Universidade Portucalense

Infante D. Henrique



|  |  |
| --- | --- |
| Disciplina: | Tecnologia de Base de Dados |
| Número do trabalho: | 2 |
| Número do problema: | 40 |
| Curso: | Informática |
| Número do aluno: | 24972 |
| Nome do aluno: | João Moreira Dias |
| E-mail: | joao.moreira.dias1@gmail.com |
| Assinatura: |  |
| Data de entrega: | Sexta-feira, 4 de Janeiro de 2013 |

# 

# Índice

[Índice 1](#_Toc345105878)

[Projeto do Trabalho 2](#_Toc345105879)

[Modelo Logico 2](#_Toc345105880)

[Modelo Físico 3](#_Toc345105881)

[Modelo Conceptual 3](#_Toc345105882)

[Justificação das Opções Tomadas 4](#_Toc345105883)

[Manual de Utilizador Sucinto 4](#_Toc345105884)

[Questões 5](#_Toc345105885)

[A- 5](#_Toc345105886)

[B- 6](#_Toc345105887)

[C- 6](#_Toc345105888)

[D- 6](#_Toc345105889)

[E- 6](#_Toc345105890)

[F- 7](#_Toc345105891)

[G- 7](#_Toc345105892)

[H- 7](#_Toc345105893)

[Código Visual Basic .NET 8](#_Toc345105894)

[clsIndiv.vb 8](#_Toc345105895)

[Geral.vb 13](#_Toc345105896)

[INDIV.vb 13](#_Toc345105897)

[LISTA.vb 16](#_Toc345105898)

[Código de Rotinas SQL 17](#_Toc345105899)

[dbo.INDIV\_DEL 17](#_Toc345105900)

[dbo.Indiv\_Edit 17](#_Toc345105901)

[dbo.Individuo\_add 17](#_Toc345105902)

[dbo.Individuo\_get\_all 18](#_Toc345105903)

[dbo.Individuo\_get\_detencoes 18](#_Toc345105904)

[dbo.Individuo\_get\_indiv 18](#_Toc345105905)

# Projeto do Trabalho

|  |  |  |
| --- | --- | --- |
| **Fase do Trabalho** | **Horas de trabalho estimadas** | **Horas de trabalho realizadas** |
| Esquemas | 4 | 3 |
| Construção da Base de Dados e Interface VB.net | 10 | 8 |
| Questões | 10 | 10 |

