

PROGRAMAÇÃO COM ACESSO A BANCO DE DADOS

C# .NET: Acessando um
Banco de Dados
PostgreSql



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

eliezio.soares@ifrn.edu.br | <https://docente.ifrn.edu.br/elieziosoares>

Msc. Eliezio Soares

ADO.NET

- A ADO.NET provê acesso a fontes de dados, tais como: SQL Server, XML, ou outras fontes de dados.
- Pode-se utilizá-la para conectar a essas fontes de dados e recuperar, manipular e atualizar os dados.
- ADO.NET possui tipos para facilitar a manipulação dos dados:
 - Os dados obtidos são colocados em um objeto **DataSet**.
 - Isso padroniza os dados obtidos, independente da fonte.
- As classes da ADO.NET estão em System.Data.dll;

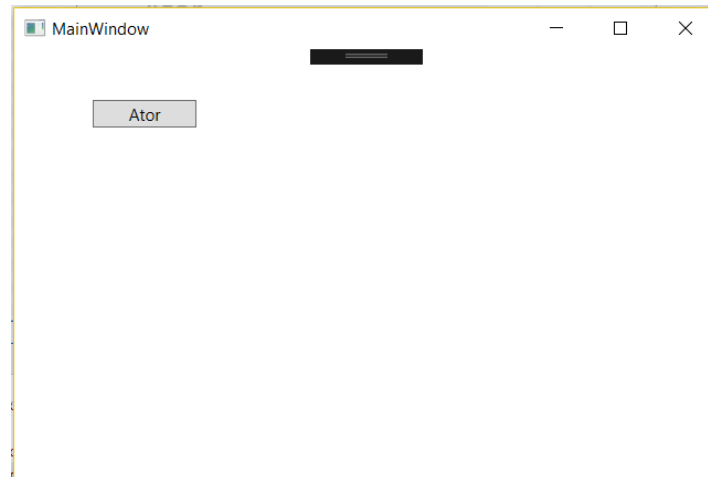
NPGSQL.DLL

- Npgsql é uma biblioteca compatível com a ADO.Net;
- Npgsql é uma implementação que possibilita a comunicação com uma fonte de dados PostgreSQL.
- O objetivo dessa biblioteca é fornecer base para que programas acessem bancos de dados PostgreSQL;

PRÁTICA

INSERINDO UM ATOR - 1

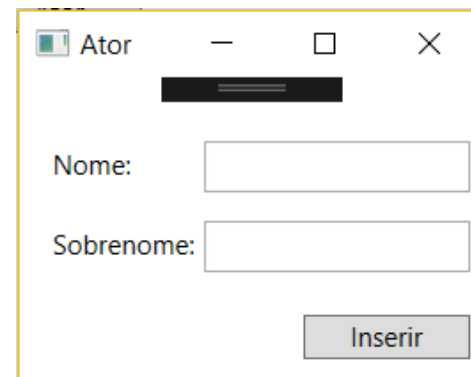
1. Criar um novo projeto WPF: **LocadoraDVD**
2. Adicionar em “MainWindow.xaml” um botão contendo o texto “Ator”.
3. Adicionar nova janela (botão direito no projeto > Add > Window...) de nome “Ator.xaml”.



PRÁTICA

INSERINDO UM ATOR - 2

- No banco de dados **LocadoraDVD** a tabela **Ator** contém os seguintes campos:
 - ator_id (serial)
 - nome
 - sobrenome
 - ultima_atualização (uso futuro)
- Adicionar os campos nome e sobrenome na janela Ator.



A screenshot of a graphical user interface window titled "Ator". The window has a standard title bar with a minimize button, a maximize button, and a close button. Below the title bar, there is a black rectangular area, possibly a logo or a placeholder. The main content area contains two text input fields. The first field is labeled "Nome:" and the second field is labeled "Sobrenome:". Below these fields is a button labeled "Inserir".

