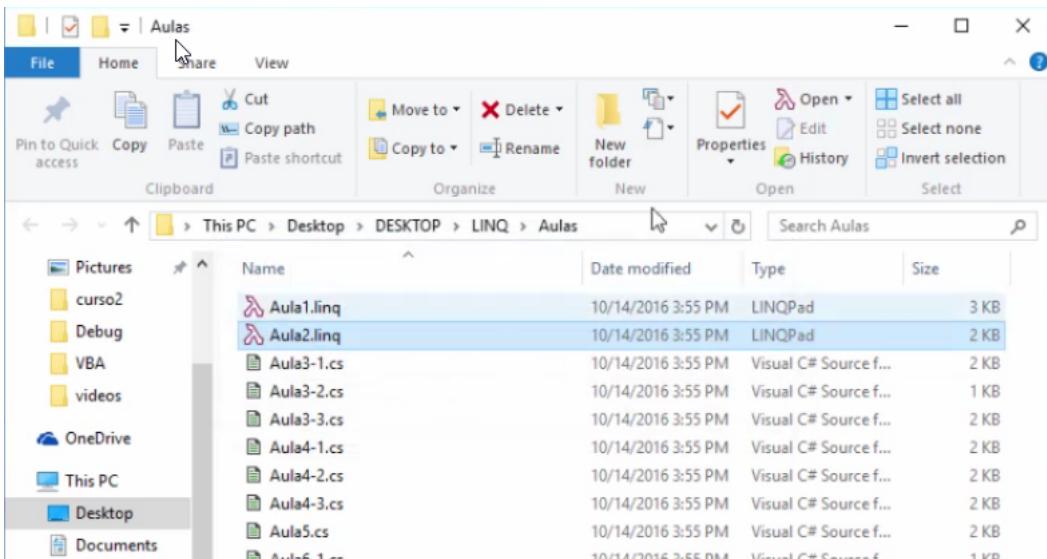


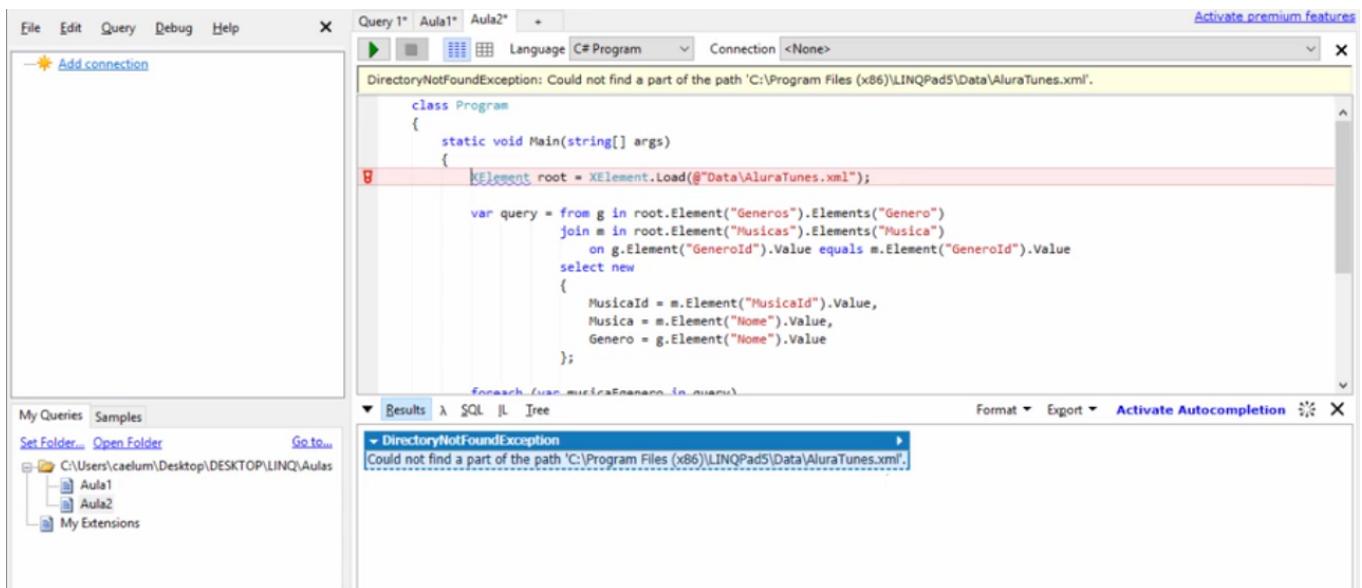
□ 02
2 - Linqpad

Transcrição

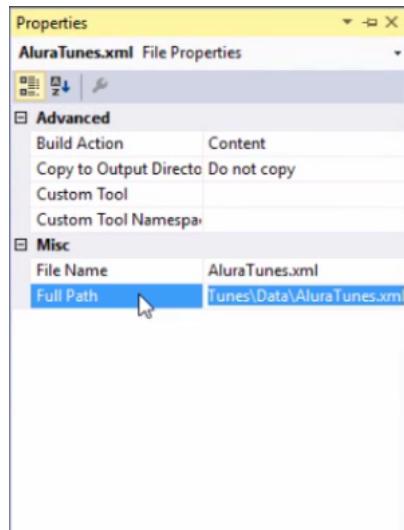
Nesta aula, vamos trabalhar com o arquivo que contenha o código no qual trabalhamos consulta LINQ utilizando o xml . Iremos renomear a extensão do documento para "Aula2.linq":



Para rodar o código é preciso apagar o namespace e modificar a linguagem para "C# Program". Mesmo com tais alterações, quando rodarmos o código, ocorrerá um erro:



A falha ocorre, porque não é possível encontrar o arquivo xml que utilizamos. Isso acontece, pois o caminho que passamos é relativo, precisaremos indicar o caminho absoluto. Em seguida, abriremos o projeto e clicaremos no arquivo AluraTunes.xml . Podemos copiar o caminho do arquivo:



Após colarmos o caminho, teremos o seguinte:

```
