitor Madeira Reis

AT

Professor: Reginaldo Ferreira
Data: 22/06/2025

Explicações:
Olha, eu fiz meu melhor, mas acabei dando um passo maior que a perna. Eu tinha uma ideia boa até pro projeto, mas fiquei inventando muito, e vários erros no banco de dados me fizeram perder horas e horas. No fim, eu desisti de dar update no banco e simplesmente, a cada alteração, eu deletava ele e criava dnv pra evitar mais problemas kkk. Sinceramente, me desculpa pelo projeto mal acabado.

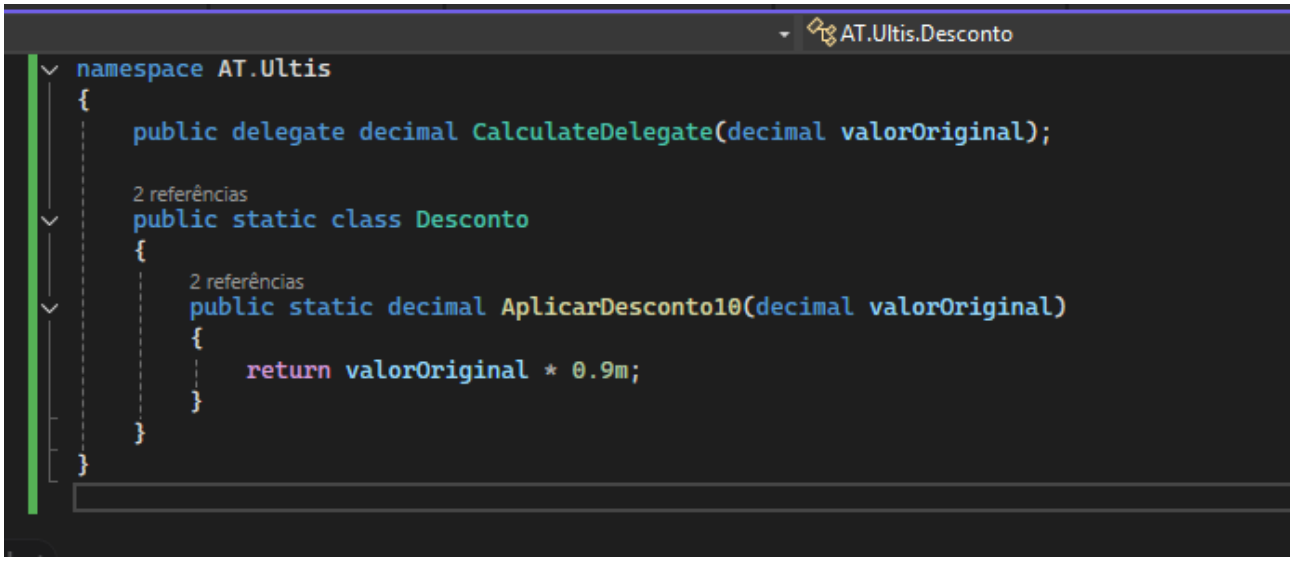
Olha, abaixo eu tirei um print e o caminho que "completei o enunciado", mas achei meio confusos esses enunciados. Então tentei aplicar tudo da aula + os enunciados. Pode ser que n tenha ficado 100% igual, mas fiz meu melhor.

Em alguns casos, tipo quando usei JavaScript, precisei pesquisar na internet, mas deixei comentado no código (coloquei um botão nas entidades “preencher” para ajudar na correção).

Eu n lembro se o senhor ensinou como retornar dois tipos de valores diferentes de um método, então procurei na net e achei o out, que parece ser algo bem simples e funciona. Então, em alguns lugares, vou usar ele, tipo pra retornar true e false e a mensagem com o tipo de erro.

