

Fundamentos de Ingeniería de Software

Profesor: Raimundo Vega/Juan Pablo Salazar

Ingeniería Civil Informática.

Universidad Austral de Chile

**Todo Transportes Código fuente: Matricular Cliente**

**Integrantes:** José Luis Acuña Oyarce.

Luis Barrientos Fajardo.

Kurt Poehler Widemann.

**Fecha**: 05 de marzo de 2015.

1. **Bienvenido.vb:** Ventana que se encarga de ingresar a un usuario registrado, y desplegar la pantalla principal “Principal.vb”.

Public Class Bienvenido

Dim con As Conexion = New Conexion

Dim resultado As String = ""

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

If validar() Then

Dim login As Boolean = con.iniciarSesion(tbox\_Usuario.Text, tbox\_Contra.Text)

Dim d As DataTable = con.doQuery("SELECT Tipo FROM Usuario WHERE Nombre='" & tbox\_Usuario.Text & "'")

If d.Rows.Count > 0 Then

resultado = d.Rows(0).Item(0).ToString

Else

resultado = ""

End If

If login And resultado = "Desarrollador" Then

Dim prueba As Prueba = New Prueba(tbox\_Usuario.Text, con)

prueba.Show()

prueba.SendToBack()

Dim principal As Principal = New Principal(tbox\_Usuario.Text, con)

principal.Show()

Me.Close()

ElseIf login And resultado = "Administrador" Then

Dim principal As Principal = New Principal(tbox\_Usuario.Text, con)

principal.Show()

Me.Close()

End If

'If login Then

' Dim prueba As Prueba = New Prueba(tbox\_Usuario.Text, con)

' prueba.Show()

' prueba.SendToBack()

' Dim principal As Principal = New Principal(tbox\_Usuario.Text, con)

' principal.Show()

' Me.Close()

'Else

' Estado.Text = "El Usuario y Contraseña ingresados no son validos."

'End If

End If

End Sub

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

'MOMENTANEO POR DESARROLLO

tbox\_Usuario.Text = "Desarrollador"

tbox\_Contra.Text = "desarrollador"

End Sub

Function validar() As Boolean

If tbox\_Usuario.Text.Trim.Equals("") And tbox\_Contra.Text.Trim.Equals("") Then

Estado.Text = "Debe ingresar sus datos de inicio de sesión para continuar."

tbox\_Usuario.Focus()

Return False

ElseIf tbox\_Usuario.Text.Trim.Equals("") Then

Estado.Text = "Debe ingresar sus datos de usuario de sesión para continuar."

tbox\_Usuario.Focus()

Return False

ElseIf tbox\_Contra.Text.Trim.Equals("") Then

Estado.Text = "Debe ingresar sus datos de usuario de sesión para continuar."

tbox\_Contra.Focus()

Return False

End If

Return True

End Function

End Class

1. **Coneccion.vb:** Clase que se encarga de administrar la conexión con el motor de base de datos MySQL.

Imports MySql.Data.MySqlClient

Public Class Conexion

Dim cs As String = "Database=ttpt;Data Source=localhost;" \_

& "User Id=testUser;Password=12Test34@"

Dim conn As New MySqlConnection(cs)

Dim adaptador As MySqlDataAdapter

Dim lector As MySqlDataReader

Dim comando As New MySqlCommand

Dim transaccion As MySqlTransaction

Sub New()

comando.CommandType = CommandType.Text

comando.Connection = conn

End Sub

Function doQuery(ByVal SQLQuery As String) As DataTable

'Realiza cualquier consulta y retorna los resultados en una tabla (DataTable).

Dim Data As DataTable = New DataTable

Using comando As New MySqlCommand()

comando.Connection = conn

comando.CommandText = SQLQuery

Try

Dim SQLAdapter As New MySqlDataAdapter(comando)

SQLAdapter.Fill(Data)

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

Return Data

End Function

Function doInsert(ByVal Tabla As String, ByVal Columnas() As String, ByVal Parametros() As String) As Integer

'Realiza cualquier insert y retorna la PK si es que es autoincrementable

'Si no es autoincrementable o si ocurre algún error, retorna -1.

Dim r As Integer = -1

Using comando As New MySqlCommand()

comando.Connection = conn

comando.CommandType = CommandType.Text

comando.Transaction = transaccion

Dim SQLQuery As String = "INSERT INTO " & Tabla & " ("

Dim Values As String = ""

For i As Integer = 0 To Columnas.Length - 1

If i < Columnas.Length - 1 Then

SQLQuery = SQLQuery & Columnas(i) & ", "

Values = Values & "@param" & i & ", "

Else

SQLQuery = SQLQuery & Columnas(i) & ") VALUES ("

Values = Values & "@param" & i & ") "

SQLQuery = SQLQuery & Values

End If

Next

comando.CommandText = SQLQuery

For i As Integer = 0 To Parametros.Length - 1

comando.Parameters.AddWithValue("@param" & i, Parametros(i))

Next

Try

comando.ExecuteNonQuery()

r = comando.LastInsertedId()

Catch ex As Exception

rollbackTransaction()

MsgBox(ex.Message.ToString)

Return r

End Try

End Using

Return r

End Function

Function doUpdate(ByVal Tabla As String, ByVal Columnas() As String, ByVal Parametros() As String, ByVal Condicion As String) As Integer

'Realiza Updates y retorna 0 en caso de éxito y -1 en caso de error

Dim ret As Integer = 0

Dim SQLQuery As String = "UPDATE " & Tabla & " SET"

For Each par As String() In Columnas.Zip(Parametros, Function(a, b) {a, b}).ToArray()

SQLQuery = SQLQuery & " " & par(0) & "='" & par(1) & "',"

Next

SQLQuery = SQLQuery.TrimEnd(",")

SQLQuery = SQLQuery & " " & Condicion

Using comando As New MySqlCommand()

comando.Connection = conn

comando.CommandText = SQLQuery

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

ret = -1

End Try

End Using

Return ret

End Function

Sub beginTransaction()

transaccion = conn.BeginTransaction

End Sub

Sub commitTransaction()

transaccion.Commit()