XElement root = XElement.Load(@"C:\Users\caelum\Documents\Visual Studio 2015\Projects\AluraTunes\Dat
```

Rodaremos a aplicação e vamos ter o seguinte:

```
class Program
{
    static void Main(string[] args)
    {
        XElement root = XElement.Load(@"C:\Users\caelum\Documents\Visual Studio 2015\Projects\AluraTunes\Data\AluraTunes.xml");

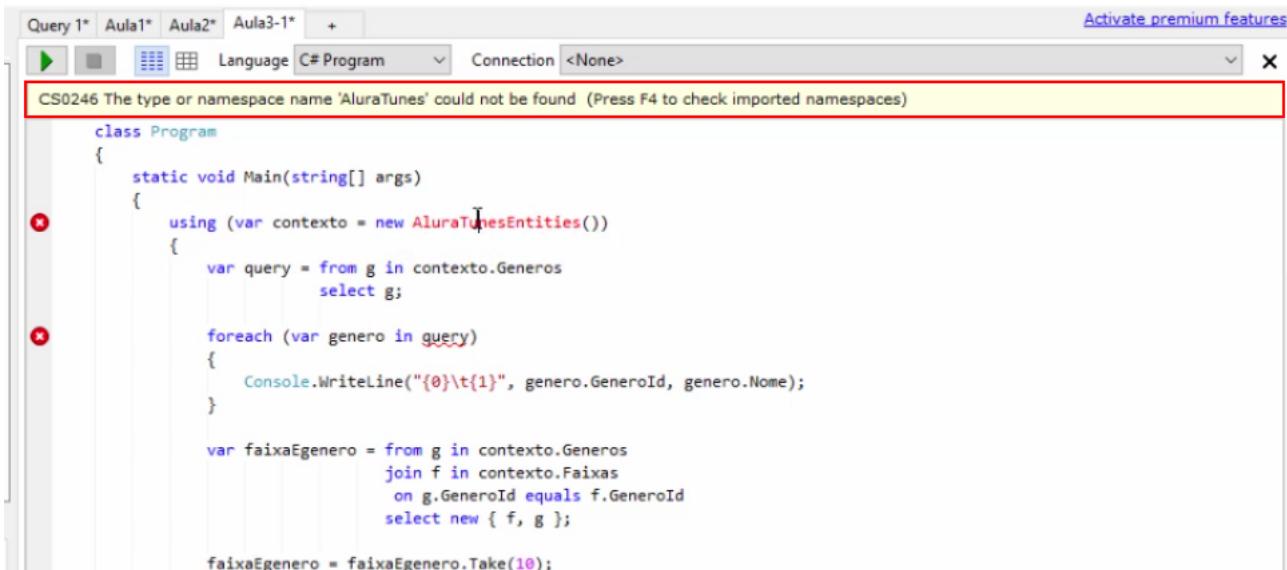
        var query = from g in root.Element("Generos").Elements("Genero")
                    join m in root.Element("Musicas").Elements("Musica")
                    on g.Element("GeneroId").Value equals m.Element("GeneroId").Value
                    select new
                    {
                        MusicaId = m.Element("MusicaId").Value,
                        Musica = m.Element("Nome").Value,
                        Genero = g.Element("Nome").Value
                    };

        foreach (var musicaEgenero in query)
    
```

