* Portada bien elaborada
* Tabla de contenido generada automáticamente
* Capítulo I
* 1.1 Introducción
* 1.2 Problema investigado
* 1.3 Justificación
* 1.4 Objetivos de la investigación (un objetivo general y por lo menos 3 específicos)
* Capítulo II
* **Marco teórico /base teórica** (Distribuido en subtítulos)
* 2.1 Información sobre base de datos (Gestor utilizado, ya sea SQL Server o MySQL)

**SQL Server:**

SQL Server es un sistema de gestión de bases de datos relacionales (RDBMS) desarrollado por Microsoft Corporation. Es una herramienta de software que permite a los usuarios almacenar, administrar y recuperar datos en una base de datos.

SQL Server utiliza el lenguaje SQL (Structured Query Language) para administrar las bases de datos y proporciona una interfaz gráfica de usuario (GUI) para realizar tareas de administración, como la creación de bases de datos, tablas, vistas y procedimientos almacenados.

SQL Server es muy utilizado en entornos empresariales y de negocios, donde se requiere una gran cantidad de datos y una alta capacidad de procesamiento y almacenamiento. También es popular entre los desarrolladores de aplicaciones que necesitan una base de datos confiable y escalable para sus aplicaciones.

SQL Server se ejecuta en sistemas operativos Windows y está disponible en diferentes versiones, desde la versión gratuita SQL Server Express hasta la versión empresarial SQL Server Enterprise.

**Ventajas:**

1. Escalabilidad: SQL Server es capaz de manejar grandes cantidades de datos y puede escalar para satisfacer las necesidades de crecimiento de una empresa.
2. Integración con otras herramientas de Microsoft: SQL Server se integra fácilmente con otras herramientas de software de Microsoft, como Visual Studio y Excel.
3. Seguridad: Cuenta de SQL Server
4. Alta disponibilidad: SQL Server ofrece opciones de alta disponibilidad, como la replicación y la agrupación de conmutación por error, para minimizar el tiempo de inactividad y mantener la disponibilidad de los datos.

**Desventajas:**

1. Costo: SQL Server puede ser costoso en comparación con otros sistemas de gestión de bases de datos.
2. Requerimientos de hardware: SQL Server requiere hardware potente para ejecutarse de manera efectiva, lo que puede ser un desafío para empresas más pequeñas con recursos limitados.
3. Licencias y restricciones de uso: Las licencias de SQL Server pueden ser restrictivas y limitar el uso del software en determinadas circunscripciones
4. No es compatible con todos los sistemas operativos: SQL Server sólo se ejecuta en sistemas operativos de Microsoft, lo que puede ser un problema para empresas que utilizan diferentes sistemas operativos en su infraestructura.

