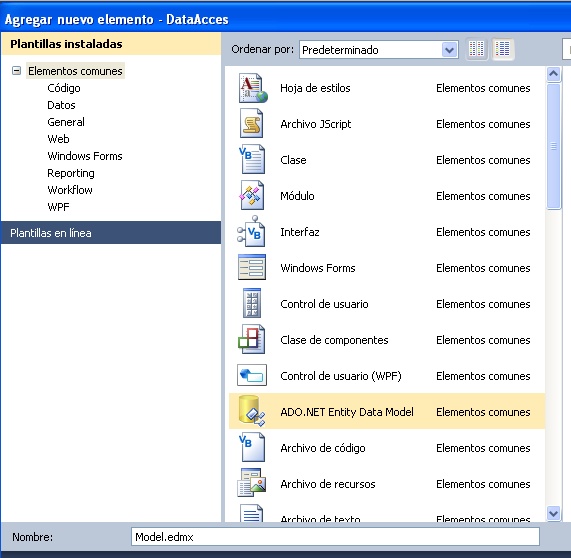
[Mantenimiento a BD Usando Entity Framework y Linq](http://cristiantorresalfaro.blogspot.com/2012/06/mantenimiento-bd-usando-entity.html)

lunes, 25 de junio de 2012

Siempre que desarrollamos aplicaciones que usan **bases** **de datos** nos vemos en la necesidad de **agregar**, **editar**, **eliminar** o **buscar registros** pues bien ahora veremos como hacer todas estas operaciones usando **entity framework**y**linq**.  
  
**Para este ejemplo desarrolle una aplicación en 2 capas y utilizo una base de datos llamada Database que tiene una tabla llamada Personas.**  
  
**El ejemplo esta desarrollado con visual estudio 2010 en c# y vb.net**



Lo primero que aremos es crear el**modelo conceptual** de nuestra base de datos para eso agregamos un nuevo elemento y elegimos **entity data model** y seguimos los pasos necesarios para que el visual estudio nos cree nuestra clase y propiedades de nuestra tabla personas.





Ahora que ya tenemos nuestro **modelo** entity de nuestra **tabla** creado empezaremos a programar todas las operaciones agregamos una clase llamada **Datos.cs**o**Datos.vb**.  
  
Para cada operación crearemos un **métod**o comenzaremos con el **método** que recupera todos los **registros** de nuestra tabla.

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | public static List<Personas> CargarDatos()  {      using (DatabaseEntities bd = new DatabaseEntities())      {         var datos = (from p in bd.Personas select p).ToList();            return datos;      }  } |

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | Public Shared Function CargarDatos() As List(Of Personas)        Using bd As New DatabaseEntities()            Dim datos = (From p In bd.Personas Select p).ToList()              Return datos        End Using    End Function |

Lo que hacemos es crear un **método** que **retornara** una **lista** de tipo **personas** y dentro del método creamos una instancia de nuestro **modelo** y con **linq** hacemos la consulta para que nos muestre todos los registros que hay en la tabla personas es como hacerle un **select** a toda la tabla.  
  
Ahora creamos el **método** que nos servirá para **buscar registros** por medio del nombre.

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | public static List<Personas> Buscar(string nombre)  {      using (DatabaseEntities bd = new DatabaseEntities())      {          var buscar = (from p in bd.Personas                        where p.Nombre.StartsWith(nombre)                        select p).ToList();            return buscar;      }  } |

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | Public Shared Function Buscar(ByVal nombre As String) As List(Of Personas)        Using bd As New DatabaseEntities()            Dim busca = (From p In bd.Personas                         Where p.Nombre.StartsWith(nombre)                         Select p).ToList()              Return busca        End Using    End Function |

Como podemos ver es parecido al anterior lo único que cambia es la consulta **linq** como lo que queremos hacer es buscar registros por nombre aremos uso de**where** para filtrar los registros por medio del **nombre** también hacemos uso de **StartsWith** que es algo similar al **like** de sql para que por ejemplo si escribimos la letra c se nos mostraran todos los registros que empiecen con la letra c.  
  