1154	Sweet Child O' Mine	Rock	
900	I Shot The Sheriff	Reggae	
3445	Danubio Azul	Classica	

O resultado trouxe o nome da música e gênero!

Agora que aprendemos a utilizar o LINQPad para abrir consultas LINQ do Objects e LINQ do xml , vamos trabalhar com o arquivo de nome "Aula 3" e para começar a trabalhar com ele é preciso modificar a extensão do arquivo cs para linq . Para rodarmos o arquivo, apagaremos o namespace AluraTunes e vamos alterar a "Language" para "C# Program". Mas ao tentarmos executar o código nos deparamos com uma falha:



The screenshot shows the LINQPad interface with a C# program. The code attempts to use `AluraTunesEntities` but gets an error: CS0246 The type or namespace name 'AluraTunes' could not be found (Press F4 to check imported namespaces). There are two red asterisks on the left margin, one above each of the two errors.

```

class Program
{
    static void Main(string[] args)
    {
        using (var contexto = new AluraTunesEntities())
        {
            var query = from g in contexto.Generos
                        select g;

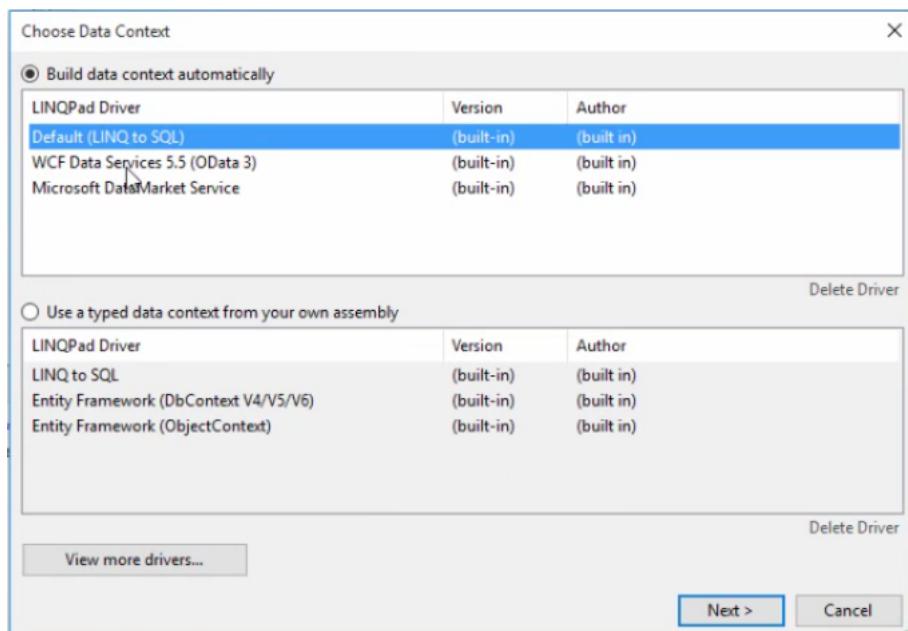
            foreach (var genero in query)
            {
                Console.WriteLine("{0}\t{1}", genero.GeneroId, genero.Nome);
            }

            var faixaEgenero = from g in contexto.Generos
                               join f in contexto.Faixas
                               on g.GeneroId equals f.GeneroId
                               select new { f, g };

            faixaEgenero = faixaEgenero.Take(10);
        }
    }
}

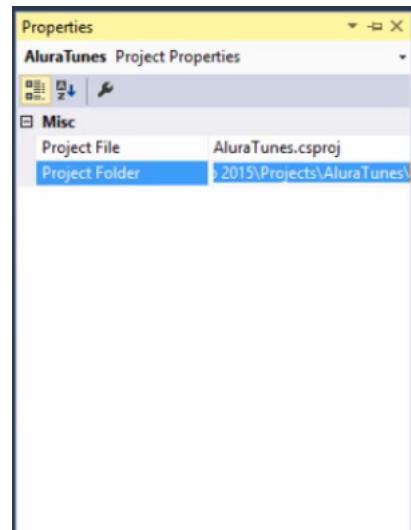
```