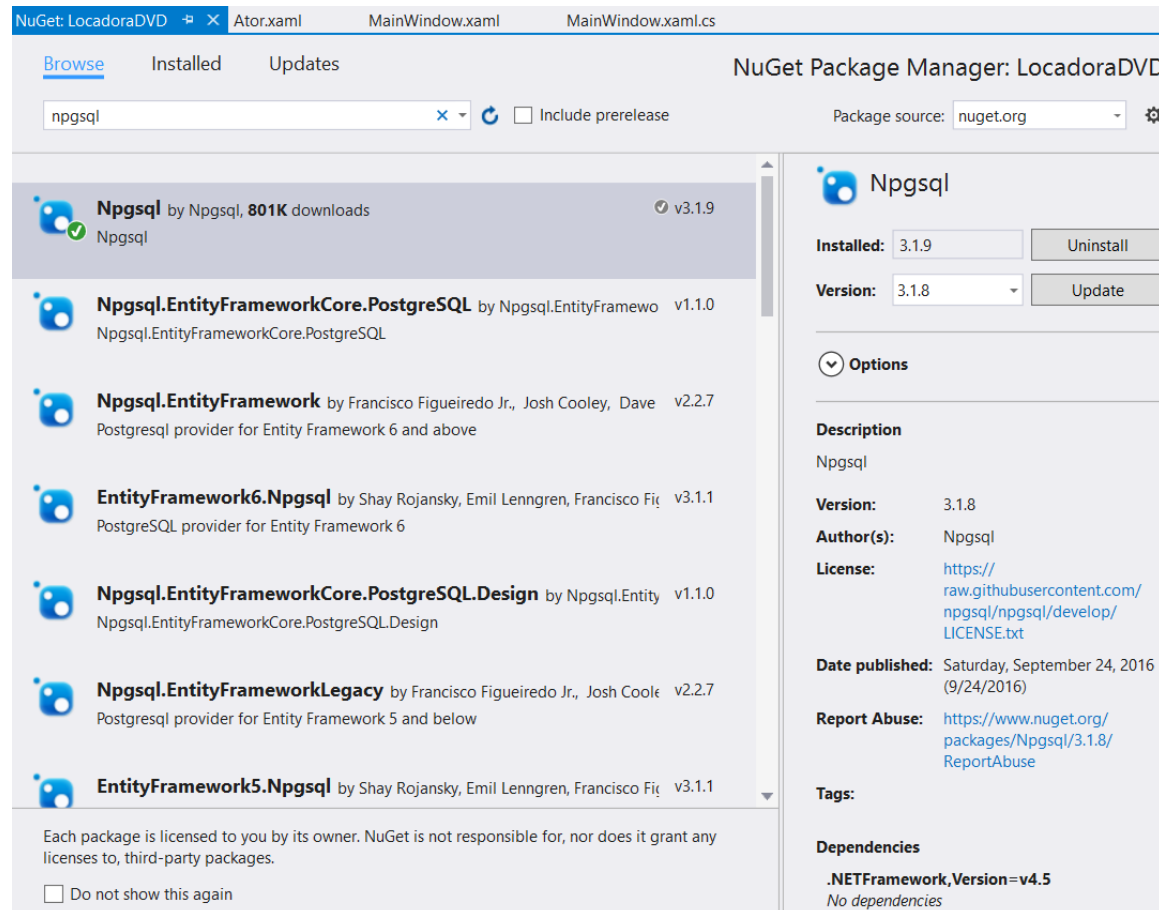
PRÁTICA

INSERINDO UM ATOR - 3

- Importar a biblioteca Npgsql ao projeto.
- Há dois modos...
- Baixando a dll e adicionando a referência ao projeto:
 - (botão direito no projeto > Add > Reference...)
- **Utilizando o NuGet – Gerenciador de Dependências**
 - (botão direito no projeto > Manage NuGet Packages...)
 - Buscar biblioteca npgsql
 - Instalar (as dependências instaladas ficam dentro das referências do projeto)

PRÁTICA

INSERINDO UM ATOR - 3



The screenshot shows the NuGet Package Manager interface for the 'LocadoraDVD' project. The search bar contains 'npgsql'. The package 'Npgsql' by Npgsql is selected, showing it is installed at version 3.1.9. The right-hand pane displays the details for the selected package, including its version (3.1.8), author (Npgsql), license, and dependencies.

Npgsql by Npgsql, 801K downloads ✓ v3.1.9

Npgsql
Npgsql

Installed: 3.1.9

Version: 3.1.8

Options

Description
Npgsql

Version: 3.1.8

Author(s): Npgsql

License: <https://raw.githubusercontent.com/npgsql/npgsql/develop/LICENSE.txt>

Date published: Saturday, September 24, 2016 (9/24/2016)

Report Abuse: <https://www.nuget.org/packages/Npgsql/3.1.8/ReportAbuse>

Tags:

Dependencies
.NETFramework,Version=v4.5
No dependencies

PRÁTICA

INSERINDO UM ATOR - 4

- Criar a camada de acesso a dados.
 - Nessa camada usaremos o Npgsql para acessar o banco de dados que está no servidor PostgreSQL.
 - Criar uma nova classe **DAL.cs** (Data Access Layer)
 - (Botão direito no projeto > Add > Class...)
 - Inicialmente utilizaremos essa classe para fazer a inserção do ator, mas o projeto será aprimorado para melhor modularização dos conceitos.