Ahora creemos el **método** que nos servirá para **agregar registros**.

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | public static void Agregar(string nombre, string apellido, int edad,string profesion)        {            using (DatabaseEntities bd = new DatabaseEntities())            {                bd.Personas.AddObject(new Personas { Nombre = nombre, Apellido = apellido, Edad = edad, Profesion = profesion });                bd.SaveChanges();            }        } |

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | Public Shared Sub Agregar(ByVal nombre As String, ByVal apellido As String, ByVal edad As Integer, ByVal profesion As String)        Using bd As New DatabaseEntities()            bd.Personas.AddObject(New Personas() With {.Nombre = nombre, .Apellido = apellido, .Edad = edad, .Profesion = profesion})            bd.SaveChanges()        End Using    End Sub |

Lo que hacemos aquí es crear un **método** que recibirá los **parámetros** necesarios para agregar un registro, luego como en todos los métodos creamos una instancia de nuestro modelo y hacemos uso del método **AddObject** para agregar un un nuevo objeto(registro) de tipo persona y guardamos los datos en nuestra bd con el método **SaveChanges**.  
  
Ahora creamos el**método** para **editar** los registros.

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | public static void Editar(int id,string nombre, string apellido, int edad, string profesion)        {            using (DatabaseEntities bd = new DatabaseEntities())            {                var editar = (from p in bd.Personas                              where p.Id == id                              select p).Single();                  editar.Nombre = nombre;                editar.Apellido = apellido;                editar.Edad = edad;                editar.Profesion = profesion;                bd.SaveChanges();            }        } |

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | Public Shared Sub Editar(ByVal id As Integer, ByVal nombre As String, ByVal apellido As String, ByVal edad As Integer, ByVal profesion As String)        Using bd As New DatabaseEntities()            Dim editar = (From p In bd.Personas                          Where p.Id = id                          Select p).Single()              editar.Nombre = nombre            editar.Apellido = apellido            editar.Edad = edad            editar.Profesion = profesion            bd.SaveChanges()        End Using    End Sub |

Al igual que el anterior creamos un **método** que recibirá los **parámetros** necesarios para editar un registro y luego haciendo uso de **linq** hacemos una consulta para recuperar el registro a editar y le asignamos los nuevos valores.  
  
Ahora veremos el **método** que nos servirá para**eliminar** un registro.

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | public static void Eliminar(int id)        {            using (DatabaseEntities bd = new DatabaseEntities())            {                var eliminar = (from p in bd.Personas                                where p.Id==id                                select p).Single();                  bd.DeleteObject(eliminar);                bd.SaveChanges();            }        } |

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | Public Shared Sub Eliminar(ByVal id As Integer)        Using bd As New DatabaseEntities()            Dim eliminar = (From p In bd.Personas                            Where p.Id = id                            Select p).Single()              bd.DeleteObject(eliminar)            bd.SaveChanges()        End Using    End Sub |

Creamos un**método** que recibirá un **parámetro** que sera el**id** por el cual eliminaremos el registro y haciendo uso de**linq** recuperamos el registro por medio del id y luego eliminamos el registro con el método **DeleteObject**.  
  
Por ultimo crearemos un **método** que nos servirá para **cargar** en nuestro formulario de edición los datos del registro que queremos editar.