# Modelo Logico

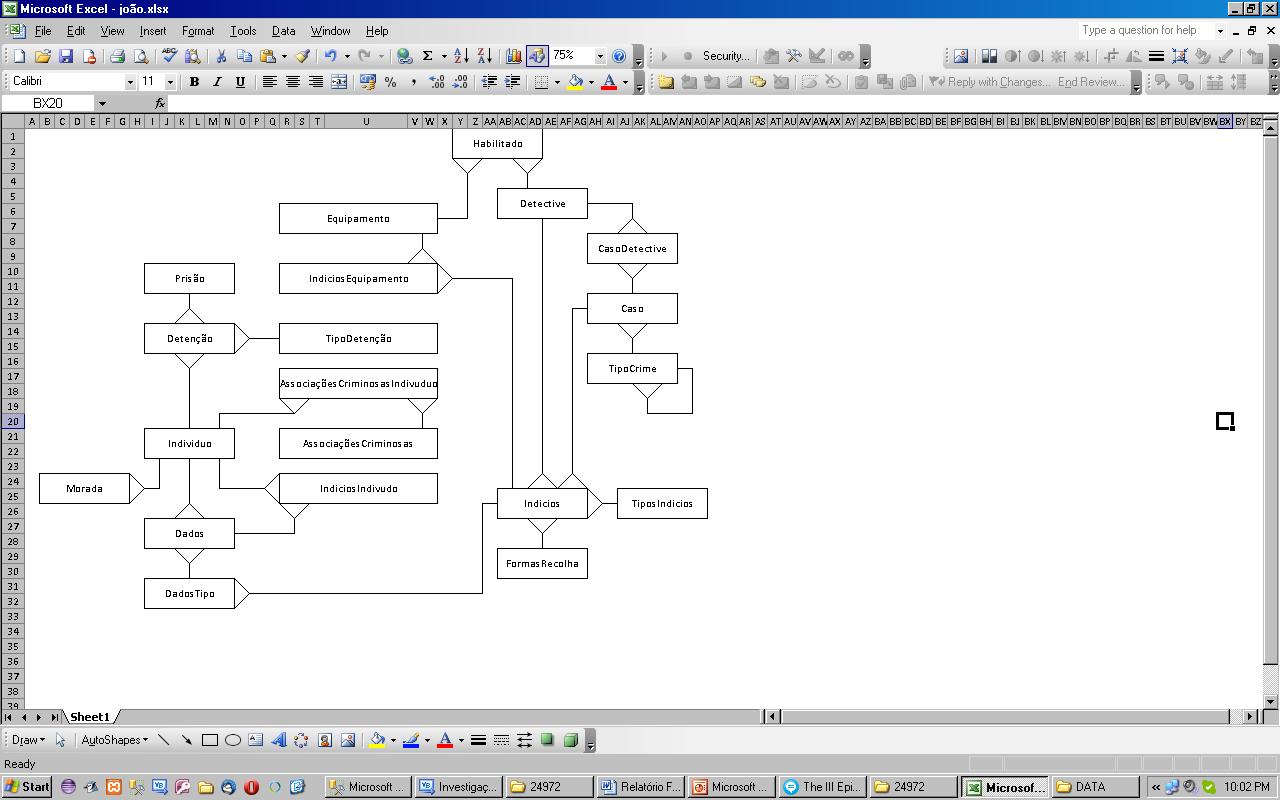


Ilustração – Modelo Logico

# Modelo Físico

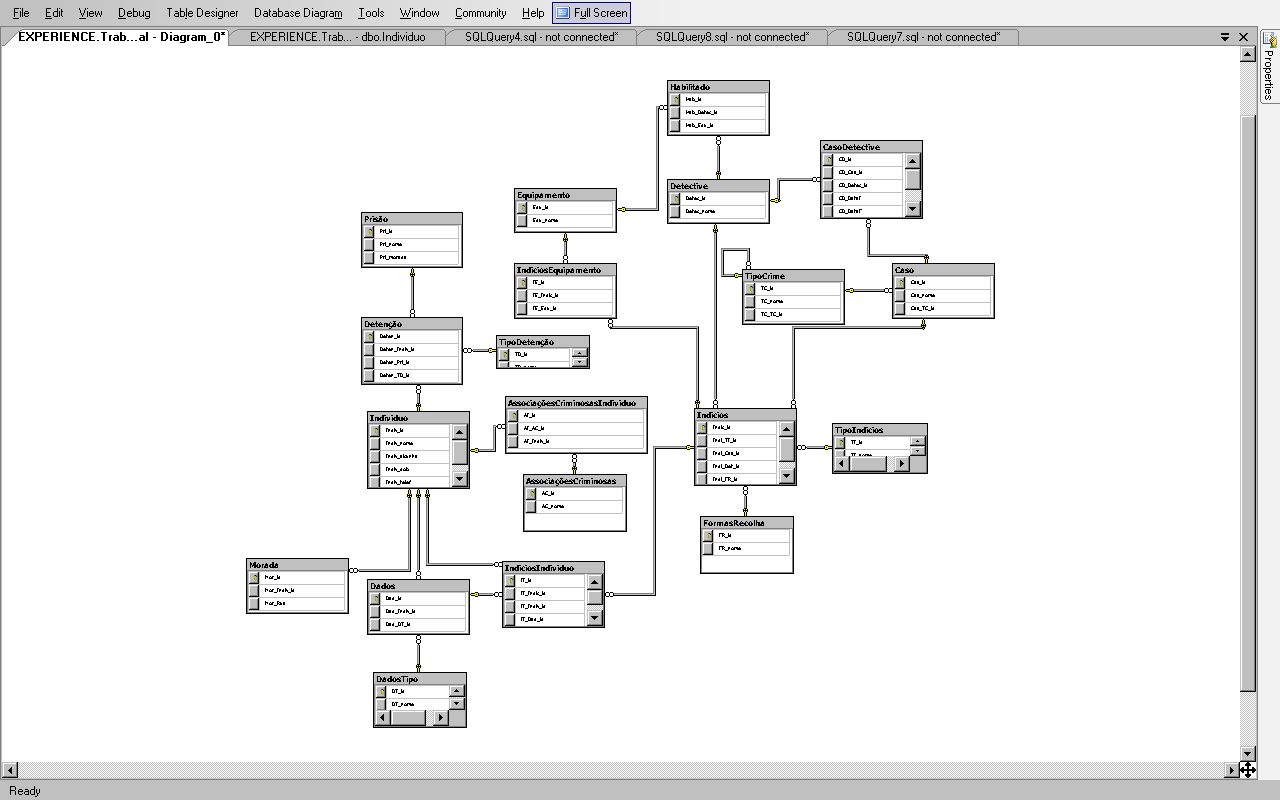


Ilustração – Modelo Físico (Microsoft SQL Server Management Studio R2)