End Sub

Sub rollbackTransaction()

transaccion.Rollback()

End Sub

#Region "DESARROLLO"

Sub iSesion()

'INICIA SESIÓN SIN PEDIR USER/PASS

Using comando As New MySqlCommand()

With comando

.CommandType = CommandType.Text

.Connection = conn

End With

Try

conn.Open()

Catch ex As Exception

MsgBox(ex.ToString)

End Try

End Using

End Sub

#End Region

#Region "HERRAMIENTAS"

Function count(ByVal NombreTabla As String) As String

'Retorna el numero de registros en 'NombreTabla'

Dim cant As String = ""

Using comando As New MySqlCommand()

With comando

.CommandText = "SELECT COUNT(\*) FROM " & NombreTabla

.CommandType = CommandType.Text

.Connection = conn

End With

Try

comando.ExecuteNonQuery()

cant = comando.ExecuteScalar()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

Return cant

End Function

Function countWhere(ByVal NombreTabla As String, ByVal Condicion As String) As String

'Retorna el numero de registros en 'NombreTabla'

Dim cant As String = ""

Using comando As New MySqlCommand()

With comando

.CommandText = "SELECT COUNT(\*) FROM " & NombreTabla & " WHERE " & Condicion

.CommandType = CommandType.Text

.Connection = conn

End With

Try

comando.ExecuteNonQuery()

cant = comando.ExecuteScalar()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

Return cant

End Function

Function last(ByVal columna As String, ByVal tabla As String) As String

'Funcion que retorna el último 'columna' de la tabla 'tabla'

Dim res As String = ""

Using comando As New MySqlCommand()

comando.CommandText = "SELECT max(" & columna & ") AS " & columna & " FROM " & tabla

comando.CommandType = CommandType.Text

comando.Connection = conn

Try

comando.ExecuteNonQuery()

res = Convert.ToString(comando.ExecuteScalar)

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

Return res

End Function

Function toArray(ByVal n As Integer, ByVal NombreCampo As String, ByVal NombreTabla As String) As String()

'Retorna un arreglo con los datos 'NombreCampo' de la tabla 'NombreTabla'

Dim arreglo(n) As String

Using comando As New MySqlCommand()

With comando

.CommandText = "SELECT " & NombreCampo & " FROM " & NombreTabla

.CommandType = CommandType.Text

.Connection = conn

End With

Try

Dim i As Integer = 0

Using lector As MySqlDataReader = comando.ExecuteReader()

While lector.Read()

arreglo(i) = lector.GetString(0)

i = i + 1

End While

End Using

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

Return arreglo

End Function

Function toArrayWhere(ByVal n As Integer, ByVal NombreCampo As String, ByVal NombreTabla As String, ByVal Condicion As String) As String()

'Retorna un arreglo con los datos 'NombreCampo' de la tabla 'NombreTabla'

Dim arreglo(n) As String

Using comando As New MySqlCommand()

With comando

.CommandText = "SELECT " & NombreCampo & " FROM " & NombreTabla & " WHERE " & Condicion

.CommandType = CommandType.Text

.Connection = conn

End With

Try

Dim i As Integer = 0

Using lector As MySqlDataReader = comando.ExecuteReader()

While lector.Read()

arreglo(i) = lector.GetString(0)

i = i + 1

End While

End Using

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

Return arreglo

End Function

Function toArray2Where(ByVal n As Integer, ByVal NombreCampos As String, ByVal NombreTabla As String, ByVal Condicion As String) As String(,)

'Retorna un arreglo con los datos 'NombreCampo' de la tabla 'NombreTabla'

Dim arreglo(n, 2) As String

Using comando As New MySqlCommand()

With comando

.CommandText = "SELECT " & NombreCampos & " FROM " & NombreTabla & " WHERE " & Condicion

.CommandType = CommandType.Text

.Connection = conn

End With

Try

Dim i As Integer = 0

Using lector As MySqlDataReader = comando.ExecuteReader()

While lector.Read()

arreglo(i, 0) = lector.GetString(0)

arreglo(i, 1) = lector.GetString(1)

MsgBox(arreglo(i, 1))

i = i + 1

End While

End Using

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

Return arreglo

End Function

Function selectWhereQuery(ByVal columna As String, ByVal tabla As String, ByVal condicion As String) As String

'DEVUELVE UN DATO EN ESPECIFICO EN LA BD

Dim res As String = ""

Using comando As New MySqlCommand()

With comando

.CommandText = "SELECT " & columna & " FROM " & tabla & " WHERE " & condicion

.CommandType = CommandType.Text

.Connection = conn

End With

Try

comando.ExecuteNonQuery()

res = comando.ExecuteScalar

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

Return res

End Function

Function iniciarSesion(ByVal usuario As String, ByVal contra As String) As Boolean

'Verifica USUARIO/CONTRASEÑA y permite el ingreso al sistema

'Abre la conexión

Using comando As New MySqlCommand()

With comando

.CommandText = "SELECT Contraseña FROM USUARIO WHERE Nombre = '" & usuario & "'"

.CommandType = CommandType.Text

.Connection = conn

End With

Try

conn.Open()

comando.ExecuteNonQuery()

Dim pass As String = Convert.ToString(comando.ExecuteScalar)

If pass.Equals(contra) Then

Return True

Else

conn.Close()

Return False

End If

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

Return False

End Function

Sub Close()

'CIERRA LA BD

Using comando As New MySqlCommand()

comando.Connection = conn

End Using

Try

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

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Sub

#End Region

#Region "INSERTS"

Sub regFuncionario(ByVal Nombre As String, ByVal Tipo As String)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO FUNCIONARIO (Nombre, Tipo) VALUES(@Nombre, @Tipo)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Nombre", Nombre)

.Parameters.AddWithValue("@Tipo", Tipo)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regUsuario(ByVal Nombre As String, ByVal Contra As String, ByVal Tipo As String, ByVal Funcionario As String)