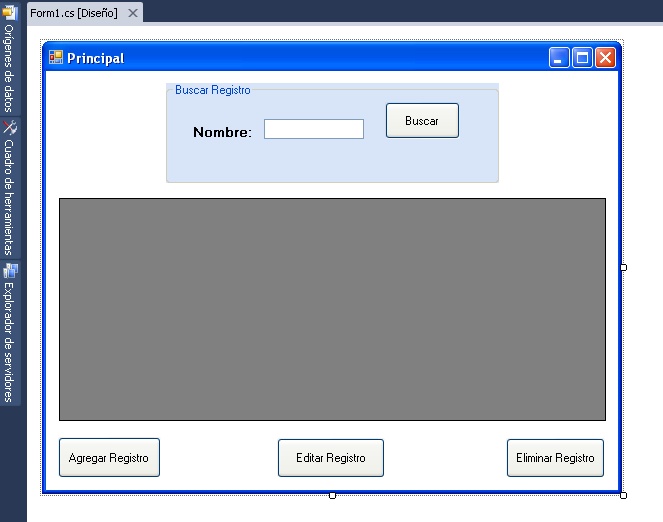
[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | public static Personas ObtenerId(int id)        {            Personas persona = new Personas();            using (DatabaseEntities bd = new DatabaseEntities())            {                var regis = (from p in bd.Personas                              where p.Id==id                              select p).Single();                  persona.Id = regis.Id;                persona.Nombre = regis.Nombre;                persona.Apellido = regis.Apellido;                persona.Edad = regis.Edad;                  return persona;            }        } |

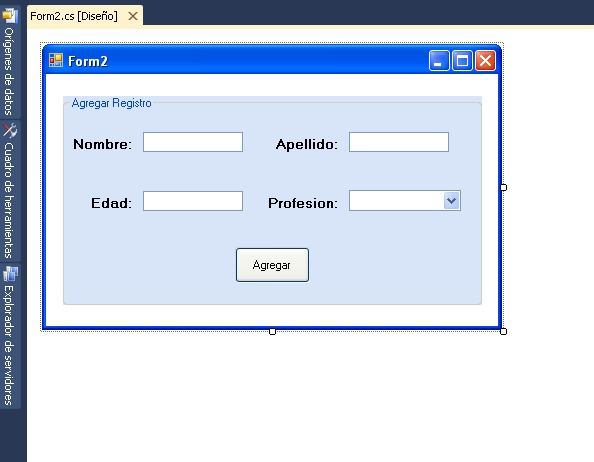
[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

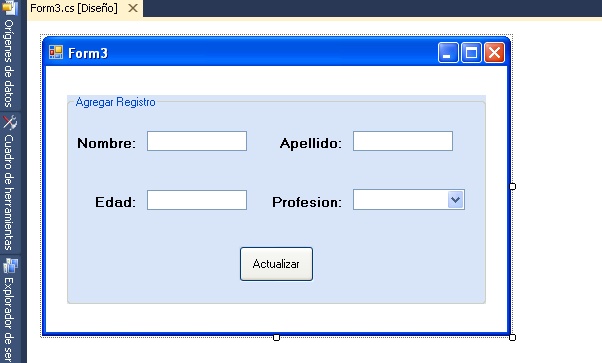
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | Public Shared Function ObtenerId(ByVal id As Integer) As Personas        Dim persona As New Personas()        Using bd As New DatabaseEntities()            Dim regis = (From p In bd.Personas                         Where p.Id = id                         Select p).Single()              persona.Id = regis.Id            persona.Nombre = regis.Nombre            persona.Apellido = regis.Apellido            persona.Edad = regis.Edad              Return persona        End Using    End Function |

Lo que hacemos es crear un **método** de tipo **personas** que recibirá el **id** como parámetro y luego haciendo uso de**linq** recuperamos el registro y le asignamos a cada propiedad de nuestra clase persona los datos obtenidos en la consulta y por ultimo retornamos los datos.  
  
**Eso seria todo en nuestra clase datos.**  
  
Ahora pasemos a diseñar nuestros formularios para este caso serán **3** el **principal** que sera donde se mostraran los datos en un **datagrid** ademas se podrá hacer la **búsqueda** de registros y tendrá **3 botones** que serán para agregar editar y eliminar un registro.



También creamos los formulario que nos servirán para **agregar** y **editar** los registros.





Ahora empezaremos a programar en nuestros formularios empecemos con el **form1** o formulario principal, comenzaremos con el código que nos servirá para **cargar** los datos y hacer la**búsqueda**.