# Modelo Conceptual

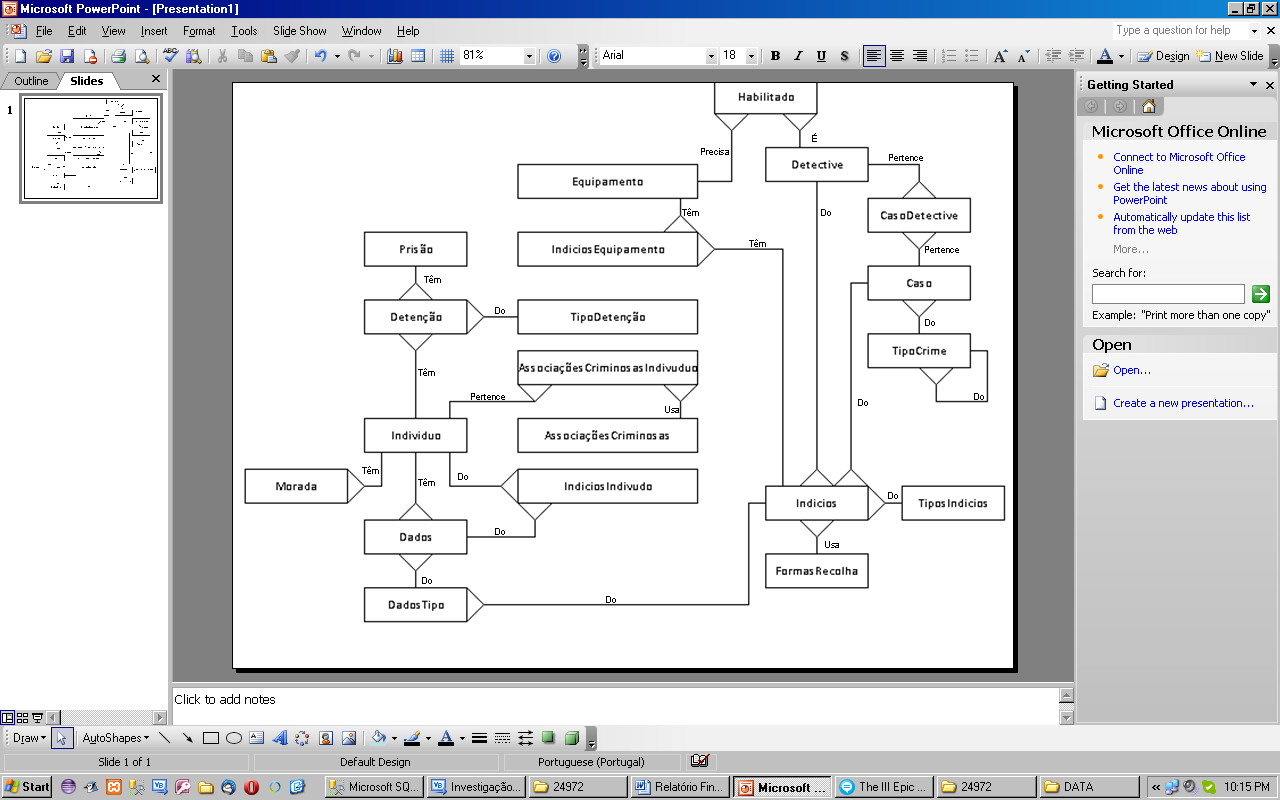


Ilustração – Modelo Conceptual

# Justificação das Opções Tomadas

|  |  |
| --- | --- |
| **Tipo de Dado** | **Uso** |
| Bigint | IDs únicos, valores inteiros. Chaves primárias e externas (FK, Foreign key) |
| nvarchar(50) | Variáveis de tipo “String” |
| Date | Data, necessário para cálculos relativos a datas, dai ser preferível a inteiros. |
| numeric(9, 0) | Formato escolhido para guardar números de telefone, podia ser mais desenvolvido para incluir indicativos nacionais e validar o número da operadora etc… |

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

# Manual de Utilizador Sucinto

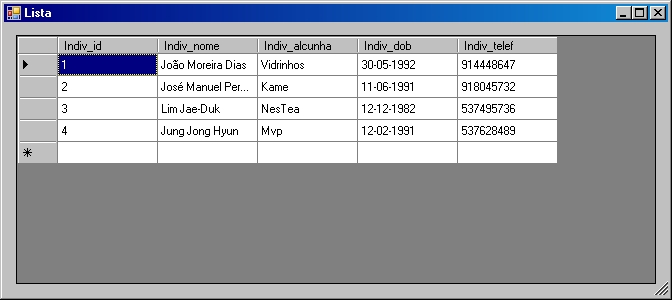


Ilustração – Lista

O programa inicializa-se com uma Lista de inviduos, ao selecionar uma linha, o registo do individuo abre em outra janela.

