Universidad Autónoma De San Luis Potosí

Facultad de Ingeniería

Manual del Programador

Estructura de archivos

Jacob Alejandro Loredo De La Rosa

Área de Ciencias De La Computación

Semestre: Enero - Junio 2020

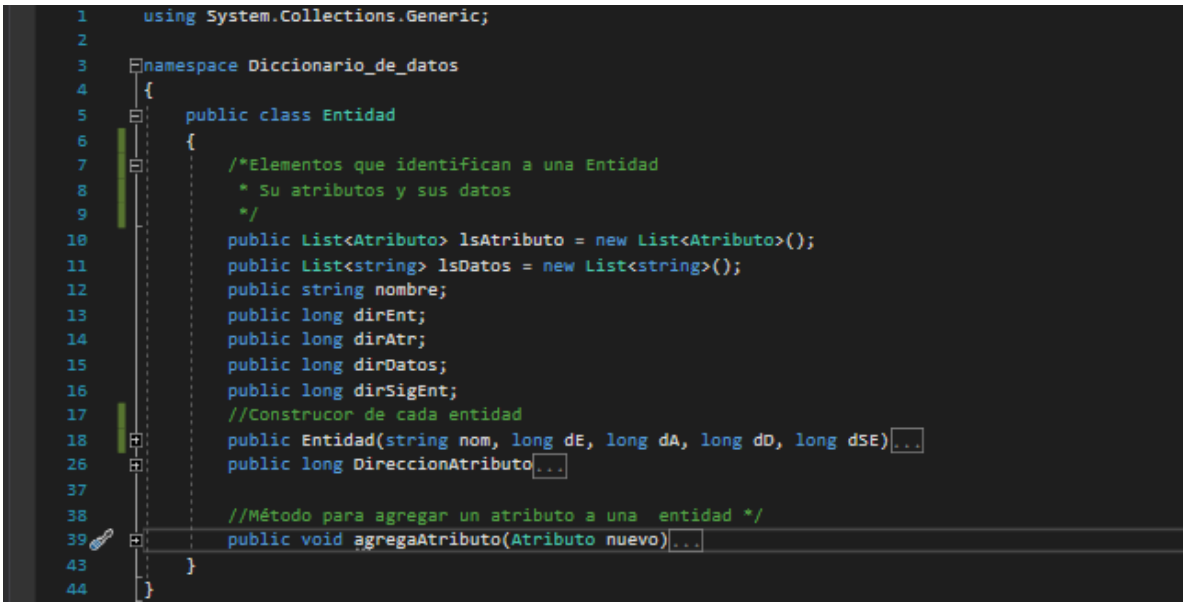
Profesor: ING. PACHECO ESTRADA MARIA CARMEN

Contenido

No se encontraron entradas de tabla de contenido.

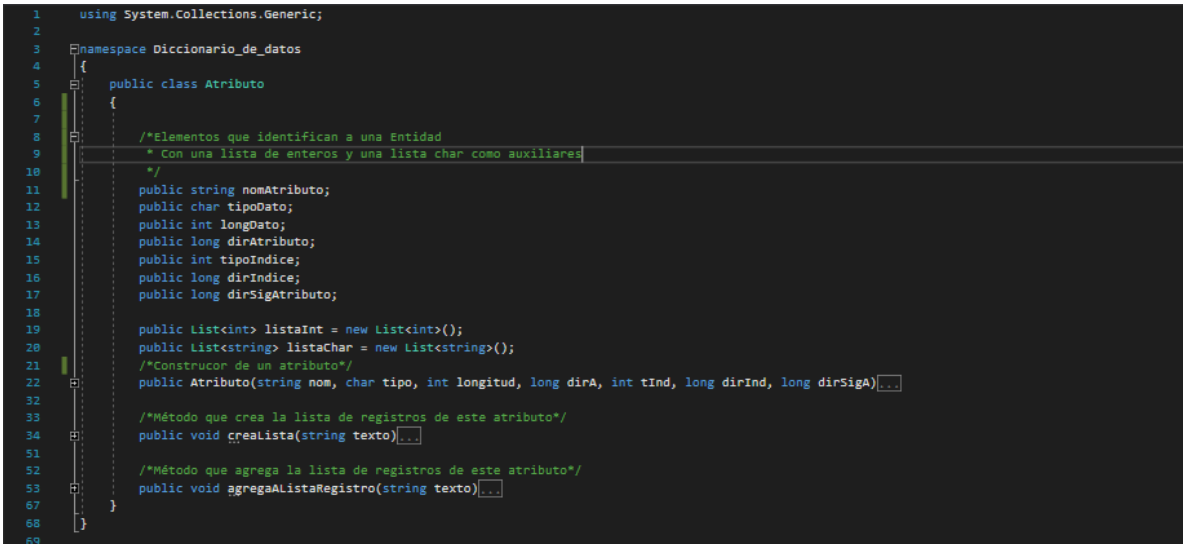
Clases

Entidad



```
using System.Collections.Generic;

namespace Diccionario_de_datos
{
    public class Entidad
    {
        /*Elementos que identifican a una Entidad
         * Su atributos y sus datos
         */
        public List<Atributo> lsAtributo = new List<Atributo>();
        public List<string> lsDatos = new List<string>();
        public string nombre;
        public long dirEnt;
        public long dirAtr;
        public long dirDatos;
        public long dirSigEnt;
        //Construcor de cada entidad
        public Entidad(string nom, long dE, long dA, long dD, long dSE)...
        public long DireccionAtributo...

        //Método para agregar un atributo a una  entidad */
        public void agregaAtributo(Atributo nuevo)...
    }
}
```