Dim idFuncionario As String = Me.selectWhereQuery("idFUNCIONARIO", "FUNCIONARIO", "Nombre = '" & Funcionario & "'")

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO USUARIO (Nombre, Contraseña, Tipo, Funcionario) VALUES(@Nombre, @Contraseña, @Tipo, @Funcionario)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Nombre", Nombre)

.Parameters.AddWithValue("@Contraseña", Contra)

.Parameters.AddWithValue("@Tipo", Tipo)

.Parameters.AddWithValue("@Funcionario", CInt(idFuncionario))

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regAtencion(ByVal usuario As String, ByVal idcliente As Integer, ByVal fecha As String, ByVal Tipo As String)

'Realiza el registro en la tabla ATENCION\_CLIENTE

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Atencion\_cliente\_potencial (Usuario, Cliente\_Potencial, Fecha, Tipo) VALUES(@Usuario,@Cliente,@Fecha,@Tipo)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Usuario", usuario)

.Parameters.AddWithValue("@Cliente", idcliente)

.Parameters.AddWithValue("@Fecha", fecha)

.Parameters.AddWithValue("@Tipo", Tipo) 'momentaneo'

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Function regClientePotencial(ByVal Nombre As String, ByVal Telefono As Integer, ByVal Producto As String, ByVal Extra As String) As Integer

'Realiza el registro en la tabla CLIENTE

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Cliente\_potencial (Nombre, Telefono, Producto, Extra) VALUES(@Nombre, @Telefono, @Producto, @Extra)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Nombre", Nombre)

.Parameters.AddWithValue("@Telefono", Telefono)

.Parameters.AddWithValue("@Producto", Producto)

.Parameters.AddWithValue("@Extra", Extra)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

Return comando.LastInsertedId

End Using

End Function

Sub regVehiculo(ByVal Patente As String, ByVal Modelo As String, ByVal Estado As String)

'Realiza el registro en la tabla CLIENTE

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Auto\_Escuela (Matricula, Modelo, Estado) VALUES(@Patente, @Modelo, @Estado)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Patente", Patente)

.Parameters.AddWithValue("@Modelo", Modelo)

.Parameters.AddWithValue("@Estado", Estado)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regDocente(ByVal ID As Integer, ByVal Telefono As Integer, ByVal Tipo As String)

'Realiza el registro en la tabla Docente

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Docente (idDocente, Telefono, Tipo) VALUES(@id, @Telefono, @Tipo)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@id", ID)

.Parameters.AddWithValue("@Telefono", Telefono)

.Parameters.AddWithValue("@Tipo", Tipo)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regInstructor(ByVal id As Integer, ByVal Auto As String)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Instructor (idInstructor, Auto) VALUES(@id, @Auto)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@id", id)

.Parameters.AddWithValue("@Auto", Auto)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regProfesor(ByVal id As Integer)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Profesor (idProfesor) VALUES(@id)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@id", id)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regCliente(ByVal Nombre As String, ByVal RUT As String, ByVal Fecha As String, ByVal Telefono As String, ByVal Email As String)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Cliente (Nombre,Rut,Fecha\_Nac,Telefono,Email) VALUES(@Nombre,@Rut,@Fecha,@Telefono,@Email)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Nombre", Nombre)

.Parameters.AddWithValue("@Rut", RUT)

.Parameters.AddWithValue("@Fecha", Fecha)

.Parameters.AddWithValue("@Telefono", Telefono)

.Parameters.AddWithValue("@Email", Email)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regCompra(ByVal Cliente As Integer, ByVal Producto As String, ByVal Fecha As String)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Compra (Cliente,Producto,Fecha) VALUES(@Cliente,@Producto,@Fecha)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Cliente", Cliente)

.Parameters.AddWithValue("@Producto", Producto)

.Parameters.AddWithValue("@Fecha", Fecha)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regPago(ByVal Codigo As String, ByVal Monto As Integer, ByVal Medio\_Pago As String, ByVal Compra As Integer, ByVal Fecha As String)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Pago (Codigo,Monto,Medio\_Pago,Compra,Fecha) VALUES(@Codigo,@Monto,@Medio\_Pago,@Compra,@Fecha)"

.Connection = conn

.Parameters.AddWithValue("@Codigo", Codigo)

.Parameters.AddWithValue("@Monto", Monto)

.Parameters.AddWithValue("@Medio\_Pago", Medio\_Pago)

.Parameters.AddWithValue("@Compra", Compra)

.Parameters.AddWithValue("@Fecha", Fecha)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regMatricula(ByVal Codigo As String, ByVal Nombre As String, ByVal RUT As String, ByVal Edad As Integer, ByVal Fecha As String, ByVal Telefono As Integer, ByVal CodigoCompra As Integer, ByVal Curso As String, ByVal HoraTeorica As String, ByVal HoraPractica As String, ByVal Documento As String, ByVal Estudiante As Integer)

'Registra Matricula'

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO MATRICULA (Codigo,Nombre,RUT,Edad,Fecha,Telefono,CodigoCompra,Curso,HoraTeoria,HoraPractica,Documento,Estudiante) VALUES(@Codigo,@Nombre,@RUT,@Edad,@Fecha,@Telefono,@CodigoCompra,@Curso,@HoraTeorica,@HoraPractica,@Documento,@Estudiante)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Codigo", Codigo)

.Parameters.AddWithValue("@Nombre", Nombre)

.Parameters.AddWithValue("@RUT", RUT)

.Parameters.AddWithValue("@Edad", Edad)

.Parameters.AddWithValue("@Fecha", Fecha)

.Parameters.AddWithValue("@Telefono", Telefono)

.Parameters.AddWithValue("@CodigoCompra", CodigoCompra)

.Parameters.AddWithValue("@Curso", Curso)

.Parameters.AddWithValue("@HoraTeorica", HoraTeorica)

.Parameters.AddWithValue("@HoraPractica", HoraPractica)

.Parameters.AddWithValue("@Documento", Documento)

