

**ACTIVIDAD INDIVIDUAL
UNIDAD 3: PASO 3 DISEÑO BASE DE DATOS**

**JULIE MARCELA GUTIERREZ PACHECO
Código. 1030546949
LEONARDO PUERTO
Código. 80251924
DAVID FELIPE HERRERA MARTÍNES
Código. 1000689402
OSCAR RICARDO RODRIGUEZ MARTINEZ
Código. 79671858**

VISUAL BASIC BASICO

GRUPO 10

PRESENTADO A

JORGE ENRIQUE PORTELA

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA

INGENIERÍA DE SISTEMAS

2019

INTRODUCTION

The purpose of this work is to show the analysis, design, and code used to create the application needed in step 3 of the Basic Visual Basic subject. This application seeks the registration, modification, and deletion of users, developed using the Visual Basic programming language together with a database created on SQL Server. In the first instance, an analysis of the need will be made, determining what elements are necessary to fulfill the objectives, later in the design, the disposition of the resources will be presented, finally we will be describing the methods used to execute the various operations.

OBJETIVOS

Objetivo General

- Develop an application for the creation, editing, and elimination of students under the Visual Basic programming language.

Objetivos Específicos

- Design a graphical interface that allows the user to use the application.
- Create the necessary program to create users.
- Design the student information database.
- Establish the operating protocols of the program to search and modify users. • Build the code necessary for user deletion operation.

CONTENIDO

Para el presente trabajo escogimos realizar una base de datos en SQL express los datos de conexión a esta base de datos son:

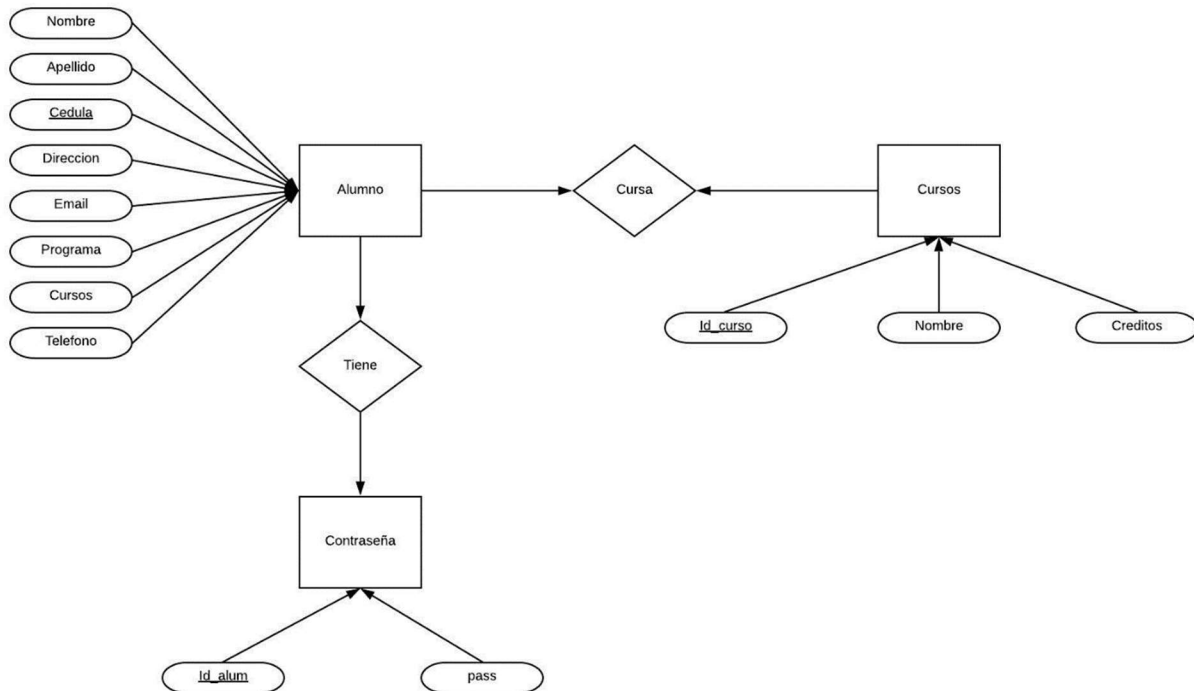
Servidor: servidor Local (localhost)
Usuario: sql_user
Password : Cajon76
Base de datos: Alumnos
Archivos físicos: Alumnos.mdf y Alumnos_log.ldf