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | public void cargar()        {            dataGridView1.DataSource = Datos.CargarDatos();        }          private void Form1\_Load(object sender, EventArgs e)        {            cargar();        }          private void btnbuscar\_Click(object sender, EventArgs e)        {            dataGridView1.DataSource = Datos.Buscar(txtnombre.Text);        } |

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | Public Sub cargar()        dataGridView1.DataSource = Datos.CargarDatos()    End Sub      Private Sub Form1\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load        cargar()    End Sub      Private Sub btnbuscar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnbuscar.Click        dataGridView1.DataSource = Datos.Buscar(txtnombre.Text)    End Sub |

El código es sencillo y fácil de entender así que no entrare en mayor explicación primero creamos un **método** donde se cargan los **datos** y luego en el**evento load** cargamos los datos y luego en el **evento** del**botón** buscar llamamos el **método** buscar y le enviamos el parámetro para obtener los datos.  
  
Ahora veamos como **agregar** un nuevo registro.

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | //cargar datos al cerrar form agregarregistro y actualizar        private void obj\_FormClosed(object sender, FormClosedEventArgs e)        {            cargar();        }          private void btnagregar\_Click(object sender, EventArgs e)        {            Form2 obj = new Form2();            //abrir el otro formulario y actualizar datagrid al cerrarlo            obj.FormClosed += new System.Windows.Forms.FormClosedEventHandler(obj\_FormClosed);            obj.Show();        } |

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | 'cargar datos al cerrar form agregarregistro y actualizar    Private Sub obj\_FormClosed(ByVal sender As Object, ByVal e As FormClosedEventArgs)        cargar()    End Sub      Private Sub btnagregar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnagregar.Click        Dim obj As New Form2()        'abrir el otro formulario y actualizar datagrid al cerrarlo        AddHandler obj.FormClosed, New System.Windows.Forms.FormClosedEventHandler(AddressOf obj\_FormClosed)        obj.Show()    End Sub |

Vemos que primero creamos un**método** que nos servirá para detectar el cierre tanto del formulario para agregar registro como el de editar esto lo hacemos para actualizar los datos siempre que hagamos cambios desde los 2 formularios, luego en el **evento** del **botón** llamamos al **form2** que es donde agregamos un nuevo registro.  
  
Ahora veamos como **editamos** y **eliminamos** un registro.

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23 | private void btneditar\_Click(object sender, EventArgs e)        {            DataGridViewRow row = dataGridView1.CurrentRow;            int id = Convert.ToInt32(row.Cells[0].Value);              Form3 obj = new Form3(id);            obj.FormClosed += new System.Windows.Forms.FormClosedEventHandler(obj\_FormClosed);            obj.Show();          }          private void btneliminar\_Click(object sender, EventArgs e)        {            DialogResult resul = MessageBox.Show("Seguro que quiere eliminar el Registro?", "Eliminar Registro", MessageBoxButtons.YesNo);            if (resul == DialogResult.Yes)            {                DataGridViewRow row = dataGridView1.CurrentRow;                int idpersona = Convert.ToInt32(row.Cells[0].Value);                Datos.Eliminar(idpersona);                MessageBox.Show("Se Elimino el Registro Con Numero ID: " + idpersona, "Eliminar Registro");                cargar();            }        } |

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19 | Private Sub btneditar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btneditar.Click        Dim row As DataGridViewRow = dataGridView1.CurrentRow        Dim id As Integer = Convert.ToInt32(row.Cells(0).Value)          Dim obj As New Form3(id)        AddHandler obj.FormClosed, New System.Windows.Forms.FormClosedEventHandler(AddressOf obj\_FormClosed)        obj.Show()    End Sub      Private Sub btneliminar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btneliminar.Click        Dim resul As DialogResult = MessageBox.Show("Seguro que quiere eliminar el Registro?", "Eliminar Registro", MessageBoxButtons.YesNo)        If resul = DialogResult.Yes Then            Dim row As DataGridViewRow = dataGridView1.CurrentRow            Dim idpersona As Integer = Convert.ToInt32(row.Cells(0).Value)            Datos.Eliminar(idpersona)            MessageBox.Show("Se Elimino el Registro Con Numero ID: " & idpersona, "Eliminar Registro")            cargar()        End If    End Sub |

