

# **Desenvolvimento de Aplicações para Internet**

**Criando um Web Site em  
ASP.NET**

## Exemplo – Banco de Dados (**Tabela: dados**)

	Column Name	Data Type	Allow Nulls
PK	rgm	varchar(7)	<input type="checkbox"/>
	nome	varchar(50)	<input checked="" type="checkbox"/>
	nota1	float	<input checked="" type="checkbox"/>
	nota2	float	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

# Default.aspx

SISTEMA DE CONTROLE DE NOTAS

Cadastro de Notas      Sobre

MainContent (Custom)

RGM:

Nome:

Nota1:

Nota2:

Incluir   Alterar   Excluir

[lblMensagem]

```
<asp:Content ID="BodyContent" runat="server" ContentPlaceHolderID="MainContent">
    <div>
        <asp:Label ID="Label1" runat="server" Text="RGM"></asp:Label>
        <asp:TextBox ID="txtRGM" runat="server"></asp:TextBox>
    </div>

    <div>
        <asp:Label ID="Label2" runat="server" Text="Nome"></asp:Label>
        <asp:TextBox ID="txtNome" runat="server"></asp:TextBox>
    </div>

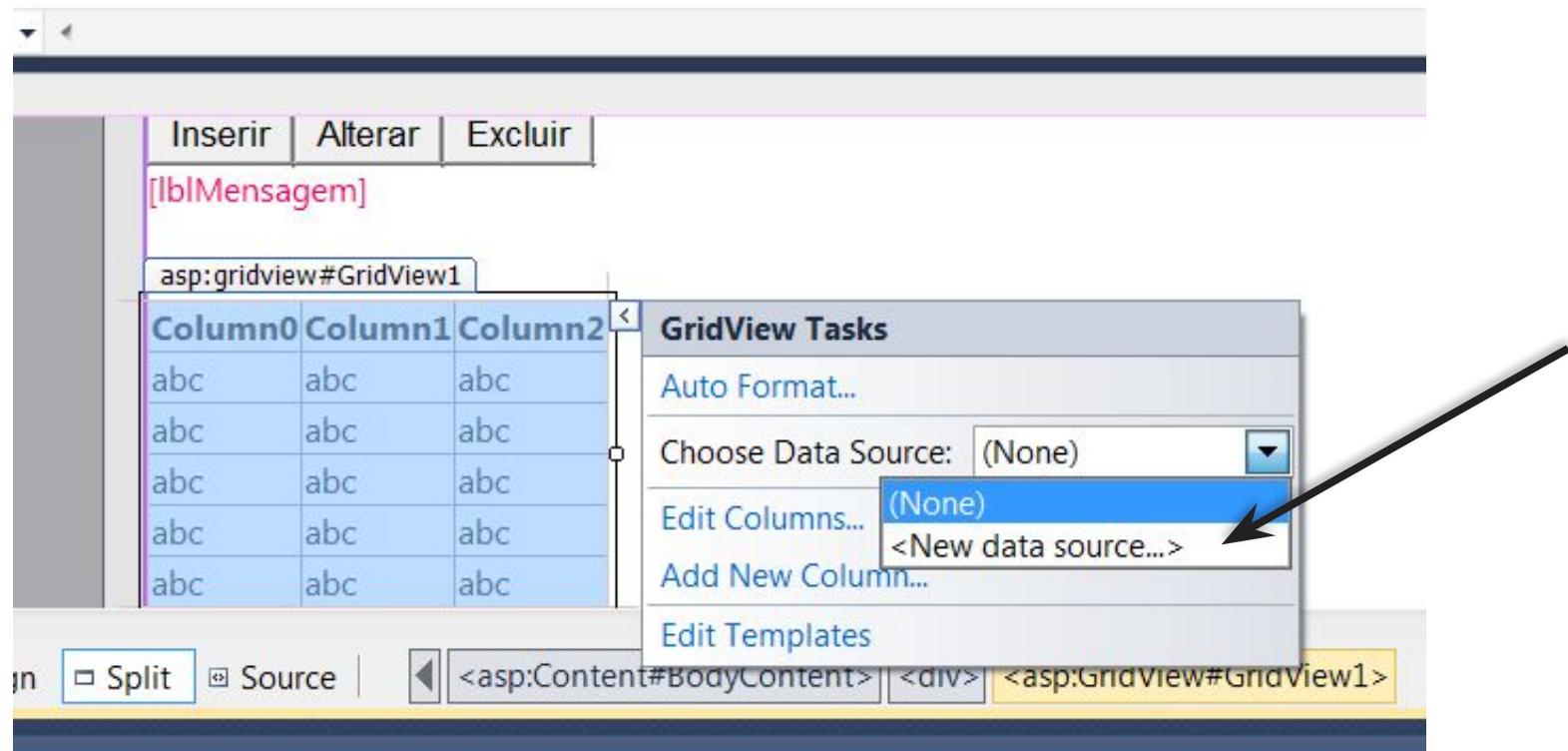
    <div>
        <asp:Label ID="Label3" runat="server" Text="Nota 1"></asp:Label>
        <asp:TextBox ID="txtNota1" runat="server"></asp:TextBox>
    </div>

    <div>
        <asp:Label ID="Label4" runat="server" Text="Nota 2"></asp:Label>
        <asp:TextBox ID="txtNota2" runat="server"></asp:TextBox>
    </div>

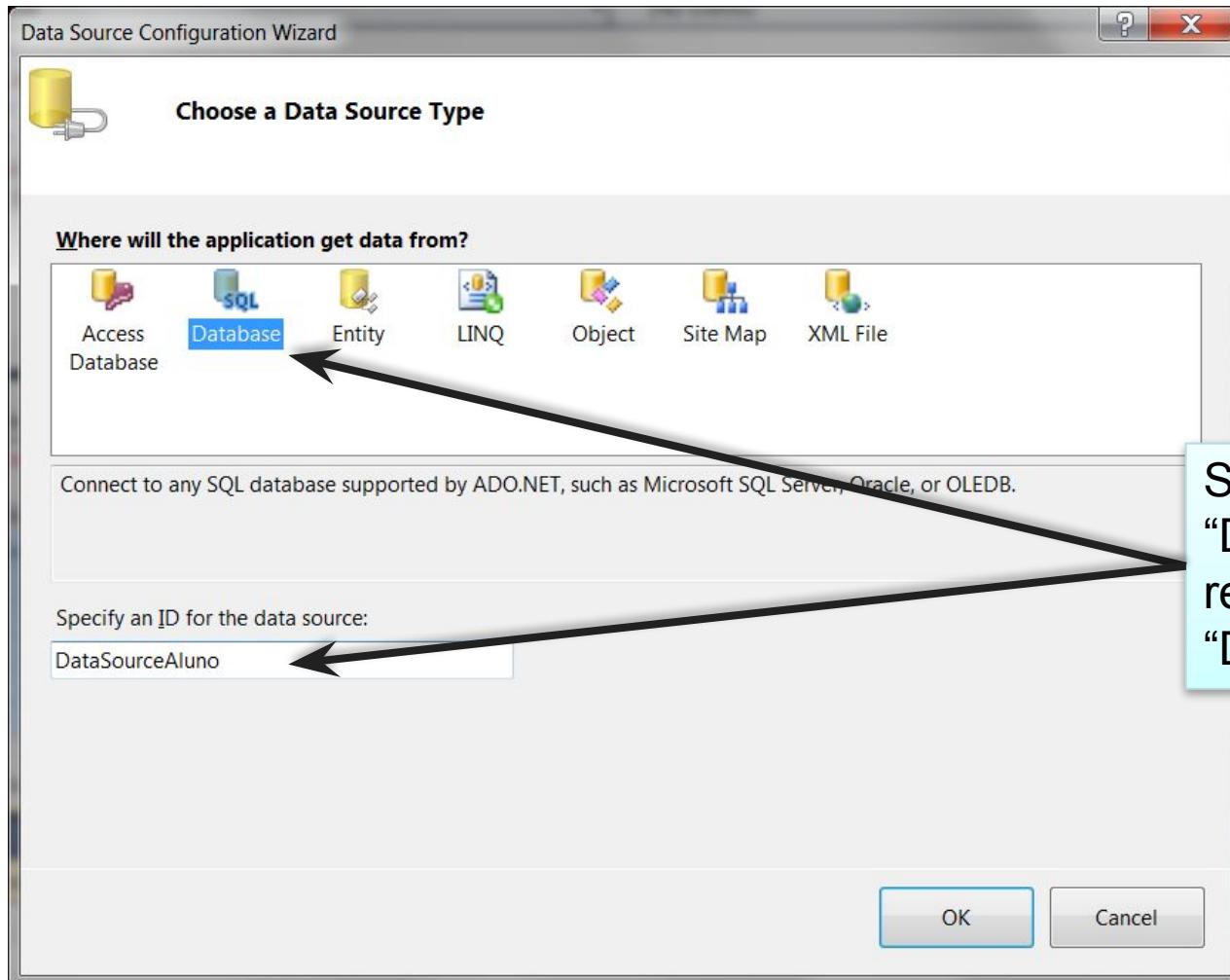
    <div>
        <asp:Button ID="Button1" runat="server" Text="Inserir" />
        <asp:Button ID="Button2" runat="server" Text="Alterar" />
        <asp:Button ID="Button3" runat="server" Text="Excluir" />
    </div>
</asp:Content>
```

