

# Manual de programador

Autor: Mario Alberto Macias Escobedo

Universidad Autónoma De San Luis Potosí

## Introducción

En este anexo se describe la documentación técnica de programación, incluyendo la instalación del entorno de desarrollo, la estructura de la aplicación, su compilación, la configuración entre otros.

## Estructura de Librerías

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

En especial usaremos: `using System.IO;` Ya que es el encargado de poder crear, abrir archivos que se encuentran en nuestro sistema.

Ya que los otros, por lo general Visual Studio los genera automáticamente.

## Manual del programador

El siguiente manual tiene como objetivo servir de referencia a futuros programadores que desean trabajar con esta aplicación. En él se explicara como es el entorno de desarrollo, diferentes clases utilizadas, y como se podría testear.

### Entorno de desarrollo

Para trabajar con el proyecto se necesita tener instalados los siguientes programas y dependencias:

- Windows
- GIT
- Visual Studio 2019 o mayor.

Este proyecto se basa únicamente en el sistema operativo Windows, desarrollado en Windows 10, 2020.

## Obtener el código fuente

El código esta abierto en GitHub, con el nombre de: MarioMacias/Archivos

Esta para poder descargarlo o simplemente si ya se tiene con el archivo en cuestión, podrás encontrar el archivo .sln, con el que se abrirá en Visual Studio.

## Forms (Windows) del programa

Aclaración: Todas las clases, tienen llamado a su respectivo \_Load(), el cual se ejecuta al inicial

ConsultaRegistro.cs

Esta es la ventana donde se despliega al momento de que el usuario ingrese a este apartado, aquí solo se muestran datos, generados por las otras clases.

FormAtributo.cs

En este apartado, estaremos guardando, modificando y borrando los atributos.

Algunos de los métodos mas importantes son los siguientes:

```
public bool agregaAtributoArchivo(string nombreArchivo, int pos, List<Entidad> entidades)
```

Este método es llamado al momento de presionar agregar, para poder agregar un atributo y guardarlo en un archivo binario.

```
private void btn_modifAtributo_Click(object sender, EventArgs e)
```

Evento para poder modificar un atributo, aquí cabe aclarar que habrá otros botones para modificar el atributo dependiendo de lo que se quiera modificar, cada nombre identifica lo que se modificará.

```
private void btn_eliminarAtributo_Click(object sender, EventArgs e)
```

Este evento eliminara un atributo del archivo binario, en dado caso que el archivo se quede sin atributos, saldrá un mensaje identificando que no existe ningún tipo de atributo que se pueda eliminar.

### ConsultaAtributo.cs

Esta es la ventana donde se despliega al momento de que el usuario ingrese a este apartado, aquí solo se muestran datos, generados por las otras clases.

```
private void tb_Buscar_TextChanged(object sender, EventArgs e)
```

Este método va buscando entre la lista de los atributos, los diferentes metadatos de la entidad seleccionada que contenga ciertos caracteres del textbox.

### FormEntidad.cs

Ventana principal, donde se deben agregar las entidades, esta ventana siempre regresa al principio ya que contiene en la parte superior los accesos a diferentes ventanas.

```
private void btn_Modificar_Click(object sender, EventArgs e)
```

Modifica la entidad seleccionada, solo se puede, debe, modificar el nombre de la entidad, ya que los demás apartados son direcciones.

```
private void btn_Borrar_Click(object sender, EventArgs e)
```

Elimina la entidad seleccionada, no se puede eliminar si existe registros guardados en el.

```
private void abrirArchivoToolStripMenuItem_Click(object sender, EventArgs e)
```

Abrir un archivo existente, el archivo que se debe abrir, o puede abrir, son los archivos con terminación, dd.

```
private void cerrarArchivoToolStripMenuItem_Click(object sender, EventArgs e)
```

Cerrar el archivo actual para poder abrir algún otro.

```
private void primarioToolStripMenuItem_Click(object sender, EventArgs e)
```

Te abrirá la ventana de índice primario, y la ventana de entidades se ocultará hasta que se necesite de nuevo

```
private void secundarioToolStripMenuItem_Click(object sender, EventArgs e)
```

Te abrirá la ventana de índice secundario, y la ventana de entidades se ocultará hasta que se necesite de nuevo

```
private void consultaToolStripMenuItem_Click(object sender, EventArgs e)
```

Abre la ventana de consulta entidad, para mostrar los datos de la entidad

```
private void hashToolStripMenuItem_Click(object sender, EventArgs e)
```

Abrir la ventana de un índice hash, para ver sus datos.

```
private void registrosToolStripMenuItem_Click(object sender, EventArgs e)
```

Este abre la ventana donde se puede agregar, modificar o ver los registros existentes.

ConsultaEntidad.cs

Esta es la ventana donde se despliega al momento de que el usuario ingrese a este apartado, aquí solo se muestran datos, generados por las otras clases.

### FormIndiceHash.cs

Esta es la ventana donde se visualizará toda la información de los índices hash

```
private void escribeDireccionesCajones()
```

Aquí pondrá en el datagrid los valores correspondientes del hash, todos estos datos ya están guardados en memoria, una vez que se van ingresando los registros.

```
private void escribirIndice()
```

Escribir en el datagrid de índice, los datos correspondientes, que en este caso serían las direcciones

### FormIndicePrimario.cs

Esta ventana mostrara los índices primarios correspondientes, guardados, se crean cada vez que se están agregando los registros.

```
private void llenaData()
```

Llena el datagrid con la información correspondiente a los índices primarios.

### FormIndiceSecundario.cs

Esta ventana mostrara los índices Secundarios correspondientes, guardados, se crean cada vez que se están agregando los registros.

```
private void llenaData()
```

Llena el datagrid con la información correspondiente a los índices Secundarios.

### BusquedaIndice.cs

Esta es la ventana donde se despliega al momento de que el usuario ingrese a este apartado, aquí solo se muestran datos, generados por las otras clases.