Atributo



```
using System.Collections.Generic;

namespace Diccionario_de_datos
{
    public class Atributo
    {

        /*Elementos que identifican a una Entidad
         * Con una lista de enteros y una lista char como auxiliares
         */
        public string nomAtributo;
        public char tipoDato;
        public int longDato;
        public long dirAtributo;
        public int tipoIndice;
        public long dirIndice;
        public long dirSigAtributo;

        public List<int> listaInt = new List<int>();
        public List<string> listaChar = new List<string>();
        /*Construcor de un atributo*/
        public Atributo(string nom, char tipo, int longitud, long dirA, int tInd, long dirInd, long dirSigA)...

        /*Método que crea la lista de registros de este atributo*/
        public void creaLista(string texto)...

        /*Método que agrega la lista de registros de este atributo*/
        public void agregaAListaRegistro(string texto)...
    }
}
```

Índice

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;

namespace Diccionario_de_datos
{
    public class Indice
    {
        Entidad entidadActual;
        Atributo atr;
        string nombreEntidad;
        long cabecera = 0;
        long tam = 0;

        long tamBloque = 0;
        long dirRegistro = 0;

        /*Constructor de la clase Indice */
        public Indice(Atributo atributo, string nombreEnt)...

        public Indice(Entidad ent, Atributo atributo, string nombreEnt, long dirReg)...

        /*Método que regresa el tamaño del bloque de un indice con su bloque de desbordamiento*/
        private long regresaTamañodelBloque()...

        /*Método que crea el archivo .idx y llena el bloque con datos que son iguales a -1*/
        public void escribeIndice()...

        /*Método que lee el archivo y genera una lista con la informacion ordenada*/
        public void guardaDatoEnteroIndicePrimario(int valor, long direccion)...

        /*Método que lee el archivo y genera una lista con las cadenas de manera ordenada*/
        public void guardaDatoStringIndicePrimario(string valor, long direccion)...

        /*Método principal para agregar un dato entero de indice secundario al archivo.*/
        public void indiceSecundario(int valor, long direccion)...

        /*Método principal para agregar una cadena de indice secundario al archivo*/
        public void indiceSecundario(string valor, long direccion)...
```

```
        /*Método que guarda la direccion de un registro en el cajon del indice secundario*/
        //El primer parametro es la direccion del cajon, el segundo el valor a escribir en el archivo, el tercero es la opcion para guardar
        //o eliminar un valor, 0 = Guardar valor 1 = Eliminar valor
        public void guardaDireccionenCajon(long dirBloque, long valor, int operacion)...

        /*Método que crea los cajones del indice secundario y los llena con valores de -1*/
        public long creaCajonesIndiceSecundario()...

        /*Método que escribe la informacion en el archivo idx cuando el valor es entero*/
        private void escribeDatosidx(List<int> val, List<long> dir)...

        /*Método que escribe la informacion en el archivo idx cuando el valor es una cadena*/
        private void escribeDatosidx(List<string> val, List<long> dir)...

        /*Método para eliminar un un indice en el archivo idx*/
        public void eliminaIndice(int valor)...
        public void eliminaIndice(int valor,ref Entidad ent)...
        public long regresaCabe() ...
        /*Método para eliminar un indice  primario de tipo string*/
        public void eliminaIndice(string valor)...

        /*Método para eliminar un valor de indice secundario en el archivo idx de tipo entero*/
        public void eliminaIndiceSec(int valor, long direccion)...

        /*Método para eliminar un valor de indice secundario en el archivo idx de tipo string*/
        public void eliminaIndiceSec(string valor, long direccion)...
```

```
/*Método que crea el bloque en el archivo idx para guardar los apuntadores a los cajones*/
public void creaIndiceHash()...

public long creaIndiceHash(FileStream file)...


/*Método que escribe un valor con Indice Hash*/
public void Hash(string valorBinario, List<Atributo> lsAtr, Entidad ent)...

private List<long> mueveRegistrosInd(int indiceTabla, long direccionCajon, List<Atributo> lsAtr, List<long> direcciones, string numActual, int ind

/*Método que actualiza el indice de una cajon que se ha llenado*/
private void actualizaIndices(long dirCajon, int ind)...


private void actualizaIndices(FileStream guardar, long dirCajon, int ind)...

