Universidade Portucalense

Infante D. Henrique



|  |  |
| --- | --- |
| Disciplina: | Tecnologia de Base de Dados |
| Número do trabalho: | 2 |
| Número do problema: | 38 |
| Curso: | Informática |
| Número do aluno: | 25201 |
| Nome do aluno: | José Manuel Guimarães Pereira |
| E-mail: | jose\_pereira\_4@hotmail.com |
| Assinatura: |  |
| Data de entrega: | Sexta-feira, 4 de Janeiro de 2013 |

# 

# Índice

[Índice 1](#_Toc345099409)

[Projeto do Trabalho 3](#_Toc345099410)

[Modelo Logico 3](#_Toc345099411)

[Modelo Físico 3](#_Toc345099412)

[Modelo Conceptual 4](#_Toc345099413)

[Justificação das Opções Tomadas 4](#_Toc345099414)

[Manual de Utilizador Sucinto 5](#_Toc345099415)

[Questões 8](#_Toc345099416)

[A- 8](#_Toc345099417)

[B- 8](#_Toc345099418)

[C- 8](#_Toc345099419)

[D- 9](#_Toc345099420)

[Código Visual Basic .NET 9](#_Toc345099421)

[clsAutocarro.vb 9](#_Toc345099422)

[clsCarta.vb 13](#_Toc345099423)

[clsDiaSemana.vb 14](#_Toc345099424)

[clsHorario.vb 16](#_Toc345099425)

[clsLinha.vb 18](#_Toc345099426)

[clsMarca.vb 20](#_Toc345099427)

[clsPessoa.vb 23](#_Toc345099428)

[clsTipo.vb 29](#_Toc345099429)

[Geral.vb 31](#_Toc345099430)

[AutocarrosMarca.vb 32](#_Toc345099431)

[Habilitacao.vb 32](#_Toc345099432)

[Inicial.vb 32](#_Toc345099433)

[Jornada.vb 33](#_Toc345099434)

[LISTA.vb 36](#_Toc345099435)

[Código de Rotinas SQL 38](#_Toc345099436)

[dbo.Autocarro\_del 38](#_Toc345099437)

[dbo.Autocarro\_edit 38](#_Toc345099438)

[dbo.Autocarro\_ins 38](#_Toc345099439)

[dbo.Carta\_get\_all 39](#_Toc345099440)

[dbo.Carta\_SEL1 39](#_Toc345099441)

[dbo.DiaSemana\_del 39](#_Toc345099442)

[dbo.DiaSemana\_edit 39](#_Toc345099443)

[dbo.diaSemana\_ins 39](#_Toc345099444)

[dbo.Linha\_del 39](#_Toc345099445)

[dbo.Linha\_edit 40](#_Toc345099446)

[dbo.Linha\_ins 40](#_Toc345099447)

[dbo.Marca\_del 40](#_Toc345099448)

[dbo.Marca\_edit 40](#_Toc345099449)

[dbo.Marca\_get\_all 40](#_Toc345099450)

[dbo.Marca\_ins 41](#_Toc345099451)

[dbo.Marca\_SEL1 41](#_Toc345099452)

[dbo.Pessoa\_del 41](#_Toc345099453)

[dbo.Pessoa\_edit 42](#_Toc345099454)

[dbo.Pessoa\_get\_all 42](#_Toc345099455)

[dbo.Pessoa\_ins 42](#_Toc345099456)

[dbo.Pessoa\_SEL\_Jornada 43](#_Toc345099457)

[dbo.Pessoa\_SEL1 43](#_Toc345099458)

[dbo.Pessoa\_SELTIPO 43](#_Toc345099459)

[dbo.Tipo\_get\_all 43](#_Toc345099460)

[dbo.Tipo\_SEL1 43](#_Toc345099461)

# Projeto do Trabalho

|  |  |  |
| --- | --- | --- |
| **Fase do Trabalho** | **Horas de trabalho estimadas** | **Horas de trabalho realizadas** |
| Esquemas | 3 | 2 |
| Construção da Base de Dados e Interface VB.net | 10 | 7 |
| Questões | 10 | 8 |

