Indice

[Clase Login 3](#_Toc171583052)

[Clase Conexión 6](#_Toc171583053)

**Indice de Figuras**

[Figura1 Codigo Login 3](#_Toc171583078)

[Figura2 Codigo btnCancelar 4](#_Toc171583079)

[Figura3 Codigo Login\_Load 5](#_Toc171583080)

[Figura4 Codigo Variables 6](#_Toc171583081)

[Figura5 Codigo Conectar 7](#_Toc171583082)

[Figura6 Codigo Validar Correo 7](#_Toc171583083)

[Figura7 Codigo Valida ID 8](#_Toc171583084)

[Figura8 Codigo Validar Usuario 9](#_Toc171583085)

[Figura9 Codigo Consulta 9](#_Toc171583086)

[Figura 10 Codigo Insertar Datos 10](#_Toc171583087)

[Figura11 Codigo Encriptar 11](#_Toc171583088)

[Figura12 Codigo Desencriptar Clave 12](#_Toc171583089)

# Introducción

En el mundo actual, la gestión eficiente de datos y la seguridad de la información son aspectos críticos para cualquier organización. Las bases de datos se han convertido en el núcleo de muchas aplicaciones, permitiendo el almacenamiento, acceso y manipulación de grandes volúmenes de datos. Visual Basic, con su facilidad de uso y potente capacidad de integración con bases de datos, es una herramienta valiosa para desarrolladores que buscan crear aplicaciones robustas y seguras.

El propósito de este documento es proporcionar una guía técnica detallada sobre un conjunto específico de código en Visual Basic que maneja la conexión a una base de datos SQL Server y la ejecución de varias operaciones esenciales. Este código es parte de una clase denominada Conexion, la cual facilita la gestión de usuarios y la validación de datos mediante métodos que permiten conectar a la base de datos, validar correos electrónicos, validar identificaciones, realizar consultas SQL, insertar datos y encriptar/desencriptar contraseñas.

La clase Conexion es fundamental para garantizar la integridad y seguridad de los datos almacenados en la base de datos. A través de una explicación minuciosa de cada método y sus funcionalidades, este documento pretende ofrecer una comprensión clara y completa del funcionamiento interno de la clase. Además, se incluyen capturas de pantalla del entorno de desarrollo en Visual Studio y del código en ejecución para proporcionar una referencia visual que facilite la comprensión del contenido técnico.

La estructura del documento está organizada para seguir un enfoque lógico y pedagógico, comenzando con una visión general del código, seguida de una explicación detallada de cada método y culminando con las capturas de pantalla que ilustran la implementación práctica del código. Esta presentación permitirá a los lectores no solo entender el código, sino también aplicarlo en sus propios proyectos de desarrollo de software.

# Descripción General del Código

## Clase Login

La clase Login contiene los métodos necesarios para gestionar el proceso de autenticación de un usuario. A continuación, se detallan los métodos contenidos en esta clase:

* **Método btnAceptar\_Click**

Este método se ejecuta cuando el usuario hace clic en el botón Aceptar.

**Código**

Figura1  
Codigo Login



Explicación:

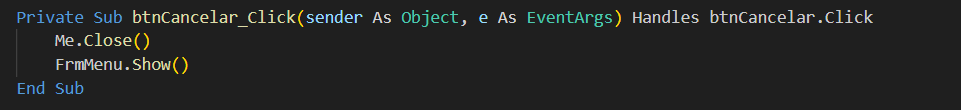
1. **Declaración de Variables**: Se declaran tres variables vusuario, vclave y strsql para almacenar el nombre de usuario, la contraseña y la cadena SQL respectivamente.
2. **Validación de Campos Vacíos**: Se verifica si los campos de usuario o contraseña están vacíos. Si es así, muestra un mensaje de error y enfoca el campo de usuario.
3. **Validación de Usuario**: Se valida si el usuario ya existe mediante el método valida\_usuario de la clase Conexion.
4. **Encriptación de Contraseña**: Si el usuario no existe, se encripta la contraseña usando el método Encriptar\_clave.
5. **Inserción en la Base de Datos**: Se crea una consulta SQL para insertar los datos del usuario en la base de datos y se ejecuta mediante el método inserta\_datos de la clase Conexion.
6. **Mensaje de Confirmación o Error**: Dependiendo del resultado de la inserción, se muestra un mensaje de confirmación o error y se limpian los campos de entrada.

