EF 6 → Migrations

1. Criar uma solução (Migrations) com 2 projetos:

1.1. Data access layer:

1.1.1. DAL.csproj

1.1.2. Class Library (.NET Framework)

1.2. Client console application

1.2.1. Client.csproj

1.2.2. Console App (.NET Framework)

2. Projeto DAL:

2.1. Adicionar a referência à EF6:

- Botão direito no projeto > Manage NuGet Packages > Browse > Entity Framework > Instalar.
- Ver na pasta References do projeto que foi incluída a library EntityFramework.

2.2. Classe Editora:

```
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

namespace DAL
{

    public class Editora
    {

        #region Scalar properties
        // Chave primária
        [Key]
        public int EditoraID { get; set; }

        // Obrigatório e tamanho máximo
        [Required]
        [StringLength(100, ErrorMessage = "Limite de 100 carateres.")]
        [MaxLength(100)]
        public string EditoraNome { get; set; }
        #endregion

        #region Navigation properties
        // 1 editora pode editar mais do que 1 livro
        public ICollection<Livro> Livro { get; set; }
        #endregion

    }

}
```

2.3. Classe Livro:

```
using System.ComponentModel.DataAnnotations;

namespace DAL
{

    public class Livro
    {

        #region Scalar properties
        // Chave primária
        [Key]
        public int LivroID { get; set; }

        // Obrigatório e tamanho máximo
        [Required]
        [StringLength(9, ErrorMessage = "Limite de 9 carateres.")]
        [MaxLength(9)]
        public string ISBN { get; set; }

        // Obrigatório e tamanho máximo
        [Required]
        [StringLength(100, ErrorMessage = "Limite de 100 carateres.")]
        [MaxLength(100)]
        public string Titulo{ get; set; }

        // Foreign key, mas sem data annotation
        public int EditoraID { get; set; }
        #endregion

        #region Navigation properties
        // 1 livro, com ISBN, é editado por uma única editora
        public Editora Editora { get; set; }
        #endregion

    }

}
```

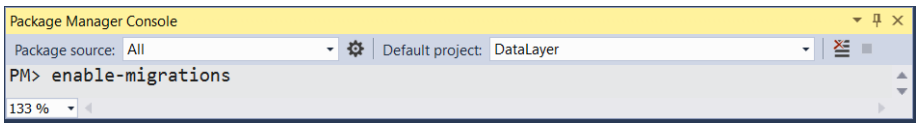
2.4. Classe BibliotecaContext:

```
using System.Data.Entity;
using System.Data.Entity.ModelConfiguration.Conventions;

namespace DAL
{

    public class BibliotecaContext : DbContext
    {

        public BibliotecaContext() : base("BibliotecaContext")
        {
        }

        protected override void OnModelCreating(DbModelBuilder modelBuilder)
        {
            // Desativar a pluralização das tabelas
            modelBuilder.Conventions.Remove<PluralizingTableNameConvention>();
        }

        public DbSet<Editora> Editora { get; set; }
        public DbSet<Livro> Livro { get; set; }

    }

}
```

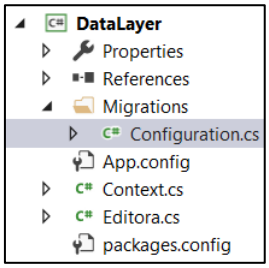
2.5. Acrescentar ao ficheiro de configuração App.config a connection string:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>

  <configSections> … </configSections>

  <entityFramework> … </entityFramework>

  <connectionStrings>
    <add
      name="BibliotecaContext"
      providerName="System.Data.SqlClient"
      connectionString=
        "Data Source=localhost;
        Initial Catalog=BiliotecaDb;
        Integrated Security=True;"/>
  </connectionStrings>

</configuration>
```

2.6. Package Manager Console para ativar o mecanismo de migrations: enable-manual.

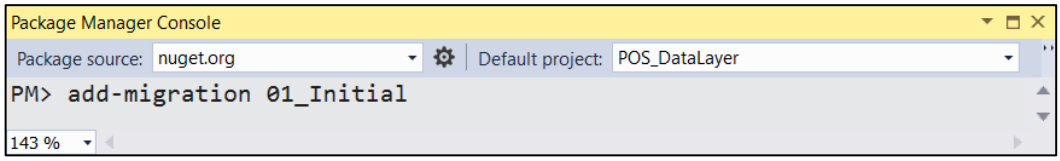


- É criada a pasta Migrations, com o ficheiro Configuration.cs:

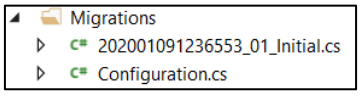



```
namespace DataLayer.Migrations
{
    using System;
    using System.Data.Entity;
    using System.Data.Entity.Migrations;
    using System.Linq;

    internal sealed class Configuration :
DbMigrationsConfiguration<DataLayer.BibliotecaContext>
    {
        public Configuration()
        {
            AutomaticMigrationsEnabled = false;
        }

        protected override void
Seed(DataLayer.BibliotecaContext context)
        {
            //  This method will be called after migrating
to the latest version.