# Modelo Logico

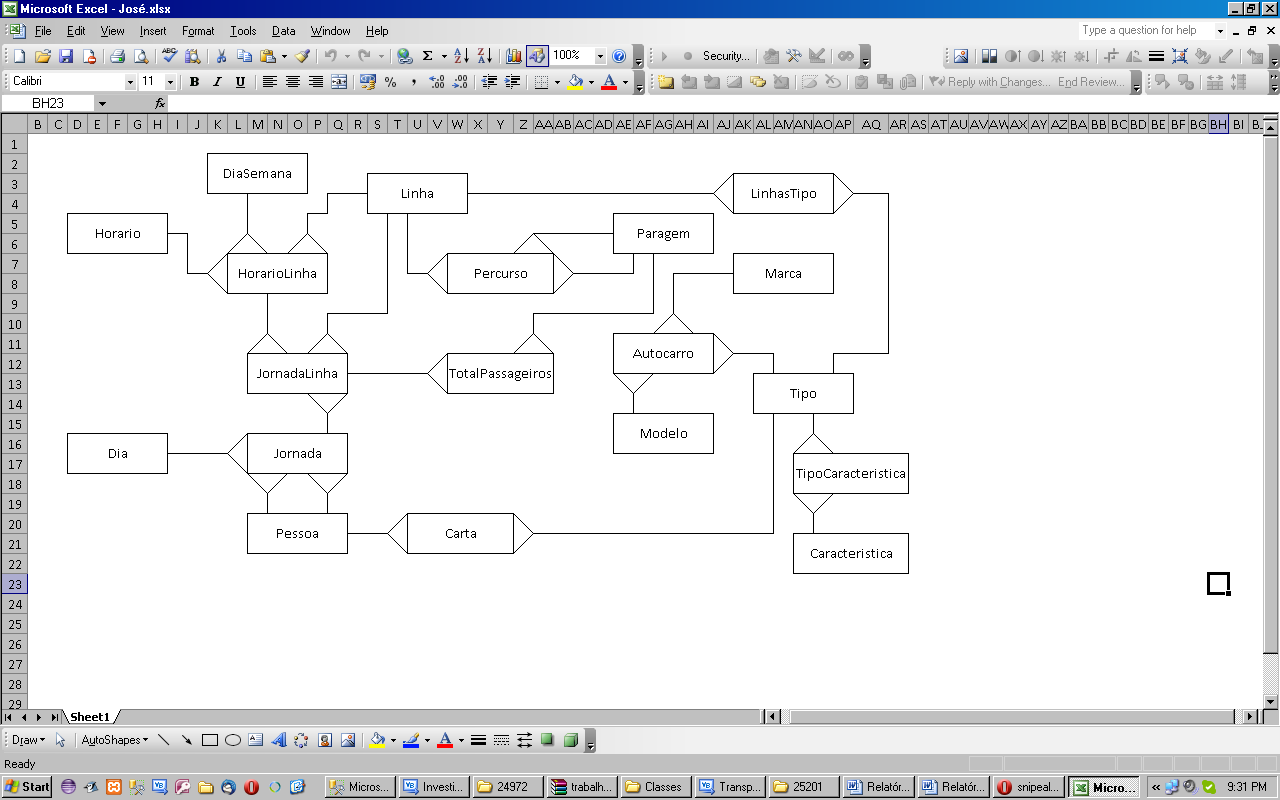


Ilustração – Modelo Lógico

# Modelo Físico

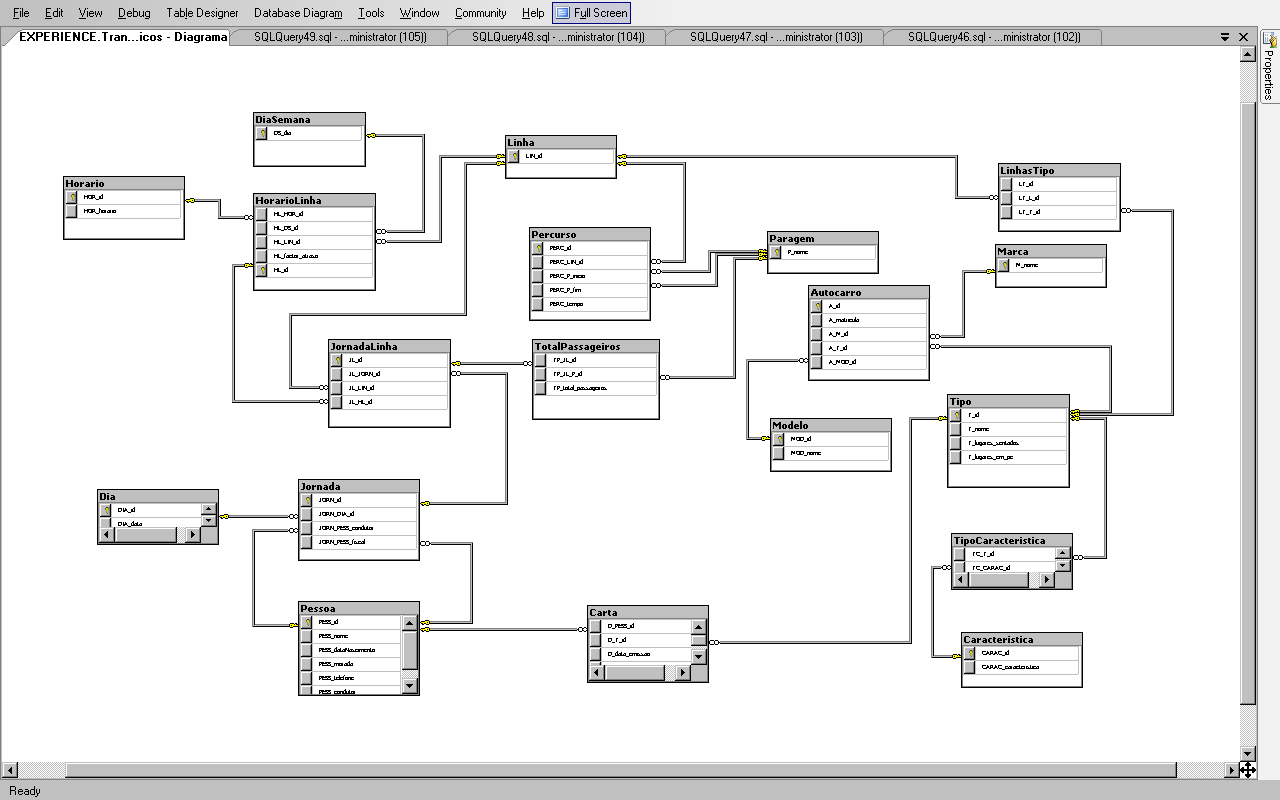


Ilustração – Modelo Físico (Microsoft SQL Server Management Studio R2)

# Modelo Conceptual

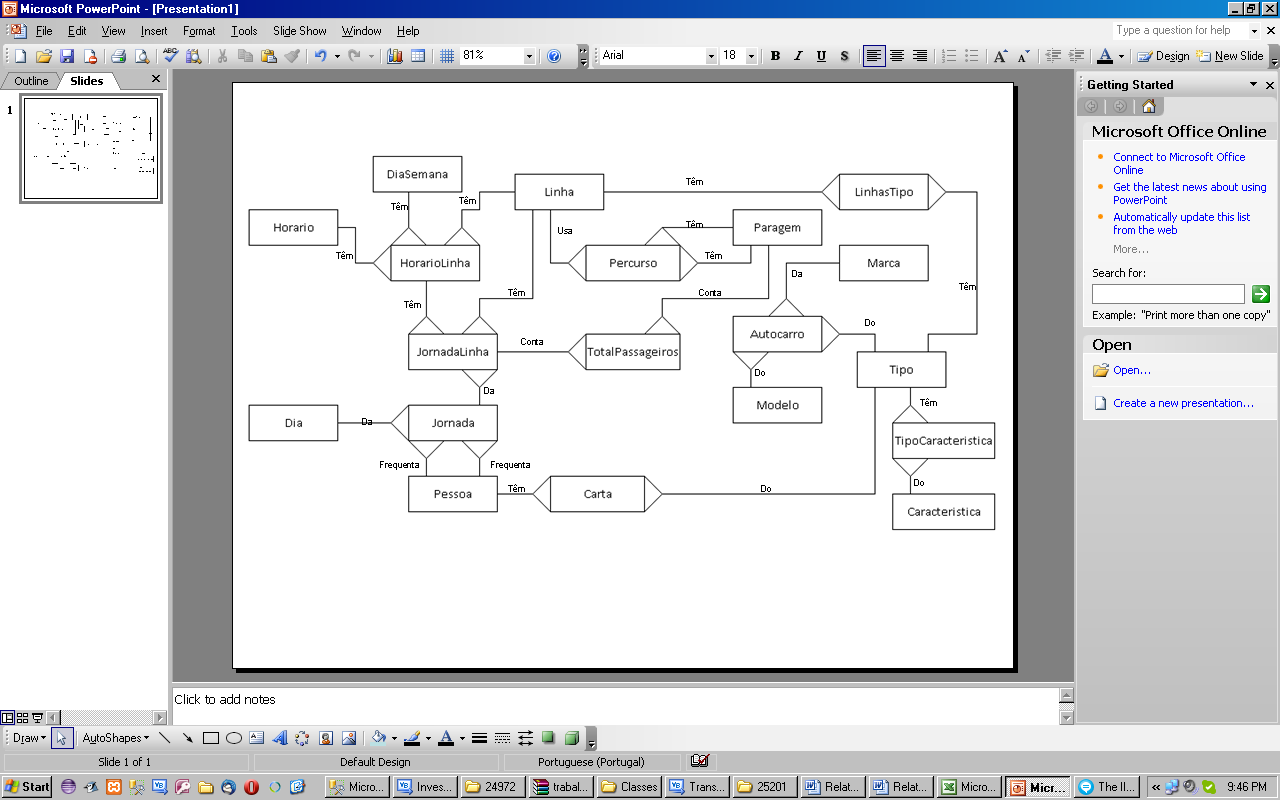


Ilustração – Modelo Conceptual

# Justificação das Opções Tomadas

|  |  |
| --- | --- |
| **Tipo de Dado** | **Uso** |
| Bigint | IDs únicos, valores inteiros. Chaves primárias e externas (FK, Foreign key) |
| nvarchar(30) | Variáveis de tipo “String” |
| Date | Data, necessário para cálculos relativos a datas, dai ser preferível a inteiros. |

Nota: Só valores de tabelas com Auto relação é que podem ter atributos que “Allow Nulls”.