# DataGridView

```
<asp:GridView ID="GridView1" runat="server">
</asp:GridView>
</div>
</asp:Content>
```

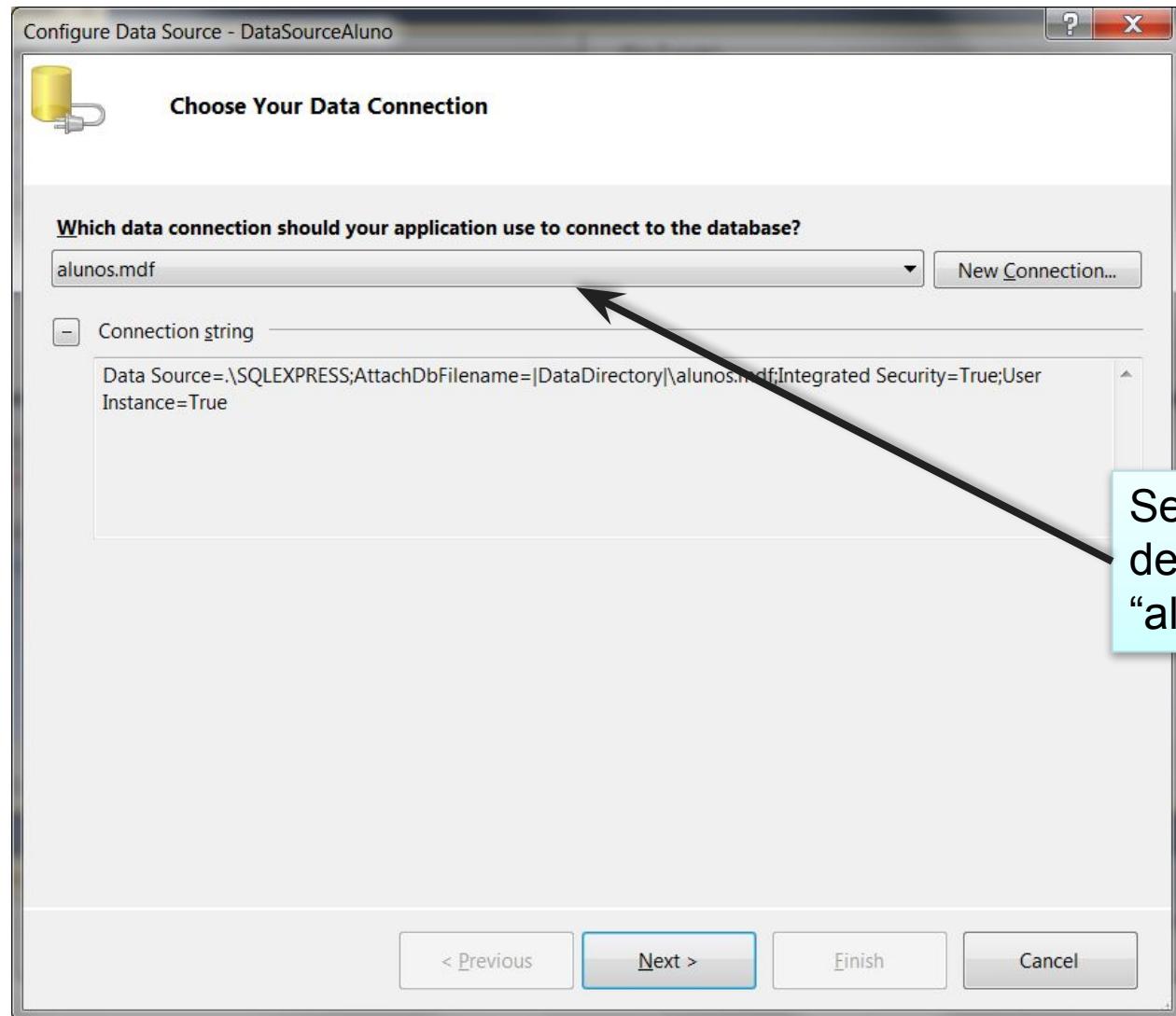


# DataGridView



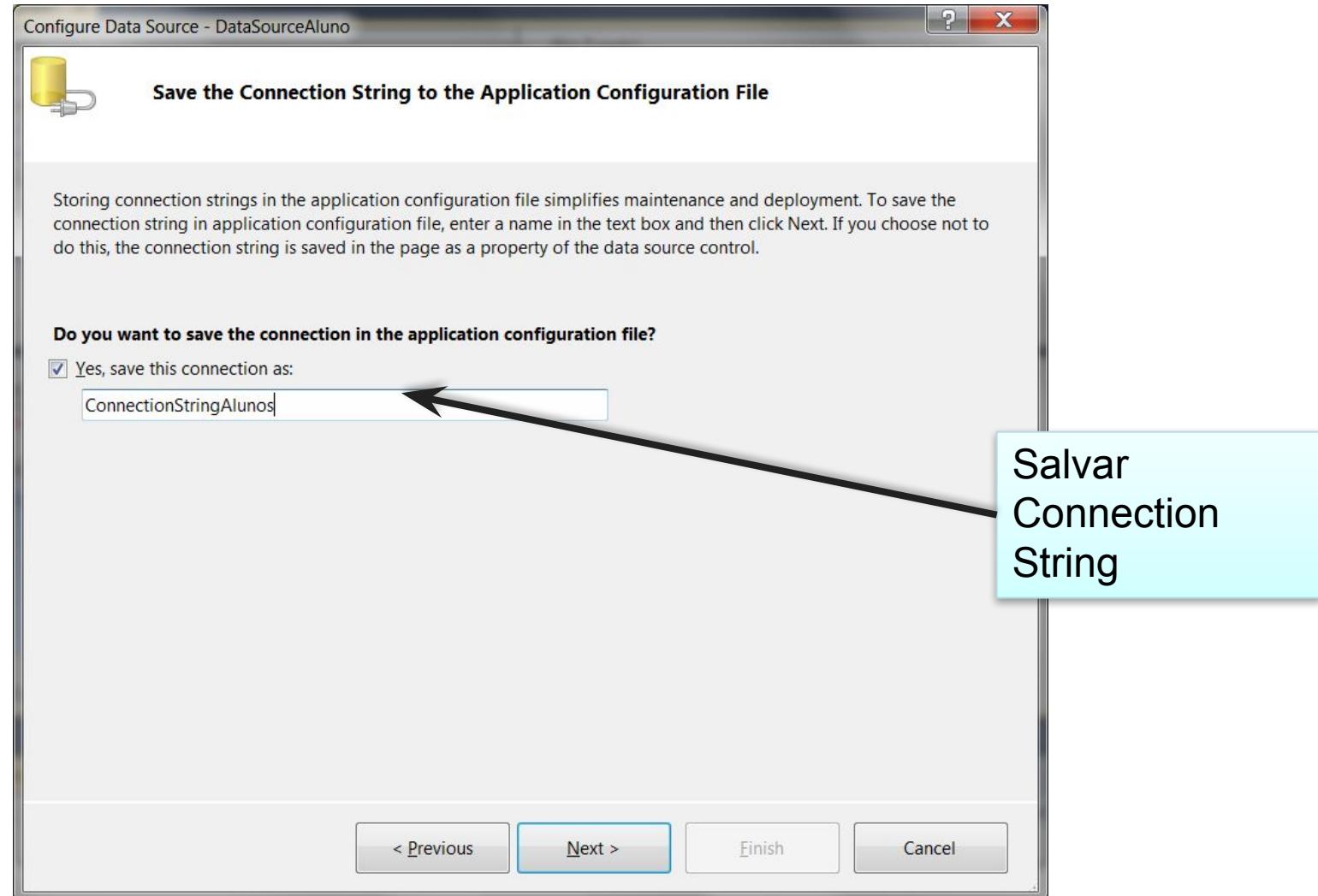
Selecionar  
“Database” e  
renomear para  
“DataSourceAluno”