Estos parámetros pueden ser modificados de acuerdo con las necesidades puntuales, los cambios se deben realizar en el procedimiento denominado conectar_sql() el cual se encuentra en el módulo Base_datos.vb

```
Public Sub conectar_sql()  
    Try  
        'creo el objeto conexión con sql y establezco parametros de conexión  
        conexion = New SqlConnection("Initial Catalog=Alumnos;" & "Data Source=localhost;Integrated Security=SSPI;" & "User Id=sql_user;Password=cajon76")  
        conexion.Open()  
    Catch exc As Exception  
        MsgBox(exc)  
    End Try  
End Sub  
  
Public Sub eliminar_registro(ByVal id As String) ...
```

La base de datos presenta la siguiente estructura:

Modelo Entidad Relación



Descripción de Entidades

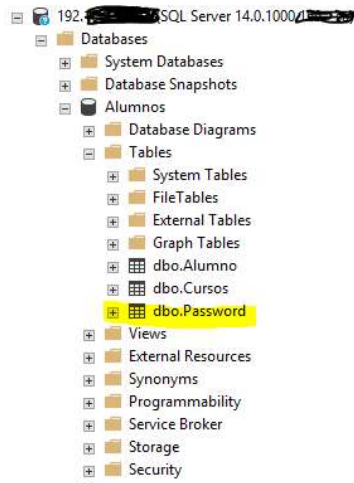
Entidad	Justificación, explicación de su existencia en el Mundo del Problema	Ejemplo de Ejemplares (Instancias)	Extensión	Intención
Alumno	Datos personales sobre los estudiantes	Oscar Ricardo Fernando Aragon	cedula (79671858)	Almacenar los datos personales de cada estudiante
Contraseña	Almacenar las contraseñas de acceso en una tabla diferente a la de datos personales		pass (Alf4*)	Mantener las contraseñas separadas de la información general para aumentar la seguridad de la base de datos
Cursos	Información sobre los cursos ofrecidos	MySQL SCRUM Base de datos	Código del curso	Individualizar los cursos ofrecidos en el presente aplicativo

Listado de atributos y sus restricciones naturales (Tipo de Dato) y restricciones adicionales, Dominios (Lista de Valores, Rangos Restricciones generales)

Nombre Entidad	Atributos	Identificador Único (UID)	Nombre dominio	Tipo de Dato (Texto, Número, Fecha)	Tamaño	Obligatoriedad (Si/No)	Cardinalidad	Restricciones adicionales		
							(1 o Muchos)	Lista de Valores	Rango de Valores	Restricción general
Alumno	Nombre		Alfabeto	T	50	si	1	N/A	N/A	N/A
	Apellido		Alfabeto	T	50	si		N/A	N/A	N/A
	Cedula	X	Números Positivos	N	15	si		N/A	N/A	>0
	Direccion		Alfanumérico	T	50	si		N/A	N/A	N/A
	Email		Alfanumérico	T	50	si		N/A	N/A	N/A
	Programa		Alfabeto	T	50	si		N/A	N/A	N/A
	Cursos		Números Positivos	N	15	si		Valores definidos en tabla cursos	N/A	>0
	Telefono		Números Positivos	N	15	si		N/A	N/A	N/A
Cursos	Id_Curso	X	Números Positivos	N	12	si	1	N/A	N/A	> 0
	Nombre		Alfabeto	T	10	si			N/A	N/A
	Código			T	30	si		N/A	N/A	N/A
Password	Id_Alum	X	Números Positivos	N	12	SI	M	N/A	N/A	N/A
	pass		Alfanumérico	T	12	SI		N/A	N/A	N/A

Modelo Relacional

La aplicación inicia su funcionamiento con un form que solicita un usuario y una contraseña de ingreso una vez digitados los datos estos son verificados con los datos almacenados en la BD específicamente en la tabla dbo.password,