/*Método que crea un cajon para el indice hash*/
private long creaCajonHash(int ind, List<Atributo> lsAtr)...
```

```
/*Método que organiza el reacomodo de los registrsos al duplicarse la tabla*/
private int reacomodaRegistros(long dirCajonLleno, List<Atributo> atributos, List<string> posicionBinaria, List<long> dirCajones, int indTabla)...

/*Método que llena de -1's una direccion especifica de un registro en el indice idx*/
public void eliminaRegistro(FileStream file, long posicionElimina, List<Atributo> lsAtr)...

/*Método que encuentra el registro a eliminar en el archivo idx*/
public void encuentraRegistroaEliminar(int valorElimina, long dirTablaHash, List<Atributo> atributos)...

public int regresaIndiceCajonHash(long dirCajon)...
```

Cajón Hash

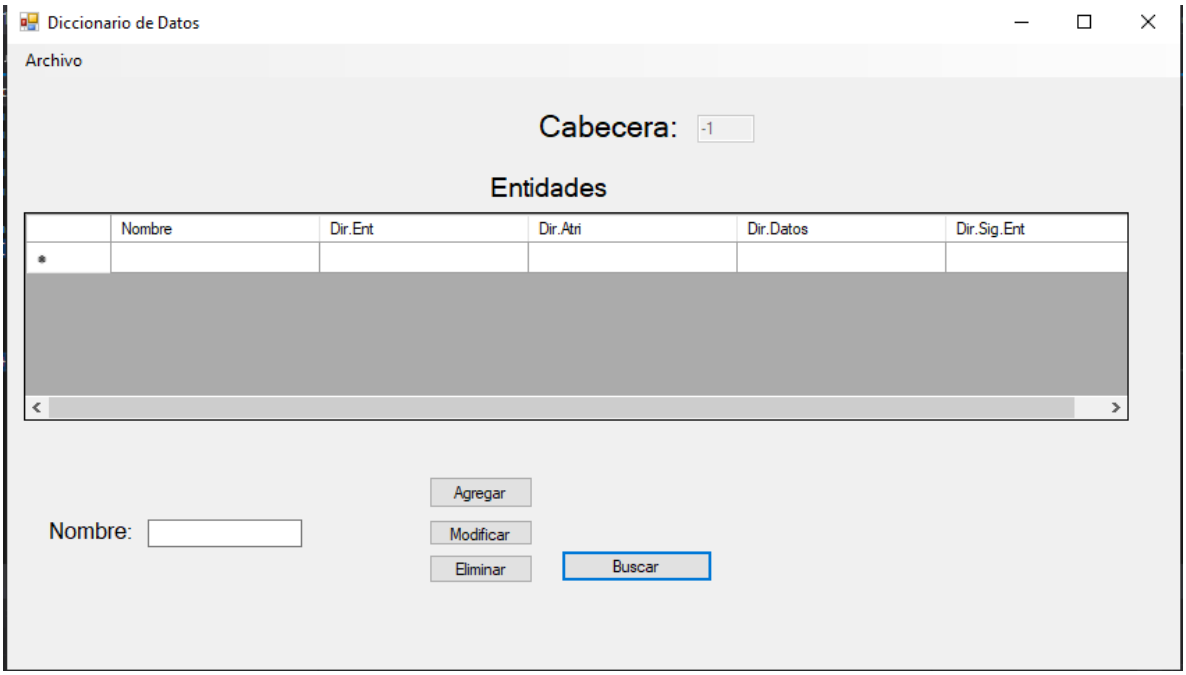
```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Diccionario_de_datos
{
    /*Funcion para manejar los cajones hash*/
    public class CajonHash
    {
        public List<DatoCajonHash> Cajon = new List<DatoCajonHash>();
        public long dircajon;
    }
}
```

Dato Cajón Hash

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Diccionario_de_datos
{
    public class DatoCajonHash
    {
        public int valint;
        public long dir;
    }
}
```