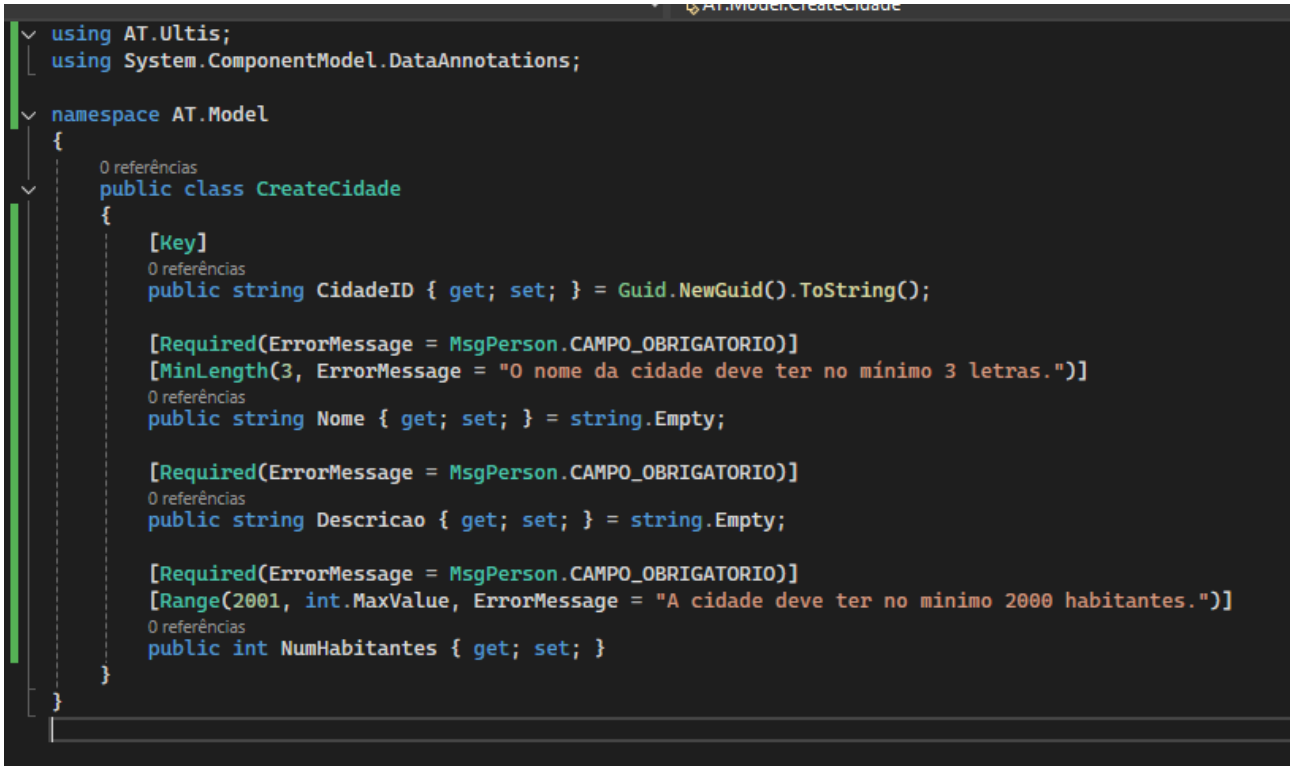
Eu usei o Guid.NewGuid() pra gerar ID. Eu até poderia usar tipo "Guid", mas teria que fazer alguns tratamentos e conversões. Aí, pra n perder tempo, deixei tudo como string pra facilitar e n perder muito tempo.

1. Fazer lógica de “CalculateDelegate” para dar desconto em propriedades.
   R: Eu eu não fiz propriedades, aí para n deixar sem fazer coloquei a lógica de delegate para todos.



5. a) Criar página Razor com form que valida campo obrigatório e mínimo de caracteres.
b) Validar que o nome da cidade tenha no mínimo 3 letras.

R: Todos Created entidade que fiz em “AT\Model” e inclusive em “AT\Model\CreateCidade.cs” tem verificações, caso queira verificar.



6. Criar uma página de cadastro para uma entidade com ao menos dois campos validados por Data Annotations.

R: Todas minhas entidades dentro do “Model” já faz isso.