* **Método btnCancelar\_Click**

Este método se ejecuta cuando el usuario hace clic en el botón Cancelar.

**Código**:

Figura2   
Codigo btnCancelar



**Explicación**:

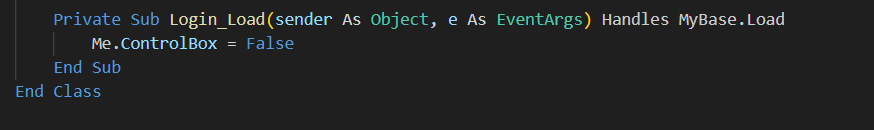
1. **Cerrar Ventana de Login**: Cierra la ventana de login actual.
2. **Mostrar Menú Principal**: Muestra el formulario principal de la aplicación (FrmMenu).

* **Método Login\_Load**

Este método se ejecuta cuando la ventana de login se carga y es donde finaliza la clase

**Código**:

Figura3   
Codigo Login\_Load



**Explicación**:

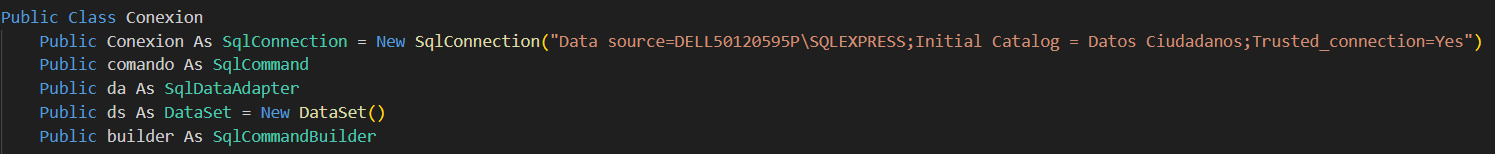
1. **Deshabilitar ControlBox**: Deshabilita la barra de control de la ventana (minimizar, maximizar y cerrar) y carga el login.

## Clase Conexión

La clase Conexion contiene métodos necesarios para gestionar la conexión a una base de datos SQL Server y operaciones relacionadas. A continuación, se detallan los métodos contenidos en esta clase.

* **Declaración de variables y conexión**

Figura4  
Codigo Variables

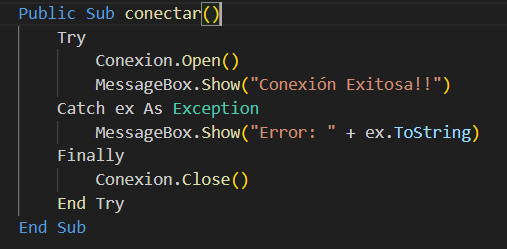
****

**Explicación**:

* **SqlConnection**: Se declara una variable de conexión a la base de datos SQL Server.
* **SqlCommand**: Se declara una variable para ejecutar comandos SQL.
* **SqlDataAdapter**: Se declara una variable para adaptar datos.
* **DataSet**: Se declara un conjunto de datos para almacenar los resultados de las consultas.
* **SqlCommandBuilder**: Se declara un constructor de comandos para construir y gestionar comandos SQL.
* **Método conectar**

Este método abre y cierra la conexión a la base de datos, mostrando un mensaje de éxito o error.

Figura5   
Codigo Conectar

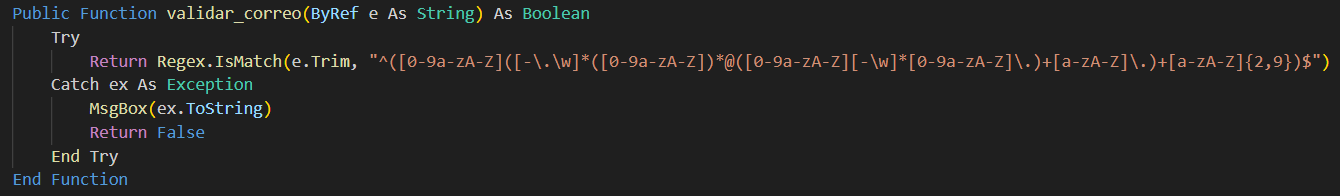


**Explicación**:

* **Abrir Conexión**: Intenta abrir la conexión a la base de datos.
* **Mostrar Mensaje**: Muestra un mensaje indicando el éxito o error de la conexión.
* **Cerrar Conexión**: Asegura que la conexión se cierre después de intentar abrirla.
* **Método validar\_correo**

Este método valida si una dirección de correo electrónico tiene el formato correcto utilizando expresiones regulares.