# DataGridView

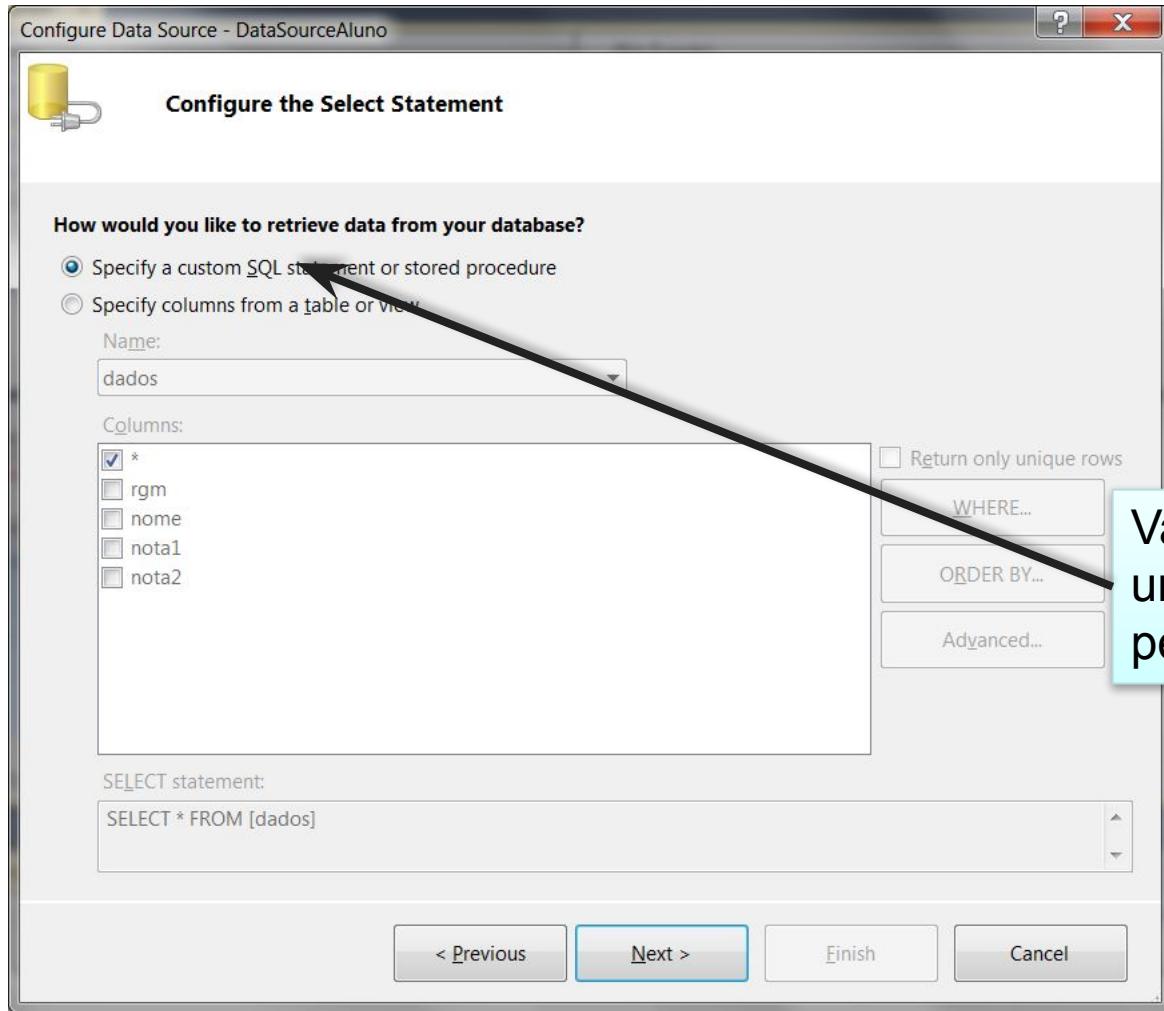


Selecionar o banco  
de dados do projeto  
“alunos.mdf”

