***Sistema Gestión de Clientes***

***Tecnologías (herramientas) utilizados :***

- Visual Basic (lenguaje de programación)

- MySql (Sistema gestor de base de datos)

***Programas:***

- Microsoft Visual Studio ( IDE entorno de desarrollo integrado)

-Xampp:  
 (Paquete de software que proporciona un entorno de desarrollo local.

**X** = representa la compatibilidad con múltiples Sistemas Operativos como Windows, Linux, etc.

**A** = Apache, un servidor web.

**M** = MySQL, un sistema de gestión de bases de datos.

**P** = Perl, lenguaje de Programación.

**P** = PHP, lenguaje de Programación.)

***El Sistema:***

El sistema en resumen se encarga de la gestión de clientes en la empresa, creando, eliminando y modificando a dichos clientes.

También esta la funcionalidad de poder dar de baja a un cliente especifico o darle de alta nuevamente. Por ejemplo cuando un cliente paga una cuota mensual esta en ‘Activo’ o cuando no paga las cuotas esta en ‘Inactivo’.

Aparte, el sistema, tiene la funcionalidad de generar un reporte en pdf con todos los clientes guardados en la base de datos.

------------------------------------------------------------------------------------------------------------------------

***Librerías o partes concretas del código:***

**Imports MySql.Data.MySqlClient:** indica que se están utilizando clases y funcionalidades proporcionadas por la biblioteca de MySQL para .NET. En este caso, específicamente, se está importando la funcionalidad relacionada con el acceso a bases de datos MySQL.

Algunas de las clases clave proporcionadas por MySql.Data.MySqlClient incluyen:

**MySqlConnection**: Representa una conexión a una base de datos MySQL.

**MySqlCommand:** Ejecuta comandos SQL en una base de datos MySQL.

**MySqlDataReader**: Lee un conjunto de resultados de una consulta SQL.

**MySqlDataAdapter**: Representa un conjunto de comandos de datos y una conexión a una base de datos que se utilizan para llenar un **DataSet** y actualizar una base de datos de acuerdo con cambios en el conjunto de datos.

Se ocupa **Imports** para importar una librería externa o interna al entorno.

Cabe aclarar que en este caso es una librería externa (no esta en el entorno de visual) ya que hay que descargar, instalar e integrar un driver de mysql que se llama así :

**Mysql-connector-net**

**Me. =** Se ocupa para referenciar a la actual clase donde se encuentra. Por ej en:

Me**.Dispose()**

**Dispose** = libera los recursos no administrados utilizados por el formulario. Esto puede incluir la liberación de memoria y la limpieza de recursos antes de que el formulario sea eliminado.

**ESTRUCTURA TRY...CATCH:**

Se utiliza para manejar excepciones o errores que pueden ocurrir durante la ejecución de un bloque de código. Proporciona un mecanismo para controlar y responder a situaciones excepcionales, evitando que el programa se detenga abruptamente.

**Try:** En este bloque, colocas el código que podría generar una excepción. El bloque Try define la región donde se supervisan las excepciones.

**Catch:** En este bloque, especificas cómo manejar las excepciones que se podrían haber producido en el bloque Try. Puedes capturar tipos específicos de excepciones y proporcionar código personalizado para manejar cada tipo de excepción. Por ejemplo, ex As Exception es la variable que captura la excepción.

**Ej:**

Try

‘’Código que podría generar excepción

Catch ex As MySqlException

MsgBox("Error en la base de datos: " & ex.Message, vbCritical, "Error")

‘’Excepción de mysql.

Catch ex As Exception

MsgBox("Error general: " & ex.Message, vbCritical, "Error")

‘’ Excepción del programa.

End Try

**Estructura Using :**

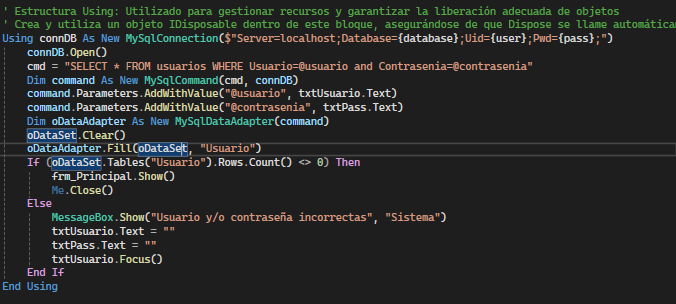
Esto es algo que agregue yo xd pero sirve bastante, en resumen, garantiza la correcta liberación de recursos y manejar objetos que implementan la interfaz IDisposable. La interfaz IDisposable requiere que una clase proporcione un método llamado Dispose, que se utiliza para liberar recursos no administrados, como conexiones a bases de datos, archivos o cualquier otro recurso que pueda agotar los recursos del sistema.

La principal utilidad de Using radica en simplificar y asegurar la liberación de recursos al garantizar que el método Dispose se llame automáticamente al salir del bloque Using. Esto es especialmente útil para evitar problemas de pérdida de memoria o degradación del rendimiento debido a la acumulación de recursos no liberados.

En resumen sirve para garantizar que funcione de manera optimizada el programa y que se cierre la base de datos aun si se llama al metodo .Close().

EJ:  
 ' Estructura Using: Utilizado para gestionar recursos y garantizar la liberación adecuada de objetos

' Crea y utiliza un objeto IDisposable dentro de este bloque, asegurándose de que Dispose se llame automáticamente al salir.

**ConDB =** Es una instancia de MsqlConnection que se usa para conectar la base de datos con el programa, se pasa como parámetro lo siguiente:  
**$”Server=localhost;Database={database};Uid={User};Pwd={pass};”**

El signo **$** peso se usa para que funciones las variables como por ej {database}

**Server= localhost** //Esto indica que se esta conectando a una base de datos de manera local (en tu compu)

**Database={database}** // Acá se indica el nombre de la base de datos con una variable que se declara un poco mas arriba.

**Uid={user}** //// Acá se indica el nombre de usuario de la base de datos con una variable que se declara un poco mas arriba.

**Pwd={pass}** // Acá se indica la contraseña de usuario de la base de datos con una variable que se declara un poco mas arriba.

***MySqlAdapter:*** es parte del proveedor de datos de MySQL para .NET y se utiliza comúnmente para conectar conjuntos de datos de un origen de datos, como una base de datos MySQL, con controles de datos en una interfaz de usuario. Ej:

Dim oDataAdapter As New MySqlDataAdapter(command)

***oDataSet.Clear():*** Esta línea de código borra cualquier contenido existente en el objeto DataSet llamado oDataSet. Un DataSet es una estructura de datos en memoria que puede contener tablas, relaciones y restricciones. Al llamar a Clear(), se eliminan todas las tablas y datos existentes en el DataSet, preparándolo para recibir nuevos datos.

***oDataAdapter.Fill(oDataSet, "Usuario"):*** Esta línea de código utiliza el adaptador de datos (oDataAdapter) para llenar el DataSet (oDataSet) con los resultados de la consulta SQL definida en el adaptador. En este caso, los resultados se llenan en una tabla llamada "Usuario". La operación Fill toma los datos devueltos por la consulta y los carga en el DataSet, creando o actualizando la tabla llamada "Usuario" dentro del DataSet.