.Parameters.AddWithValue("@Estudiante", Estudiante)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regEstudiante(ByVal FotosCarnet As Integer, ByVal CertEstudios As Boolean, ByVal CertAlumnRegular As Boolean, ByVal AutNotarial As Boolean)

'Registra Estudiante'

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO ESTUDIANTE (FotosCarnet, CertEstudios, CertAlumnRegular, AutNotarial) VALUES( @FotosCarnet, @CertEstudios, @CertAlumnRegular, @AutNotarial)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@FotosCarnet", FotosCarnet)

.Parameters.AddWithValue("@CertEstudios", CertEstudios)

.Parameters.AddWithValue("@CertAlumnRegular", CertAlumnRegular)

.Parameters.AddWithValue("@AutNotarial", AutNotarial)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regDocumento(ByVal Documento As String)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Documento (Tipo) VALUES(@Tipo)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Tipo", Documento)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regDocumento2(ByVal Tipo As String, ByVal Funcionario As Integer, ByVal Fecha As String, ByVal Estado As String)

'Registra Documento'

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Documento (Tipo, Funcionario, Fecha, Estado) VALUES(@Tipo, @Funcionario, @Fecha, @Estado)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Tipo", Tipo)

.Parameters.AddWithValue("@Funcionario", Funcionario)

.Parameters.AddWithValue("@Fecha", Fecha)

.Parameters.AddWithValue("@Estado", Estado)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regCurso(ByVal Codigo As String, ByVal Producto As String, ByVal FechaInicio As String, ByVal FechaTermino As String)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Curso (Codigo, Producto, FechaInicio, FechaTermino) VALUES(@Codigo, @Producto, @FechaInicio, @FechaTermino)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Codigo", Codigo)

.Parameters.AddWithValue("@Producto", Producto)

.Parameters.AddWithValue("@FechaInicio", FechaInicio)

.Parameters.AddWithValue("@FechaTermino", FechaTermino)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regTeoria(ByVal Codigo As String, ByVal Curso As String, ByVal Horario As String, ByVal Duracion As Integer, ByVal idProfesor As Integer)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Teoria (idTeoria,Clase,Horario,Duracion,Profesor) VALUES(@Codigo,@Curso,@Horario,@Duracion,@idProfesor)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Codigo", Codigo)

.Parameters.AddWithValue("@Curso", Curso)

.Parameters.AddWithValue("@Horario", Horario)

.Parameters.AddWithValue("@Duracion", Duracion)

.Parameters.AddWithValue("@idProfesor", idProfesor)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regRueda(ByVal Documento As Integer, ByVal Horario As String)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO CAMBIO\_RUEDA (Documento,Horario) VALUES(@Documento,@Horario)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Documento", Documento)

.Parameters.AddWithValue("@Horario", Horario)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regPsico(ByVal Documento As Integer)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO PSICOTECNICO (Documento) VALUES(@Documento)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Documento", Documento)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regVisual(ByVal Documento As Integer, ByVal Certificado As Boolean)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO EXAMEN\_VISUAL (Documento,Certificado) VALUES(@Documento,@Certificado)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Documento", Documento)

.Parameters.AddWithValue("@Certificado", Certificado)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regPractica(ByVal Codigo As String, ByVal Clase As String, ByVal Instructor As Integer, ByVal Horario As String)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO Practica (idPractica,Clase,Instructor,Horario) VALUES(@Codigo,@Clase,@Instructor,@Horario)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Codigo", Codigo)

.Parameters.AddWithValue("@Clase", Clase)

.Parameters.AddWithValue("@Instructor", Instructor)

.Parameters.AddWithValue("@Horario", Horario)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regExTeo(ByVal Documento As Integer, ByVal Calificacion As Integer)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO EXAMEN\_TEORICO (Documento,Calificacion) VALUES(@Documento,@Calificacion)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Documento", Documento)

.Parameters.AddWithValue("@Calificacion", Calificacion)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regExPract(ByVal Documento As Integer, ByVal Calificacion As Integer)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO EXAMEN\_PRACTICO (Documento,Calificacion) VALUES(@Documento,@Calificacion)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Documento", Documento)

.Parameters.AddWithValue("@Calificacion", Calificacion)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regExMun(ByVal Documento As Integer, ByVal Comentario As String)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO EXAMEN\_MUNICIPAL (Documento,Comentario) VALUES(@Documento,@Comentario)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Documento", Documento)

.Parameters.AddWithValue("@Comentario", Comentario)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

Sub regEstDoc(ByVal Estudiante As String, ByVal Documento As Integer)

Using comando As New MySqlCommand()

With comando

.CommandText = "INSERT INTO ESTUDIANTE\_DOCUMENTO (Estudiante, Documento) VALUES(@Estudiante, @Documento)"

.CommandType = CommandType.Text

.Connection = conn

.Parameters.AddWithValue("@Estudiante", Estudiante)

.Parameters.AddWithValue("@Documento", Documento)

End With

Try

comando.ExecuteNonQuery()

Catch ex As Exception

MsgBox(ex.Message.ToString)

End Try

End Using

End Sub

#End Region

End Class

1. **Principal.vb:** Ventana que se encarga de mostrar todos los componentes del software mediante un sistema de pestañas o “Tabs”, desplegada desde “Bienvenido.vb”

Public Class Principal

Dim con As Conexion

Dim USER As String = ""

Sub New(ByVal Usuario As String, ByVal conexion As Conexion)

'Guarda el nombre del usuario y la conexión

USER = Usuario

con = conexion

InitializeComponent()

End Sub

Private Sub Principal\_FormClosing(ByVal sender As Object, ByVal e As System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

con.Close()

Application.Exit()

End Sub

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

loadTAB(MainMenu.TabPages(MainMenu.SelectedIndex).Text, MainMenu.SelectedIndex)

ESTADO.Text = "Usuario: " & USER

End Sub

Private Sub MainMenu\_DrawItem(ByVal sender As Object, ByVal e As System.Windows.Forms.DrawItemEventArgs) Handles MainMenu.DrawItem