O problema ocorreu porque o programa não reconhecerá nem o `AluraTunesEntities()`, nem o contexto. A falha é não informar a configuração do LINQ ao Entity Framework que deve instanciar usando o `AluraTunesEntities()`. Vamos fazer isso utilizando uma nova conexão para o LINQ do Entities , assim, clicamos no Add connections :

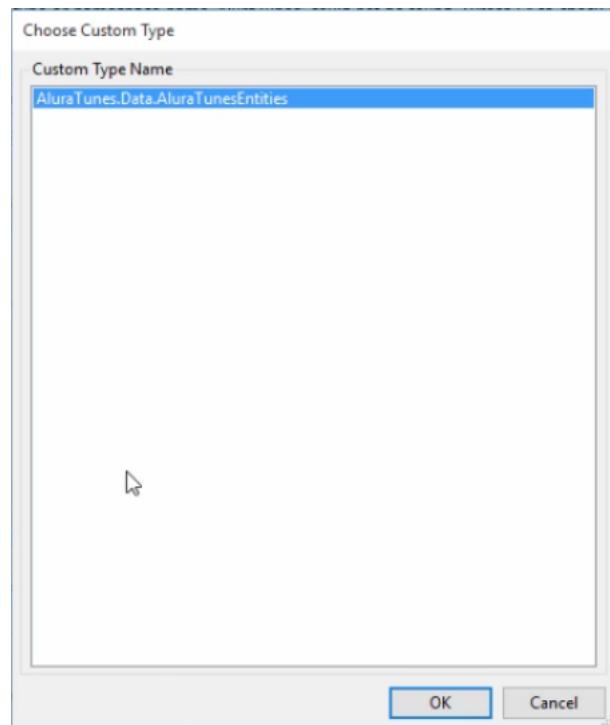


Uma janela é aberta e aparecem as opções para uma nova conexão. Escolheremos a opção de estabelecer uma conexão com o Entity Framework e clicamos em "Next".

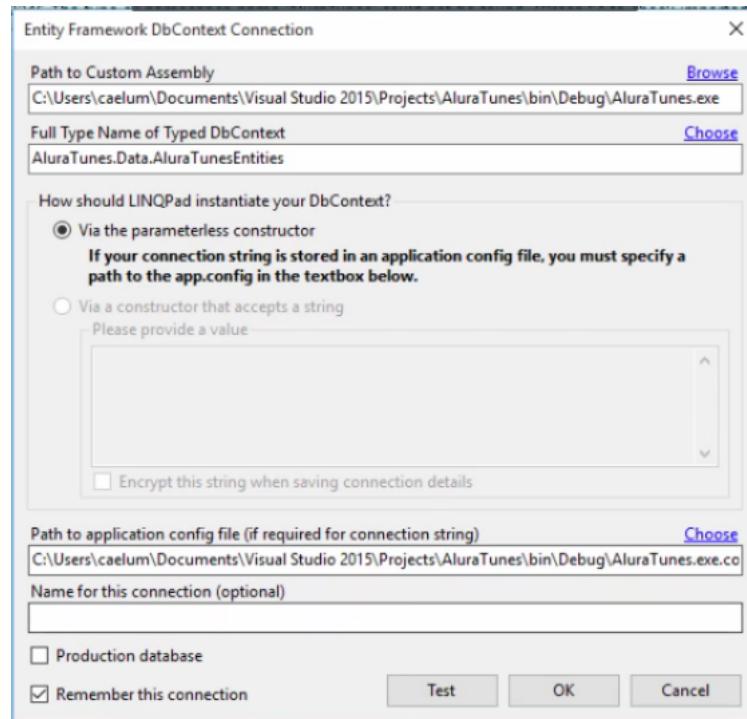
A próxima janela que será aberta servirá para escolhermos o caminho do Assembly , que é onde está o executável (ou a DLL) que contém as referências do projeto do Entity Framework . No campo "Path to Costume Assembly", vamos clicar em "Browse" e escolher o caminho do executável da aplicação. Para obter tal informação basta acessar o Visual Studio e copiar o caminho da pasta do projeto AluraTunes :