Form

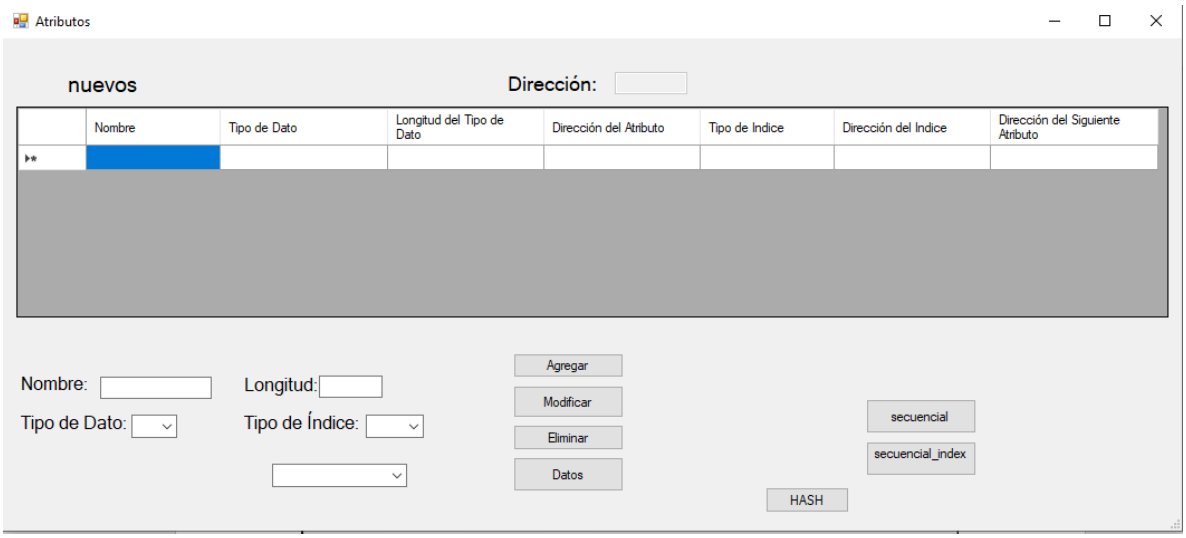
Form1




```
using System;
using System.Collections.Generic;
using System.IO;
using System.Windows.Forms;

namespace Diccionario_de_datos
{
    public partial class Form1 : Form
    {
        List<Entidad> lsEntidad = new List<Entidad>();
        public List<Entidad> lsEntidadAux = new List<Entidad>();
        FileStream archivo;
        public long cabecera;
        public long tamArchivo = 0;
        string nombreArchivo;
        string entSeleccionada = "";
        /*Constructor del form princial*/
        public Form1()...
        private void Form1_Load(object sender, EventArgs e)...
        //Método que cierra la ventana al hacer clic en la opcion cerrar
        private void cerrarToolStripMenuItem_Click(object sender, EventArgs e)...
        //Evento que inicia al hacer clic en abrir
        private void abrirArchivoToolStripMenuItem_Click(object sender, EventArgs e)...
        //Método que abre un archivo existente.
        public void AbrirArchivo()...
        //Método que lee la informacion del archivo que se va a abrir
        public void leeArchivo(FileStream file)...
        //Evento que inicia al hacer clic en crear
        private void crearArchivoToolStripMenuItem_Click(object sender, EventArgs e)...
        //Método que guarda el archivo por entidad
        private void escribeArchivo(string nomArchivo)...
        //Evento de hacer clic en el boton de Crear Entidad
        private void bt_CrearEntidad_Click(object sender, EventArgs e)...
        /*Evento de hacer clic en el boton de agregar*/
        private void bt_AgregarEnt_Click(object sender, EventArgs e)...
        /*Funcion que lee los atributos directamente del archivo*/
        private void leeAtributos(Entidad entidad, FileStream fileStream)...
        /*Método que compara si la entidad a agregar ya se encuentra en la lista de Entidades*/
        public int comparaEntidades(string nombre)...
```

```
//Funcion que ordena las entidades de manera logica
private void ordenaEntidades()...
/*Evento de modificar el nombre de una entidad, lo camnbia  y lo guarda en la entidad*/
private void bt_ModificarEnt_Click(object sender, EventArgs e)...
/*Evento de eliminar una entidad*/
private void bt_EliminarEnt_Click(object sender, EventArgs e)...
/*Funcion qur actualiza los atributos*/
public void ActualizaAtributos(List<Entidad> entidads, Entidad ent)...
/*Evento que abre la ventana para agregar atributos de una entidad*/
private void bt_AgregaAtr_Click(object sender, EventArgs e)...
/*Metodo que actulaiza los valores del DGV cuando hay un cambio*/
private void actualizaDGV()...
/*Funcion que recupera los Registros de datos directamente del archivo*/
public void RecuperaDatos(Entidad entidad)...
public void RecuperaRegistros()...
/*Evento que al dar click en el boton te manda al form busqueda para buscar entidades*/
private void btn_buscarEntidades_Click(object sender, EventArgs e)...
```