# Manual de Utilizador Sucinto



Ilustração - Menu inicial



Ilustração - Lista (de marcas)

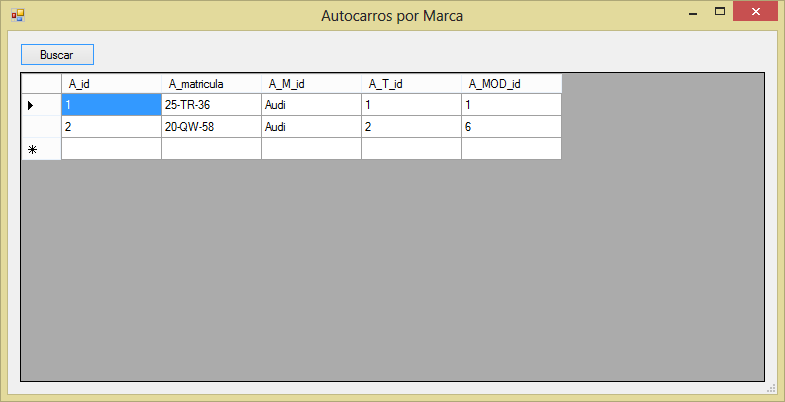


Ilustração - Autocarros por Marca

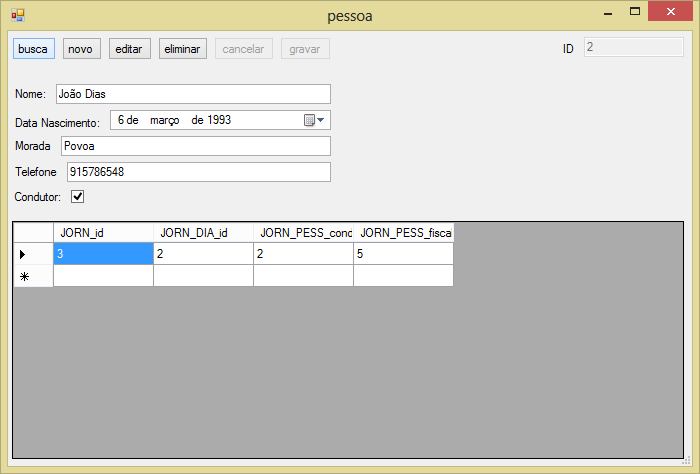


Ilustração - Pessoa (com um registo aberto)

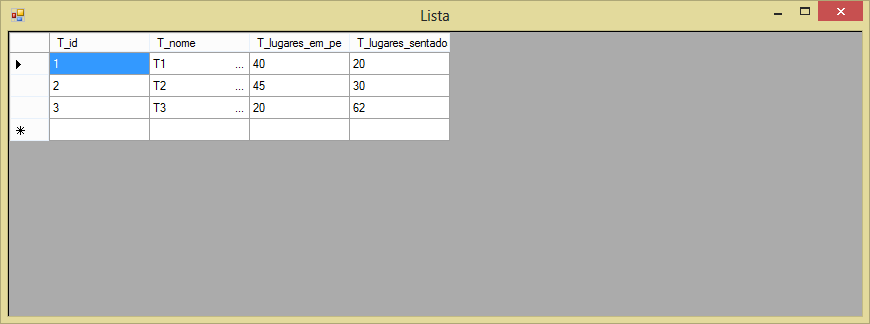


Ilustração - Lista de tipos

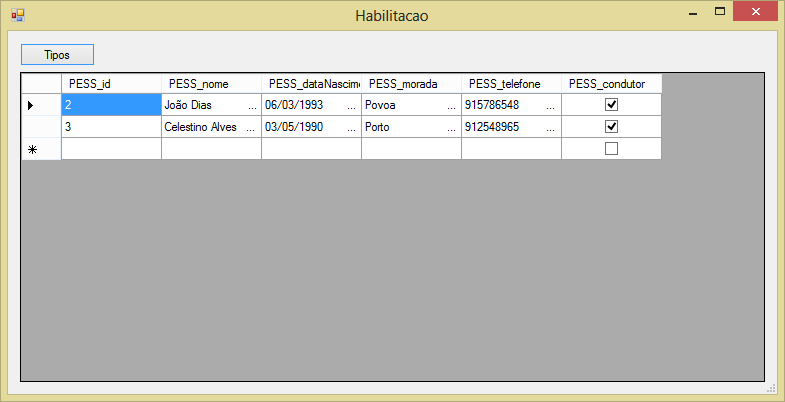


Ilustração - Tipo de autocarro eu cada utilizador pode conduzir

# Questões

## A-

ALTER PROCEDURE [dbo].[Autocarro\_SELMARCA]

@autocarro\_Marca nvarchar(30)

AS

BEGIN

SELECT Autocarro.A\_id, Autocarro.A\_matricula, Autocarro.A\_M\_id, Autocarro.A\_T\_id, Autocarro.A\_MOD\_id from Autocarro

where Autocarro.A\_M\_id = @autocarro\_Marca

END

## B-

ALTER PROCEDURE [dbo].[Tipo\_SEL1]

@id bigint

AS

BEGIN

select Tipo.T\_nome from Tipo where Tipo.T\_id=@id

END

## C-

ALTER PROCEDURE [dbo].[C]

@pessId bigint,

@data\_minima date,

@data\_maxima date

AS

BEGIN

SELECT Linha.LIN\_id

FROM Pessoa INNER JOIN

Jornada ON Pessoa.PESS\_id = Jornada.JORN\_PESS\_fiscal INNER JOIN

Dia ON Jornada.JORN\_DIA\_id = Dia.DIA\_id INNER JOIN

JornadaLinha ON Jornada.JORN\_id = JornadaLinha.JL\_JORN\_id INNER JOIN

Linha ON JornadaLinha.JL\_LIN\_id = Linha.LIN\_id

where PESS\_id=@pessId and Dia.DIA\_data>@data\_minima and Dia.DIA\_data<@data\_maxima

END

## D-

ALTER PROCEDURE [dbo].[D]

@autocarro\_matricula nvarchar(20),

@data\_minima date,

@data\_maxima date

AS

BEGIN

SELECT COUNT(Linha.LIN\_id) AS 'Total linhas'

FROM Autocarro INNER JOIN

Tipo ON Autocarro.A\_T\_id = Tipo.T\_id INNER JOIN

JornadaLinha INNER JOIN

Jornada ON JornadaLinha.JL\_JORN\_id = Jornada.JORN\_id INNER JOIN

Linha ON JornadaLinha.JL\_LIN\_id = Linha.LIN\_id INNER JOIN

Pessoa ON Jornada.JORN\_PESS\_condutor = Pessoa.PESS\_id INNER JOIN

Carta ON Pessoa.PESS\_id = Carta.D\_PESS\_id ON Tipo.T\_id = Carta.D\_T\_id INNER JOIN

Dia ON Jornada.JORN\_DIA\_id = Dia.DIA\_id

WHERE Autocarro.A\_matricula = @autocarro\_matricula and Dia.DIA\_data>@data\_minima and Dia.DIA\_data<@data\_maxima

END

# Código Visual Basic .NET

### clsAutocarro.vb

Public Class clsAutocarro

Private myCnn As SqlClient.SqlConnection

Private myAutocarroId As String

Public Event Action(ByVal estado As Integer)

Public Event Errors(ByVal mensagem As Object)

Public Sub New(ByVal dbConStr As String)

myCnn = New SqlClient.SqlConnection(dbConStr)

End Sub

Public ReadOnly Property autocarroId As Long

Get

autocarroId = myAutocarroId

End Get

End Property

'executa a stored procedure Autocarro\_ins, para inserir um autocarro novo

Public Sub addAutocarro(ByVal matricula As String, ByVal marca As String, ByVal tipo As Long, ByVal modelo As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Autocarro\_ins"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@autocarroMatricula"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = matricula

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@autocarroMarca"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = marca

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@autocarroTipo"

.DbType = DbType.Int64

.Direction = ParameterDirection.Input

.Value = tipo

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@autocarroModelo"

.DbType = DbType.Int64

.Direction = ParameterDirection.Input

.Value = modelo

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

myCnn.Close()

Catch ex As SqlClient.SqlException

If Left(ex.Message, 35) = "Violação da chave primária" Then

RaiseEvent Action(5)