'Método extraido desde 'http://msdn.microsoft.com/en-us/library/ms404305(v=vs.90).aspx'

Dim g As Graphics = e.Graphics

Dim \_TextBrush As Brush

Dim \_TabPage As TabPage = MainMenu.TabPages(e.Index)

Dim \_TabBounds As Rectangle = MainMenu.GetTabRect(e.Index)

If (e.State = DrawItemState.Selected) Then

\_TextBrush = New SolidBrush(System.Drawing.SystemColors.ControlText)

g.FillRectangle(System.Drawing.SystemBrushes.ControlLight, e.Bounds)

Else

\_TextBrush = New System.Drawing.SolidBrush(System.Drawing.SystemColors.ControlText)

End If

Dim \_TabFont As New Font("Microsoft Sans Serif", 12.0, FontStyle.Regular, GraphicsUnit.Pixel)

Dim \_StringFlags As New StringFormat()

\_StringFlags.Alignment = StringAlignment.Center

\_StringFlags.LineAlignment = StringAlignment.Center

g.DrawString(\_TabPage.Text, \_TabFont, \_TextBrush, \_TabBounds, New StringFormat(\_StringFlags))

End Sub

Private Sub MainMenu\_SelectedIndexChanged(ByVal sender As Object, ByVal e As System.EventArgs) Handles MainMenu.SelectedIndexChanged

Dim tab As Integer = MainMenu.SelectedIndex

Dim Nombre As String = MainMenu.TabPages(tab).Text

loadTAB(Nombre, tab)

End Sub

Sub loadTAB(ByVal Nombre As String, ByVal tab As Integer)

'Se crea este método para poder editar el orden de las pestañas y no alterar el contenido

Dim Form As Form

Me.MainMenu.TabPages(tab).Controls.Clear()

If Nombre.Equals("Atención al Cliente") Then

Form = New RegistrarCliente(USER, con, ESTADO)

Form.TopLevel = False

Me.MainMenu.TabPages(tab).Controls.Add(Form)

Form.Show()

ElseIf Nombre.Equals("Matrícula") Then

Form = New Matricula(USER, con, ESTADO)

Form.TopLevel = False

Me.MainMenu.TabPages(tab).Controls.Add(Form)

Form.Show()

ElseIf Nombre.Equals("Profesores") Then

Form = New DocenteTabs(USER, con, ESTADO)

Form.TopLevel = False

Me.MainMenu.TabPages(tab).Controls.Add(Form)

Form.Show()

ElseIf Nombre.Equals("Encuesta de Calidad") Then

Form = New Encuesta(USER, con, ESTADO)

Form.TopLevel = False

Me.MainMenu.TabPages(tab).Controls.Add(Form)

Form.Show()

ElseIf Nombre.Equals("Vehículos") Then

Form = New newFlota(USER, con, ESTADO)

Form.TopLevel = False

Me.MainMenu.TabPages(tab).Controls.Add(Form)

Form.Show()

ElseIf Nombre.Equals("Configuración") Then

Form = New tab\_Config(USER, con, ESTADO)

Form.TopLevel = False

Me.MainMenu.TabPages(tab).Controls.Add(Form)

Form.Show()

ElseIf Nombre.Equals("Examenes") Then

Form = New newExamen(USER, con, ESTADO)

Form.TopLevel = False

Me.MainMenu.TabPages(tab).Controls.Add(Form)

Form.Show()

ElseIf Nombre.Equals("Cursos") Then

Form = New Curso(USER, con, ESTADO)

Form.TopLevel = False

Me.MainMenu.TabPages(tab).Controls.Add(Form)

Form.Show()

ElseIf Nombre.Equals("Resumen") Then

Form = New ResumenTabs(USER, con, ESTADO)

Form.TopLevel = False

Me.MainMenu.TabPages(tab).Controls.Add(Form)

Form.Show()

End If

End Sub

End Class

1. **Matricula.vb:** Ventana que se encarga de administrar el caso de uso “Matricular Cliente”, es llamada desde “Príncipal.vb” y llama a las clases “DataCBOX.vb” y “Coneccion.vb” para poder funcionar

Imports System.Globalization

Public Class Matricula

Private USER As String

Private con As Conexion

Private STATUS As ToolStripStatusLabel

Dim dc As DataCBOX

Dim MontoAPagar As Integer

Sub New(ByVal usuario As String, ByVal conexion As Conexion, ByVal estado As ToolStripStatusLabel)

con = conexion

USER = usuario

STATUS = estado

dc = New DataCBOX(conexion)

InitializeComponent()

End Sub

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

loadCBOX("FranqTributaria")

loadCBOX("Producto")

loadCBOX("MedioPago")

Me.LimpiarBtn.PerformClick()

End Sub

#Region "METODOS"

Sub loadCBOX(ByVal Nombre As String)

If Nombre.Equals("Producto") Then

ProductoCbBx.DataSource = dc.Productos

ProductoCbBx.DisplayMember = "Nombre"

ProductoCbBx.ValueMember = "Nombre"

ProductoCbBx.SelectedIndex = -1

ElseIf Nombre.Equals("FranqTributaria") Then

ClienteCbBx.DataSource = dc.Clientes

ClienteCbBx.DisplayMember = "Nombre"

ClienteCbBx.ValueMember = "Nombre"

ElseIf Nombre.Equals("CursosXProductos") Then

CursoCbBx.DataSource = dc.CursosDeProducto(ProductoCbBx.Text)

CursoCbBx.DisplayMember = "Codigo"

CursoCbBx.ValueMember = "Codigo"

CursoCbBx.SelectedItem = 0

HraTeoriaCbBx.Enabled = True

HraPracticaCbBx.Enabled = True

DiasGBx.Enabled = True

ElseIf Nombre.Equals("HTeoriaXCurso") Then 'terminar

HraTeoriaCbBx.DataSource = dc.HTeoriaDeCurso(CursoCbBx.Text)

HraTeoriaCbBx.DisplayMember = "Horario"

HraTeoriaCbBx.ValueMember = "Horario"