Form2

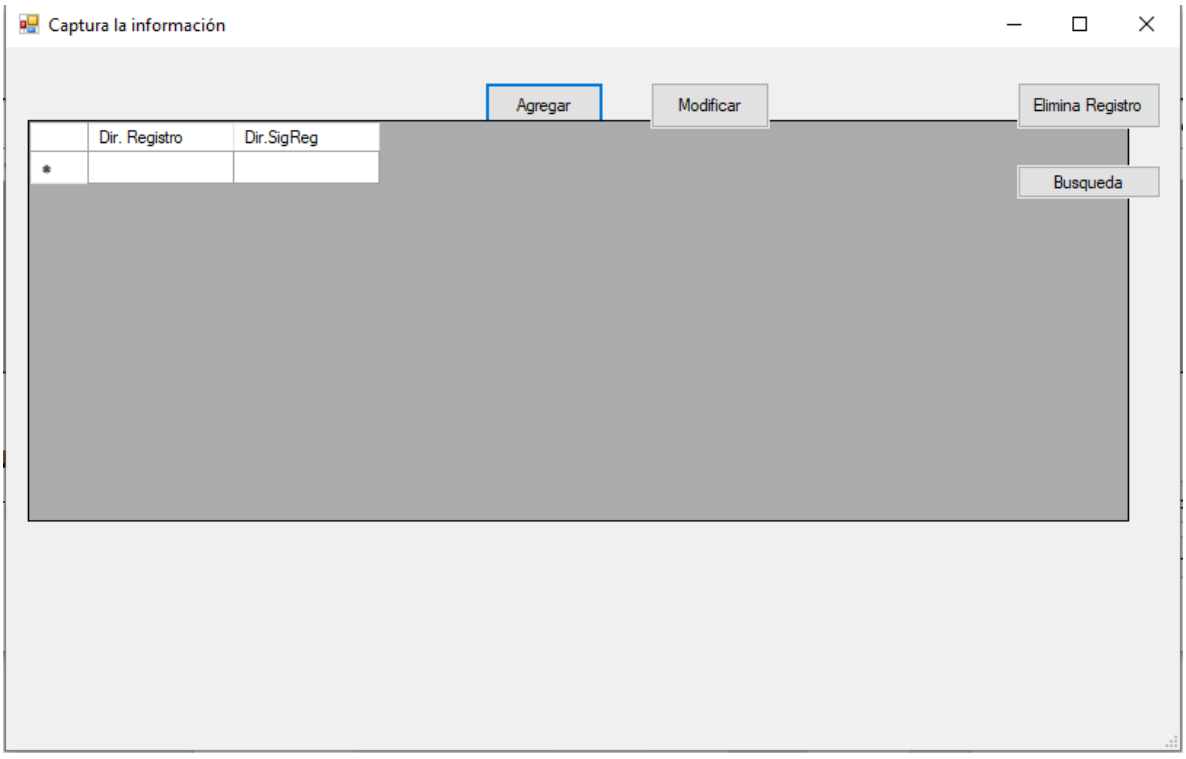


```
using System;
using System.Collections.Generic;
using System.IO;
using System.Windows.Forms;


namespace Diccionario_de_datos
{
    public partial class Form2 : Form
    {

        public Entidad entidadSel;
        public string nomArchivo;
        public long tamArchivo;
        public long cabeceraAtr;
        public List<Entidad> Aux = new List<Entidad>();
        public List<Entidad> Aux2 = new List<Entidad>();
        public List<CajonHash> CajonesHash = new List<CajonHash>();
        public List<CajonHash> CajonesHashAUX = new List<CajonHash>();

        public List<long> CajPrinHach = new List<long>();
        public List<long> CajPrinHachAUX = new List<long>();
         public List<long> CajPrinHachAUX2 = new List<long>();
        public List<int> vs = new List<int>();
        public List<long> vs2 = new List<long>();
        FileStream fileidx;

        /*Constructor del Form2*/
        public Form2(Entidad agrega, string archivo, long tam, List<Entidad> lsEntidad)...

        private void Form2_Load(object sender, EventArgs e)...

        /*Método que se encarga de leer los atributos que esten guardados en el archivo*/
        private void leeAtributos()...
        /*Metodo que verifica que un atributo no este repetido*/
        private bool checaAtributoRepetido()...
        /*Metodo que detecta el indice de una entidad en una lista*/
        public int detectaInd()...
        /*Metodo que suma el tamaño de cada atributo de una entidad*/
```

```
        /*Metodo que suma el tamaño de cada atributo de una entidad*/
        public int sumaTamAtributos(Entidad entida)...
        /*Evento que agrega la informacion a un atributio*/
        private void bt_AgregarAtr_Click(object sender, EventArgs e)...

        /*Método que escribe el atributo en el archivo*/
        private void escribeAtributo()...

        /*Método que actuliza el DGV en cada cambio que hay*/
        private void actualizaDGVAtr()...

        /*Evento de apretar el boton de modificar atributo*/
        private void bt_ModificarAtr_Click(object sender, EventArgs e)...

        /*Evento de eliminar un atributo*/
        private void bt_EliminarAtr_Click(object sender, EventArgs e)...
        public List<string> regresaDatos()...
        /*Evento de insertar un nuevo registro de información*/
        private void bt_InsertaRegistro_Click(object sender, EventArgs e)...

        /*Evento de cerrar la ventana*/
        private void Form2_FormClosing(object sender, FormClosingEventArgs e)...

        /*Método que regresa el valor de un registro con todos sus atributos*/
        private int tamañoDeRegistro()...

        /*Método que muestra los indices de un atributo*/
        private void bt_VerIndices_Click(object sender, EventArgs e)...

        internal void ActualizaAtributos(ref List<Entidad> lsEntidadAux, Entidad ent)...

        /*Evento para detectar cuando cambia de tipo Entero o Cadena, si es entero se le da un valor por default de 4*/
        private void Cb_TipoDato_SelectedIndexChanged(object sender, EventArgs e)...
```

```
private void comboBox1_SelectedIndexChanged(object sender, EventArgs e)...
/*Evento para detectar cuando cambia de tipo Entero o Cadena, si es entero se le da un valor por default de 4*/
private void cb_TipoIndice_SelectedIndexChanged(object sender, EventArgs e)...
public void creaCajonHASHpRINCIPAL()...
//Funcion que abre mi archivo de datos para asi usarlos despues en el hash
private void abreArchivoRegistros()...

private void cargaIDXHash()...
//checar clase cajon y DatoCajon
private void hashEsbtn_Click(object sender, EventArgs e)...
/*Metodo que detecta y crea un cajon solo si se necesita*/
private void creaCajonBajoDemanda(int res, DatoCajonHash auxcaj)...
}
```