            //  You can use the DbSet<T>.AddOrUpdate()
helper extension method
            //  to avoid creating duplicate seed data.
        }
    }
}
```

2.7. Package Manager Console para criar o 1º ficheiro de migrations: add-migration 01_Initial.

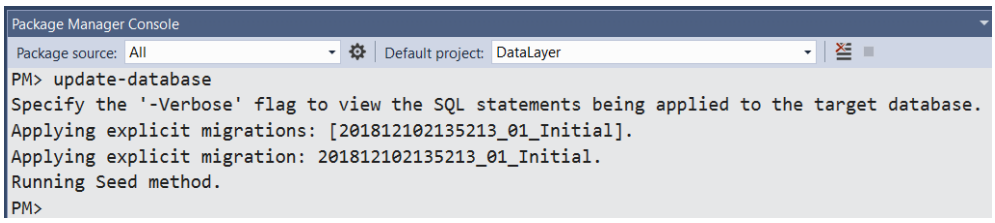


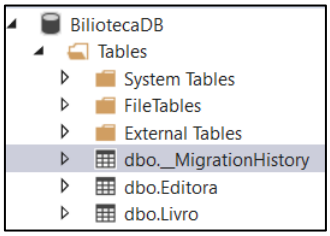
- Na pasta Migrations é criado o ficheiro 202001091236553_01_Initial.cs:

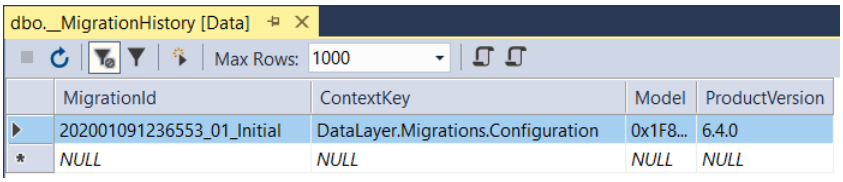



```
namespace DataLayer.Migrations
{
    using System;
    using System.Data.Entity.Migrations;

    public partial class _01_Initial : DbMigration
    {
        public override void Up()
        {
            CreateTable(
                "dbo.Editora",
                c => new
                    {
                        EditoraID = c.Int(nullable: false, identity: true),
                        EditoraNome = c.String(nullable: false, maxLength: 100),
                    })
                .PrimaryKey(t => t.EditoraID);

            CreateTable(
                "dbo.Livro",
                c => new
                    {
                        LivroID = c.Int(nullable: false, identity: true),
                        ISBN = c.String(nullable: false, maxLength: 9),
                        Titulo = c.String(nullable: false, maxLength: 100),
                        EditoraID = c.Int(nullable: false),
                    })
                .PrimaryKey(t => t.LivroID)
                .ForeignKey("dbo.Editora", t => t.EditoraID, cascadeDelete: true)
                .Index(t => t.EditoraID);