Else

RaiseEvent Errors(ex.Message)

End If

Catch ex2 As Exception

End Try

If myCnn.State = ConnectionState.Open Then

myCnn.Close()

End If

End Sub

'executa a stored procedure Autocaro\_del, apaga um autocarro pela matricula

Public Sub delAutocarro(ByVal matricula As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Autocarro\_del"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@autocarroMatricula"

.DbType = DbType.String

.Value = matricula

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(6)

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

RaiseEvent Errors(ex.Message)

End Try

End Sub

'executa a stored procedure Autocarro\_ins, para editar o autocarro

Public Sub editMarca(ByVal matricula As String, ByVal marca As String, ByVal tipo As Long, ByVal modelo As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Autocarro\_ins"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@autocarroMatricula"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = matricula

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@autocarroMarca"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = marca

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@autocarroTipo"

.DbType = DbType.Int64

.Direction = ParameterDirection.Input

.Value = tipo

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@autocarroModelo"

.DbType = DbType.Int64

.Direction = ParameterDirection.Input

.Value = modelo

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(7)

Catch ex As Exception

RaiseEvent Errors(ex.Message)

End Try

End Sub

'executa a stored procedure Autocarro\_SELMARCA

Public Function autocarroSelMarca(ByVal marca As String) As DataTable

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandText = "Autocarro\_SELMARCA"

.CommandType = CommandType.StoredProcedure

End With

Dim prm As SqlClient.SqlParameter

prm = New SqlClient.SqlParameter

With prm

.Direction = ParameterDirection.Input

.DbType = DbType.String

.ParameterName = "@autocarro\_Marca"

.Value = marca

End With

cmd.Parameters.Add(prm)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

Dim tbl As New DataTable

tbl.Load(cmd.ExecuteReader)

autocarroSelMarca = tbl

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

autocarroSelMarca = Nothing

RaiseEvent Errors(ex.Message)

End Try

End Function

End Class

### clsCarta.vb

Public Class clsCarta

Private myCnn As SqlClient.SqlConnection

Private myCartaTipoId As Long

Private myCartaDataEmissao As String

Private myCartaDataValidade As String

Public Event Action(ByVal estado As Integer)

Public Event Errors(ByVal mensagem As Object)

Public Property cartaTipoId As Long

Get

cartaTipoId = myCartaTipoId

End Get

Set(value As Long)

myCartaTipoId = value

End Set

End Property

Public Property cartaDataEmissao As String

Get

cartaDataEmissao = myCartaDataEmissao

End Get

Set(value As String)

myCartaDataEmissao = value

End Set

End Property

Public Property cartaDataValidade As String

Get

cartaDataValidade = myCartaDataValidade

End Get

Set(value As String)

myCartaDataValidade = value

End Set

End Property

Public Sub New(ByVal dbConStr As String)

myCnn = New SqlClient.SqlConnection(dbConStr)

End Sub

'escreve o quadro das cartas, executa a stored procedure Carta\_SEL1

Public Function pessCarta(ByVal pessId As Long) As DataTable

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandText = "Carta\_SEL1"

.CommandType = CommandType.StoredProcedure

End With

Dim prm As SqlClient.SqlParameter

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@pessId"

.DbType = DbType.Int64

.Direction = ParameterDirection.Input

.Value = pessId

End With

cmd.Parameters.Add(prm)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

Dim tbl As New DataTable

tbl.Load(cmd.ExecuteReader)

pessCarta = tbl

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

pessCarta = Nothing

RaiseEvent Errors(ex.Message)

End Try

End Function

End Class

### clsDiaSemana.vb

Public Class clsDiaSemana

Private myCnn As SqlClient.SqlConnection

Private myDiaSemanaId As String

Public Event Action(ByVal estado As Integer)

Public Event Errors(ByVal mensagem As Object)

Public ReadOnly Property diaSemanaId As String

Get

diaSemanaId = myDiaSemanaId

End Get

End Property

'executa a stored procedure DiaSemana\_ins, adiciona um dia da semana

Public Sub addDiaSemana(ByVal dia As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "DiaSemana\_ins"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@diaSemanaId"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = dia

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

myCnn.Close()

Catch ex As SqlClient.SqlException

If Left(ex.Message, 35) = "Violação da chave primária" Then

RaiseEvent Action(5)

Else

RaiseEvent Errors(ex.Message)

End If

Catch ex2 As Exception

End Try

If myCnn.State = ConnectionState.Open Then

myCnn.Close()

End If

End Sub

'executa a stored procedure DiaSemana\_del, apaga um dia da semana

Public Sub delDiaSemana(ByVal dia As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "DiaSemana\_del"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@diaSemanaId"

.DbType = DbType.String

.Value = dia

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(6)

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

RaiseEvent Errors(ex.Message)

End Try

End Sub

'executa a stored procedure editDiaSemana, edita o dia da semana

Public Sub editDiaSemana(ByVal dia)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "DiaSemana\_edit"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@diaSemanaId"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = dia

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(7)

Catch ex As Exception

RaiseEvent Errors(ex.Message)

End Try

End Sub

End Class

### clsHorario.vb

Public Class clsHorario

Private myCnn As SqlClient.SqlConnection

Private myHorario As String

Private myHorarioId As Long

Public Event Action(ByVal estado As Integer)

Public Event Errors(ByVal mensagem As Object)

Public ReadOnly Property horarioId As String

Get

horarioId = myHorarioId

End Get

End Property

'executa a stored procedure Horarios\_ins, insere um horario

Public Sub addHorario(ByVal horario As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Horario\_ins"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@horarioId"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = horario

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

myCnn.Close()

Catch ex As SqlClient.SqlException

If Left(ex.Message, 35) = "Violação da chave primária" Then

RaiseEvent Action(5)

Else

RaiseEvent Errors(ex.Message)

End If

Catch ex2 As Exception

End Try

If myCnn.State = ConnectionState.Open Then

myCnn.Close()

End If

End Sub

'executa a stored procedure Horario\_del, apaga um horario

Public Sub delHorario(ByVal horario As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Horario\_del"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@horarioId"

.DbType = DbType.String

.Value = horario

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(6)

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

RaiseEvent Errors(ex.Message)

End Try

End Sub

'executa a stored procedure Horario\_edit, edita o horario

Public Sub editHorario(ByVal horario)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Horario\_edit"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@horarioId"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = horario

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(7)

Catch ex As Exception

RaiseEvent Errors(ex.Message)

End Try

End Sub

End Class

### clsLinha.vb

Public Class clsLinha

Private myCnn As SqlClient.SqlConnection

Private mylinId As String

Public Event Action(ByVal estado As Integer)

Public Event Errors(ByVal mensagem As Object)

Public ReadOnly Property linId As String

Get

linId = mylinId

End Get

End Property

'executa a stored procedure Linha\_ins, insere uma linha

Public Sub addLinha(ByVal nome As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Linha\_ins"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@linhaId"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = nome

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

myCnn.Close()

Catch ex As SqlClient.SqlException

If Left(ex.Message, 35) = "Violação da chave primária" Then

RaiseEvent Action(5)

Else

RaiseEvent Errors(ex.Message)

End If

Catch ex2 As Exception

End Try

If myCnn.State = ConnectionState.Open Then

myCnn.Close()

End If

End Sub

'executa a stored procedure Linha\_del, apaga uma linha

Public Sub delLinha(ByVal nome As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Linha\_del"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@linhaId"

.DbType = DbType.String

.Value = nome

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(6)

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

RaiseEvent Errors(ex.Message)

End Try

End Sub

'executa a stored procedure Linha\_edit, edita a linha

Public Sub editLinha(ByVal nome)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Linha\_edit"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@linhaId"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = nome

