

## Configurando nulidade das colunas

### Transcrição

O script gerado pelo Entity para a tabela `Ator` ainda apresenta diferenças com relação à tabela original, precisamos configurar a parte de nulidade das colunas `first_name` e `last_name`. É interessante destacar que a propriedade `Id` não foi classificada como nula. Estamos nos deparando, portanto, com mais uma convenção do Entity para determinar nulidade dos objetos. A nulidade de uma coluna é determinada pelo tipo da **CLR** da propriedade. Caso seja permitido o valor `null` pelo tipo de **CLR**, a coluna terá o valor `null`.

Para demonstrarmos isso, criaremos uma variável do tipo `int` com o nome `idade` e tentaremos colocar o valor nulo. Perceberemos que o programa irá apontar o erro com a mensagem: **"não é possível converter o valor nulo em 'int' porque ele é um tipo de valor não nulo"**

Ao tentarmos realizar o mesmo procedimento com uma propriedade do tipo `string`, a mensagem de erro não ocorrerá. A `string` é uma classe, por isso ela aceita o valor nulo. Para averiguar se um tipo de **CLR** aceita valor `null`, basta realizar esse teste simples.

Para quebrarmos a convenção de nulidade do Entity, faremos mais uma anotação denominada `Required`. A `Required` assinala a obrigatoriedade das colunas, ou seja, não permite que os valores sejam nulos.

```
using System.ComponentModel.DataAnnotations.Schema;

namespace Alura.Filmes.App.Negocio
{
    [Table("actor")]
    public class Ator
    {
        [Column("actor_id")]
        public int Id { get; set; }

        [Required]
        [Column("first_name", TypeName = "varchar(45)")]
        public string PrimeiroNome { get; set; }

        [Required]
        [Column("last_name", TypeName = "varchar(45)")]
        public string UltimoNome { get; set; }

        public override string ToString()
        {
            return $"Ator ({Id}): {PrimeiroNome} {UltimoNome}";
        }
    }
}
```

No console, acionaremos o `Remove-Migration` para apagar a migração anterior e iniciarmos uma nova. Para isso, utilizamos a `Add-Migration Inicial`. Após esses procedimentos, iremos gerar o script (`Script-Migration`). Veremos que o novo script está muito mais parecido com o original do banco de dados.

```
CREATE TABLE [actor] (  
    [actor_id] int NOT NULL IDENTITY  
    [first_name] nvarchar(45) NOT NULL,  
    [last_name] nvarchar(45) NOT NULL,  
    CONSTRAINT [PK_actor] PRIMARY KEY ([actor_id])  
);
```