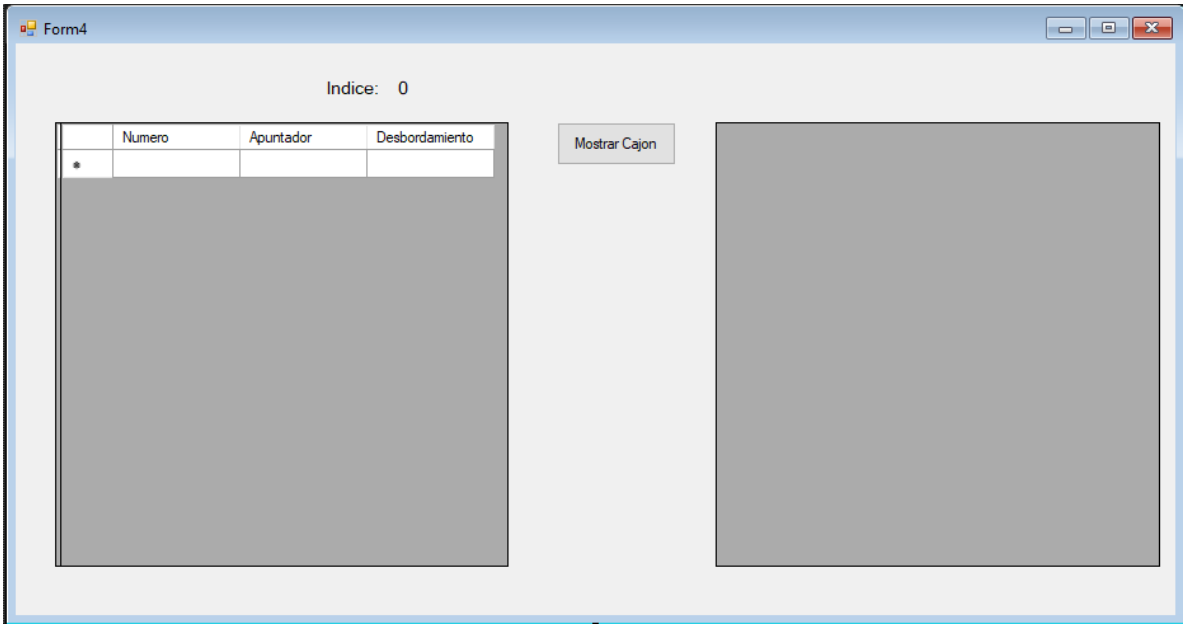
Form3



```
private void Form3_Load(object sender, EventArgs e)...
private bool checa()...
private bool checa2()...
/*Evento al hacer clic en el boton de guardar*/
private void bt_Guardar_Click(object sender, EventArgs e)...
private bool ChecaTamaño()...
/*Método que solamente escribe los datos en el DGV*/
private void escribeRegistros()...
/*Método que escribe el los datos por indice 2 */
private void escribeRegistrosPorIndice2()...
/*Método que elige si la clave de busqueda se hara con datos int o string*/
private void claveDeBusqueda(Atributo atr, int col)...
/*Método que crea una instancia de la clase Indice para crear el archivo idx y guardar los datos*/
private void indicePrimarioChido(Atributo atr, string valor, long direccion)...
/*Método que crea una instancia de la clase indice e inserta el dato en el indice secundario*/
private void indiceSecundario(Atributo atr, string valor, long direccion)...
/*Método que crea una instancia de la clase indice e inserta el dato de tipo hash dinamico*/
private void hashDinamico(Atributo atr, string valor, long direccionReg)...
/*Método que regresa una cadena que contiene el valor en binario de un número entero*/
private string regresavalorBinario(int valor)...
/*Método que guarda la informacion del DGV en el archivo '.dat'*/
private void guardaDGV()...
private void abreArchivoRegistrosClave8(Atributo nom, ComboBox comboBox)...
/*Método que abre el archivo .dat*/
private void abreArchivoRegistros()...
/*Método que ordena segun el atributo que tenga la clave de busqueda en su tipo de indice, en este caso solo lo hace
 con valores de tipo int*/
private void ordenaClavedeBusquedaInt(List<int> listaInt, int col)...
/*Método que ordena segun el atributo que tenga la clave de busqueda en su tipo de indice, en este caso solo lo hace
 con valores de tipo string*/
private void ordenaClavedeBusquedaString(List<string> listaStr, int col)...
/*Evento de hacer clic en el boton de modificar*/
private void bt_ModificarReg_Click(object sender, EventArgs e)...
/*Evento de hacer clic en el boton de eliminar*/
private void bt_EliminaReg_Click(object sender, EventArgs e)...
/*Método que regresa la cabecera del registro de datos*/
public long regresaCabecera()...
public List<string> regresaDatos()...
private void Form3_FormClosing(object sender, FormClosingEventArgs e)...
/*Evento click que al precionar el boton te lleva al form para buscar datos */
private void button1_Click(object sender, EventArgs e)...
```

```
using System;
using System.Collections.Generic;
using System.Drawing;
using System.IO;
using System.Windows.Forms;

namespace Diccionario_de_datos
{
    public partial class Form3 : Form
    {
        string nombreEntidad;
        public List<Entidad> lssentidads = new List<Entidad>();
        public List<Atributo> atributos = new List<Atributo>();
        public List<TextBox> textBoxes = new List<TextBox>();//Crea una lista de TextBoxes para manipularlos y tener acceso a su informacion.
        public ComboBox ComboBoxRRR = new ComboBox();
        Point p1 = new Point(15, 30);//Posicion X,Y de los label.
        Point p2 = new Point(125, 30);//Posicion X,Y de los TextBox.
        int cambiar = 6;
        long tamArchivoData = 0;
        int tamRegistro = 0;
        long cabecera = 0;
        Entidad entActual;
        Entidad entActualCla1;
        Entidad entActualCla2;
        Entidad entActualCla3;
        Entidad entActualCla6;

        /*Constructor del Form 3*/
        public Form3(string nEnt, List<Atributo> lista, int tam, long cabRegistros, Entidad ent,List<Entidad> lsentidadessss)...
```