# DataGridView

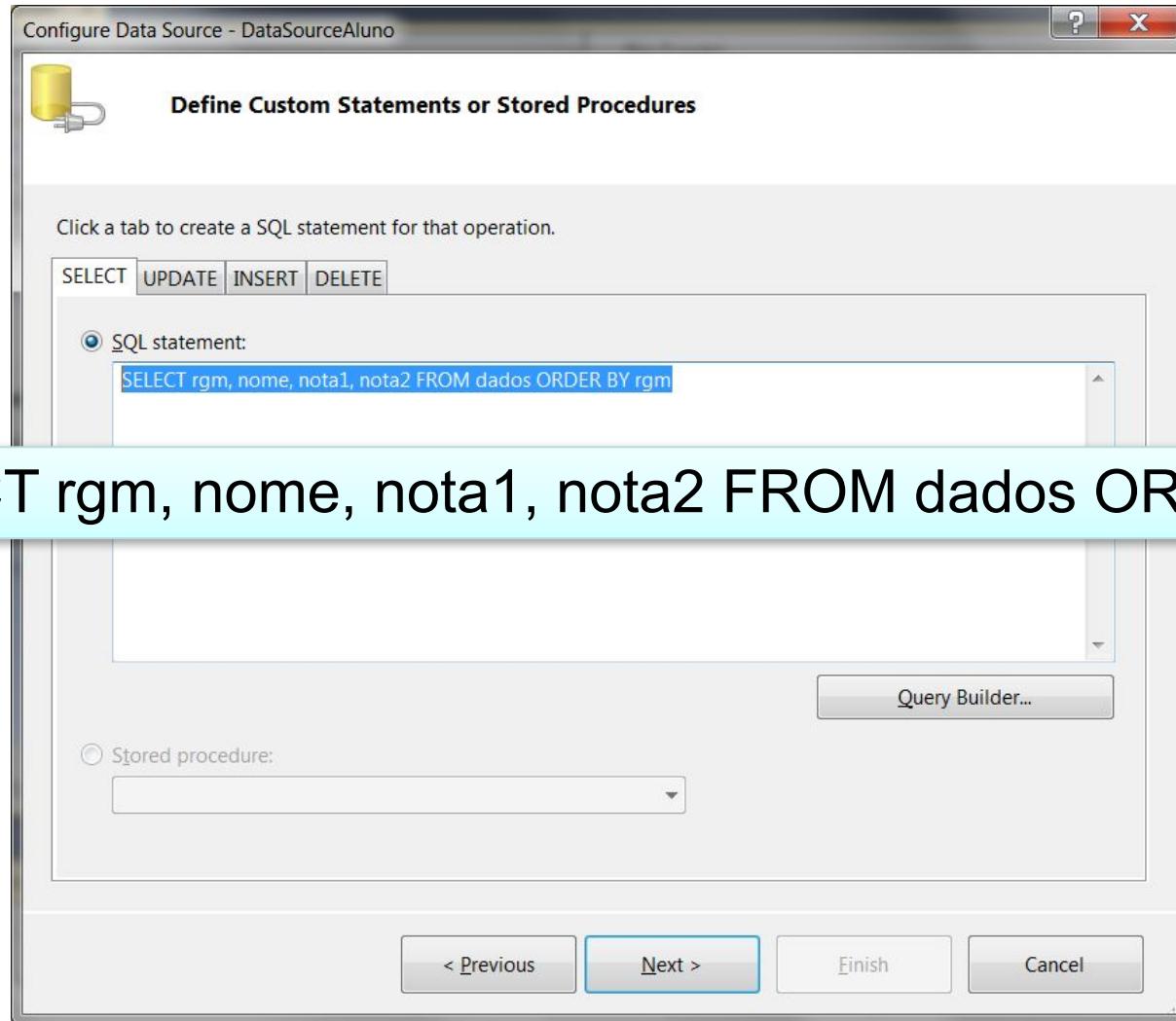


# DataGridView

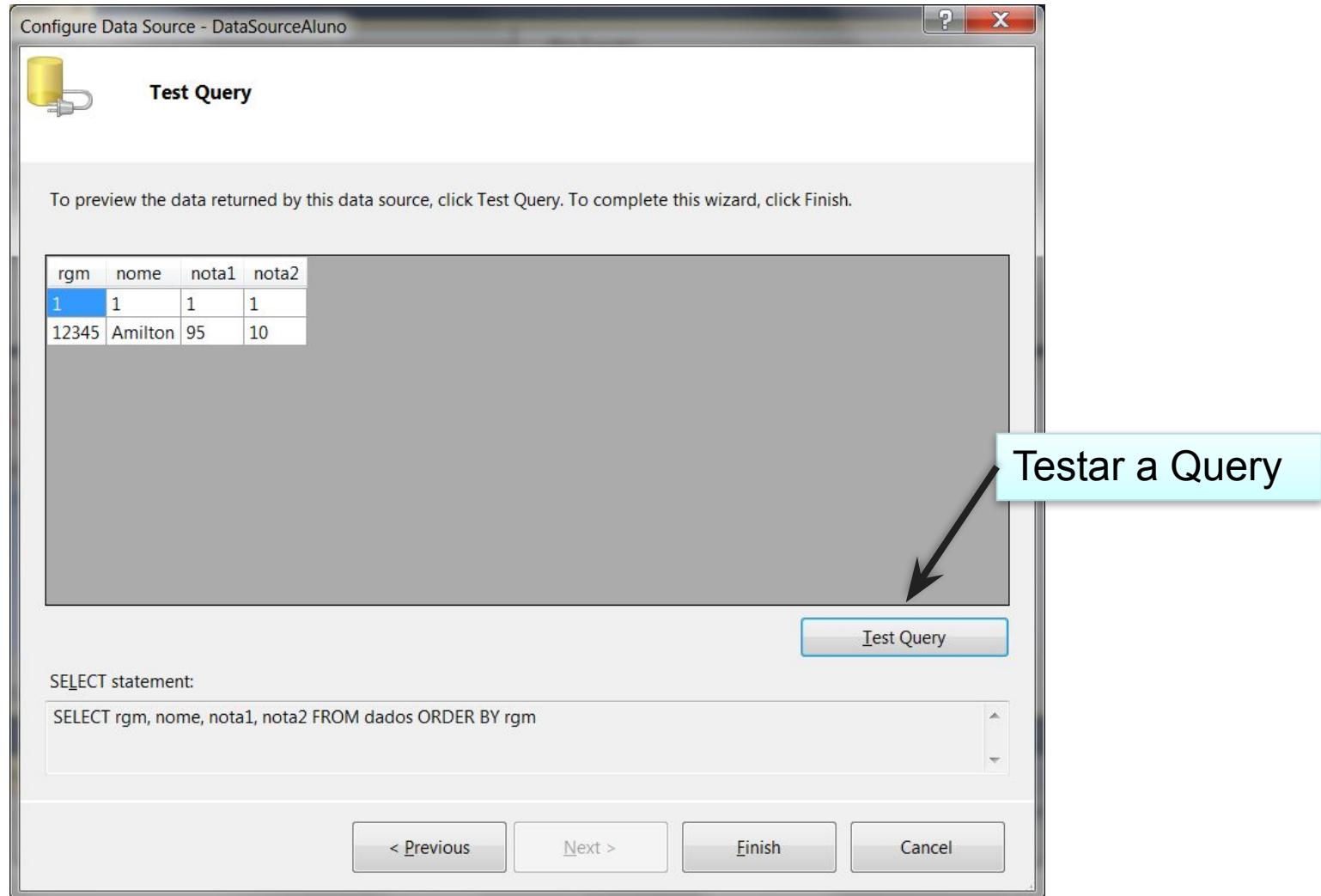


Vamos montar  
uma Query  
personalizada

# DataGridView



# DataGridView



# DataGridView

## SISTEMA DE CONTROLE DE NOTAS

Cadastro de Notas      Sobre

RGM

Nome

Nota 1

Nota 2

rgm	nome	nota1	nota2
1	1	1	1
12345	Amilton	95	10

# Atualizar DataGridView

```
Aluno aluno = new Aluno(txtRGM.Text, txtNome.Text, float.Parse(txtM  
String retorno = new AlunoDAO().salvarAluno(aluno);

if (retorno == "OK")
{
    lblMensagem.Text = "Sucesso na inserção";
    GridView1.DataBind();
}
else
{
    lblMensagem.Text = retorno;
}
```

```
Aluno aluno = new Aluno(txtRGM.Text, txtNome.Text, float.Parse(txtM  
String retorno = new AlunoDAO().atualizarAluno(aluno);
```

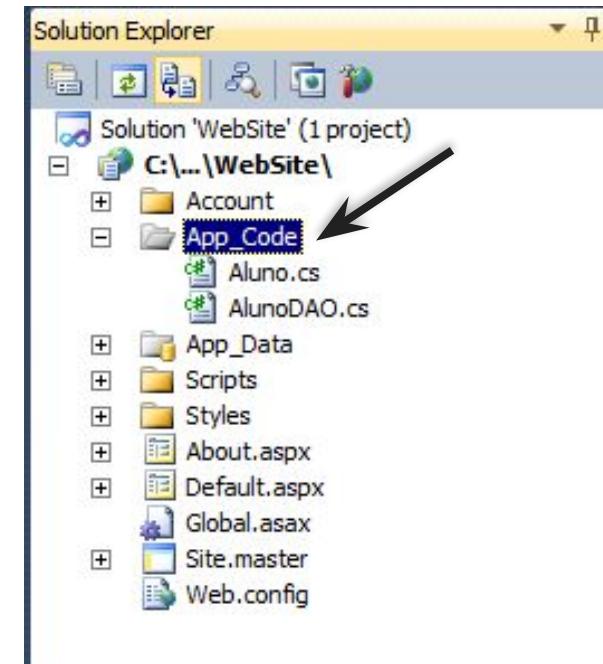
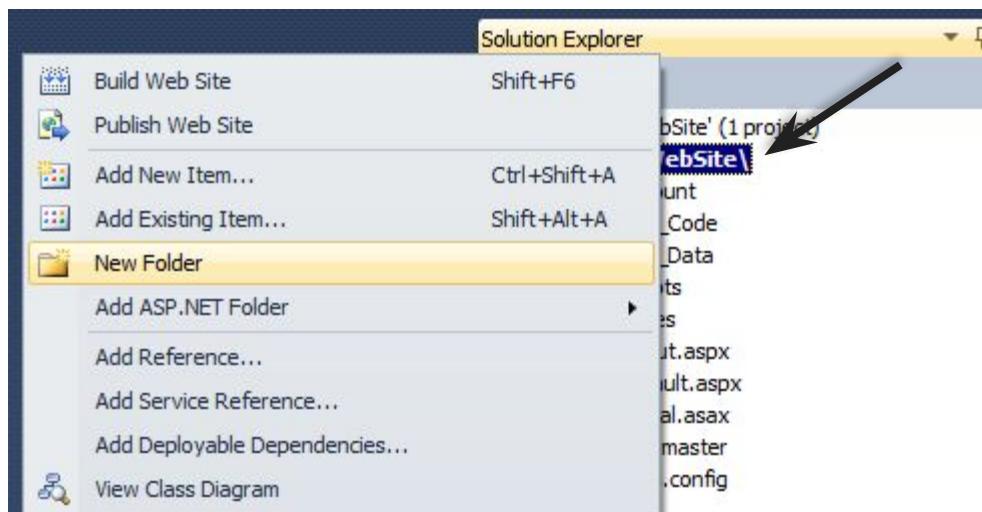
```
if (retorno == "OK")
{
    lblMensagem.Text = "Sucesso na alteração";
    GridView1.DataBind();
}
else
{
    lblMensagem.Text = retorno;
}
```

GridView1.DataBind();

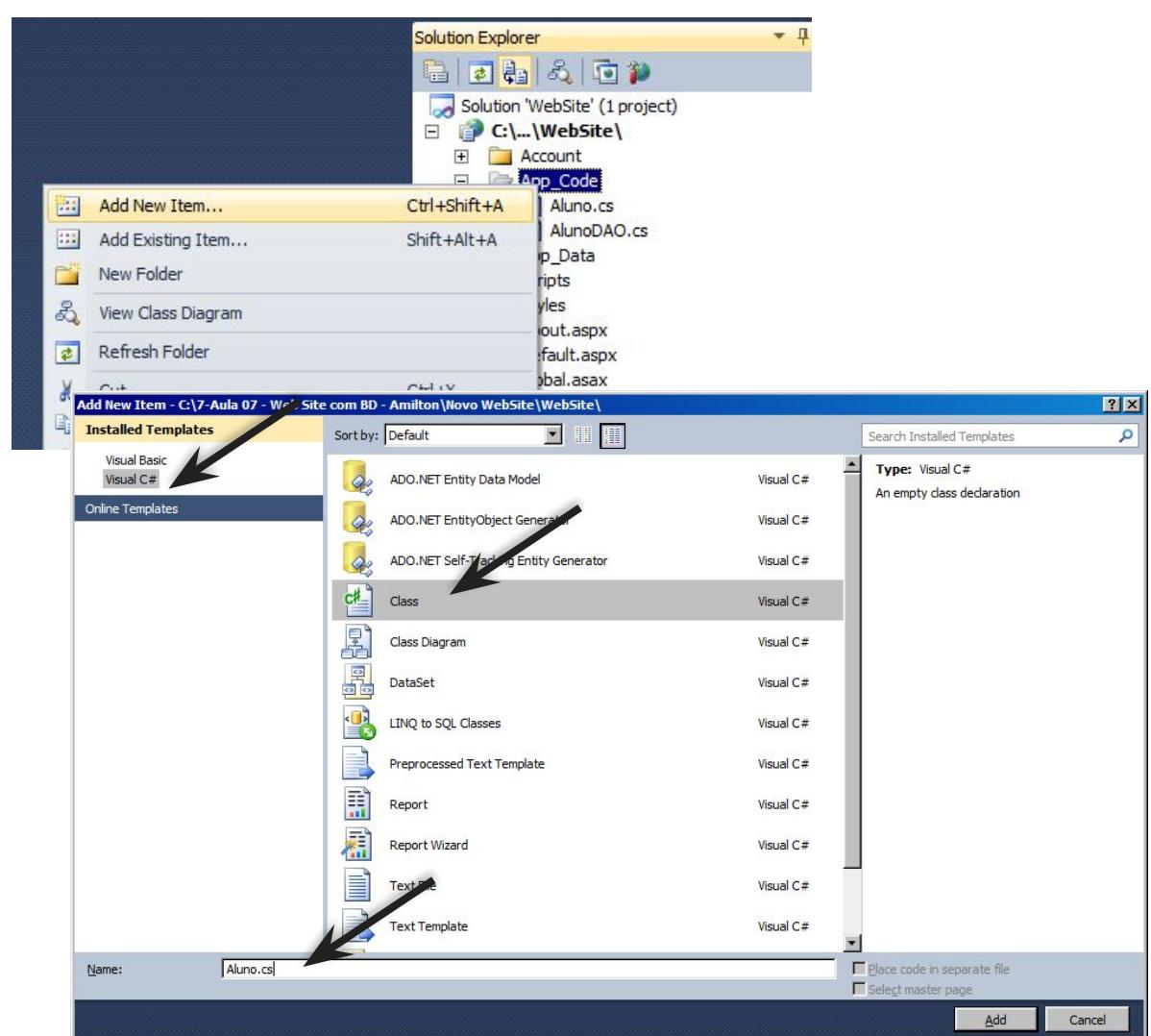
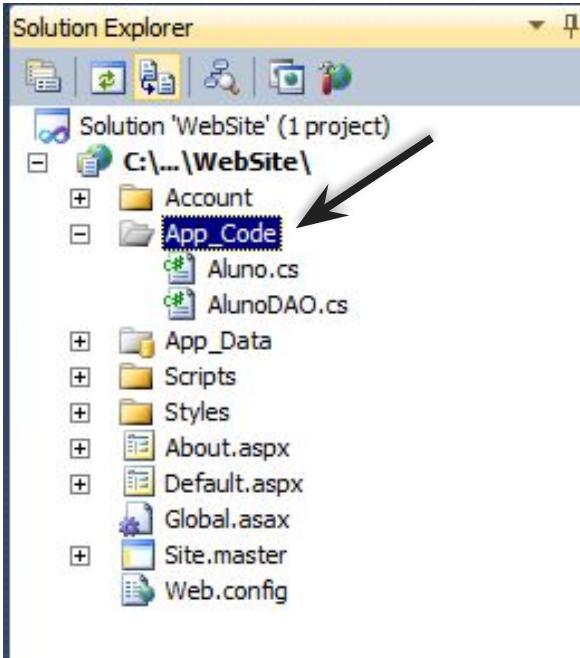
```
Aluno aluno = new Aluno();
aluno.RGM = rgm;
String retorno = new AlunoDAO().excluirAluno(aluno);

if (retorno == "OK")
{
    lblMensagem.Text = "Sucesso na exclusão";
    GridView1.DataBind();
}
else
{
    lblMensagem.Text = retorno;
}
```