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(7)

Catch ex As Exception

RaiseEvent Errors(ex.Message)

End Try

End Sub

End Class

### clsMarca.vb

Public Class clsMarca

Private myCnn As SqlClient.SqlConnection

Private myMarca As String

Public Event Action(ByVal estado As Integer)

Public Event Errors(ByVal mensagem As Object)

Public Sub New(ByVal dbConStr As String)

myCnn = New SqlClient.SqlConnection(dbConStr)

End Sub

Public ReadOnly Property marca As String

Get

marca = myMarca

End Get

End Property

'executa a stored procedure Marca\_ins, para inserir uma marca

Public Sub addMarca(ByVal nome As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Marca\_ins"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@marcaId"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = nome

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

myCnn.Close()

Catch ex As SqlClient.SqlException

If Left(ex.Message, 35) = "Violação da chave primária" Then

RaiseEvent Action(5)

Else

RaiseEvent Errors(ex.Message)

End If

Catch ex2 As Exception

End Try

If myCnn.State = ConnectionState.Open Then

myCnn.Close()

End If

End Sub

'executa a stored procedure Marca\_del, para apagar uma marca

Public Sub delMarca(ByVal nome As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Marca\_del"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@marcaId"

.DbType = DbType.String

.Value = nome

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(6)

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

RaiseEvent Errors(ex.Message)

End Try

End Sub

'executa a stored procedure Marca\_edit, para editar uma marca

Public Sub editMarca(ByVal nome)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Marca\_edit"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@marcaId"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = nome

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(7)

Catch ex As Exception

RaiseEvent Errors(ex.Message)

End Try

End Sub

'executa a stored procedure Marca\_get\_all, para listar todas as marcas

Public Function listaMarca() As DataTable

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandText = "Marca\_get\_all"

.CommandType = CommandType.StoredProcedure

End With

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

Dim tbl As New DataTable

tbl.Load(cmd.ExecuteReader)

listaMarca = tbl

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

listaMarca = Nothing

RaiseEvent Errors(ex.Message)

End Try

End Function

'executa a stored procedure Marca\_SEL1

Public Sub getMarca(ByVal nome As String)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Marca\_SEL1"

End With

Dim prm As SqlClient.SqlParameter

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@nome"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = nome

End With

cmd.Parameters.Add(prm)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

Dim tbl As New DataTable

tbl.Load(cmd.ExecuteReader)

If tbl.Rows.Count > 0 Then

With tbl.Rows(0)

Me.myMarca = .Item("M\_nome")

End With

RaiseEvent Action(-1)

Else

Me.myMarca = vbNull

RaiseEvent Action(2)

End If

Catch ex As Exception

RaiseEvent Errors(ex.Message)

End Try

End Sub

End Class

### clsPessoa.vb

Public Class clsPessoa

Private myCnn As SqlClient.SqlConnection

Private myPessId As Long

Private myPessNome As String

Private myPessDataNascimento As String

Private myPessMorada As String

Private myPessTelefone As String

Private myPessCondutor As Boolean

Public Event Action(ByVal estado As Integer)

Public Event Errors(ByVal mensagem As Object)

Public Sub New(ByVal dbConStr As String)

myCnn = New SqlClient.SqlConnection(dbConStr)

End Sub

Public ReadOnly Property pessId As Long

Get

pessId = myPessId

End Get

End Property

Public Property pessNome As String

Get

pessNome = myPessNome

End Get

Set(value As String)

myPessNome = value

End Set

End Property

Public Property pessDataNascimento As String

Get

pessDataNascimento = myPessDataNascimento

End Get

Set(value As String)

myPessDataNascimento = value

End Set

End Property

Public Property pessMorada As String

Get

pessMorada = myPessMorada

End Get

Set(value As String)

myPessMorada = value

End Set

End Property

Public Property pessTelefone As String

Get

pessTelefone = myPessTelefone

End Get

Set(value As String)

myPessTelefone = value

End Set

End Property

Public Property pessCondutor As Boolean

Get

pessCondutor = myPessCondutor

End Get

Set(value As Boolean)

myPessCondutor = value

End Set

End Property

'executa a stored procedure Pessoa\_get\_all, para buscar a lista de todas as pessoas

Public Function pessLista() As DataTable

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandText = "Pessoa\_get\_all"

.CommandType = CommandType.StoredProcedure

End With

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

Dim tbl As New DataTable

tbl.Load(cmd.ExecuteReader)

pessLista = tbl

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

pessLista = Nothing

RaiseEvent Errors(ex.Message)

End Try

End Function

'executa a stored procedure Pessoa\_ins, para inserir uma pessoa nova

Public Sub addPessoa(ByVal nome As String, ByVal dataNascimento As String, ByVal morada As String, ByVal telefone As String, ByVal condutor As Boolean)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Pessoa\_ins"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessNome"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = nome

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessDataNascimento"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = dataNascimento

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessMorada"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = morada

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessTelefone"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = telefone

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessCondutor"

.DbType = DbType.Byte

.Direction = ParameterDirection.Input

.Value = condutor

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(8)

myCnn.Close()

Catch ex As SqlClient.SqlException

If Left(ex.Message, 35) = "Violação da chave primária" Then

RaiseEvent Action(5)

Else

RaiseEvent Errors(ex.Message)

End If

Catch ex2 As Exception

End Try

If myCnn.State = ConnectionState.Open Then

myCnn.Close()

End If

End Sub

'executa a stored procedure Pessoa\_del, para apagar a pessoa pelo Id da pessoa

Public Sub delPessoa(ByVal id As Long)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Pessoa\_del"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessId"

.DbType = DbType.Int64

.Value = id

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(6)

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

RaiseEvent Errors(ex.Message)

End Try

End Sub

'executa a stored procedure Pessoa\_edit, para editar a pessoa

Public Sub editPessoa(ByVal id As Long, ByVal nome As String, ByVal dataNascimento As String, ByVal morada As String, ByVal telefone As String, ByVal condutor As Boolean)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Pessoa\_edit"

End With

Dim parametro As SqlClient.SqlParameter

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessId"

.DbType = DbType.Int64

.Direction = ParameterDirection.Input

.Value = id

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessNome"

.Direction = ParameterDirection.Input

.DbType = DbType.String

.Value = nome

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessDataNascimento"

.Direction = ParameterDirection.Input

.DbType = DbType.String

.Value = dataNascimento

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessMorada"

.Direction = ParameterDirection.Input

.DbType = DbType.String

.Value = morada

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessTelefone"

.Direction = ParameterDirection.Input

.DbType = DbType.String

.Value = telefone

End With

cmd.Parameters.Add(parametro)

parametro = New SqlClient.SqlParameter

With parametro

.ParameterName = "@pessCondutor"

.Direction = ParameterDirection.Input

.DbType = DbType.Byte

.Value = condutor

End With

cmd.Parameters.Add(parametro)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

cmd.ExecuteNonQuery()

RaiseEvent Action(7)

Catch ex As Exception

RaiseEvent Errors(ex.Message)

End Try

End Sub

'executa a stored procedure Pessoa\_SELTIPO

Public Function pessoaSelTipo(ByVal tipoId As Long) As DataTable

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandText = "Pessoa\_SELTIPO"

.CommandType = CommandType.StoredProcedure

End With

Dim prm As SqlClient.SqlParameter

prm = New SqlClient.SqlParameter

With prm

.Direction = ParameterDirection.Input

.DbType = DbType.Int64

.ParameterName = "@PessoaTipo"

.Value = tipoId

End With

cmd.Parameters.Add(prm)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

Dim tbl As New DataTable

tbl.Load(cmd.ExecuteReader)

pessoaSelTipo = tbl

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

pessoaSelTipo = Nothing

RaiseEvent Errors(ex.Message)