Figura6   
Codigo Validar Correo

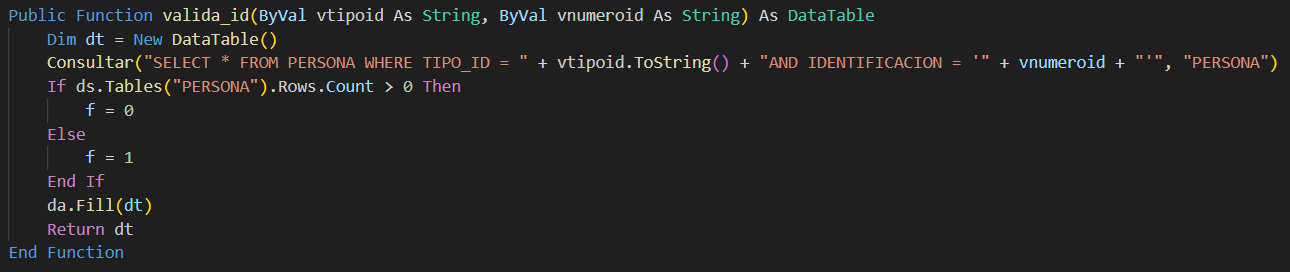


**Explicación**:

* **Expresión Regular**: Utiliza una expresión regular para verificar el formato del correo electrónico.
* **Captura de Excepciones**: Captura y muestra cualquier excepción que ocurra durante la validación.
* **Método valida\_id**

Este método valida si una identificación ya existe en la base de datos.

Figura7   
Codigo Valida ID

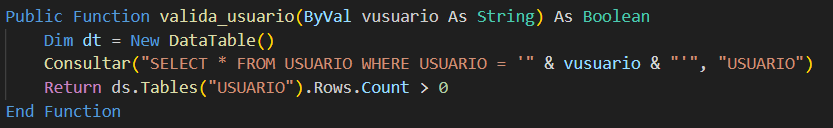


**Explicación**:

* **Consulta SQL**: Ejecuta una consulta para verificar si la identificación existe en la tabla PERSONA.
* **Validación de Resultados**: Verifica si se encontraron resultados y ajusta el valor de f en consecuencia.
* **Método valida\_usuario**

Este método valida si un nombre de usuario ya existe en la base de datos.

Figura8   
Codigo Validar Usuario

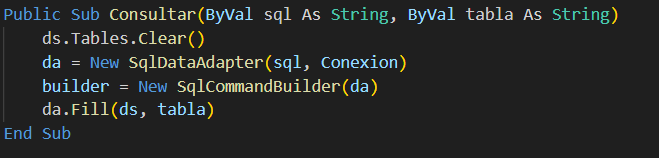


**Explicación**:

* **Consulta SQL**: Ejecuta una consulta para verificar si el usuario existe en la tabla USUARIO.
* **Retorno de Resultados**: Retorna True si se encontraron resultados, False en caso contrario.
* **Método Consultar**

Este método ejecuta una consulta SQL y llena el conjunto de datos con los resultados.

Figura9   
Codigo Consulta

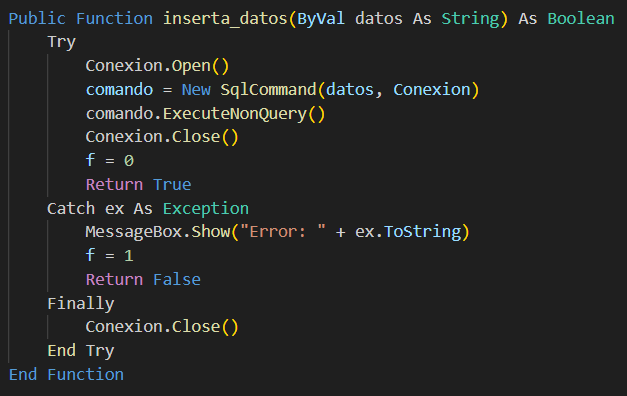


**Explicación**:

* **Limpieza de DataSet**: Limpia las tablas existentes en el DataSet.
* **Ejecución de Consulta**: Ejecuta la consulta SQL y llena el DataSet con los resultados.
* **Método inserta\_datos**

Este método inserta datos en la base de datos utilizando una consulta SQL.