Form4

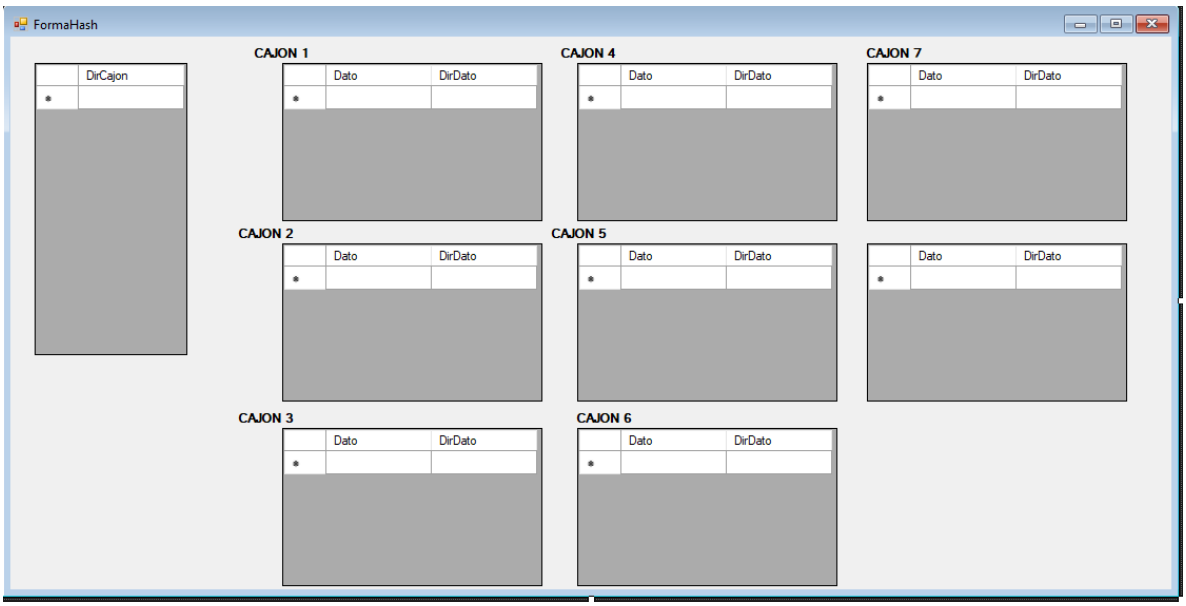


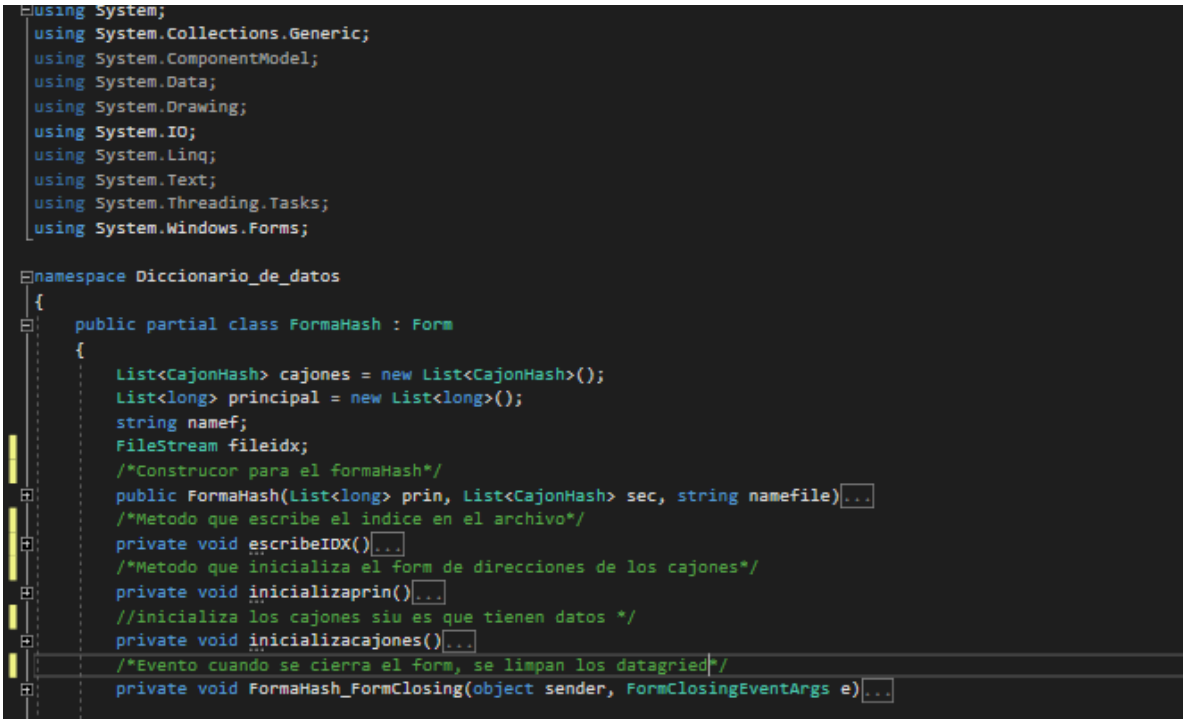

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Windows.Forms;

namespace Diccionario_de_datos
{
    public partial class Form4 : Form
    {
        Atributo atr;
        string nomEntidad;
        List<Atributo> lsAtr;
        /*Constructor del Form 4*/
        public Form4(string entidad, Atributo atri)...
        /*Constructor del Form 4*/
        public Form4(string entidad, List<Atributo> atributos, Atributo atri)...
        /*Constructor del Form 4*/
        private void Form4_Load(object sender, EventArgs e)...
        /*Método que segun el tipo de dato crea listas para guardar informacion del indice*/
        private void creaListas()...
        /*Método que lee los valores enteros y los escribe en un dgv*/
        private void leeArchivoidx(List<int> lsInt, List<long> lsDir, long reg)...
        /*Método que lee los valores de tipo string y los escribe en un dgv*/
        private void leeArchivoidx(List<string> lsStr, List<long> lsDir, long reg)...
        /*Evento para mostrar el contenido de un cajon en un dgv*/
        private void bt_Cajon_Click(object sender, EventArgs e)...
        /*Método que lee el contenido de un cajon del indice secundario*///recuerdaquehacer
        private void leeidxCajon()...
        /*Método que lee la caja principal del indice Hash dinámico y lo muestra en eun DGV*/
        private void leeidxHash()...
        /*Métodp que lee un cajon de tipo Hash Dinámico y lo muestra en un DGV*/
        private void leeidxCajonesHash()...
        /*Método que regresa un valor con la longitud de los atributos*/
        private int regresaLongituddeAtributos()...
```

Form Búsqueda


```
using System.Collections.Generic;
using System.Windows.Forms;

namespace Diccionario_de_datos