# Criar uma pasta nova (App\_Code) para alocar as classes que iremos criar



# Criar uma classe nova Aluno.cs no diretório App\_Code



# Classe Aluno

The screenshot shows a code editor window titled "App\_Code/Aluno.cs\*". The window displays the following C# code:

```
Aluno
/// <summary>
/// Summary description for Aluno
/// </summary>
public class Aluno
{
    public Aluno()
    {

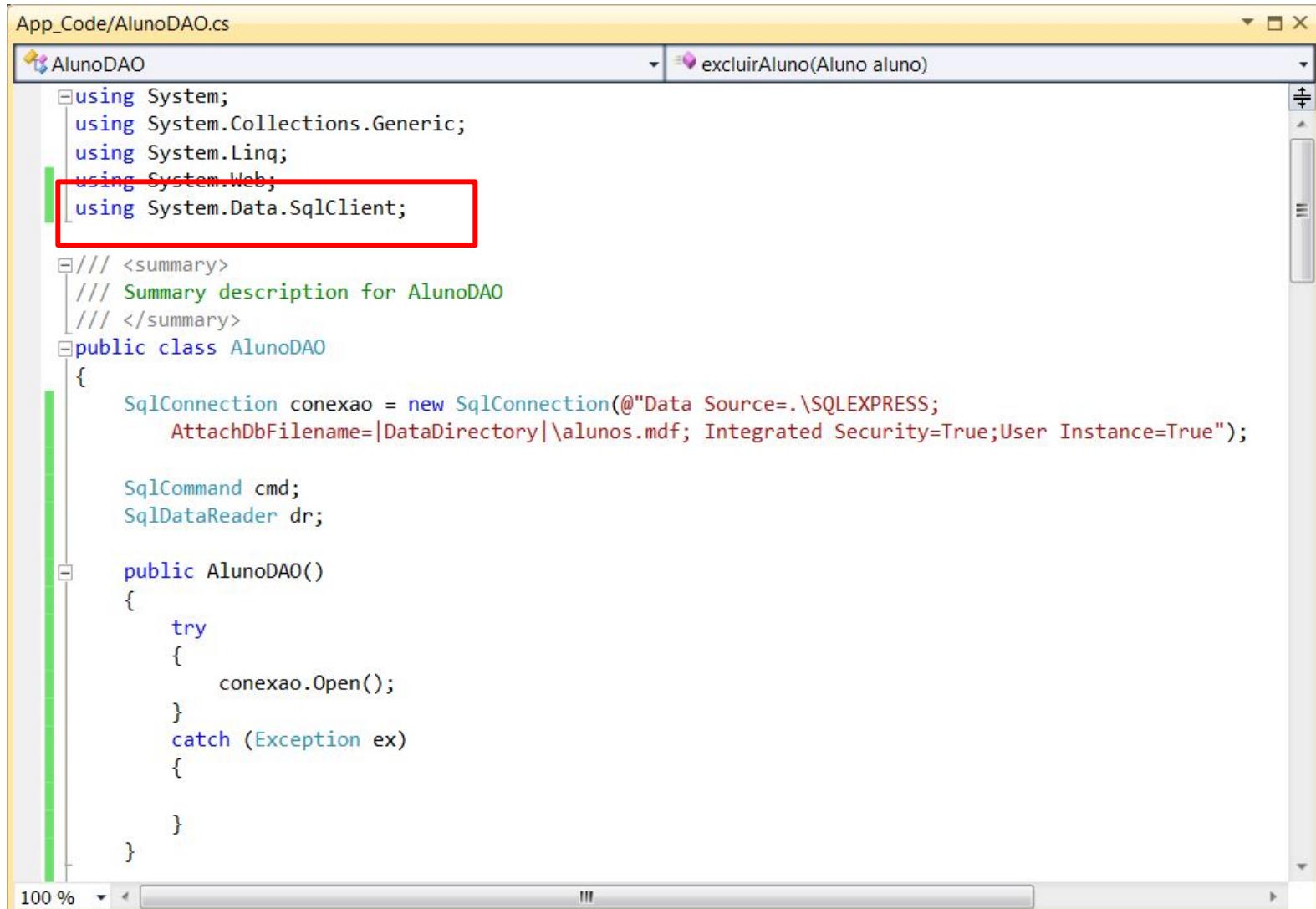
    }

    public Aluno(string rgm, string nome, float nota1, float nota2)
    {
        this.RGM = rgm;
        this.Nome = nome;
        this.Nota1 = nota1;
        this.Nota2 = nota2;
    }

    public string RGM { set; get; }
    public string Nome { set; get; }
    public float Nota1 { set; get; }
    public float Nota2 { set; get; }
}
```

The code editor interface includes a toolbar at the top with icons for file operations, and a status bar at the bottom showing "100 %".

# Classe AlunoDAO - Conexão



The screenshot shows a code editor window with the following details:

- Title Bar:** App\_Code/AlunoDAO.cs
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and Close.
- Code Area:** Displays the C# code for the AlunoDAO class. A red rectangle highlights the `using System.Data.SqlClient;` statement.
- Toolbars:** On the right side, there are several toolbars with icons for file operations, search, and other development tools.
- Status Bar:** Shows "100 %".

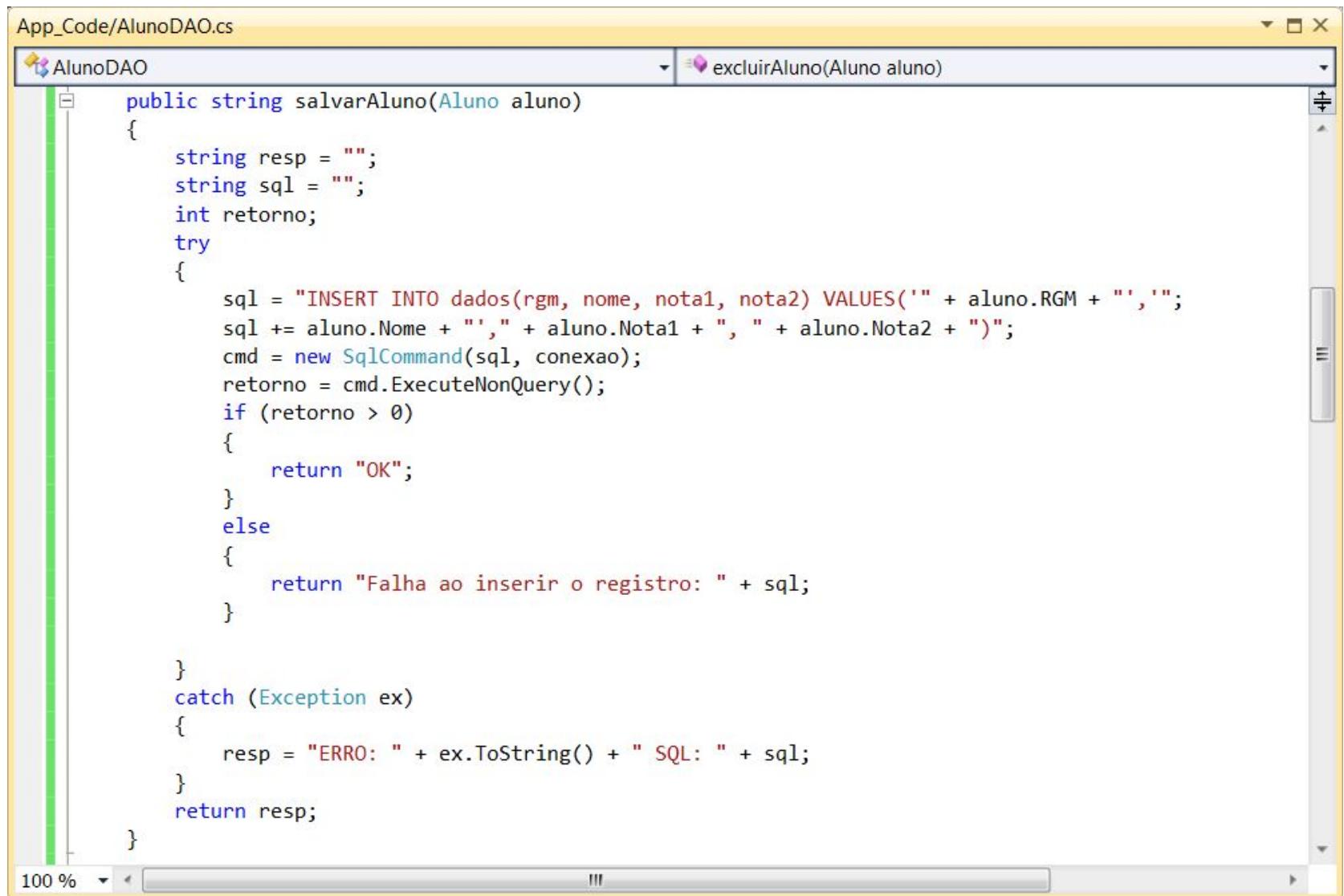
```
App_Code/AlunoDAO.cs
AlunoDAO
AlunoDAO.cs
public class AlunoDAO
{
    SqlConnection conexao = new SqlConnection(@"Data Source=.\SQLEXPRESS;
        AttachDbFilename=|DataDirectory|\alunos.mdf; Integrated Security=True;User Instance=True");

    SqlCommand cmd;
    SqlDataReader dr;

    public AlunoDAO()
    {
        try
        {
            conexao.Open();
        }
        catch (Exception ex)
        {
        }
    }

    public void excluirAluno(Aluno aluno)
    {
        // Implementation of the method
    }
}
```