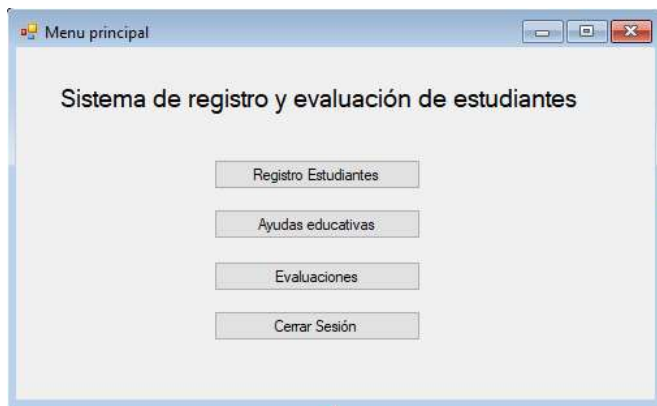


La validación de contraseña se realiza mediante el llamado a un procedimiento denominado validar_psw() el cual realiza un query a la base de datos y valida que la contraseña almacenada sea igual a la digitada por el usuario si esto es así hace el llamado al form_menu que contiene un breve menú de opciones para navegar por la aplicación

```
'funcion para validar el usuario y contraseña de ingreso
Public Sub validar_psw(ByVal id_al As String, pswd As String)
    Dim resultado1 As SqlDataReader
    resultados = ""

    comandos = conexion.CreateCommand
    comandos.CommandText = "SELECT pass FROM Password where id_alumn='" & id_al & "'"
    resultado1 = comandos.ExecuteReader()
    Do While resultado1.Read()
        resultados = resultado1.GetString(0)
    Loop
    If resultados = pswd Then

        Form1.Hide()
        Form_menu.Show()
    Else
        MsgBox("usuario o contraseña erronea")
    End If
    resultado1.Close()
End Sub
```



El formulario menú principal por el momento solo tiene habilitada la opción de registro de estudiantes, este botón nos lleva al formulario utilizado para realizar las diferentes tareas sobre la base de datos

Este formulario consta de 5 botones cada uno de los cuales realiza las siguientes tareas:

Botón Agregar: Este botón llama el procedimiento agregar registro, este procedimiento recolecta los datos que se encuentran en el formulario y los organiza en un string, paso seguido ejecuta la sentencia INSERT INTO para agregar el registro a la tabla Alumno


```

Public Sub agregar_registro()
    Dim valores As String = ""
    Dim codigo_curso = 0
    'asigno un los codigos relacionados al curso
    Select Case Form_registro.ComboBox1.SelectedIndex
        Case 0
            codigo_curso = 9081
        Case 1
            codigo_curso = 9082
        Case 2
            codigo_curso = 9083
        Case 3
            codigo_curso = 9084
        Case 4
            codigo_curso = 9085
        Case Else
            MsgBox("error al asignar codigo al curso seleccionado")
    End Select
    'armo el string de valores a utilizar con la sentencia de insercion
    valores = "" & Form_registro.TextBox1.Text & "," & Form_registro.TextBox2.Text & "," & Form_registro.TextBox3.Text & "," & Form_registro.TextBox4.Text & "," & For
Try
    ' ejecuto comando insert sobre la base de datos
    comandos = conexion.CreateCommand
    comandos.CommandText = "INSERT INTO Alumno VALUES (" & valores & ")"
    resultado = comandos.ExecuteReader()
    resultado.Close()
    MsgBox("Registro añadido correctamente")
Catch ex As Exception
    MsgBox(ex)
End Try
limpiar_form_registro()
End Sub

Public Sub buscar_sql(ByVal id As String) ...

```

Botón Buscar: Realiza un llamado a un formulario temporal denominado form_búsqueda, el cual es utilizado para solicitar el número de identificación de un estudiante, realizar una búsqueda de este en la base de datos y publicar los datos encontrados en el formulario de registro

The image shows a Windows application window titled "Busqueda". The window's title bar includes standard Windows controls (minimize, maximize, close). The main content area is titled "Busqueda de Alumnos". It features a label "Codigo" next to a text input field. Below the input field, there are two buttons: "Buscar" and "Cancelar".

```

Public Sub buscar_sql(ByVal id As String)
    comandos = conexion.CreateCommand
    comandos.CommandText = "SELECT * FROM Alumno where Cedula='" & id & "'"
    resultado = comandos.ExecuteReader()
    Do While resultado.Read()
        Form_registro.TextBox1.Text = resultado.GetString(0)
        Form_registro.TextBox2.Text = resultado.GetString(1)
        Form_registro.TextBox3.Text = resultado.GetString(2)
        Form_registro.TextBox4.Text = resultado.GetString(4)
        Form_registro.TextBox5.Text = resultado.GetString(7)
        Form_registro.TextBox6.Text = resultado.GetString(3)
        Form_registro.TextBox7.Text = resultado.GetString(5)

    Loop
    resultado.Close()
End Sub

```

Botón Modificar: Hace un llamado al procedimiento modificar_sql() el cual revisa que este escrito por lo menos el código de un estudiante en el formulario y modifica el registro asociado a este código en la base de datos con los valores escritos en el formulario mediante la sentencia UPDATE de sql.

