

```
using System;
//Using para Banco de Dados
using System.Data;
using System.Data.SqlClient;
using AcessoBancoDados.Properties;

namespace AcessoBancoDados
{
    public class AcessoDadosSQLServer
    {
        //Criar a conexão com o BD
        private SqlConnection criarConexao()
        {
            return new SqlConnection(Settings.Default.conectaSQLServer);
        }

        //Parâmetros para o Banco de Dados
        private SqlParameterCollection sqlParameterCollection = new SqlCommand
            ().Parameters;

        public void LimparParametros()
        {
            sqlParameterCollection.Clear();
        }

        public void AdicionarParametros(string nomeParametro, object
            valorParametro)
        {
            sqlParameterCollection.Add(new SqlParameter(nomeParametro,
                valorParametro));
        }

        //Preenche os dados na SqlCommand para Manipulação e Consultas
        private SqlCommand PreencheSqlCommand(CommandType commandType, string
            minhaUSP)
        {
            try
            {
                //Criar e Abri a conexão com o Banco de Dados
                SqlConnection sqlConnection = criarConexao();
                sqlConnection.Open();
                //Criar e Enviar os comandos ao Banco de Dados
                SqlCommand sqlCommand = sqlConnection.CreateCommand();
                sqlCommand.CommandType = commandType;
                sqlCommand.CommandText = minhaUSP;
                //Define o tempo de execução desses comandos (Timeout em segundos)
                sqlCommand.CommandTimeout = 3600;
                //Adicionar os parâmetros na coleção
                foreach (SqlParameter sqlParameter in sqlParameterCollection)
                {
                    sqlCommand.Parameters.Add(new SqlParameter
                        (sqlParameter.ParameterName, sqlParameter.Value));
                }
                //Executar o comando e pegar o retorno da Stored Procedure
                return sqlCommand;
            }
        }
    }
}
```

```
        catch (Exception ex)
        {
            //Dispara um novo evento de exceção para ser utilizada na tela
            //O formato de saída é diferente de String
            throw new Exception(ex.Message);
        }
    }

    //Inserir, Alterar, Excluir
    public object ManipulaDados(CommandType commandType, string minhaUSP)
    {
        try
        {
            SqlCommand sqlCommand = new SqlCommand();
            //Chama a função que preenche o SqlCommand
            sqlCommand = PreencheSqlCommand(commandType, minhaUSP);
            //Executar o comando e pegar o retorno da Stored Procedure
            return sqlCommand.ExecuteScalar();
        }
        catch (Exception ex)
        {
            //Dispara um novo evento de exceção para ser utilizada na tela
            throw new Exception(ex.Message);
        }
    }

    //Consultar registros
    public DataTable ExecutaConsulta(CommandType commandType, string minhaUSP)
    {
        try
        {
            SqlCommand sqlCommand = PreencheSqlCommand(commandType, minhaUSP);
            //Criar um adaptador para 'traduzir' o retorno do BD
            SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(sqlCommand);
            //Criar uma tabela vazia
            DataTable dataTable = new DataTable();
            //Preencher a tabela de dados
            sqlDataAdapter.Fill(dataTable);
            //Retorno da Stored Procedure
            return dataTable;
        }
        catch (Exception ex)
        {
            //Dispara um novo evento de exceção para ser utilizada na tela
            throw new Exception(ex.Message);
        }
    }
}
```