End Try

End Function

'executa a stored procedure Pessoa\_SEL1

Public Sub getPessoa(ByVal id As Long)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Pessoa\_SEL1"

End With

Dim prm As SqlClient.SqlParameter

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@id"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = id

End With

cmd.Parameters.Add(prm)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

Dim tbl As New DataTable

tbl.Load(cmd.ExecuteReader)

If tbl.Rows.Count > 0 Then

With tbl.Rows(0)

Me.myPessId = .Item("PESS\_id")

Me.myPessNome = .Item("PESS\_nome")

Me.pessDataNascimento = .Item("PESS\_dataNascimento")

Me.myPessTelefone = .Item("PESS\_telefone")

Me.pessCondutor = .Item("PESS\_condutor")

Me.myPessMorada = .Item("PESS\_morada")

End With

RaiseEvent Action(-1)

Else

Me.myPessId = vbNull

Me.myPessNome = vbNull

Me.pessDataNascimento = vbNull

Me.myPessTelefone = vbNull

RaiseEvent Action(2)

End If

Catch ex As Exception

RaiseEvent Errors(ex.Message)

End Try

End Sub

'executa a stored procedure Pessoa\_SEL\_Jornada

Public Function pessoaGetJornada(ByVal id As Long) As DataTable

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandText = "Pessoa\_SEL\_Jornada"

.CommandType = CommandType.StoredProcedure

End With

Dim prm As SqlClient.SqlParameter

prm = New SqlClient.SqlParameter

With prm

.Direction = ParameterDirection.Input

.DbType = DbType.Int64

.ParameterName = "@id"

.Value = id

End With

cmd.Parameters.Add(prm)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

Dim tbl As New DataTable

tbl.Load(cmd.ExecuteReader)

pessoaGetJornada = tbl

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

pessoaGetJornada = Nothing

RaiseEvent Errors(ex.Message)

End Try

End Function

End Class

### clsTipo.vb

Public Class clsTipo

Private myCnn As SqlClient.SqlConnection

Private myTipoId As Long

Private myTipoNome As String

Private myTipoLugaresSentados As Integer

Private myTipoLugaresEmPe As Integer

Public Event Action(ByVal estado As Integer)

Public Event Errors(ByVal mensagem As Object)

Public Sub New(ByVal dbConStr As String)

myCnn = New SqlClient.SqlConnection(dbConStr)

End Sub

Public ReadOnly Property id As Long

Get

id = myTipoId

End Get

End Property

Public Property tipoNome As String

Get

tipoNome = myTipoNome

End Get

Set(value As String)

myTipoNome = value

End Set

End Property

Public Property tipoLugaresSentados As Integer

Get

tipoLugaresSentados = myTipoLugaresSentados

End Get

Set(value As Integer)

myTipoLugaresSentados = value

End Set

End Property

Public Property tipoLugaresEmPe As Integer

Get

tipoLugaresEmPe = myTipoLugaresEmPe

End Get

Set(value As Integer)

myTipoLugaresEmPe = value

End Set

End Property

'executa a stored procedure Tipo\_get\_all, busca todos os tipos

Public Function listaTipo() As DataTable

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandText = "Tipo\_get\_all"

.CommandType = CommandType.StoredProcedure

End With

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

Dim tbl As New DataTable

tbl.Load(cmd.ExecuteReader)

listaTipo = tbl

myCnn.Close()

Catch ex As Exception

If myCnn.State = ConnectionState.Open Then myCnn.Close()

listaTipo = Nothing

RaiseEvent Errors(ex.Message)

End Try

End Function

'executa a stored procedure Tipo\_SEL1

Public Sub getTipo(ByVal id As Long)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Tipo\_SEL1"

End With

Dim prm As SqlClient.SqlParameter

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@id"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = id

End With

cmd.Parameters.Add(prm)

Try

If myCnn.State = ConnectionState.Closed Then myCnn.Open()

Dim tbl As New DataTable

tbl.Load(cmd.ExecuteReader)

If tbl.Rows.Count > 0 Then

With tbl.Rows(0)

Me.myTipoId = .Item("T\_id")

Me.myTipoNome = .Item("T\_nome")

Me.myTipoLugaresSentados = .Item("T\_lugares\_sentados")

Me.myTipoLugaresEmPe = .Item("T\_lugares\_em\_pe")

End With

RaiseEvent Action(-1)

Else

Me.myTipoId = vbNull

Me.myTipoNome = vbNull

Me.myTipoLugaresSentados = vbNull

Me.myTipoLugaresEmPe = vbNull

RaiseEvent Action(2)

End If

Catch ex As Exception

RaiseEvent Errors(ex.Message)

End Try

End Sub

End Class

### Geral.vb

Module Geral

'database connection string para a base de dados

Public dbCNNstr As String = "Data Source=(local);Initial Catalog=""Transportes Publicos"";Integrated Security=True"

End Module

### AutocarrosMarca.vb

Public Class AutocarrosMarca

Private WithEvents myMarca As clsMarca

Private WithEvents myAutocarro As clsAutocarro

Public Sub AutocarrosMarca\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

myMarca = New clsMarca(dbCNNstr)

myAutocarro = New clsAutocarro(dbCNNstr)

End Sub

'busca a lista de marcas de autocarro

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Dim tbl As New DataTable

LISTA.DataGridView1.DataSource = myMarca.listaMarca

LISTA.ShowDialog()

If Not LISTA.Escolha Is Nothing Then

myMarca.getMarca(LISTA.Escolha.Cells("M\_nome").Value)

Me.dgMarcaCarro.DataSource = myAutocarro.autocarroSelMarca(LISTA.Escolha.Cells("M\_nome").Value)

End If

End Sub

End Class

### Habilitacao.vb

Public Class Habilitacao

Private WithEvents myTipo As clsTipo

Private WithEvents myPessoa As clsPessoa

Public Sub Habilitacao\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

myTipo = New clsTipo(dbCNNstr)

myPessoa = New clsPessoa(dbCNNstr)

End Sub

'lista os tipos

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles Button1.Click

Dim tbl As New DataTable

LISTA.DataGridView1.DataSource = myTipo.listaTipo

LISTA.ShowDialog()

If Not LISTA.Escolha Is Nothing Then

myTipo.getTipo(LISTA.Escolha.Cells("T\_id").Value)

Me.dgTipoPessoa.DataSource = myPessoa.pessoaSelTipo(LISTA.Escolha.Cells("T\_id").Value)

End If

End Sub

End Class

### Inicial.vb

Public Class Inicial

Private Sub AutocarrosToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles AutocarrosToolStripMenuItem.Click

AutocarrosMarca.ShowDialog()

End Sub

Private Sub HabilitaçãoToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles HabilitaçãoToolStripMenuItem.Click

Habilitacao.ShowDialog()

End Sub

Private Sub PessoasToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles PessoasToolStripMenuItem.Click

Jornada.ShowDialog()

End Sub

Private Sub SairToolStripMenuItem\_Click(sender As Object, e As EventArgs) Handles SairToolStripMenuItem.Click

Me.Close()

End Sub

End Class

### Jornada.vb

Public Class Jornada

Private daPessoa As New SqlClient.SqlDataAdapter

Private daDOCT As New SqlClient.SqlDataAdapter

Private daDOC As New SqlClient.SqlDataAdapter

Private daDOClin As New SqlClient.SqlDataAdapter

Private dsMain As New DataSet

Private WithEvents myPessoa As clsPessoa

Private WithEvents myCarta As clsCarta

Private listaCartas As DataGridView

Private novoREG As Boolean

Private Sub Pessoa\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

myPessoa = New clsPessoa(dbCNNstr)

myCarta = New clsCarta(dbCNNstr)