{
    public partial class FormBusqueda : Form
    {
        List<Entidad> Entidads = new List<Entidad>();
        List<Atributo> atribu = new List<Atributo>();
        bool atributoOentidad;
        DataGridView DATOS = new DataGridView();
        /*Constructor en caso de que solo se busque una entidad*/
        public FormBusqueda(List<Entidad> lsentidads)...
        /*Constructor en caso de que se desee consultar algun dato*/
        public FormBusqueda(List<Atributo> atributossss, DataGridView dataGridViewDatos)...
        /*Metodo que se encarga de encontrar una entidad*/
        public void BuscaEntidades()...
        /*Metodo que se encarga de encontrar un dato o  varios*/
        public void BuscaDato() ...
        /*Metodo que se encarga de mostrar los datos que se encontraron en un datagrid*/
        public string imprimeInfoDato(DataGridViewRow row ) ...
```

FormHash





```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.IO;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Diccionario_de_datos
{
    public partial class FormaHash : Form
    {
        List<CajonHash> cajones = new List<CajonHash>();
        List<long> principal = new List<long>();
        string namef;
        FileStream fileidx;
        /*Construcor para el formaHash*/
        public FormaHash(List<long> prin, List<CajonHash> sec, string namefile)...
        /*Metodo que escribe el indice en el archivo*/
        private void escribeIDX()...
        /*Metodo que inicializa el form de direcciones de los cajones*/
        private void inicializaprin()...
        //inicializa los cajones siu es que tienen datos */
        private void inicializacajones()...
        /*Evento cuando se cierra el form, se limpan los datagried*/
        private void FormaHash_FormClosing(object sender, FormClosingEventArgs e)...
```