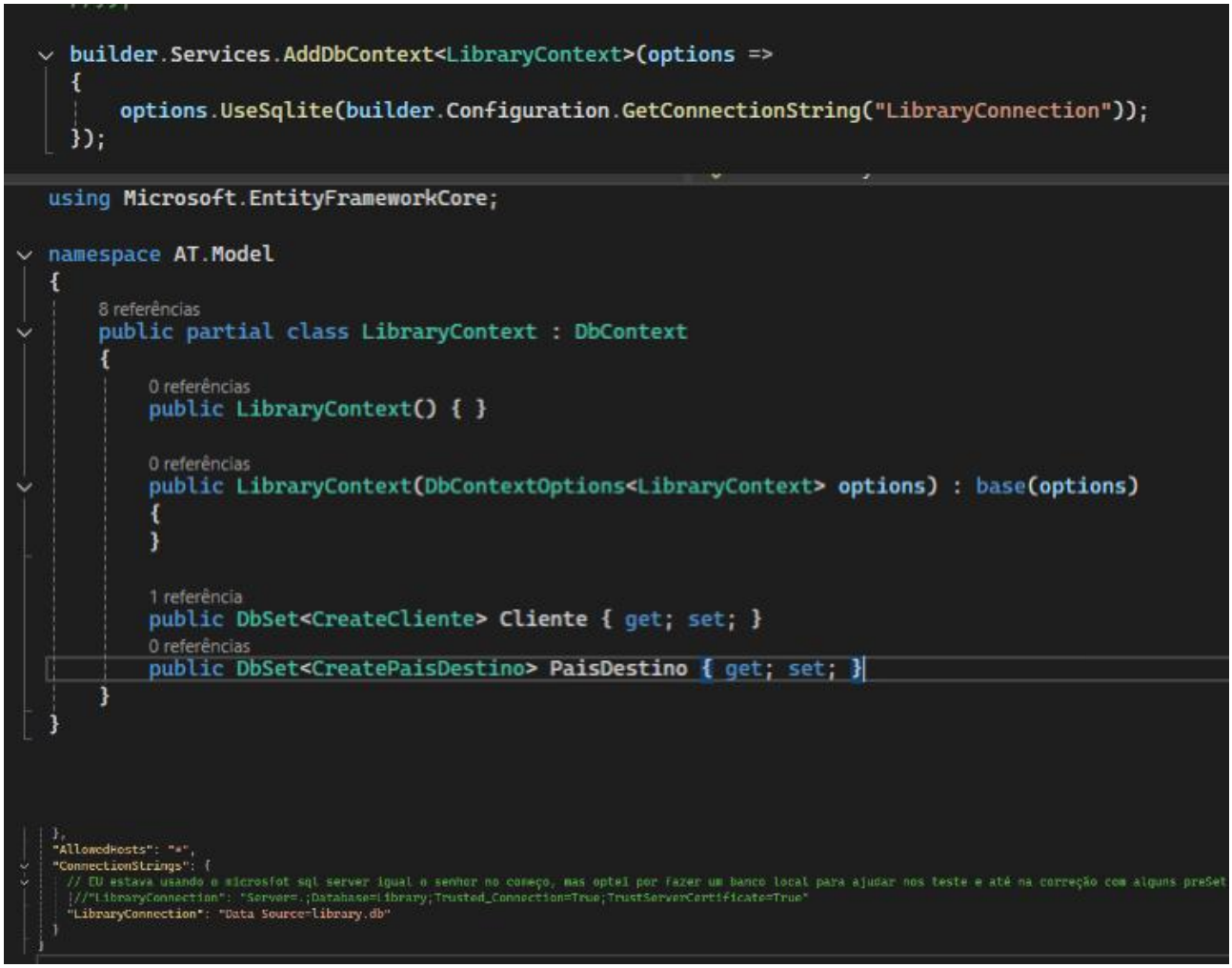
7. Crie uma página [Entidade]Details.cshtml que use parâmetros na URL (ex: /EntidadeDetails/123) para exibir os dados completos da entidade correspondente, conforme as regras do sistema.

R: Eu n usei os mesmo nomes igual do enunciado por ser um exemplo, todas minhas entidades tem o botão “Ver “Entidade” e lá tem uma opção e editar que ve todos detalhes e edita exemplo: “https://localhost:7141/Cadastro/EditarCadastro/4d831d57-5848-4bc0-bb21-60ebb4cbeffc”.

9. Criar classe de contexto com DbSets, registrar no Program.cs e configurar conexão com banco relacional via EF Core.

R: Eu havia feito igual o senhor com “Microsoft Sql Server” porém para melhor correção e testes fiz um local no próprio arquivo mesmo.

Os demais arquivos são os nomes e locais padrões e o contexto ta em “AT\Model\LibraryContext”.