ElseIf Nombre.Equals("HPracticaXCurso") Then 'terminar

HraPracticaCbBx.DataSource = dc.HPracticaDeCurso(CursoCbBx.Text)

HraPracticaCbBx.DisplayMember = "Horario"

HraPracticaCbBx.ValueMember = "Horario"

ElseIf Nombre.Equals("DocumentoPago") Then 'terminar

DocPagoCbBx.DataSource = dc.DocPago

DocPagoCbBx.DisplayMember = "Value"

DocPagoCbBx.ValueMember = "Value"

ElseIf Nombre.Equals("MedioPago") Then 'terminar

MedioPagoChBx.DataSource = dc.MedioPago

MedioPagoChBx.ValueMember = "Nombre"

End If

End Sub

Function validar() As Boolean

STATUS.Text = ""

STATUS.ForeColor = System.Drawing.SystemColors.ControlText

If Me.NRegCTxBx.Text.Trim.Equals("") Then

STATUS.Text = "ERROR: Ingrese el Nro de Registro del cliente."

Me.NRegCTxBx.Focus()

ElseIf Me.NombreCTxBx.Text.Trim.Equals("") Then

STATUS.Text = "ERROR: Ingrese el Nombre del cliente."

Me.NombreCTxBx.Focus()

ElseIf Not isRut(Me.RutCTxBx.Text, Me.DVCTxBx.Text) Then

STATUS.Text = "ERROR: Ingrese un RUT válido."

Me.RutCTxBx.Focus()

ElseIf Me.TelCTxBx.Text.Trim.Equals("") Then

STATUS.Text = "ERROR: Ingrese el Teléfono del cliente."

Me.TelCTxBx.Focus()

ElseIf Me.EmailCTxBx.Text.Trim.Equals("") Then

STATUS.Text = "ERROR: Ingrese el Email del cliente."

Me.EmailCTxBx.Focus()

ElseIf ((Date.Now - Me.FNacDTiPckr.Value).TotalDays / 365) < 17 Then

STATUS.Text = "ERROR: El cliente no puede ser menor de 17 años."

Me.FNacDTiPckr.Focus()

ElseIf Me.CuposTxtLbl.Equals("0") Then

STATUS.Text = "ERROR: No hay cupos."

Me.CursoCbBx.Focus()

Else

Return True

End If

MsgBox("", MsgBoxStyle.Critical, AcceptButton)

STATUS.ForeColor = Color.Red

Return False

End Function

#End Region

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

'Se reinician todos los controles a su estado inicial

'primero se vacían los campos de textos

Me.NRegCTxBx.Text = ""

Me.NombreCTxBx.Text = ""

Me.RutCTxBx.Text = ""

Me.DVCTxBx.Text = ""

Me.TelCTxBx.Text = ""

Me.EmailCTxBx.Text = ""

Me.CuposTxtLbl.Text = ""

Me.ValorTxtLbl.Text = ""

Me.APagarTxtLbl.Text = ""

'Luego se reestablece el campo de fecha a la actual

Me.FNacDTiPckr.Value = Date.Today

'Después se deseleccionan los checkbox

Me.LuSaRaBtn.Checked = True

Me.FotosChBx.Checked = False

Me.FotocopiaCIChBx.Checked = False

Me.FotocopiaLicenciaChBx.Checked = False

Me.CertEstudiosChBx.Checked = False

Me.CertNotarialChBx.Checked = False

'Se vacían los combobox secundarios

'Me.CursoCbBx.DataSource = DBNull.Value

'Me.HraPracticaCbBx.DataSource = DBNull.Value

'Me.HraTeoriaCbBx.DataSource = DBNull.Value

'Luego se reestablece el valor del spinBox y algunos comboBox

Me.DctoSpinBx.Value = 0

Me.DocPagoCbBx.SelectedIndex = 0

Me.MedioPagoChBx.SelectedIndex = 0

'Después la desabilitación

Me.CursoCbBx.Enabled = False

Me.HraTeoriaCbBx.Enabled = False

Me.HraPracticaCbBx.Enabled = False

Me.DctoSpinBx.Enabled = False

Me.MedioPagoChBx.Enabled = False

Me.PagoRealizChBx.Enabled = False

Me.CodPagoTxBx.Enabled = False

'Terminando, Se establecen los valores predeterminados necesarios

Me.ClienteCbBx.SelectedIndex = 0

Me.ProductoCbBx.SelectedIndex = 0

Me.NRegCTxBx.Focus()

End Sub

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

'Format(Date.Today.ToString, "HH:mm:ss") & "','" & \_

'con.doQuery(

'comprobar variables nombre, rut, telefono

If validar() Then

Dim Query As String = "call MATRICULAR('" & Me.NRegCTxBx.Text & "', '" & \_

Me.NombreCTxBx.Text & "', '" & \_

Me.RutCTxBx.Text & "-" & Me.DVCTxBx.Text & "','" & \_

Format(Me.FNacDTiPckr.Value, "yyyy-MM-dd") & "','" & \_

Me.TelCTxBx.Text & "','" & \_

Me.EmailCTxBx.Text & "','" & \_

Me.ClienteCbBx.Text & "','" & \_

Me.ProductoCbBx.Text & "','" & \_

Format(Date.Today, "yyyy-MM-dd") & "','" & \_

Me.USER & "','" & \_

Me.CursoCbBx.Text & "','" & \_

Me.HraTeoriaCbBx.Text & "','" & \_

Me.HraPracticaCbBx.Text & "'," & \_

Me.LuViRaBtn.Enabled.ToString.ToLower() & "," & \_

Me.FotosChBx.Checked.ToString.ToLower() & "," & \_

Me.CertEstudiosChBx.Checked.ToString.ToLower() & "," & \_

Me.CertNotarialChBx.Checked.ToString.ToLower() & "," & \_

Me.FotocopiaCIChBx.Checked.ToString.ToLower() & ", '" & \_

Me.DctoSpinBx.Value.ToString & "','" & \_

Me.DocPagoCbBx.Text & "','" & \_

Me.MedioPagoChBx.Text & "'," & \_

Me.PagoRealizChBx.Checked.ToString.ToLower() & ",'" & \_

Me.CodPagoTxBx.Text & "');" ')

'Me.LimpiarBtn.PerformClick()

Me.QueryTb.Text = Query

con.doQuery(Query)

Dim res As String = con.selectWhereQuery("1", "MATRICULA", "Codigo = '" & Me.NRegCTxBx.Text & "'")

If res.Trim.Equals("") Then

STATUS.Text = "ERROR: Hubo un error al agregar la matrícula, por favor intente más tarde."