# Classe AlunoDAO - Inclusão

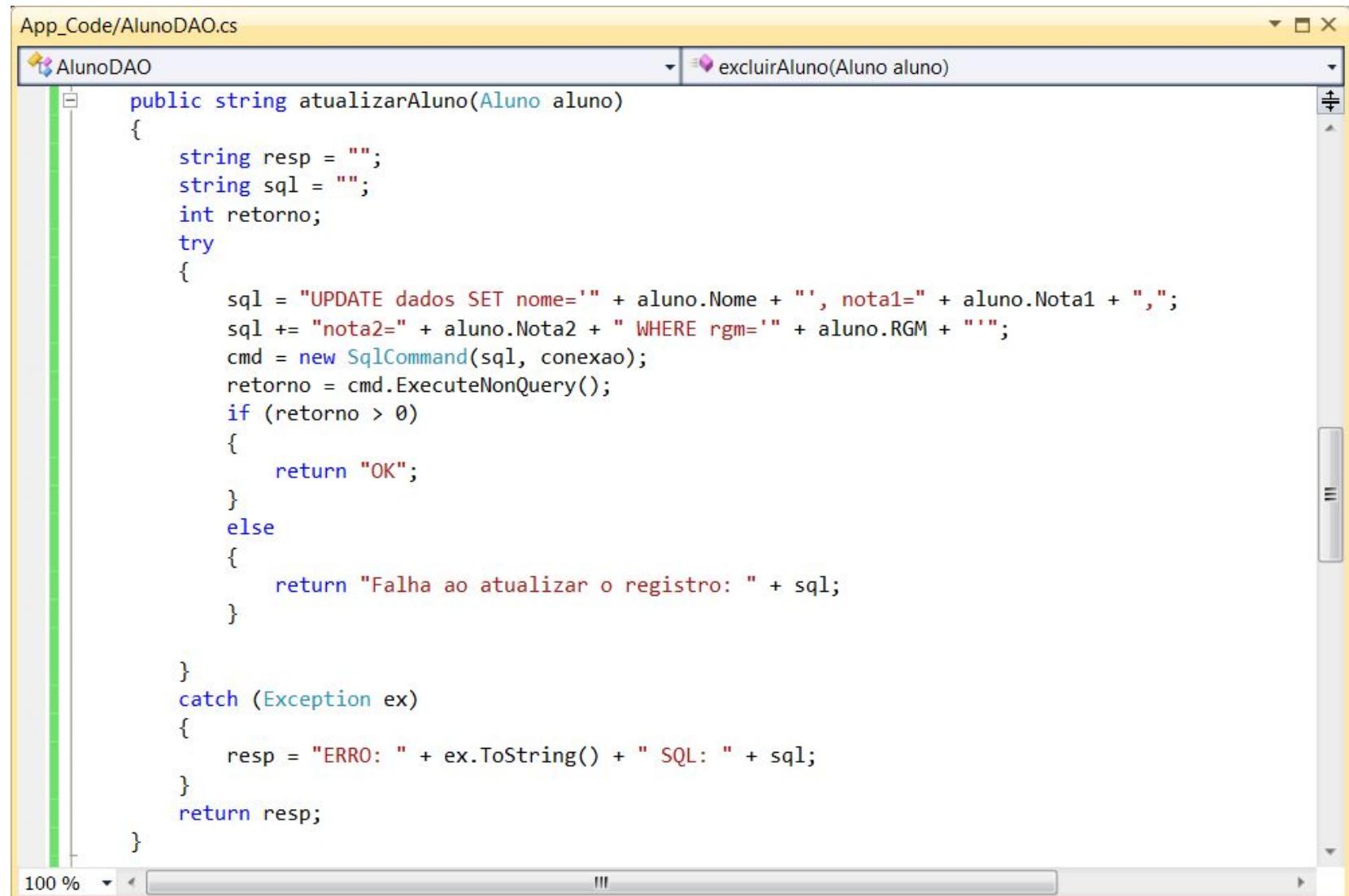


The screenshot shows a code editor window with the following details:

- Title Bar:** App\_Code/AlunoDAO.cs
- Toolbar:** Shows icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Code Area:** Displays the C# code for the `salvarAluno` method of the `AlunoDAO` class. The code handles inserting a new student record into a database table named `dados`.
- Method Signature:** `public string salvarAluno(Aluno aluno)`
- Code Content:**

```
public string salvarAluno(Aluno aluno)
{
    string resp = "";
    string sql = "";
    int retorno;
    try
    {
        sql = "INSERT INTO dados(rgm, nome, nota1, nota2) VALUES('" + aluno.RGM + "','" +
        sql += aluno.Nome + "','" + aluno.Nota1 + "','" + aluno.Nota2 + ")";
        cmd = new SqlCommand(sql, conexao);
        retorno = cmd.ExecuteNonQuery();
        if (retorno > 0)
        {
            return "OK";
        }
        else
        {
            return "Falha ao inserir o registro: " + sql;
        }
    }
    catch (Exception ex)
    {
        resp = "ERRO: " + ex.ToString() + " SQL: " + sql;
    }
    return resp;
}
```
- Status Bar:** Shows "100 %".

# Classe AlunoDAO - Alteração

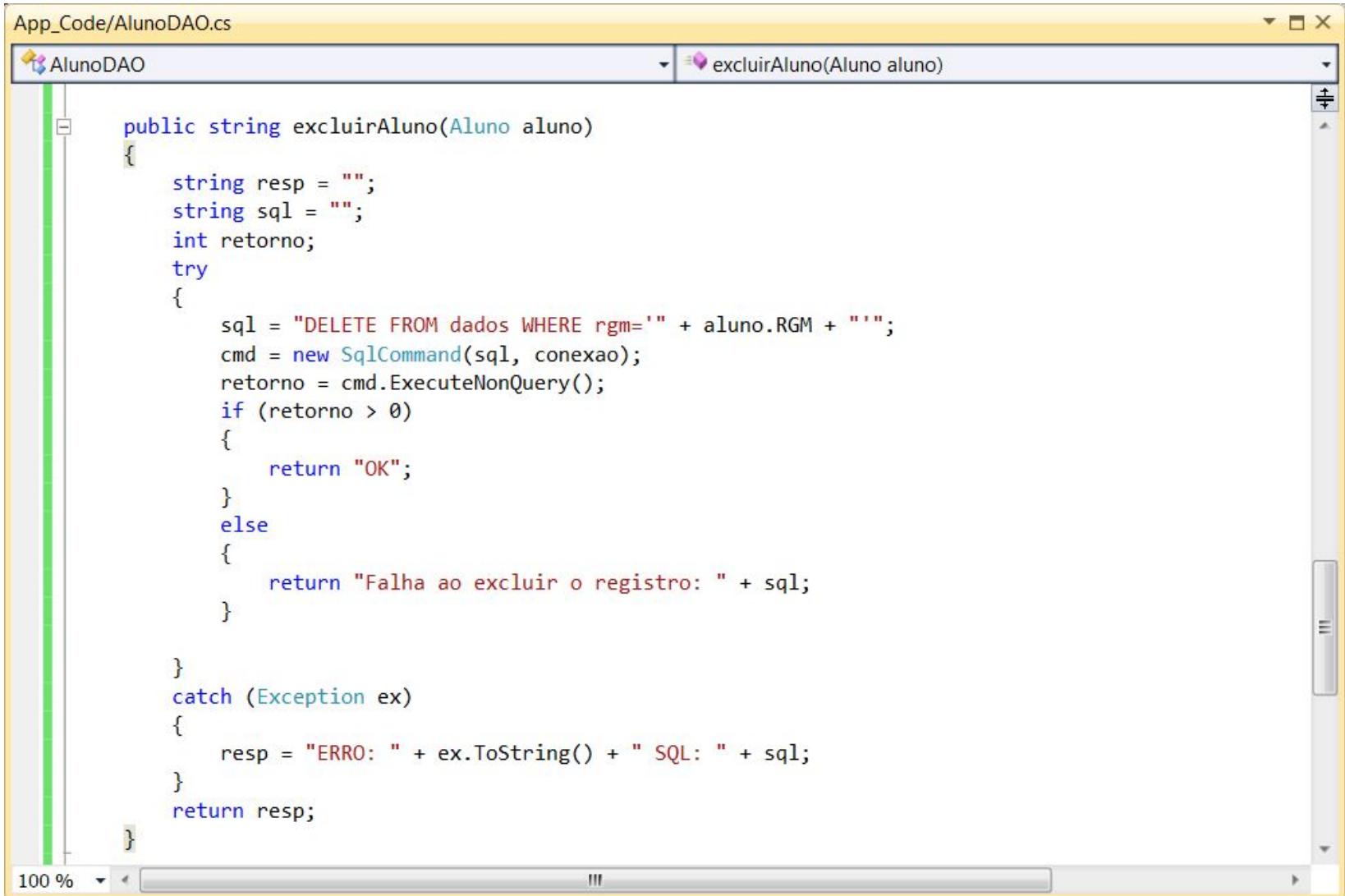


The screenshot shows a code editor window with the following details:

- Title Bar:** App\_Code/AlunoDAO.cs
- Toolbar:** Includes icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Code Area:** Displays the C# code for the `AlunoDAO` class, specifically the `atualizarAluno` method.
- Method Signature:** `public string atualizarAluno(Aluno aluno)`
- Code Content:**

```
public string atualizarAluno(Aluno aluno)
{
    string resp = "";
    string sql = "";
    int retorno;
    try
    {
        sql = "UPDATE dados SET nome='" + aluno.Nome + "', nota1=" + aluno.Nota1 + ",";
        sql += "nota2=" + aluno.Nota2 + " WHERE rgm='" + aluno.RGM + "'";
        cmd = new SqlCommand(sql, conexao);
        retorno = cmd.ExecuteNonQuery();
        if (retorno > 0)
        {
            return "OK";
        }
        else
        {
            return "Falha ao atualizar o registro: " + sql;
        }
    }
    catch (Exception ex)
    {
        resp = "ERRO: " + ex.ToString() + " SQL: " + sql;
    }
    return resp;
}
```
- Status Bar:** Shows "100 %".

# Classe AlunoDAO - Exclusão

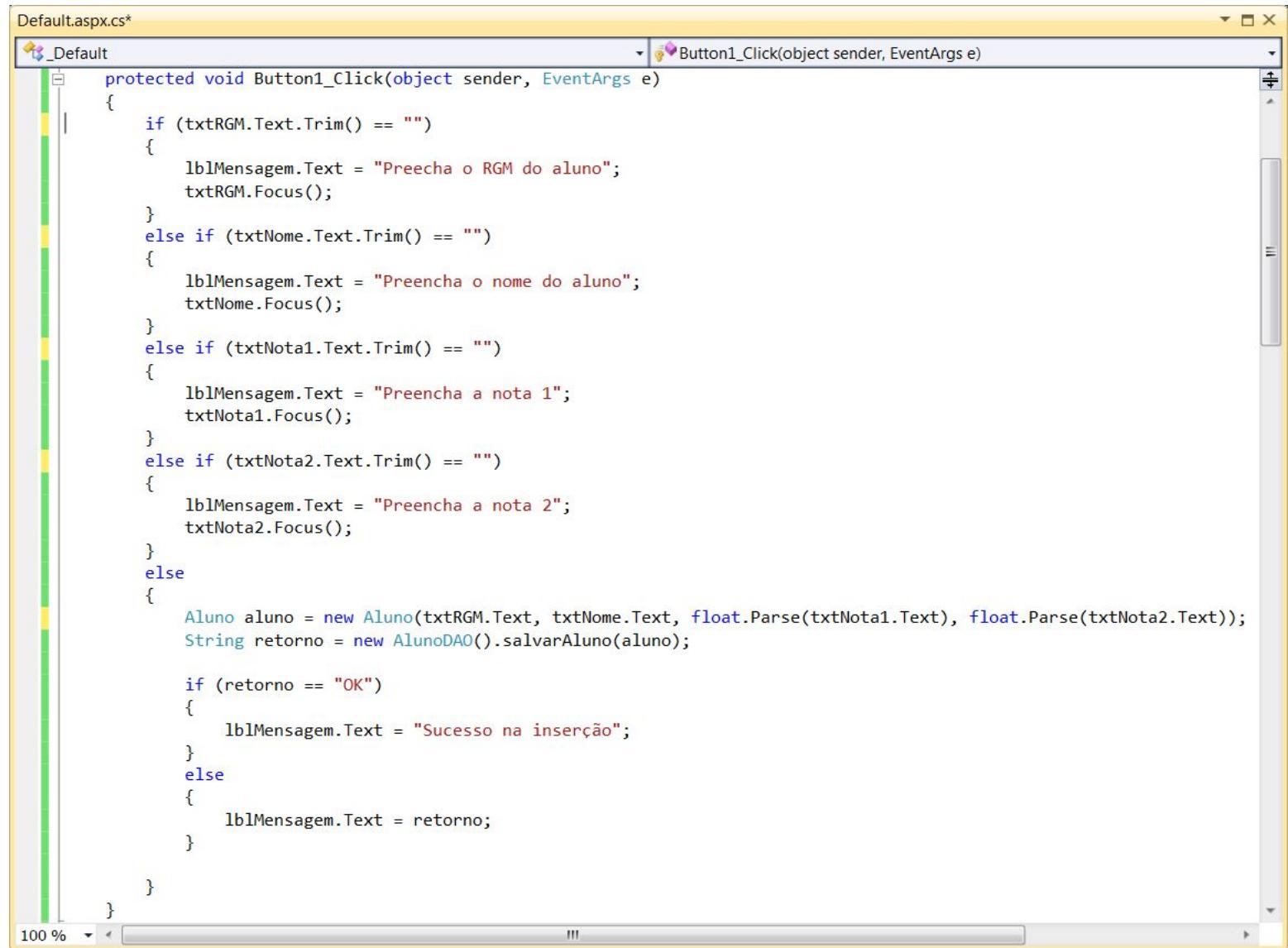


The screenshot shows a code editor window with the following details:

- Title Bar:** App\_Code/AlunoDAO.cs
- Toolbar:** Shows icons for Save, Undo, Redo, Cut, Copy, Paste, Find, Replace, and others.
- Code Area:** Displays the C# code for the `excluirAluno` method in the `AlunoDAO` class.
- Method Signature:** `public string excluirAluno(Aluno aluno)`
- Code Content:**

```
public string excluirAluno(Aluno aluno)
{
    string resp = "";
    string sql = "";
    int retorno;
    try
    {
        sql = "DELETE FROM dados WHERE rgm='" + aluno.RGM + "'";
        cmd = new SqlCommand(sql, conexao);
        retorno = cmd.ExecuteNonQuery();
        if (retorno > 0)
        {
            return "OK";
        }
        else
        {
            return "Falha ao excluir o registro: " + sql;
        }
    }
    catch (Exception ex)
    {
        resp = "ERRO: " + ex.ToString() + " SQL: " + sql;
    }
    return resp;
}
```
- Status Bar:** Shows "100 %".