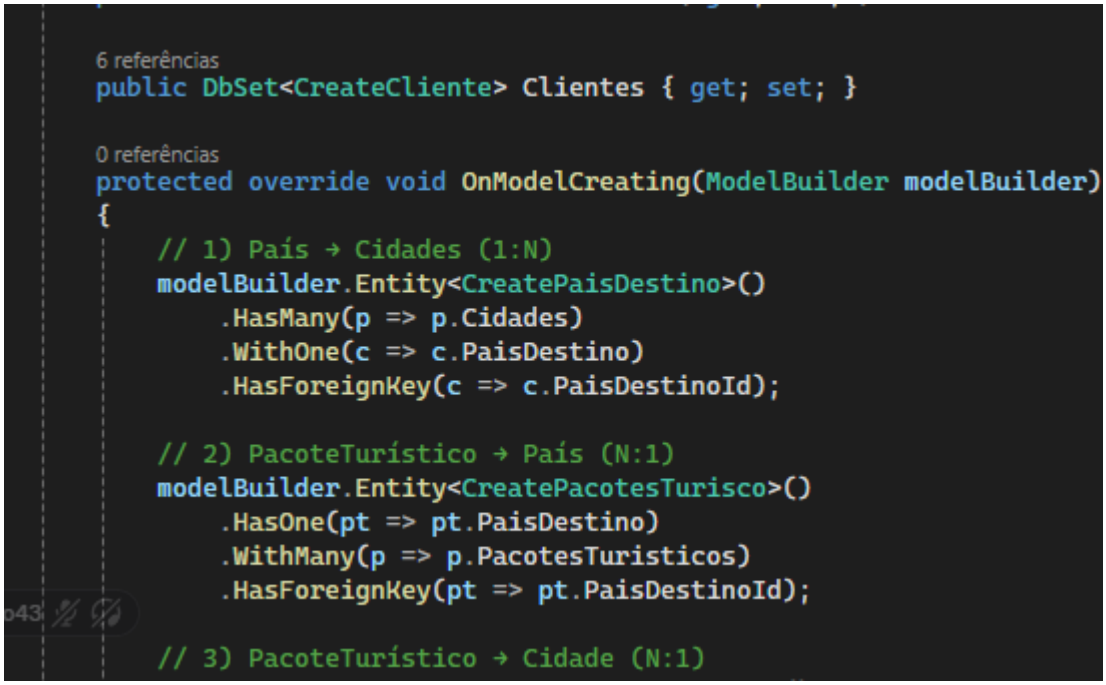
```
builder.Services.AddDbContext<LibraryContext>(options =>
{
    options.UseSqlite(builder.Configuration.GetConnectionString("LibraryConnection"));
});
```

```
using Microsoft.EntityFrameworkCore;

namespace AT.Model
{
    8 referências
    public partial class LibraryContext : DbContext
    {
        0 referências
        public LibraryContext() { }

        0 referências
        public LibraryContext(DbContextOptions<LibraryContext> options) : base(options)
        {
        }

        1 referência
        public DbSet<CreateCliente> Cliente { get; set; }
        0 referências
        public DbSet<CreatePaisDestino> PaisDestino { get; set; }
    }
}
```

```
},
"AllowedHosts": "*",
"ConnectionStrings": {
  // EU estava usando o microsfot sql server igual o senhor no começo, mas optei por fazer um banco local para ajudar nos teste e até na correção com alguns preSet
  //"LibraryConnection": "Server=.;Database=Library;Trusted_Connection=True;TrustServerCertificate=True"
  "LibraryConnection": "Data Source=library.db"
}
}
```

10. Modele as entidades e seus relacionamentos no EF Core, usando Fluent API ou DataAnnotations, com chaves estrangeiras e navegação.

R: Eu fui revendo as aulas e por algum motivo algumas coisas estavam dando erro, ai eu fui procurando na net oq poderia ser e fui ajustando, pode ser que n esteja 100% como o senhor esperava, mas fiz meu melhor”



```
6 referências
public DbSet<CreateCliente> Clientes { get; set; }

0 referências
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    // 1) País → Cidades (1:N)
    modelBuilder.Entity<CreatePaisDestino>()
        .HasMany(p => p.Cidades)
        .WithOne(c => c.PaisDestino)
        .HasForeignKey(c => c.PaisDestinoId);

    // 2) PacoteTurístico → País (N:1)
    modelBuilder.Entity<CreatePacotesTurisco>()
        .HasOne(pt => pt.PaisDestino)
        .WithMany(p => p.PacotesTuristicos)
        .HasForeignKey(pt => pt.PaisDestinoId);

    // 3) PacoteTurístico → Cidade (N:1)
```

11. Gerar páginas CRUD no Visual Studio com scaffolding para uma entidade do sistema, garantindo que o DbContext esteja configurado. Depois, personalizar as views geradas para mostrar dados relacionados, formatar valores (como datas e moedas) e melhorar a interface visual.

R: Eu acho que até fugi um pouco do enunciado, mas essa geração automática de CRUD eu achei muito ruim. Era muito mais difícil gerar automaticamente e arrumar na mão. Então eu fiz diretamente na mão. Todas entidades tem Index que é a principal e “Ver[entidade]” e no ver tem o botão de delete simples que remove do banco e o de editar que redireciona pro “site” de edição”. Tecnicamente fiz oq o enunciado pediu.