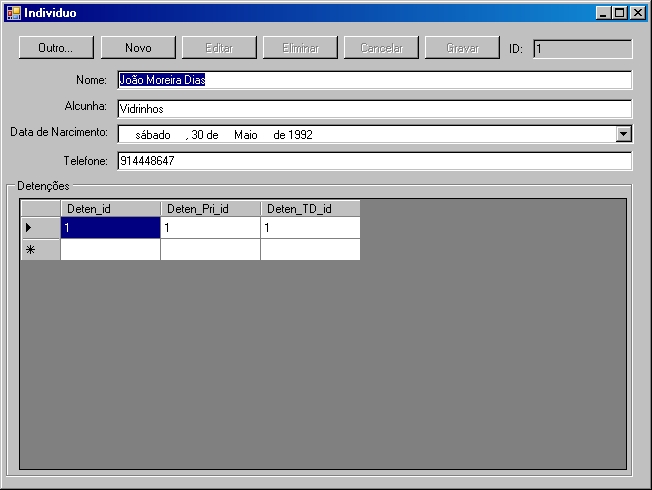


Ilustração – Individuo

O registo do individuo mostra toda a informação relativa ao individuo. Os botões fazem a manutenção do registo (do individuo).

# Questões

## A-

ALTER Procedure [dbo].[A]

@id bigint

AS

BEGIN

SELECT Individuo.Indiv\_nome

FROM Caso INNER JOIN

Indicios ON Caso.Cas\_id = Indicios.Indi\_Cas\_id INNER JOIN

IndiciosIndividuo ON Indicios.Indic\_id = IndiciosIndividuo.II\_Indic\_id INNER JOIN

Individuo ON IndiciosIndividuo.II\_Indiv\_id = Individuo.Indiv\_id

WHERE Cas\_id = @id

END

## B-

ALTER Procedure [dbo].[B]

@id bigint

AS

BEGIN

SELECT TOP (5) Individuo.Indiv\_nome

FROM Detenção INNER JOIN

Individuo ON Detenção.Deten\_Indiv\_id = Individuo.Indiv\_id

END

## C-

ALTER Procedure [dbo].[C]