Figura 10  
Codigo Insertar Datos

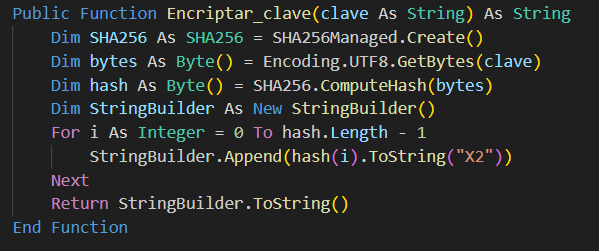


**Explicación**:

* **Apertura de Conexión**: Abre la conexión a la base de datos.
* **Ejecución de Comando**: Ejecuta el comando SQL para insertar datos.
* **Captura de Excepciones**: Captura y muestra cualquier excepción que ocurra durante la inserción.
* **Cierre de Conexión**: Asegura que la conexión se cierre después de intentar insertar datos
* **Método Encriptar\_clave**

Este método encripta una contraseña utilizando SHA-256.

Figura11   
Codigo Encriptar

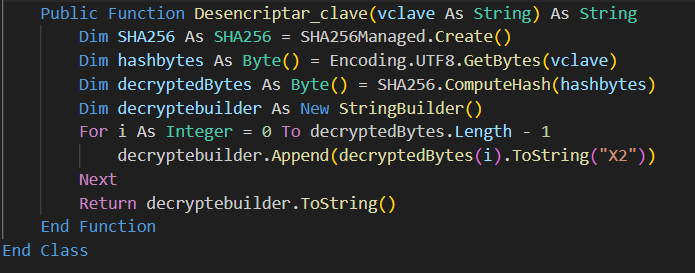


**Explicación**:

* **Creación de SHA-256**: Crea una instancia del algoritmo SHA-256.
* **Generación de Hash**: Genera el hash de la contraseña y lo convierte en una cadena hexadecimal.
* **Método Desencriptar\_clave**

Este método (incorrectamente nombrado) recalcula el hash de una contraseña.

Figura12   
Codigo Desencriptar Clave



**Explicación**:

* **Creación de SHA-256**: Crea una instancia del algoritmo SHA-256.
* **Recalculo de Hash**: Genera nuevamente el hash de la contraseña y lo convierte en una cadena hexadecimal. (Nota: Este método no desencripta, solo recalcula el hash).

# Conclusión

El desarrollo de aplicaciones**No index entries found.** que interactúan con bases de datos es una tarea esencial para muchas organizaciones que dependen de la integridad y la seguridad de los datos. A lo largo de este documento, se ha proporcionado una guía técnica detallada sobre la clase Conexion en Visual Basic, que permite la conexión a una base de datos SQL Server y la ejecución de diversas operaciones cruciales, como la validación de usuarios y la encriptación de contraseñas.

La clase Conexion incluye métodos clave para abrir y cerrar conexiones, validar correos electrónicos e identificaciones, realizar consultas e insertar datos en la base de datos. Cada uno de estos métodos ha sido explicado minuciosamente, destacando su importancia y funcionamiento dentro del contexto de la gestión de datos. Además, se han incluido capturas de pantalla del entorno de desarrollo y del código en ejecución para proporcionar una referencia visual que facilite la comprensión de los conceptos técnicos.

Este documento no solo pretende ofrecer una comprensión detallada del código, sino también servir como un recurso práctico para desarrolladores avanzados y estudiantes de informática. Al comprender y aplicar los métodos descritos, los desarrolladores pueden asegurar la integridad y seguridad de los datos en sus propias aplicaciones, adoptando prácticas de programación robustas y seguras.

La clase Conexion es una herramienta poderosa que facilita la gestión de bases de datos en aplicaciones desarrolladas en Visual Basic. La correcta implementación y comprensión de sus métodos son fundamentales para garantizar operaciones eficientes y seguras. Este manual técnico, presentado en formato APA 7ma edición, busca ser una referencia invaluable para aquellos que buscan profundizar en el desarrollo de aplicaciones con Visual Basic y la integración con bases de datos SQL Server, promoviendo prácticas óptimas en la gestión de datos.