



# Programação Orienta à Objetos II - Aula 3

 **Professor:** Salatiel Luz Marinho

 [salatiel.marinho@docente.unip.br](mailto:salatiel.marinho@docente.unip.br)

## Tabela de Aluno

```
CREATE TABLE Aluno (  
    ID INT PRIMARY KEY,  
    Nome VARCHAR(100) NOT NULL,  
    Idade INT,  
    Curso VARCHAR(100),  
    Email VARCHAR(100) UNIQUE,  
    DataDeMatricula DATE,  
    Endereco VARCHAR(255),  
    Telefone VARCHAR(15)  
);
```

Neste script, a tabela `Aluno` foi expandida para incluir mais colunas que podem ser úteis para armazenar informações sobre os alunos:

- `DataDeMatricula`: Armazena a data em que o aluno se matriculou.
- `Endereco`: Armazena o endereço do aluno.
- `Telefone`: Armazena o número de telefone do aluno.

Essas colunas adicionais ajudam a fornecer um perfil mais completo de cada aluno na tabela. Ajuste os tipos de dados e as restrições conforme necessário, dependendo das necessidades específicas do seu banco de dados.

# Stored Procedure

```
CREATE PROCEDURE InserirAluno
    @Nome VARCHAR(100),
    @Idade INT,
    @Curso VARCHAR(100),
    @Email VARCHAR(100),
    @DataDeMatricula DATE,
    @Endereco VARCHAR(255),
    @Telefone VARCHAR(15)
AS
BEGIN
    INSERT INTO Aluno (Nome, Idade, Curso, Email, DataDeMatricula, Endereco, Telefone)
    VALUES (@Nome, @Idade, @Curso, @Email, @DataDeMatricula, @Endereco, @Telefone);
END;
```

```
CREATE PROCEDURE AtualizarAluno
    @ID INT,
    @Nome VARCHAR(100),
    @Idade INT,
    @Curso VARCHAR(100),
    @Email VARCHAR(100),
    @DataDeMatricula DATE,
    @Endereco VARCHAR(255),
    @Telefone VARCHAR(15)
AS
BEGIN
    UPDATE Aluno
    SET Nome = @Nome,
        Idade = @Idade,
        Curso = @Curso,
        Email = @Email,
        DataDeMatricula = @DataDeMatricula,
        Endereco = @Endereco,
```

```
        Telefone = @Telefone
    WHERE ID = @ID;
END;
```

```
CREATE PROCEDURE ExcluirAluno
    @ID INT
AS
BEGIN
    DELETE FROM Aluno
    WHERE ID = @ID;
END;
```

```
CREATE PROCEDURE ConsultarAlunos
AS
BEGIN
    SELECT Nome, Matricula
    FROM Alunos;
END;
```

## Exemplo de Implementação Stored Procedure - C#

```
using System.Data;
using System.Data.SqlClient;

// ...
//Exemplo Insert
string connectionString = "sua_string_de_conexão";
using (SqlConnection connection = new SqlConnection(connectionString))
{
    connection.Open();
```

```
// Crie um comando para a stored procedure
using (SqlCommand command = new SqlCommand("InserirAluno", connection))
{
    command.CommandType = CommandType.StoredProcedure;

    // Adicione os parâmetros necessários
    command.Parameters.AddWithValue("@Nome", "João");
    command.Parameters.AddWithValue("@Matricula", 12345);

    // Execute a stored procedure
    command.ExecuteNonQuery();
}
}
```

```
using System;
using System.Data;
using System.Data.SqlClient;

// Exemplo Select
string connectionString = "sua_string_de_conexão";
using (SqlConnection connection = new SqlConnection(connectionString))
{
    connection.Open();
    // Crie um comando para a stored procedure
    using (SqlCommand command =
        new SqlCommand("uspEmployeeInfo", connection))
    {
        command.CommandType = CommandType.StoredProcedure;

        // Adicione os parâmetros necessários (por exemplo, @employeeID)
        command.Parameters.AddWithValue("@employeeID", 12
```

```

3);

    // Execute a stored procedure e leia os resultados
    using (SqlDataReader reader = command.ExecuteReader
    ())
    {
        while (reader.Read())
        {
            // Acesse os dados retornados
            string nome =
                reader["firstName"].ToString();
            int matricula =
                Convert.ToInt32(reader["code"]);
            Console.WriteLine($"Nome: {nome}, Matrícula: {matricula}");
        }
    }
}

```

## Exemplo de Implementação DataGridView - C#

```

using (SqlDataAdapter adapter = new SqlDataAdapter(command))
{
    DataTable dataTable = new DataTable();
    adapter.Fill(dataTable);

    dataGridView1.DataSource = dataTable;
}

```

## 1. Selecionar a Linha no DataGridView:C#

- Certifique-se de que o `DataGridView` está configurado com a propriedade `SelectionMode` definida como `FullRowSelect` ou `RowHeaderSelect`. Isso permite que o usuário selecione a linha inteira.
- Para obter a linha selecionada, você pode usar a propriedade `SelectedRows`. Por exemplo:

```
DataGridViewRow selectedRow =  
dataGridView1.SelectedRows[0];
```

## 2. Acessar o Conteúdo da Linha Selecionada:C#

- Para obter o conteúdo de células específicas na linha selecionada, você pode acessar as células da linha. Por exemplo, se você tiver colunas chamadas "Nome" e "Matrícula":

```
string nome =  
selectedRow.Cells["Nome"].Value.ToString();  
int matricula =  
Convert.ToInt32(selectedRow.Cells["Matricula"].Value);
```

## 3. Exibir o Conteúdo em um TextBox ou MessageBox:C#

- Você pode exibir o conteúdo em um `TextBox` ou em uma caixa de mensagem (`MessageBox`). Por exemplo:

```
textBoxNome.Text = nome;  
textBoxMatricula.Text = matricula.ToString();  
  
// Ou exiba em uma MessageBox:  
MessageBox.Show($"Nome: {nome}, Matrícula: {matricula}");
```

# Exemplo de Método para validar e-mail - C#

```
static bool IsValidEmail(string email)
{
    string emailPattern =
@"^[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$";
    if (string.IsNullOrEmpty(email))
        return false;

    Regex regex = new Regex(emailPattern);
    return regex.IsMatch(email);
}
```

## 1. Impedir a Digitação de Números:

- No evento `KeyPress`, você pode verificar se o caractere digitado é um número e, se for, impedir que ele seja inserido no campo de texto. Aqui está um exemplo:

```
private void textBox1_KeyPress(object sender,
    KeyPressEventArgs e)
{
    if (char.IsDigit(e.KeyChar) ||
        e.KeyChar == (char)Keys.Back)
    {
        // Permite apenas letras e
        // a tecla Backspace (para apagar)
        e.Handled = false;
    }
    else
    {
        // Bloqueia outros caracteres
        e.Handled = true;
    }
}
```

```
}  
}
```

## 2. Impedir a Digitação de Letras e Caracteres Especiais:

- Se você deseja permitir apenas números e espaços, pode usar o seguinte código:

```
private void textBox1_KeyPress(object sender,  
KeyPressEventArgs e)  
{  
    if (!char.IsDigit(e.KeyChar) &&  
        e.KeyChar != (char)Keys.Back &&  
        e.KeyChar != ' ' )  
    {  
        // Bloqueia letras e caracteres especiais  
        e.Handled = true;  
    }  
}
```