@id bigint

AS

BEGIN

SELECT Equipamento.Equ\_nome

FROM Equipamento INNER JOIN

Habilitado ON Equipamento.Equ\_id = Habilitado.Hab\_Equ\_id INNER JOIN

Detective ON Habilitado.Hab\_Detec\_id = Detective.Detec\_id INNER JOIN

CasoDetective ON Detective.Detec\_id = CasoDetective.CD\_Detec\_id INNER JOIN

Caso ON CasoDetective.CD\_Cas\_id = Caso.Cas\_id

WHERE (Caso.Cas\_id = @id)

END

## D-

ALTER Procedure [dbo].[D]

@id bigint

AS

BEGIN

SELECT TipoDetenção.TD\_id, TipoDetenção.TD\_nome

FROM AssociaçõesCriminosas INNER JOIN

AssociaçõesCriminosasIndividuo ON AssociaçõesCriminosas.AC\_id = AssociaçõesCriminosasIndividuo.AI\_AC\_id INNER JOIN

Individuo ON AssociaçõesCriminosasIndividuo.AI\_Indiv\_id = Individuo.Indiv\_id INNER JOIN

Detenção ON Individuo.Indiv\_id = Detenção.Deten\_Indiv\_id INNER JOIN

TipoDetenção ON Detenção.Deten\_TD\_id = TipoDetenção.TD\_id

WHERE (AssociaçõesCriminosas.AC\_id = @id)

END

## E-

create trigger E

on Indicios

after insert

as

BEGIN

IF Habilitado(Hab\_Detec) != Indi\_Det\_id AND Habilitado(Equ\_id) != IndiciosEquipamento.IE\_Equ.id

RAISERROR ('ERRO! DETECTIVE NÃO HABILITADO!', 16, 1)

ROLLBACK TRANSACTION

UPDATE Indicios

SET Indi\_Ti\_id = Indi\_Ti\_id

FROM Equipamento INNER JOIN

Habilitado ON Equipamento.Equ\_id = Habilitado.Hab\_Equ\_id INNER JOIN

Detective ON Habilitado.Hab\_Detec\_id = Detective.Detec\_id INNER JOIN

IndiciosEquipamento ON Equipamento.Equ\_id = IndiciosEquipamento.IE\_Equ\_id

END

## F-

ALTER Procedure [dbo].[F]

@DI date,

@DF date

AS

BEGIN

SELECT Detec\_nome, DATEDIFF(day,@DI,@DF) AS 'Tempo Gasto'

FROM CasoDetective INNER JOIN

Detective ON CasoDetective.CD\_Detec\_id = Detective.Detec\_id

WHERE @DI<=CD\_DataI AND @DF>=CD\_DataF

GROUP BY Detec\_nome

END

## G-

create trigger G

on CasoDetective

after insert, update

as

BEGIN

UPDATE CasoDetective SET CasoDetective.CD\_Tempo = DATEDIFF(day,CD\_DataI,CD\_DataF)

end

## H-

create trigger H

on Indicios

after insert, update

as

BEGIN

IF UPDATE(Indi\_Cont)

UPDATE Parts

SET Indi\_Cont = INSIRTED.Indi\_Cont + ':' + Indi\_Cont

FROM Parts

INNER JOIN INSIRTED ON Indi\_Cont.Indic\_id = INSERTED.Indic\_id

END

# Código Visual Basic .NET

### clsIndiv.vb

Public Class clsIndiv

Private myCnn As SqlClient.SqlConnection

Private myIndiv\_id As Long

Private myIndiv\_nome As String

Private myIndiv\_alcunha As String

Private myIndiv\_dob As Date

Private myIndiv\_telef As Integer

Public Event Action(ByVal status As Integer)

Public Event Errors(ByVal errMessages As Object)

Public Sub New(ByVal dbConStr As String)

myCnn = New SqlClient.SqlConnection(dbConStr)