Call prepForm(0)

End Sub

'Controla o uso dos botões de manutenção

Sub prepForm(ByVal status As Integer)

Select Case status

Case 0

btnNOVO.Enabled = True

btnBUSCA.Enabled = True

btnEDITAR.Enabled = False

btnELIMINAR.Enabled = False

btnGRAVAR.Enabled = False

btnCANCELAR.Enabled = False

Me.pessId.Enabled = False

Case 1

btnNOVO.Enabled = False

btnBUSCA.Enabled = False

btnEDITAR.Enabled = False

btnELIMINAR.Enabled = False

btnGRAVAR.Enabled = True

btnCANCELAR.Enabled = True

Me.pessId.Enabled = False

Case 2

btnNOVO.Enabled = False

btnBUSCA.Enabled = False

btnEDITAR.Enabled = False

btnELIMINAR.Enabled = False

btnGRAVAR.Enabled = True

btnCANCELAR.Enabled = True

Me.pessId.Enabled = False

Case 3

btnNOVO.Enabled = True

btnBUSCA.Enabled = True

btnEDITAR.Enabled = True

btnELIMINAR.Enabled = True

btnGRAVAR.Enabled = False

btnCANCELAR.Enabled = False

Me.pessId.Enabled = False

End Select

End Sub

'limpa os campos do formulario da pessoa

Private Sub newPessoa()

Me.pessNome.Text = ""

Me.pessMorada.Text = ""

Me.pessDataNascimento.Text = ""

Me.cbCondutor.Checked = False

Me.pessTelefone.Text = ""

novoREG = True

End Sub

'botao busca

Private Sub btnBUSCA\_Click(sender As System.Object, e As System.EventArgs) Handles btnBUSCA.Click

LISTA.DataGridView1.DataSource = myPessoa.pessLista

LISTA.ShowDialog()

If Not LISTA.Escolha Is Nothing Then

myPessoa.getPessoa(LISTA.Escolha.Cells("PESS\_id").Value)

myCarta.pessCarta(LISTA.Escolha.Cells("PESS\_id").Value)

Me.pessNome.Text = myPessoa.pessNome

Me.pessDataNascimento.Value = myPessoa.pessDataNascimento

Me.pessMorada.Text = myPessoa.pessMorada

Me.pessTelefone.Text = myPessoa.pessTelefone

Me.cbCondutor.Checked = myPessoa.pessCondutor

If (myPessoa.pessCondutor = True) Then

Me.cbCondutor.CheckState = CheckState.Checked

Me.cbCondutor.Visible = True

Me.dgJORN.Visible = True

Else

Me.cbCondutor.CheckState = CheckState.Unchecked

Me.dgJORN.Visible = False

End If

Me.pessId.Text = myPessoa.pessId

Me.dgJORN.DataSource = myPessoa.pessoaGetJornada(LISTA.Escolha.Cells("PESS\_id").Value)

End If

End Sub

'gere eventos do preopForm e mostra as mensagens de feedback das acçõens

Private Sub clsPessoaAction(ByVal status As Short) Handles myPessoa.Action

Select Case status

Case -1

Call prepForm(3)

Case 2

Call prepForm(1)

novoREG = True

'txtTTDESCR.Focus()

Case 5

MsgBox("Registo com chave duplicada", MsgBoxStyle.Exclamation, "Pessoa")

'txtTTCOD.Focus()

Case 6

MsgBox("Registo Eliminado", MsgBoxStyle.Information, "Pessoa")

Call newPessoa()

Call prepForm(0)

'txtTTCOD.Focus()

Case 7

MsgBox("Registo actualizado.", MsgBoxStyle.Information, "Pessoa")

Call prepForm(3)

'txtTTCOD.Focus()

Case 8

MsgBox("Registo inserido", MsgBoxStyle.Information, "Pessoa")

Call prepForm(3)

'txtTTCOD.Focus()

End Select

End Sub

'botao novo

Private Sub btnNOVO\_Click(sender As System.Object, e As System.EventArgs) Handles btnNOVO.Click

Call newPessoa()

Call prepForm(1)

dgJORN.DataSource = Nothing

End Sub

'botao editar

Private Sub btnEDITAR\_Click(sender As System.Object, e As System.EventArgs) Handles btnEDITAR.Click

novoREG = False

Call prepForm(2)

Me.pessNome.Focus()

End Sub

'botao cancelar

Private Sub btnCANCELAR\_Click(sender As System.Object, e As System.EventArgs) Handles btnCANCELAR.Click

Call newPessoa()

Call prepForm(0)

End Sub

'botao gravar

Private Sub btnGRAVAR\_Click(sender As System.Object, e As System.EventArgs) Handles btnGRAVAR.Click

If novoREG Then

myPessoa.addPessoa(Me.pessNome.Text, Me.pessDataNascimento.Value, Me.pessMorada.Text, Me.pessTelefone.Text, Me.cbCondutor.Checked)

Else

myPessoa.editPessoa(Me.pessId.Text, Me.pessNome.Text, Me.pessDataNascimento.Value, Me.pessMorada.Text, Me.pessTelefone.Text, Me.cbCondutor.Checked)

End If

End Sub

'Serve para defenir quem é condutor

Private Sub cbCondutor\_Click(sender As Object, e As EventArgs) Handles cbCondutor.Click

If (Me.cbCondutor.Checked = True) Then

Me.cbCondutor.CheckState = CheckState.Checked

Me.cbCondutor.Checked = True

Me.cbCondutor.Visible = True

Me.dgJORN.Visible = True

Else

Me.cbCondutor.CheckState = CheckState.Unchecked

Me.cbCondutor.Checked = False

Me.dgJORN.Visible = False

End If

End Sub

'botao eliminar

Private Sub btnELIMINAR\_Click(sender As Object, e As EventArgs) Handles btnELIMINAR.Click

If MsgBox("Pretende eliminar o registo?", MsgBoxStyle.YesNo, "Pessoa") = MsgBoxResult.Yes Then

myPessoa.delPessoa(Me.pessId.Text)

Call newPessoa()

Call prepForm(0)

Me.dgJORN.DataSource = Nothing

End If

End Sub

End Class

### LISTA.vb

Public Class LISTA

Private linha As DataGridViewRow

Public ReadOnly Property Escolha As DataGridViewRow

Get

Escolha = linha

End Get

End Property

Private Sub LISTA\_Load(sender As System.Object, e As System.EventArgs) Handles MyBase.Load

linha = Nothing

End Sub

'Abre o resgisto da linha registada

Private Sub DataGridView1\_DoubleClick(sender As Object, e As System.EventArgs) Handles DataGridView1.DoubleClick

linha = New DataGridViewRow

linha = Me.DataGridView1.CurrentRow

Me.Close()

End Sub

End Class

# Código de Rotinas SQL

### dbo.Autocarro\_del

ALTER PROCEDURE [dbo].[Autocarro\_del]

@autocarroMatricula nvarchar(20)

AS

BEGIN

delete from Autocarro where Autocarro.A\_matricula=@autocarroMatricula

END

### dbo.Autocarro\_edit

ALTER PROCEDURE [dbo].[Autocarro\_edit]

@autocarroMatricula nvarchar(20),

@autocarroMarca nvarchar(30),

@autocarroTipo bigint,

@autocarroModelo bigint

AS

BEGIN

BEGIN

if not ISNULL(@autocarroMarca,'')=''

update Autocarro set A\_M\_id = @autocarroMarca where A\_matricula = @autocarroMatricula

if not ISNULL (@autocarroTipo,'')=''

update Autocarro set A\_T\_id = @autocarroTipo where A\_matricula = @autocarroMatricula

if not ISNULL (@autocarroModelo,'')=''

update Autocarro set A\_MOD\_id = @autocarroModelo where A\_matricula = @autocarroMatricula