PRÁTICA

INSERINDO UM ATOR - 4

- Atributos da classe DAL.cs:

```
static string servidor = "127.0.0.1"; //localhost
static string porta = "5432"; //porta default
static string userName = "postgres"; //nome do administrador
static string password = "sua_senha"; //senha do administrador
static string banco = "LocadoraDVD"; //nome do banco de dados

NpgsqlConnection pgsqlConnection = null;
string connectionString = null;
```

PRÁTICA

INSERINDO UM ATOR - 4

- Construtor da classe DAL.cs:

```
public DAL()  
{  
    connectionString = String.Format("Server={0};Port={1};User Id={2};Password={3};Database={4};",  
                                     servidor, porta, userName, password, banco);  
}
```

- Ou...

```
public DAL()  
{  
    connectionString = "Server=" +servidor+";Port=" +porta+ ";User Id=" +userName+  
                       ";Password=" + password+ ";Database=" +banco+ ";";  
}
```

PRÁTICA

INSERINDO UM ATOR - 5

○ Método inserirAtor:

- Inicialmente esse método será implementado na classe DAL.cs, mas essa classe deve realizar uma comunicação genérica, obedecendo o princípio da decomposição.

```
public void InserirAtor(string nome, string sobrenome)
{
    try
    {
        using (NpgsqlConnection pgsqlConnection = new NpgsqlConnection(connectionString))
        {
            pgsqlConnection.Open();
            string cmdInserir = String.Format("Insert Into Ator(nome,sobrenome) values('{0}','{1}']", nome, sobrenome);
            using (NpgsqlCommand pgsqlcommand = new NpgsqlCommand(cmdInserir, pgsqlConnection))
            {
                pgsqlcommand.ExecuteNonQuery();
            }

            pgsqlConnection.Close();
        }
    }
    catch (NpgsqlException ex)
    {
        throw;
    }
    catch (Exception ex)
    {
        throw;
    }
}
```

PRÁTICA

INSERINDO UM ATOR - 6

- Implementar o comportamento da janela `MainWindow.xaml`:
 1. Ao clicar no botão `Ator`, deve se abrir uma janela para cadastro de atores (`Ator.xaml`)
 1. Duplo clique no botão `Ator`;
 2. Um método (handler) foi gerado no code behind (`MainWindow.xaml.cs`);
 3. O método handler será executado sempre que o usuário clicar no botão `Ator`;
 4. Esse método deve instanciar uma janela `Ator.xaml` e abri-la:

```
private void AtorBT_Click(object sender, RoutedEventArgs e)
{
    Ator janelaAtor = new Ator();
    janelaAtor.Show();
}
```

PRÁTICA

INSERINDO UM ATOR - 7

- Implementar o comportamento da janela Ator.xaml:
- Ao clicar no botão Inserir, a aplicação deve recuperar os valores das caixas de texto (nome e sobrenome) e enviar para o banco de dados através da classe DAL.cs.

1. Duplo clique no botão Inserir;
2. Um método (handler) foi gerado no code behind (Ator.xaml.cs);
3. O método handler será executado sempre que o usuário clicar no botão Inserir;
4. Esse método deve instanciar ter acesso a uma instância da classe DAL e executar o método inserirAtor passando os parâmetros recuperados:

```

DAL dao = new DAL();
1 reference
private void inserirAtorBT_Click(object sender, RoutedEventArgs e)
{
    try
    {
        dao.InserirAtor(nomeTB.Text, sobrenomeTB.Text);
        MessageBox.Show("Ator inserido com sucesso.");
    }
    catch(Exception ex)
    {
        MessageBox.Show("Erro: "+ex.Message);
        return;
    }
}

```

DÚVIDAS?



EXERCÍCIO

1. Implemente a janela `Categoria.xaml`, contendo campos necessários para o cadastro de uma categoria.
2. Implemente a janela `Idioma.xaml`, contendo campos necessários para o cadastro de uma idioma.
3. Implemente na classe `DAL.cs`:
 1. O método `inserirCategoria`;
 2. O método `inserirIdioma`;
4. Implemente na janela `MainWindow.xaml`:
 1. Um botão `Categoria`;
 2. Um botão `Idioma`;
5. Implemente na classe `Categoria.xaml.cs` e `Idioma.xaml.cs` a chamada aos novos métodos da classe `DAL.cs`;

REFERÊNCIAS BIBLIOGRÁFICAS

ADO.NET Overview. Disponível em: [https://msdn.microsoft.com/en-us/library/h43ks021\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/h43ks021(v=vs.110).aspx) >. Acesso em 24 Jan. 2017.

Npgsql: Getting Started. Disponível em: <http://www.npgsql.org/doc/index.html#getting-started>>. Acesso em 24 Jan. 2017.