End Sub

Public ReadOnly Property Indiv\_id As Long

Get

Indiv\_id = myIndiv\_id

End Get

End Property

Public Property Indiv\_nome As String

Get

Indiv\_nome = myIndiv\_nome

End Get

Set(ByVal value As String)

myIndiv\_nome = value

End Set

End Property

Public Property Indiv\_alcunha As String

Get

Indiv\_alcunha = myIndiv\_alcunha

End Get

Set(ByVal value As String)

myIndiv\_alcunha = value

End Set

End Property

Public Property Indiv\_dob As Date

Get

Indiv\_dob = myIndiv\_dob

End Get

Set(ByVal value As Date)

myIndiv\_dob = value

End Set

End Property

Public Property Indiv\_telef As Integer

Get

Indiv\_telef = myIndiv\_telef

End Get

Set(ByVal value As Integer)

myIndiv\_telef = value

End Set

End Property

'Executa o stored procedure Individuo\_get\_indiv, busca o individuo por ID Unico

Public Sub getIndiv(ByVal Indiv\_id As Long)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Individuo\_get\_indiv"

End With

Dim prm As SqlClient.SqlParameter

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@Indiv\_id"

.DbType = DbType.Int64

.Direction = ParameterDirection.Input

.Value = Indiv\_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.myIndiv\_id = .Item("Indiv\_id")

Me.myIndiv\_nome = .Item("Indiv\_nome")

Me.myIndiv\_alcunha = .Item("Indiv\_alcunha")

Me.myIndiv\_dob = .Item("Indiv\_dob")

Me.myIndiv\_telef = .Item("Indiv\_telef")

End With

RaiseEvent Action(-1)

Else

Me.myIndiv\_id = vbNull

Me.myIndiv\_nome = vbNull

Me.myIndiv\_alcunha = vbNull

Me.myIndiv\_dob = Nothing

Me.myIndiv\_telef = vbNull

RaiseEvent Action(2)

End If

Catch ex As Exception

RaiseEvent Errors(ex.Message)

End Try

End Sub

'Executa o stored procedure Individuo\_get\_all, busac todos os Individuos

Public Function listIndiv() As DataTable

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandText = "Individuo\_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)

listIndiv = tbl

myCnn.Close()

Catch ex As Exception

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

listIndiv = Nothing

RaiseEvent Errors(ex.Message)

End Try

End Function

'Executa o stored procedure Individuo\_get\_detencoes, busac todas as detençõens pelo ID Unico do individuo em questão

Public Function getDeten() As DataTable

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Individuo\_get\_detencoes"

End With

Dim prm As SqlClient.SqlParameter

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@id"

.DbType = DbType.Int64

.Direction = ParameterDirection.Input

.Value = Indiv\_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)

getDeten = tbl

myCnn.Close()

Catch ex As Exception

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

getDeten = Nothing

RaiseEvent Errors(ex.Message)

End Try

End Function

'Executa o stored procedure INDIV\_DEL, para apagar o Individuo

Public Sub del(ByVal indID As Long)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "INDIV\_DEL"

End With

Dim prm As SqlClient.SqlParameter

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@indivId"

.DbType = DbType.Int64

.Direction = ParameterDirection.Input

.Value = indID

End With

cmd.Parameters.Add(prm)

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 o stored procedure, para adicionar um individuo novo

Public Sub add(ByVal nome As String, ByVal alcunha As String, ByVal dob As Date, ByVal telefone As Integer)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Individuo\_add"

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)

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@alcunha"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = alcunha

End With

cmd.Parameters.Add(prm)

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@dob"

.DbType = DbType.Date

.Direction = ParameterDirection.Input

.Value = dob

End With

cmd.Parameters.Add(prm)

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@telefone"

.DbType = DbType.Int32

.Direction = ParameterDirection.Input

.Value = telefone

End With

cmd.Parameters.Add(prm)

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 o stored procedure, para editar o Individuo

Public Sub edit(ByVal id As Long, ByVal nome As String, ByVal alcunha As String, ByVal dob As Date, ByVal telefone As Integer)

Dim cmd As New SqlClient.SqlCommand

With cmd

.Connection = myCnn

.CommandType = CommandType.StoredProcedure

.CommandText = "Indiv\_Edit"

End With

Dim prm As SqlClient.SqlParameter

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@id"

