

Beto está em apuros de novo!!

Beto resolveu isolar o código de configuração das classes `Personagem` e `Raca` conforme o vídeo anterior.

Contudo, agora seu programa voltou a apresentar o erro `Invalid object name 'Racas'`. Veja o código abaixo:

Da classe `PersonagemConfiguration`:

```
public class PersonagemConfiguration : IEntityTypeConfiguration<Personagem>
{
    public void Configure(EntityTypeBuilder<Personagem> builder)
    {
        builder
            .ToTable("lotr_characters");

        builder
            .Property(p => p.Id)
            .HasColumnName("character_id");

        builder
            .Property(p => p.Nome)
            .HasColumnName("character_name")
            .HasColumnType("varchar(60)")
            .IsRequired();
    }
}
```

Da classe `RacaConfiguration`:

```
public class RacaConfiguration : IEntityTypeConfiguration<Raca>
{
    public void Configure(EntityTypeBuilder<Raca> builder)
    {
        builder.ToTable("lotr_race");

        builder
            .Property(r => r.Id)
            .HasColumnName("race_id");

        builder
            .Property(r => r.Nome)
            .HasColumnName("race_name")
            .IsRequired();
    }
}
```

E assim ele pode limpar o método `OnModelCreating()` na classe de contexto:

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
}
```

O que Beto esqueceu de fazer?

Selecione uma alternativa

A Executar o comando `Script-Migration` .

B Executar o comando `Update-Database` .

C Aplicar a configuração no método `OnModelCreating()` :

```
protected override void OnModelCreating(ModelBuilder modelBuilder)
{
    modelBuilder.ApplyConfiguration(new PersonagemConfiguration());
    modelBuilder.ApplyConfiguration(new RacaConfiguration());
}
```