# Default.aspx – Botão Inserir

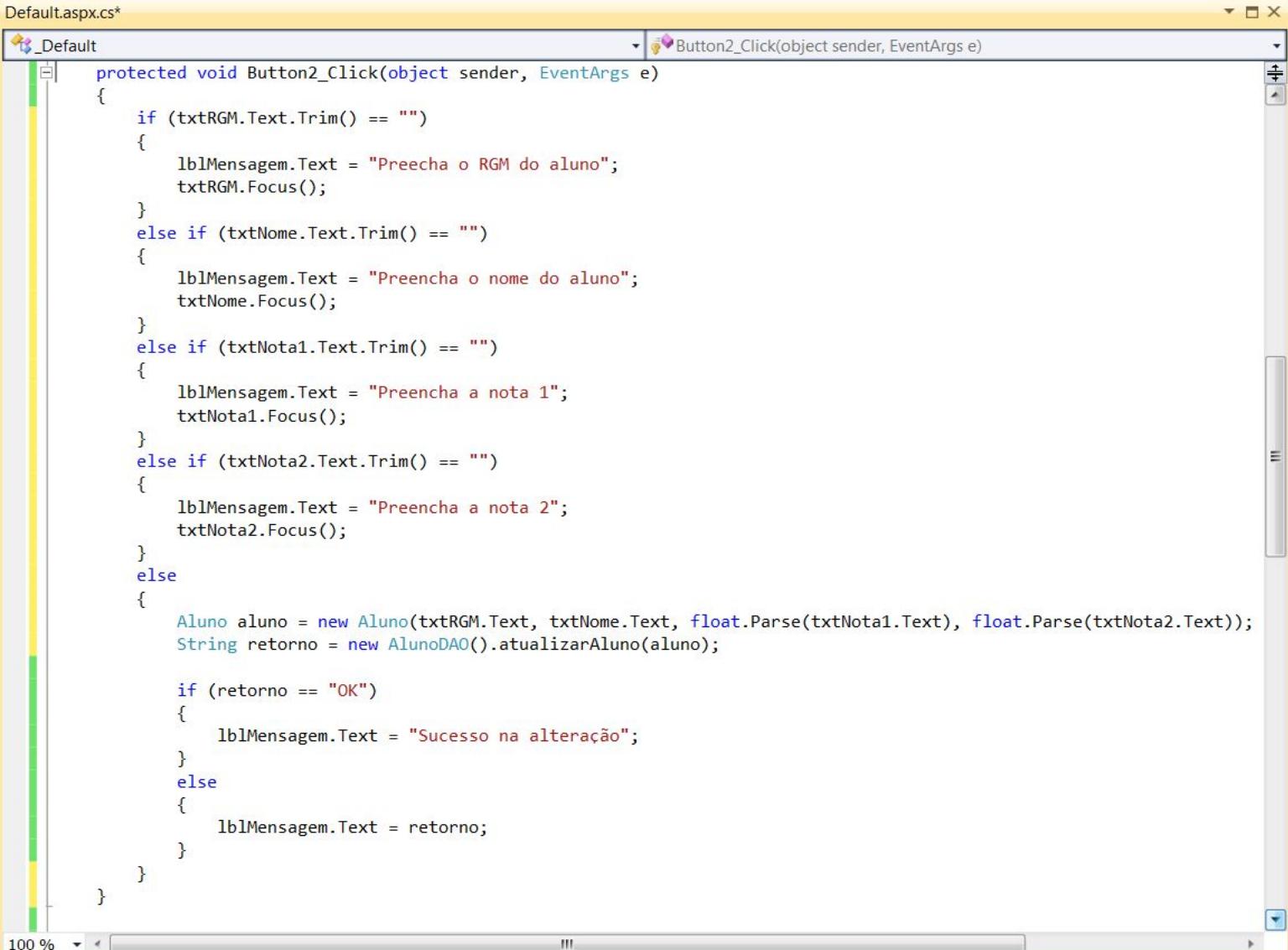


The screenshot shows the Microsoft Visual Studio IDE with the code editor open for the file Default.aspx.cs\*. The title bar indicates the file name and the current event handler being edited: Button1\_Click(object sender, EventArgs e). The code itself is written in C# and handles the click event for the 'Button1' control. It first checks if the 'txtRGM' text box is empty, displaying a message and focusing on it if so. It then checks if the 'txtNome' text box is empty, displaying a message and focusing on it if so. It also checks if the 'txtNota1' text box is empty, displaying a message and focusing on it if so. If none of these fields are empty, it creates a new 'Aluno' object with the values from the text boxes and calls a 'salvarAluno' method from a 'AlunoDAO' class. It then checks the return value ('OK') and sets the 'lblMensagem' label accordingly.

```
Default.aspx.cs*
_Default
protected void Button1_Click(object sender, EventArgs e)
{
    if (txtRGM.Text.Trim() == "")
    {
        lblMensagem.Text = "Preecha o RGM do aluno";
        txtRGM.Focus();
    }
    else if (txtNome.Text.Trim() == "")
    {
        lblMensagem.Text = "Preencha o nome do aluno";
        txtNome.Focus();
    }
    else if (txtNota1.Text.Trim() == "")
    {
        lblMensagem.Text = "Preencha a nota 1";
        txtNota1.Focus();
    }
    else if (txtNota2.Text.Trim() == "")
    {
        lblMensagem.Text = "Preencha a nota 2";
        txtNota2.Focus();
    }
    else
    {
        Aluno aluno = new Aluno(txtRGM.Text, txtNome.Text, float.Parse(txtNota1.Text), float.Parse(txtNota2.Text));
        String retorno = new AlunoDAO().salvarAluno(aluno);

        if (retorno == "OK")
        {
            lblMensagem.Text = "Sucesso na inserção";
        }
        else
        {
            lblMensagem.Text = retorno;
        }
    }
}
```

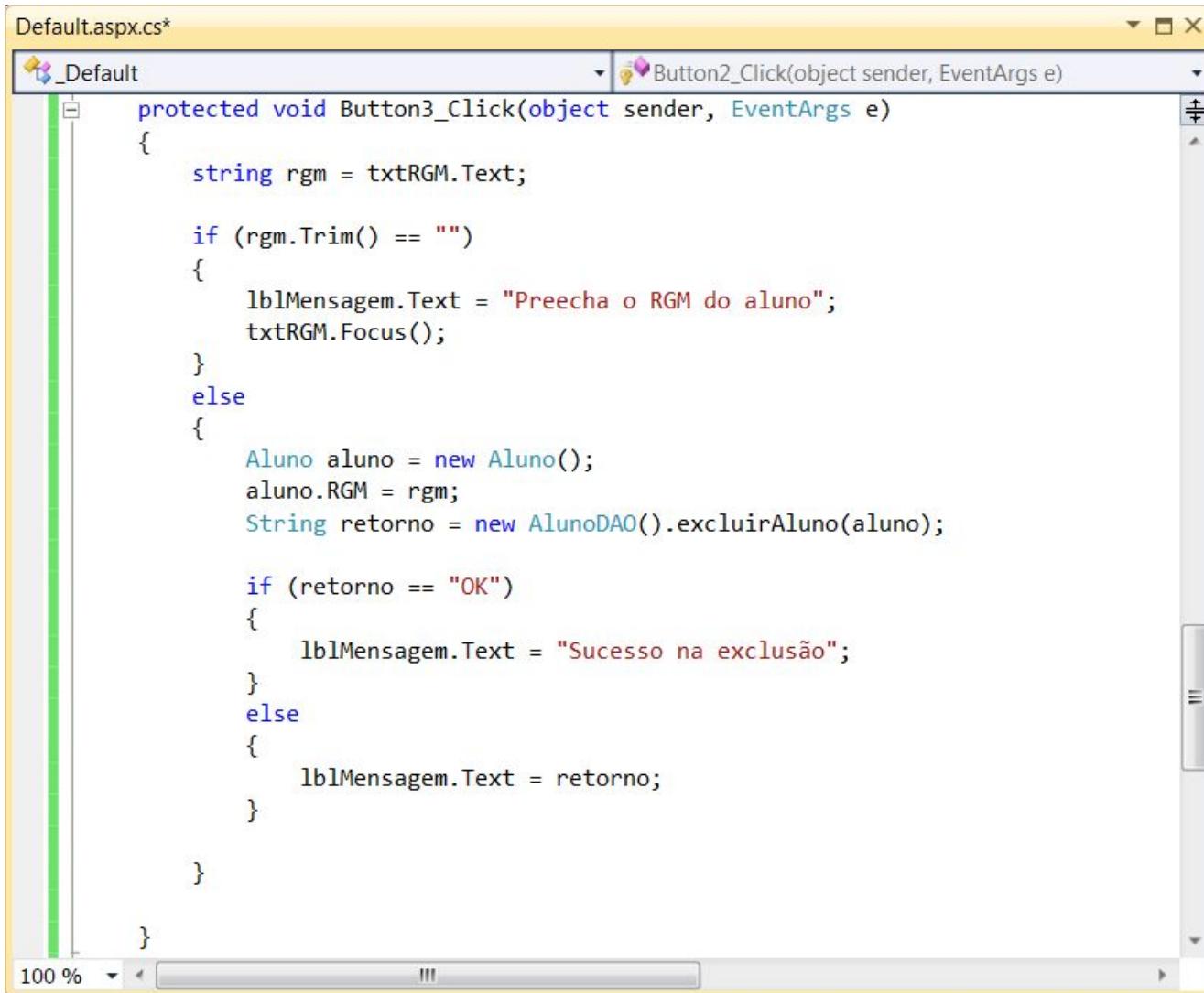
# Default.aspx – Botão Alterar



```
Default.aspx.cs*
_Default
protected void Button2_Click(object sender, EventArgs e)
{
    if (txtRGM.Text.Trim() == "")
    {
        lblMensagem.Text = "Preecha o RGM do aluno";
        txtRGM.Focus();
    }
    else if (txtNome.Text.Trim() == "")
    {
        lblMensagem.Text = "Preencha o nome do aluno";
        txtNome.Focus();
    }
    else if (txtNota1.Text.Trim() == "")
    {
        lblMensagem.Text = "Preencha a nota 1";
        txtNota1.Focus();
    }
    else if (txtNota2.Text.Trim() == "")
    {
        lblMensagem.Text = "Preencha a nota 2";
        txtNota2.Focus();
    }
    else
    {
        Aluno aluno = new Aluno(txtRGM.Text, txtNome.Text, float.Parse(txtNota1.Text), float.Parse(txtNota2.Text));
        String retorno = new AlunoDAO().atualizarAluno(aluno);

        if (retorno == "OK")
        {
            lblMensagem.Text = "Sucesso na alteração";
        }
        else
        {
            lblMensagem.Text = retorno;
        }
    }
}
```

# Default.aspx – Botão Excluir



The screenshot shows the Microsoft Visual Studio code editor with the file "Default.aspx.cs\*" open. The title bar indicates the current file and the active event handler: "\_Default" and "Button2\_Click(object sender, EventArgs e)". The code itself is written in C# and handles the click event for the Button3 control.

```
Default.aspx.cs*
private void _Default_Button2_Click(object sender, EventArgs e)
{
    protected void Button3_Click(object sender, EventArgs e)
    {
        string rgm = txtRGM.Text;

        if (rgm.Trim() == "")
        {
            lblMensagem.Text = "Preecha o RGM do aluno";
            txtRGM.Focus();
        }
        else
        {
            Aluno aluno = new Aluno();
            aluno.RGM = rgm;
            String retorno = new AlunoDAO().excluirAluno(aluno);

            if (retorno == "OK")
            {
                lblMensagem.Text = "Sucesso na exclusão";
            }
            else
            {
                lblMensagem.Text = retorno;
            }
        }
    }
}
```

The code performs the following steps:

- It checks if the text in the `txtRGM` input field is empty. If it is, it sets the message label to "Preecha o RGM do aluno" and focuses on the input field.
- If the text is not empty, it creates a new `Aluno` object and sets its `RGM` property to the value from the input field.
- It then calls the `excluirAluno` method from the `AlunoDAO` class, passing the `aluno` object.
- Finally, it checks the return value of the `excluirAluno` method. If it's "OK", it sets the message label to "Sucesso na exclusão". Otherwise, it sets the message label to the returned string.

# Referências

- Capítulo 1 até Capítulo 10  
DEITEL, H. M. **C#**: Como Programar. São Paulo: Pearson Education do Brasil, 2004