END

END

### dbo.Autocarro\_ins

ALTER PROCEDURE [dbo].[Autocarro\_ins]

@autocarroMatricula nvarchar(20),

@autocarroMarca nvarchar(30),

@autocarroTipo bigint,

@autocarroModelo bigint

AS

BEGIN

insert into Autocarro(A\_matricula, A\_M\_id, A\_T\_id, A\_MOD\_id) values(@autocarroMatricula, @autocarroMarca, @autocarroTipo, @autocarroModelo)

END

### dbo.Carta\_get\_all

ALTER PROCEDURE [dbo].[Carta\_get\_all]

as

BEGIN

SELECT Carta.\* from Carta

END

### dbo.Carta\_SEL1

ALTER PROCEDURE [dbo].[Carta\_SEL1]

@pessId as bigint

as

BEGIN

SELECT Carta.\* from Carta where Carta.D\_PESS\_id=@pessId

END

### dbo.DiaSemana\_del

ALTER PROCEDURE [dbo].[DiaSemana\_del]

@diaSemanaId nvarchar(30)

AS

BEGIN

delete from DiaSemana where DS\_dia=@diaSemanaId

END

### dbo.DiaSemana\_edit

ALTER PROCEDURE [dbo].[DiaSemana\_edit]

@diaSemanaId nvarchar(30)

AS

BEGIN

if not ISNULL(@diaSemanaId,'')=''

update DiaSemana set DS\_dia = @diaSemanaId where DS\_dia = @diaSemanaId

END

### dbo.diaSemana\_ins

ALTER PROCEDURE [dbo].[diaSemana\_ins]

@diaSemanaId nvarchar(30)

AS

BEGIN

insert into DiaSemana(DS\_dia) values(@diaSemanaId)

END

### dbo.Linha\_del

ALTER PROCEDURE [dbo].[Linha\_del]

@linhaId nvarchar(30)

AS

BEGIN

delete from Linha where LIN\_id=@linhaId

END

### dbo.Linha\_edit

ALTER PROCEDURE [dbo].[Linha\_edit]

@linhaId nvarchar(30)

AS

BEGIN

if not ISNULL(@linhaId,'')=''

update Linha set LIN\_id = @linhaId where LIN\_id = @linhaId

END

### dbo.Linha\_ins

ALTER PROCEDURE [dbo].[Linha\_ins]

@linhaId nvarchar(30)

AS

BEGIN

insert into Linha(LIN\_id) values(@linhaId)

END

### dbo.Marca\_del

ALTER PROCEDURE [dbo].[Marca\_del]

@marcaId nvarchar(30)

AS

BEGIN

delete from Marca where M\_nome=@marcaId

END

### dbo.Marca\_edit

ALTER PROCEDURE [dbo].[Marca\_edit]

@marcaId nvarchar(30)

AS

BEGIN

if not ISNULL(@marcaId,'')=''

update Marca set M\_nome = @marcaId where M\_nome = @marcaId

END

### dbo.Marca\_get\_all

ALTER PROCEDURE [dbo].[Marca\_get\_all]

as

BEGIN

SELECT Marca.M\_nome from Marca order by Marca.M\_nome

END

### dbo.Marca\_ins

ALTER PROCEDURE [dbo].[Marca\_ins]

@marcaId nvarchar(30)

AS

BEGIN

insert into Marca(M\_nome) values(@marcaId)

END

### dbo.Marca\_SEL1

ALTER PROCEDURE [dbo].[Marca\_SEL1]

@pessId as bigint

as

BEGIN

SELECT Marca.M\_nome from Marca order by Marca.M\_nome

END

### dbo.Pessoa\_del

ALTER PROCEDURE [dbo].[Pessoa\_del]

@pessId Bigint

AS

BEGIN

delete from Pessoa where PESS\_id=@pessId

END

### dbo.Pessoa\_edit

ALTER PROCEDURE [dbo].[Pessoa\_edit]

@pessId bigint,

@pessNome nvarchar(30),

@pessDataNascimento nvarchar(30),

@pessMorada nvarchar(30),

@pessTelefone nvarchar(30),

@pessCondutor bit

AS

BEGIN

if not ISNULL(@pessNome,'')=''

update Pessoa set PESS\_nome = @pessNome where PESS\_id = @pessId

if not ISNULL (@pessDataNascimento,'')=''

update Pessoa set PESS\_dataNascimento = @pessDataNascimento where PESS\_id = @pessId

if not ISNULL (@pessMorada,'')=''

update Pessoa set PESS\_morada = @pessMorada where PESS\_id = @pessId

if not ISNULL (@pessTelefone,'')=''

update Pessoa set PESS\_telefone = @pessTelefone where PESS\_id = @pessId

if not ISNULL (@pessCondutor,'')=''

update Pessoa set PESS\_condutor = @pessCondutor where PESS\_id = @pessId

END

### dbo.Pessoa\_get\_all

ALTER PROCEDURE [dbo].[Pessoa\_get\_all]

as

BEGIN

SELECT Pessoa.PESS\_id, Pessoa.PESS\_nome, Pessoa.PESS\_morada, Pessoa.PESS\_dataNascimento, Pessoa.PESS\_telefone from Pessoa order by Pessoa.PESS\_id

END

### dbo.Pessoa\_ins

ALTER PROCEDURE [dbo].[Pessoa\_ins]

@pessNome nvarchar(30),

@pessDataNascimento nvarchar(30),

@pessMorada nvarchar(30),

@pessTelefone nvarchar(30),

@pessCondutor bit

AS

BEGIN

insert into Pessoa (PESS\_nome, PESS\_dataNascimento, PESS\_morada, PESS\_telefone, PESS\_condutor) values (@pessNome, @pessDataNascimento, @pessMorada, @pessTelefone, @pessCondutor)

END

### dbo.Pessoa\_SEL\_Jornada

ALTER PROCEDURE [dbo].[Pessoa\_SEL\_Jornada]

@id bigint

AS

BEGIN

SELECT Jornada.\*

FROM Jornada INNER JOIN

Pessoa ON Jornada.JORN\_PESS\_condutor = Pessoa.PESS\_id

where Jornada.JORN\_PESS\_condutor=@id

END

### dbo.Pessoa\_SEL1

ALTER PROCEDURE [dbo].[Pessoa\_SEL1]

@id bigint

AS

BEGIN

select Pessoa.\* from Pessoa where Pessoa.PESS\_id=@id

END

### dbo.Pessoa\_SELTIPO

ALTER PROCEDURE [dbo].[Pessoa\_SELTIPO]

@PessoaTipo bigint

AS

BEGIN

SELECT Pessoa.\*

FROM Tipo INNER JOIN

Carta ON Tipo.T\_id = Carta.D\_T\_id INNER JOIN

Pessoa ON Carta.D\_PESS\_id = Pessoa.PESS\_id

where Tipo.T\_id=@PessoaTipo

END

### dbo.Tipo\_get\_all

ALTER PROCEDURE [dbo].[Tipo\_get\_all]

as

BEGIN

SELECT Tipo.T\_id, Tipo.T\_nome, Tipo.T\_lugares\_em\_pe, Tipo.T\_lugares\_sentados from Tipo order by Tipo.T\_nome

END

### dbo.Tipo\_SEL1

ALTER PROCEDURE [dbo].[Tipo\_SEL1]

@id bigint

AS

BEGIN

select Tipo.T\_nome from Tipo where Tipo.T\_id=@id

END