STATUS.ForeColor = Color.Red

Me.NRegCTxBx.Focus()

Else

STATUS.Text = "Alumno ingresado exitosamente."

STATUS.ForeColor = Color.Blue

Me.LimpiarBtn.PerformClick()

Me.NRegCTxBx.Focus()

End If

End If

End Sub

Private Sub ProductoCbBx\_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles ProductoCbBx.SelectedIndexChanged

loadCBOX("CursosXProductos")

Me.CursoCbBx.Enabled = True

End Sub

Private Sub CursoCbBx\_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles CursoCbBx.SelectedIndexChanged

'Revisa la cantidad de cupos en el curso

If dc.Cupos(Me.CursoCbBx.SelectedValue.ToString).Rows.Count > 0 Then

Me.CuposTxtLbl.Text = CInt(dc.Cupos(Me.CursoCbBx.SelectedValue.ToString).Rows(0).Item(0).ToString())

Else

Me.CuposTxtLbl.Text = 0

End If

loadCBOX("HTeoriaXCurso")

loadCBOX("HPracticaXCurso")

If dc.MontoAPagar(Me.ClienteCbBx.SelectedValue.ToString, ProductoCbBx.SelectedValue.ToString).Rows.Count > 0 Then

MontoAPagar = CInt(dc.MontoAPagar(Me.ClienteCbBx.SelectedValue.ToString, ProductoCbBx.SelectedValue.ToString).Rows(0).Item(0).ToString())

Else

MontoAPagar = 0

End If

Me.ValorTxtLbl.Text = "$" & MontoAPagar

Me.APagarTxtLbl.Text = "$" & MontoAPagar

Me.DctoSpinBx.Enabled = True

Me.DocPagoCbBx.Enabled = True

Me.MedioPagoChBx.Enabled = True

Me.PagoRealizChBx.Enabled = True

End Sub

Private Sub PagoRealizChBx\_CheckedChanged(sender As System.Object, e As System.EventArgs) Handles PagoRealizChBx.CheckedChanged

Me.CodPagoTxBx.Enabled = Me.PagoRealizChBx.Checked

End Sub

Private Sub DctoSpinBx\_ValueChanged(sender As System.Object, e As System.EventArgs) Handles DctoSpinBx.ValueChanged

'MsgBox(Me.ValorTxtLbl.Text.Substring(1))

If Me.ValorTxtLbl.Text.Length <> 0 Then

Try

Dim MtoO As Integer = Convert.ToInt32(Me.ValorTxtLbl.Text.Substring(1))

Dim Dcto As Integer = Me.DctoSpinBx.Value

Dim MtoAPa As Integer = MtoO - ((MtoO \* Dcto) / 100)

Me.APagarTxtLbl.Text = MtoAPa

Catch ex As Exception

MsgBox(Me.ValorTxtLbl.Text.Substring(1))

End Try

End If

End Sub

Private Sub ClienteCbBx\_SelectedIndexChanged(sender As System.Object, e As System.EventArgs) Handles ClienteCbBx.SelectedIndexChanged

Try

If dc.MontoAPagar(Me.ClienteCbBx.SelectedValue.ToString, ProductoCbBx.SelectedValue.ToString).Rows.Count > 0 Then

MontoAPagar = CInt(dc.MontoAPagar(Me.ClienteCbBx.SelectedValue.ToString, ProductoCbBx.SelectedValue.ToString).Rows(0).Item(0).ToString())

Else

MontoAPagar = 0

End If

Me.ValorTxtLbl.Text = "$" & MontoAPagar

Me.APagarTxtLbl.Text = "$" & MontoAPagar

Catch ex As Exception

End Try

End Sub

End Class

DataCBOX.vb

Public Class DataCBOX

Dim con As New Conexion

Sub New(ByVal conexion As Conexion)

con = conexion

End Sub

Function Profesores() As DataTable

Dim Data As DataTable = con.doQuery("SELECT f.idFuncionario, f.nombre " \_

& "FROM funcionario AS f, docente AS d " \_

& "WHERE d.tipo = 'P' AND f.idFuncionario = d.idDocente")

Return Data

End Function

Function Instructores() As DataTable

Dim Data As DataTable = con.doQuery("SELECT f.idFuncionario, f.nombre " \_

& "FROM funcionario AS f, docente AS d " \_

& "WHERE d.tipo = 'I' AND f.idFuncionario = d.idDocente")

Return Data

End Function

Function Funcionarios() As DataTable

Dim Data As DataTable = con.doQuery("SELECT \* FROM Funcionario")

Return Data

End Function

Function Area() As DataTable

Dim Data As DataTable = con.doQuery("SELECT Nombre FROM Area")

Return Data

End Function

Function ProductoDeArea(ByVal Area As String)

Dim Data As DataTable = con.doQuery("SELECT Nombre FROM Producto " \_

& "WHERE Area = '" & Area & "'")

Return Data

End Function

Function Estudiantes() As DataTable

Dim Data As DataTable = con.doQuery("SELECT \* FROM Estudiante")

Return Data

End Function

Function CursosDeProducto(ByVal Producto As String) As DataTable

Dim Data As DataTable = con.doQuery("SELECT Codigo FROM Curso " \_

& "WHERE Producto = '" & Producto & "' AND" \_

& " FechaTermino >= DATE(NOW())")

Return Data

End Function

Function Productos() As DataTable

Dim Data As DataTable = con.doQuery("SELECT \* FROM PRODUCTO")