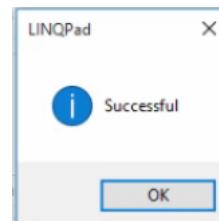
Ao copiar o caminho, retornaremos ao LINQPad e depois, passaremos para ele esta informação. Nós colamos o caminho no campo "File Name" e a partir disso vamos acessar os arquivos em "bin > Debug". Desta forma, abriremos o arquivo `AluraTunes.exe` e ao selecionarmos o documento teremos o seguinte:



Clicamos no "Ok" e voltaremos na janela que tínhamos antes, pois ainda falta preencher o campo "Path to application config file". Para escolhermos, o caminho clicaremos em "Choose" e dessa maneira, selecionarremos o `AluraTunes.exe.config`:



Agora, vamos testar! Selecionamos "Test", esperamos rodar e teremos como resposta o "Successful":



O resultado mostra os arquivos! O problema é que um erro continua acontecendo devido a que o LINQPad desconhece qual é a conexão será utilizada para acessar o "AluraTunesEntities". Por essa razão, o objeto aparece em vermelho:

```

class Program
{
    static void Main(string[] args)
    {
        using (var contexto = new AluraTunesEntities())
        {
            var query = from g in contexto.Generos
                        select g;

            foreach (var genero in query)
            {
                Console.WriteLine("{0}\t{1}", genero.GeneroId, genero.Nome);
            }

            var faixaEgenero = from g in contexto.Generos
                               join f in contexto.Faixas
                               on g.GeneroId equals f.GeneroId
                               select new { f, g };
        }
    }
}

```

O resultado mostra exatamente a lista de gêneros das músicas e também uma consulta que contém o nome da música e o nome do gênero:

The screenshot shows the LINQPad interface with the 'Results' tab selected. The results pane displays a list of genres and their associated songs:

```

▼ Results λ SQL JL Tree
19 TV Shows
20 Sci Fi & Fantasy
21 Drama
22 Comedy
23 Alternative
24 Classical
25 Opera

For Those About To Rock (We Salute You) Rock
Balls to the Wall Rock
Fast As a Shark Rock
Restless and Wild Rock
Princess of the Dawn Rock
Put The Finger On You Rock
Let's Get It Up Rock
Inject The Venom Rock
Snowballed Rock
Evil Walks Rock

```

Query successful (0.003 seconds)

Ao clicarmos no botão do SQL encontramos o seguinte resultado:

The screenshot shows the LINQPad interface with the 'SQL' tab selected. The results pane displays the generated SQL query:

```

▼ Results λ SQL JL Tree
SELECT
[Extent1].[Generoid] AS [Generoid],
[Extent1].[Nome] AS [Nome]
FROM [dbo].[Genero] AS [Extent1]
GO
SELECT TOP (10)
[Extent1].[Generoid] AS [Generoid],
[Extent2].[FaixaId] AS [FaixaId],
[Extent2].[Nome] AS [Nome],
[Extent2].[AlbumId] AS [AlbumId],
[Extent2].[TipoMidiaId] AS [TipoMidiaId],
[Extent2].[Generoid] AS [Generoid1],
[Extent2].[Compositor] AS [Compositor],
[Extent2].[Milissegundos] AS [Milissegundos],
[Extent2].[Bytes] AS [Bytes],
[Extent2].[PrecoUnitario] AS [PrecoUnitario]

```

Query successful (0.003 seconds)

Automaticamente o LINQPad traz a consulta SQL equivalente!

Vimos como utilizar o LINQPad para abrir as consultas sem rodarmos a aplicação. O resultado é mostrado de maneira simples e rápida, sem precisar iniciar a aplicação toda vez ou montar um set up, ou inserir usuário e senha. Mostramos que o LINQPad suporta tipos diferentes de consulta, por exemplo: LINQ do Objects , do xml e Entities . Para utilizarmos o LINQPad com o fim de acessar uma consulta LINQ do Entities , é preciso configurar uma conexão e passar para o LINQPad o Assembly que contem o executável ou a dll da aplicação e também é necessário passar o caminho da configuração , o webconfig , ou qualquer outro que a aplicação tiver. Por fim, abordamos também que é possível exibir o resultado no formato de tabela e que é possível obter de maneira fácil uma consulta com a linguagem SQL .