        }

        public override void Down()
        {
            DropForeignKey("dbo.Livro", "EditoraID", "dbo.Editora");
            DropIndex("dbo.Livro", new[] { "EditoraID" });
            DropTable("dbo.Livro");
            DropTable("dbo.Editora");
        }
    }
}
```

2.8. Criar a base de dados: update-database.







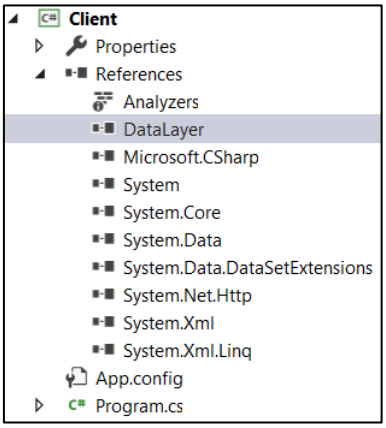
2.9. Fazer Build para compilar o projeto.

3. Usando o projeto Client:

3.1. Adicionar a referência à EF6.

3.2. Adicionar a referência ao projeto DataLayer:

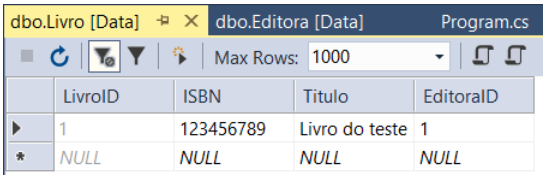
- Botão direito na pasta References > Add Reference… > Projects > DAL

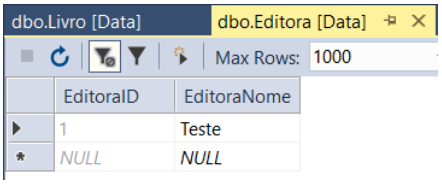


3.3. Acrescentar ao ficheiro de configuração App.config a connection string.

3.4. No Program.cs criar dois métodos para inserir uma editora e um livro dessa editora. Executar.

```
using System;
using System.Data.Entity;
using DataLayer;

namespace Client
{

    class Program
    {

        static void Main(string[] args)
        {

            Database.SetInitializer(new
NullDatabaseInitializer<BibliotecaContext>());

            // Executar só a 1ª vez
            InserirEditora();
            InserirLivro();

            Console.Write("Registos inseridos com sucesso!");

            Console.ReadKey();

        }

        public static void InserirEditora()
        {

            var editora = new Editora
            {
                EditoraNome = "Teste"
            };

            using (var context = new BibliotecaContext())
            {
                context.Editora.Add(editora);
                context.SaveChanges();
            }

        }

        public static void InserirLivro()
        {

            var livro = new Livro
            {
                ISBN = "123456789",
                Titulo = "Livro do teste",
                EditoraID = 1
            };

            using (var context = new BibliotecaContext())
            {
                context.Livro.Add(livro);
                context.SaveChanges();
            }

        }

    }

}
```



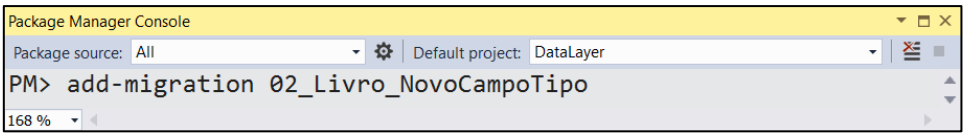


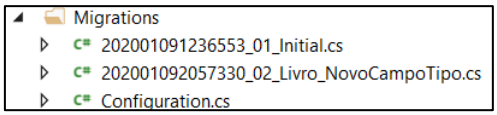
4. Usando o projeto DataLayer:

4.1. Acrescentar à tabela Livro o campo Tipo:

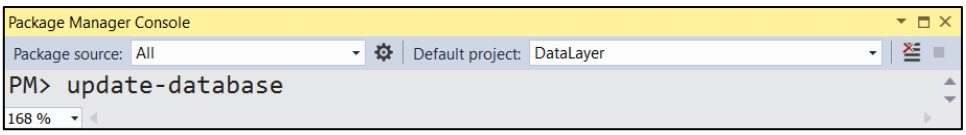
```
// Obrigatório e tamanho máximo
[Required]
[StringLength(100, ErrorMessage = "Limite de 100 carateres.")]
[MaxLength(100)]
public string Titulo{ get; set; }
```

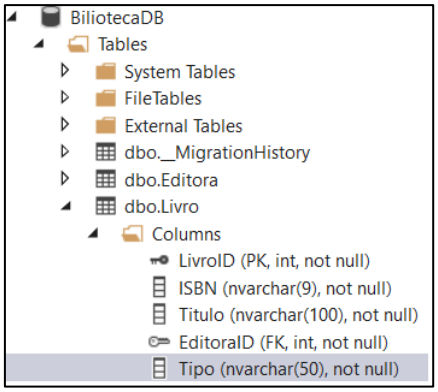
4.2. Gerar mais uma migration: add-migration 02_Livro_NovoCampoTipo:





4.3. Usar a nova migration para atualizar a base de dados: update-database.





5. Desafio:

5.1. Criar a tabela Tipo, definir uma associação entre Tipo e Livro e adicionar três tipos novos usando o método Seed (Configuration.cs).

5.2. Criar mais uma migration com as alterações.

5.3. Atualizar a base de dados.