En el **evento click** de **botón** editar igual que el anterior abrimos el **form3** que es el que nos servirá para editar los registros y actualizamos los datos al cerrarlo, algo importante es que al**form3** le enviamos el id del registro a editar.  
  
Y en el **evento** del **botón** eliminar lo único que hacemos es **eliminar** el registro seleccionado en el datagrid pasandole como parametro el id del registro seleccionado a nuestro metodo eliminar.  
  
**Ahora pasamos al código del form2 que es en el que agregamos los registros.**

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | private void btnaceptar\_Click(object sender, EventArgs e)  {      DataAcces.Datos.Agregar(txtnombre.Text,txtapellido.Text,Convert.ToInt32(txtedad.Text),cbbprofesion.SelectedItem.ToString());      MessageBox.Show("Registro Agregado...","Agregar Registro");      this.Close();  } |

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5 | Private Sub btnaceptar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnaceptar.Click        DataAcces.Datos.Agregar(txtnombre.Text, txtapellido.Text, Convert.ToInt32(txtedad.Text), cbbprofesion.SelectedItem.ToString())        MessageBox.Show("Registro Agregado...", "Agregar Registro")        Me.Close()    End Sub |

Aquí lo único que hacemos es llamar al metodo**agregar** y le enviamos los datos necesarios para agregar un registro.  
  
**Y por ultimo en nuestro form3 tendremos el siguiente codigo.**

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24 | private int id;        public Form3(int idempleado)        {            InitializeComponent();            id = idempleado;        }          private void Form3\_Load(object sender, EventArgs e)        {            // se carga el registro que se quiere editar             cbbprofesion.SelectedIndex = 0;             Personas persona = Datos.ObtenerId(id);             id = persona.Id;             txtnombre.Text = persona.Nombre;             txtapellido.Text = persona.Apellido;             txtedad.Text = persona.Edad.ToString();        }          private void btnaceptar\_Click(object sender, EventArgs e)        {            Datos.Editar(id,txtnombre.Text,txtapellido.Text,Convert.ToInt32(txtedad.Text),cbbprofesion.SelectedItem.ToString());            MessageBox.Show("Registro Actualizado...", "Actualizar Registro");            this.Close();        } |

[?](http://cristiantorresalfaro.blogspot.com/search/label/Entity)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | Private id As Integer    Public Sub New(ByVal idempleado As Integer)        InitializeComponent()        id = idempleado    End Sub      Private Sub Form3\_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load        ' se carga el registro que se quiere editar        cbbprofesion.SelectedIndex = 0        Dim persona As Personas = Datos.ObtenerId(id)        id = persona.Id        txtnombre.Text = persona.Nombre        txtapellido.Text = persona.Apellido        txtedad.Text = persona.Edad.ToString()    End Sub      Private Sub btnaceptar\_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnaceptar.Click        Datos.Editar(id, txtnombre.Text, txtapellido.Text, Convert.ToInt32(txtedad.Text), cbbprofesion.SelectedItem.ToString())        MessageBox.Show("Registro Actualizado...", "Actualizar Registro")        Me.Close()    End Sub |

Aqui lo que hacemos es primeramente **obtener** los datos del registro a **editar** en el **evento load**eso lo hacemos con el metodo **ObtenerId** y le enviamos como parametro el dato recibido del formulario principal.  
  
Y en el **evento** del **boton** llamamos a nuestro metodo Editar y le enviamos los parametros para editar el registro.  
  
**Bueno eso es todo por ahora, como podemos notar usar entity framework y linq nos facilita el trabajo para trabajar con bases de datos.**