.DbType = DbType.UInt64

.Direction = ParameterDirection.Input

.Value = id

End With

cmd.Parameters.Add(prm)

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@nome"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = nome

End With

cmd.Parameters.Add(prm)

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@alcunha"

.DbType = DbType.String

.Direction = ParameterDirection.Input

.Value = alcunha

End With

cmd.Parameters.Add(prm)

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@dob"

.DbType = DbType.Date

.Direction = ParameterDirection.Input

.Value = dob

End With

cmd.Parameters.Add(prm)

prm = New SqlClient.SqlParameter

With prm

.ParameterName = "@telefone"

.DbType = DbType.Int32

.Direction = ParameterDirection.Input

.Value = telefone

End With

cmd.Parameters.Add(prm)

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

End Class

### Geral.vb

Module Geral

'Declarar a database connection string a base de dados

Public dbCNNstr As String = "Data Source=(local);Initial Catalog=Trabalho Prático 2 - Investigação Criminal;Integrated Security=True"

End Module

### INDIV.vb

Public Class INDIV

Private WithEvents myIndiv As clsIndiv

Private tabela As DataGridView

Private novoREG As Boolean

Public Event Action(ByVal status As Integer)

Public Event Errors(ByVal errMessages As Object)

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

'Inicializar um Individuo da clase clsIndiv, com a database connection string passada parametro

myIndiv = New clsIndiv(dbCNNstr)

'Preenche a lista de individuos

LISTA.DataGridView1.DataSource = myIndiv.listIndiv

LISTA.ShowDialog()

'Preenche a ficha do individuo

If Not LISTA.Escolha Is Nothing Then

myIndiv.getIndiv(LISTA.Escolha.Cells("Indiv\_id").Value)

Me.TextBox1.Text = myIndiv.Indiv\_nome

Me.TextBox2.Text = myIndiv.Indiv\_alcunha

Me.TextBox3.Text = myIndiv.Indiv\_telef

Me.TextBox4.Text = myIndiv.Indiv\_id

Me.DateTimePicker1.Value = myIndiv.Indiv\_dob

Me.DataGridView1.DataSource = myIndiv.getDeten

End If

Call prepForm(0)

End Sub

'Este botão volta a mostrar a lista de individuos

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

'Preenche a lista de individuos

LISTA.DataGridView1.DataSource = myIndiv.listIndiv

LISTA.ShowDialog()

'Preenche a ficha do individuo

If Not LISTA.Escolha Is Nothing Then

myIndiv.getIndiv(LISTA.Escolha.Cells("Indiv\_id").Value)

Me.TextBox1.Text = myIndiv.Indiv\_nome

Me.TextBox2.Text = myIndiv.Indiv\_alcunha

Me.TextBox3.Text = myIndiv.Indiv\_telef

Me.TextBox4.Text = myIndiv.Indiv\_id

Me.DateTimePicker1.Value = myIndiv.Indiv\_dob

Me.DataGridView1.DataSource = myIndiv.getDeten

End If

End Sub

'Limpa o registo

Private Sub newIndividuo()

Me.TextBox1.Text = ""

Me.TextBox2.Text = ""

Me.TextBox3.Text = ""

Me.DateTimePicker1.Text = ""

Me.TextBox4.Text = ""

Me.TextBox1.Focus()

novoREG = True

End Sub

'Gere a utilização dos botões, controlando as operaçõens de manutenção do quadro disponiveis

Sub prepForm(ByVal status As Integer)

Select Case status

Case 0

Button2.Enabled = True

Button1.Enabled = True

Button3.Enabled = False

Button6.Enabled = False

Button5.Enabled = False

Button4.Enabled = False

Case 1 'edit

Button2.Enabled = False

Button1.Enabled = False

Button3.Enabled = False

Button6.Enabled = False

Button5.Enabled = True

Button4.Enabled = True

Case 2 'novo

Button2.Enabled = False

Button1.Enabled = False

Button3.Enabled = False

Button6.Enabled = False

Button5.Enabled = True

Button4.Enabled = True

Case 3

Button2.Enabled = True

Button1.Enabled = True

Button3.Enabled = True

Button6.Enabled = True

Button5.Enabled = False

Button4.Enabled = False

End Select

End Sub

'Faz feedback ao utilizador com uma mensagem relativa a operação realizada e interage com o prepForm