Return Data

End Function

Function ClaseTeorica() As DataTable

Dim Data As DataTable = con.doQuery("SELECT c.idClase, c.Curso FROM Clase c, Teoria t WHERE t.clase=c.idClase")

Return Data

End Function

Function ClasePractica() As DataTable

Dim Data As DataTable = con.doQuery("SELECT c.idClase, c.Curso FROM Clase c, Practica p WHERE p.clase=c.idClase")

Return Data

End Function

Function Horarios(ByVal Curso As String) As DataTable

Dim Data As DataTable = con.doQuery("SELECT Horario FROM Clase WHERE Curso = '" & Curso & "'")

Return Data

End Function

Function Vehiculos() As DataTable

Dim Data As DataTable = con.doQuery("SELECT \* FROM auto\_escuela WHERE Matricula<>'000000'")

Return Data

End Function

Function TipoUsuario() As DataTable

Dim Data As DataTable = con.doQuery("SELECT \* FROM tipo\_usuario")

Return Data

End Function

Function HTeoriaDeCurso(ByVal Curso As String) As DataTable

Dim Data As DataTable = con.doQuery("SELECT DISTINCT(horario) As 'Horario' FROM Clase " \_

& "WHERE Curso = '" & Curso & "' AND Tipo = 'T'")

Return Data

End Function

Function HPracticaDeCurso(ByVal Curso As String) As DataTable

Dim Data As DataTable = con.doQuery("SELECT DISTINCT(horario) As 'Horario' FROM Clase " \_

& "WHERE Curso = '" & Curso & "' AND Tipo = 'P'")

Return Data

End Function

Function DocPago() As DataTable

Dim Data As DataTable = con.doQuery("SELECT 'Boleta' as Value union select 'Factura' as Value ")

Return Data

End Function

Function MedioPago() As DataTable

Dim Data As DataTable = con.doQuery("SELECT \* FROM MEDIO\_PAGO")

Return Data

End Function

Function Cupos(ByVal Curso As String) As DataTable

Dim Data As DataTable = con.doQuery("SELECT Cupos FROM Curso " \_

& "WHERE Codigo = '" & Curso & "'")

Return Data

End Function

Function MontoAPagar(ByVal Cliente As String, ByVal Producto As String) As DataTable

Dim Data As DataTable = con.doQuery("SELECT Monto FROM Precio " \_

& "WHERE Producto = '" & Producto & "' AND Tipo\_Cliente = '" & Cliente & "'")

Return Data

End Function

Function Clientes() As DataTable

Dim Data As DataTable = con.doQuery("SELECT \* FROM TIPO\_CLIENTE")

Return Data

End Function

End Class

Herramientas.vb

Module Herramientas

Public ReadOnly NoEspeciales As String = "abcdefghijklmnñopqrstuvwxyzABCDEFGHIJKLMNÑOPQRSTUVWXYZ0123456789"

Function isRut(ByVal rut As String, ByVal digito As String)

'Verifica el Rut

Dim largo As Integer = Len(rut)

Dim suma As Integer = 0

Dim actual As Integer

Dim resto As Integer

For i As Integer = 1 To largo

actual = CInt(Left(Right(rut, i), 1))

If i <= 6 Then

actual = actual \* (i + 1)

suma = suma + actual

Else

actual = actual \* (i - 5)

suma = suma + actual

End If

Next

resto = 11 - (suma Mod 11)

If resto = 10 Then

If digito.Equals("K") Or digito.Equals("k") Then Return True

ElseIf resto = 11 Then

If digito.Equals("0") Then Return True

ElseIf digito.Equals(resto.ToString) Then

Return True

End If

Return False

End Function

Sub soloNumeros(ByVal e As System.Windows.Forms.KeyPressEventArgs)

'Valida la entrada de solo números

If Char.IsDigit(e.KeyChar) Then

e.Handled = False

ElseIf Char.IsControl(e.KeyChar) Then

e.Handled = False

Else

e.Handled = True

End If

End Sub

Sub soloTexto(ByVal e As System.Windows.Forms.KeyPressEventArgs)

'Valida la entrada de solo texto

If Not Char.IsDigit(e.KeyChar) Then

e.Handled = False

ElseIf Char.IsControl(e.KeyChar) Then

e.Handled = False

Else

e.Handled = True

End If

End Sub

Sub textoDeseado(ByVal e As System.Windows.Forms.KeyPressEventArgs, ByVal cadena As String)

'Valida la entrada de caracteres deseados

If InStr(cadena, e.KeyChar) Then

e.Handled = False

ElseIf Char.IsControl(e.KeyChar) Then

e.Handled = False

Else

e.Handled = True

End If

End Sub

Function edadPorNacimiento(ByVal fecha As String) As String

'Devuelve la edad a partir de la fecha de nacimiento

'Formato fecha = yyyy-MM-dd

Dim edad As Integer = 0

Dim dia\_n As Integer = Right(fecha, 2)

Dim mes\_n As Integer = Left(Right(fecha, 5), 2)

Dim ano\_n As Integer = Left(fecha, 4)

Dim fechaActual As String = Format(Date.Now, "yyyy-MM-dd")

Dim dia\_a As Integer = Right(fechaActual, 2)

Dim mes\_a As Integer = Left(Right(fechaActual, 5), 2)

Dim ano\_a As Integer = Left(fechaActual, 4)

edad = ano\_a - ano\_n

If mes\_a < mes\_n Then edad = edad - 1

If mes\_a = mes\_n Then

If dia\_a < dia\_n Then edad = edad - 1

End If

Return edad

End Function

Function filtroContiene(ByVal items() As String, ByVal compara As String) As String()

Return items.Where(Function(s) s Like "\*" & compara & "\*").ToArray

End Function

Function HoraMaxima(ByVal H As Integer) As Integer

If H > 23 Then

Return 23

End If

Return H

End Function

Function MinutoMaximo(ByVal M As Integer) As Integer

If M > 59 Then

Return 59

End If

Return M

End Function

End Module