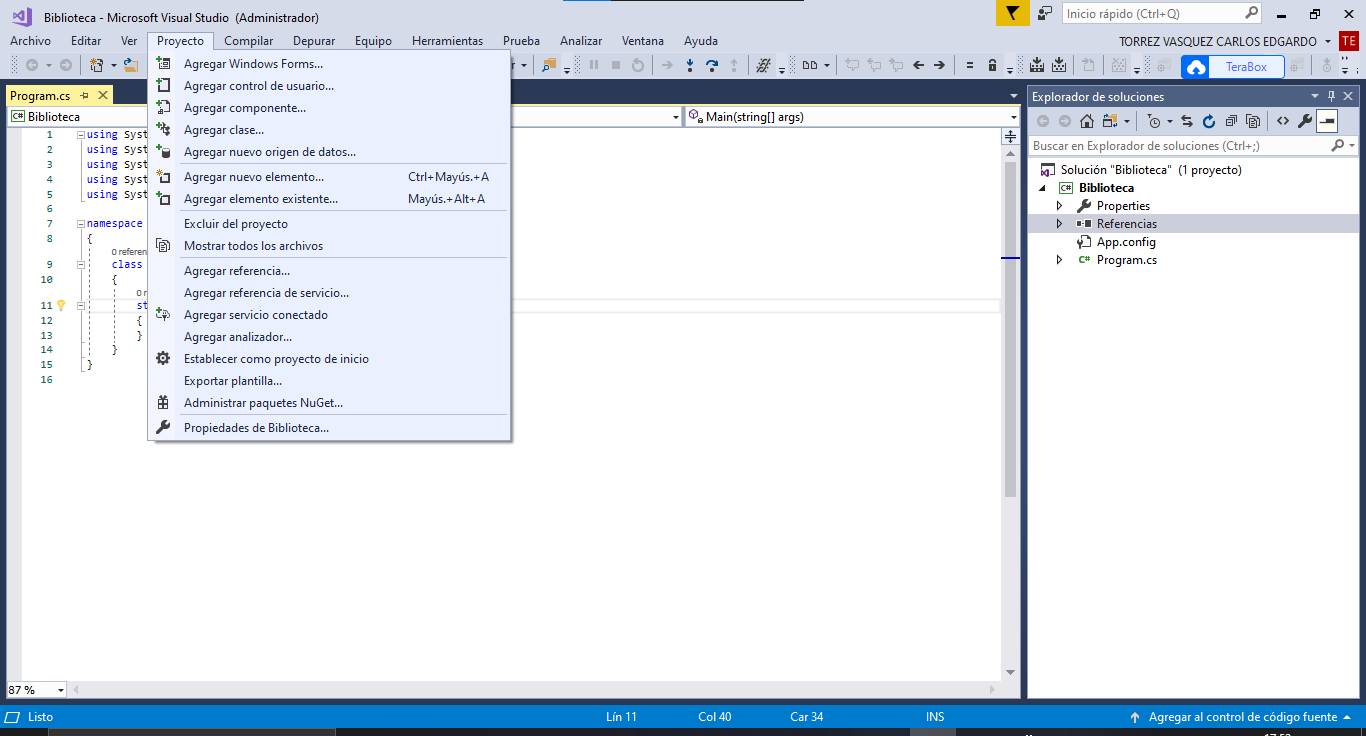
**COMO CONECTAR A UNA BASE DE DATOS EN SQL SERVER DESDE VISUAL STUDIO**

1. Interfaz de usuario gráfica, Aplicación

   Descripción generada automáticamenteAbrir Visual Studio
2. Crear o abrir un proyecto

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

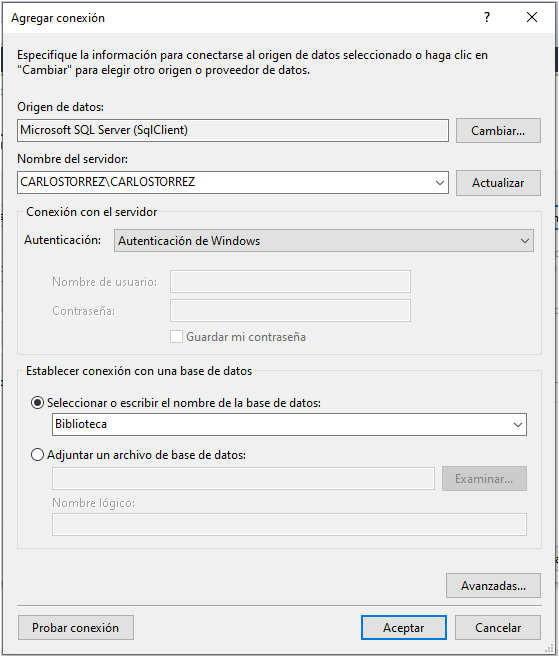
1. Seleccionamos orígenes de Datos o agregar nuevo origen de Datos
2. Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamenteSeleccionamos la opción Base de Datos, luego siguiente.
3. Seleccionamos Conjuntos de Datos y luego presionamos siguiente.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

1. Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamenteLuego seleccionamos nueva conexión
2. Agregamos el nombre de nuestro servidor de SQL server y seleccionamos la Base de Datos a Utilizar, presionamos aceptar
3. Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

   Descripción generada automáticamenteInterfaz de usuario gráfica, Aplicación

   Descripción generada automáticamenteElegimos los objetos de la Base de Datos, tablas y Procedimientos Almacenados.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

**Que son los procedimientos almacenados**

Los procedimientos almacenados son un tipo de objeto de base de datos en sistemas de gestión de bases de datos relacionales, como SQL Server, que permiten definir una serie de instrucciones SQL y lógica de programación en un único bloque de código. Estos bloques de código pueden ser almacenados en la base de datos y reutilizados varias veces, lo que los hace muy útiles para mejorar el rendimiento y la eficiencia de las aplicaciones.

Los procedimientos almacenados pueden recibir parámetros de entrada y/o de salida, lo que permite hacerlos muy flexibles y personalizables. Además, los procedimientos almacenados pueden ser utilizados para encapsular lógica de negocio y mantener la consistencia de los datos, lo que ayuda a garantizar la integridad y seguridad de la información almacenada en la base de datos.

**Para que son los procedimientos almacenados**

Los procedimientos almacenados tienen múltiples propósitos y pueden ser utilizados para diversas tareas dentro de una base de datos. Algunos de los usos más comunes de los procedimientos almacenados son:

1. Mejorar el rendimiento de la base de datos: Al compilar una serie de instrucciones SQL en un procedimiento almacenado, se puede reducir la cantidad de tiempo necesario para analizar, compilar y optimizar las consultas, lo que puede mejorar significativamente el rendimiento de la base de datos.
2. Simplificación
3. Garantizar la integridad y seguridad de los datos: Los procedimientos almacenados pueden ser utilizados para realizar operaciones complejas y/o repetitivas en la base de datos, lo que ayuda a garantizar la integridad y seguridad de la información almacenada.
4. Reutilización
5. Personalización de resultados: Los procedimientos almacenados pueden recibir parámetros de entrada y/o salida, lo que permite personalizar los resultados de las consultas SQL para satisfacer necesidades específicas de los usuarios.

En resumen, los procedimientos almacenados son una herramienta poderosa y versátil en la gestión de bases de datos y pueden utilizarse para mejorar el rendimiento, simplificar la lógica de programación, garantizar la integridad y seguridad de los datos, reutilizar código y personalizar los resultados de las consultas SQL.