Private Sub clsIndividAction(ByVal status As Short) Handles myIndiv.Action

Select Case status

Case -1

Call prepForm(3)

Case 2

Call prepForm(1)

novoREG = True

Case 5

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

Case 6

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

Call newIndividuo()

Call prepForm(0)

Case 7

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

Call prepForm(3)

Case 8

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

Call prepForm(3)

End Select

End Sub

'Botão Novo

Private Sub Button2\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button2.Click

Call newIndividuo()

Call prepForm(2)

Me.DataGridView1.DataSource = Nothing

End Sub

'Botão Eliminar

Private Sub Button6\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button6.Click

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

myIndiv.del(Me.TextBox4.Text)

Call newIndividuo()

Call prepForm(0)

Me.DataGridView1.DataSource = Nothing

End If

End Sub

'Botão Editar

Private Sub Button3\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button3.Click

novoREG = False

Call prepForm(1)

Me.TextBox1.Focus()

End Sub

'Botão Cancelar

Private Sub Button4\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button4.Click

Call newIndividuo()

Call prepForm(0)

End Sub

'Botão Gravar

Private Sub Button5\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button5.Click

If novoREG Then

myIndiv.add(Me.TextBox1.Text, Me.TextBox2.Text, Me.DateTimePicker1.Value, Me.TextBox3.Text)

Else

myIndiv.edit(Me.TextBox4.Text, Me.TextBox1.Text, Me.TextBox2.Text, Me.DateTimePicker1.Value, Me.TextBox3.Text)

End If

End Sub

End Class

### LISTA.vb

Public Class LISTA

Private linha As DataGridViewRow

Private WithEvents myIndiv As clsIndiv

Public ReadOnly Property Escolha As DataGridViewRow

Get

Escolha = linha

End Get

End Property

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

linha = Nothing

End Sub

'Este event DoubleCLick abre o resgisto relativo a linha seleciona

Private Sub DataGridView1\_DoubleClick(ByVal sender As Object, ByVal 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.INDIV\_DEL

ALTER PROCEDURE [dbo].[INDIV\_DEL]

@indivId bigint

AS

BEGIN

delete from Individuo where Individuo.Indiv\_id = @indivId

END

### dbo.Indiv\_Edit

ALTER PROCEDURE [dbo].[Individuo\_add]

@nome nvarchar(50),

@alcunha nvarchar(50),

@dob date,

@telefone numeric(9,0)

AS

BEGIN

insert into Individuo(Indiv\_nome, Indiv\_alcunha, Indiv\_dob, Indiv\_telef) values(@nome, @alcunha, @dob, @telefone)

END

### dbo.Individuo\_add

ALTER PROCEDURE [dbo].[Individuo\_add]

@nome nvarchar(50),

@alcunha nvarchar(50),

@dob date,

@telefone numeric(9,0)

AS

BEGIN

insert into Individuo(Indiv\_nome, Indiv\_alcunha, Indiv\_dob, Indiv\_telef) values(@nome, @alcunha, @dob, @telefone)

END

### dbo.Individuo\_get\_all

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

AS

BEGIN

SELECT Individuo.\*

FROM Individuo

END

### dbo.Individuo\_get\_detencoes

ALTER PROCEDURE [dbo].[Individuo\_get\_detencoes]

@id bigint

AS

BEGIN

SELECT Detenção.Deten\_id, Detenção.Deten\_Pri\_id, Detenção.Deten\_TD\_id

FROM Detenção INNER JOIN

Individuo ON Detenção.Deten\_Indiv\_id = Individuo.Indiv\_id

WHERE (Detenção.Deten\_Indiv\_id = @id)

END

### dbo.Individuo\_get\_indiv

ALTER PROCEDURE [dbo].[Individuo\_get\_indiv]

@indiv\_id bigint

AS

BEGIN

SELECT Individuo.\* FROM Individuo WHERE Indiv\_id=@indiv\_id

END