En caso de que el código no este escrito el procedimiento llama al formulario de búsqueda para buscar los datos de un usuario existente, de igual forma en caso de que algún campo este en blanco el procedimiento emitirá una alerta de esta situación esperando confirmación de actualización de datos vacíos por parte del usuario

```

Public Sub modificar_sql()
    Dim actualizar As String = ""
    Dim confirmar = vbYes
    Dim codigo_curso = 0

    Select Case Form_registro.ComboBox1.SelectedIndex
        Case 0
            codigo_curso = 9001
        Case 1
            codigo_curso = 9002
        Case 2
            codigo_curso = 9003
        Case 3
            codigo_curso = 9004
        Case 4
            codigo_curso = 9005
        Case Else
            MsgBox("error al asignar codigo al curso seleccionado")
    End Select

    Try
        If (Form_registro.TextBox3.Text IsNot "") Then
            If (vacio_registro()) Then
                confirmar = MsgBox("Existen campos en blanco esta seguro de continuar ", vbYesNo)
            End If
        Else
            MsgBox("por favor digite la cedula del alumno a modificar y luego haga click en buscar ")
            Form_búsqueda.Show()
            Return
        End If

        If confirmar = vbYes Then
            actualizar = "Nombre = '" & Form_registro.TextBox1.Text & "',Apellido = '" & Form_registro.TextBox2.Text & "',Cedula = '" & Form_registro.TextBox3.Text & "',Em"
            comandos = conexion.CreateCommand
            comandos.CommandText = "UPDATE Alumno SET " & actualizar & "Where Cedula = '" & Form_registro.TextBox3.Text & "'"
            resultado = comandos.ExecuteReader()
            resultado.Close()
            MsgBox("Registro actualizado correctamente ")
        End If
    Catch ex As Exception
        MsgBox(ex)
    End Try
End Sub

```

Botón Eliminar: Hace un llamado al procedimiento eliminar_registro, este procedimiento recibe un parámetro que corresponde al código de un alumno y ejecuta la sentencia sql DELETE FROM, antes de realizar la eliminación solicita confirmación por parte del usuario

```
Public Sub eliminar_registro(ByVal id As String)
    Dim confirmacion
    confirmacion = MsgBox("Esta operación eliminara el registro y no se podra deshacer, esta seguro?", vbYesNo)
    If confirmacion = DialogResult.Yes Then

        comandos = conexion.CreateCommand
        comandos.CommandText = "DELETE FROM Password where id_alumn='" & id & "'"
        resultado = comandos.ExecuteReader()
        resultado.Close()
        comandos.CommandText = "DELETE FROM Alumno where Cedula='" & id & "'"
        resultado = comandos.ExecuteReader()
        resultado.Close()
        MsgBox("Registro eliminado correctamemnte")

    End If
    limpiar_form_registro()
End Sub
```

CONCLUSIONS

Through the use of Windows Forms under Visual Studio, a simpler methodology of object oriented programming is established, since in this one through a drag and drop we can build an interface and then with the use of properties and actions of the different tools generate the series of operations that we like. On the other hand, the use of Visual Basic allows programming without very complex sentences, with a basic logic, which allows the use of this program and programming language in other developments.

BIBLIOGRAFIA

- GODINEZ. G. Escribir solo números en Textbox de Visual Basic (Visual Studio 2015). Disponible en: <https://www.youtube.com/watch?v=1ZZKJpp2Ciw>
- PROGRAMACIÓN TOTAL. Insertar y consultar datos de MySQL desde VB.NET. Disponible en: <https://www.youtube.com/watch?v=wURM8tgEUcs>
- Fernández, C. (2009). Visual Basic: básico. Madrid, ES: RA-MA Editorial.