

## Faça como eu fiz

Chegou a hora de você pôr em prática o que foi visto na aula. Para isso, execute os passos listados abaixo.

---

1) Programe os métodos de inclusão, alteração, exclusão, busca de um cliente e busca de todos os clientes.

2) Antes na classe `Cliente`, corrija o método `ToInsert` para:

```
public string ToInsert()
{
    string SQL;
    SQL = @"INSERT INTO TB_Cliente
        (Id
        ,Nome
        ,NomePai
        ,NomeMae
        ,NaoTemPai
        ,Cpf
        ,Genero
        ,Cep
        ,Logradouro
        ,Complemento
        ,Bairro
        ,Cidade
        ,Estado
        ,Telefone
        ,Profissao
        ,RendaFamiliar)
    VALUES ";

    SQL += "(" + this.Id + ",";
    SQL += "," + this.Nome + ",";
    SQL += "," + this.NomePai + ",";
    SQL += "," + this.NomeMae + ",";
    SQL += "," + Convert.ToString(this.NaoTemPai) + ",";
    SQL += "," + this.Cpf + ",";
    SQL += "," + Convert.ToString(this.Genero) + ",";
    SQL += "," + this.Cep + ",";
    SQL += "," + this.Logradouro + ",";
    SQL += "," + this.Complemento + ",";
    SQL += "," + this.Bairro + ",";
    SQL += "," + this.Cidade + ",";
    SQL += "," + this.Estado + ",";
    SQL += "," + this.Telefone + ",";
    SQL += "," + this.Profissao + ",";
    SQL += "," + Convert.ToString(this.RendaFamiliar) + ");";

    return SQL;
}
```

3) No método `DataRowUnit`, ajuste o código para o mostrado abaixo:

```

public Unit DataRowToUnit (DataRow dr)
{
    Unit u = new Unit();
    u.Id = dr["Id"].ToString();
    u.Nome = dr["Nome"].ToString();
    u.NomePai = dr["NomePai"].ToString();
    u.NomeMae = dr["NomeMae"].ToString();
    u.NaoTemPai = Convert.ToInt32(dr["NaoTemPai"]);
    u.Cpf = dr["Cpf"].ToString();
    u.Cep = dr["Cep"].ToString();
    u.Logradouro = dr["Logradouro"].ToString();
    u.Complemento = dr["Complemento"].ToString();
    u.Bairro = dr["Bairro"].ToString();
    u.Cidade = dr["Cidade"].ToString();
    u.Estado = dr["Estado"].ToString();
    u.Telefone = dr["Telefone"].ToString();
    u.Profissao = dr["Profissao"].ToString();
    u.RendaFamiliar = Convert.ToDouble(dr["RendaFamiliar"]);

    return u;
}

```

4) Ainda na classe `Cliente`, inclua os novos métodos:

```

public void IncluirFicharioSQLREL()
{
    try
    {
        string SQL;
        SQL = this.ToInsert();
        var db = new SQLServerClass();
        db.SQLCommand(SQL);
        db.Close();
    }
    catch (Exception ex)
    {
        throw new Exception("Inclusão não permitida. Identificador: " + this.Id + ", erro: " + ex.Message);
    }
}

public Unit BuscarFicharioSQLREL(string Id)
{
    try
    {
        string SQL = "SELECT * FROM [TB_Cliente] WHERE Id = '" + Id + "'";
        var db = new SQLServerClass();
        var Dt = db.SQLQuery(SQL);
        if (Dt.Rows.Count == 0)
        {
            db.Close();
            throw new Exception("Indentificador não existente: " + Id);
        }
    }
    else
    {
        return Dt.Rows[0];
    }
}

```

```
{
    Unit u = this.DataRowToUnit(Dt.Rows[0]);
    return u;
}
}
catch (Exception ex)
{
    throw new Exception("Erro ao buscar o conteúdo do identificador: " + ex.Message);
}
}

public void AlterarFicharioSQLREL()
{
    try
    {
        string SQL = "SELECT * FROM [TB_Cliente] WHERE Id = '" + Id + "'";
        var db = new SQLServerClass();
        var Dt = db.SQLQuery(SQL);
        if (Dt.Rows.Count == 0)
        {
            db.Close();
            throw new Exception("Indentificador não existente: " + Id);
        }
        else
        {
            SQL = this.ToUpdate(this.Id);
            db.SQLCommand(SQL);
            db.Close();
        }
    }
    catch (Exception ex)
    {
        throw new Exception("Erro ao alterar o conteúdo do identificador: " + ex.Message);
    }
}

public void ApagarFicharioSQLREL()
{
    try
    {
        string SQL = "SELECT * FROM [TB_Cliente] WHERE Id = '" + this.Id + "'";
        var db = new SQLServerClass();
        var Dt = db.SQLQuery(SQL);
        if (Dt.Rows.Count == 0)
        {
            db.Close();
            throw new Exception("Indentificador não existente: " + this.Id);
        }
        else
        {
            SQL = "DELETE FROM TB_cliente WHERE Id = '" + this.Id + "'";
            db.SQLCommand(SQL);
            db.Close();
        }
    }
    catch (Exception ex)
    {
        throw new Exception("Erro ao excluir o conteúdo do identificador: " + ex.Message);
    }
}
```

```
    }  
}  
  
public List<List<string>> BuscarFicharioDBTodosSQLREL()  
{  
    List<List<string>> ListaBusca = new List<List<string>>();  
    try  
    {  
        var SQL = "SELECT * FROM TB_Cliente";  
        var db = new SQLServerClass();  
        var Dt = db.SQLQuery(SQL);  
        for (int i = 0; i <= Dt.Rows.Count - 1; i++)  
        {  
            ListaBusca.Add(new List<string> { Dt.Rows[i]["Id"].ToString(), Dt.Rows[i]["Nome"].Tr  
        }  
        return ListaBusca;  
    }  
    catch (Exception ex)  
    {  
        throw new Exception("Conexão com a base ocasionou um erro: " + ex.Message);  
    }  
}
```

5) Vá no formulário do cadastro de cliente (**Frm\_Cadastro\_Cliente\_UC**) e mude as chamadas para os novos métodos adicionados na classe **Cliente** .

6) Salve e execute seu projeto.