

**Fundamentos de Bases de Datos**

**Proyecto II Bimestre**

**Tema:** Administrador para alquiler de Vehículos

**Integrante:**

* Luis Enrique Pérez Señalin

## Índice:

1. Introducción
2. Descripción del proyecto
3. Diseño de la base de datos
4. Desarrollo del sistema

4.1 Conexión a la base de datos.

4.2 Flujo del programa

4.3 Navegación por el sistema - Tabs

4.4 Funciones de la ventana Registros

4.5 Funciones de la ventana Actualizar datos

4.6 Funciones de la ventana Insertar datos

4.7 Funciones de la ventana Eliminar datos

1. Prueba del sistema.
2. Conclusiones

## Introducción

Se necesita un sistema que permita manejar los registros de un local de alquiler de coches, para esto se utilizó una base de datos de SQL Server, un sistema de Windows Form en C# usando Visual Studio, extra se utilizó Git para control de versiones con el fin de mantener copias de seguridad y un seguimiento de los cambios realizados en el proyecto.

## Descripción del proyecto

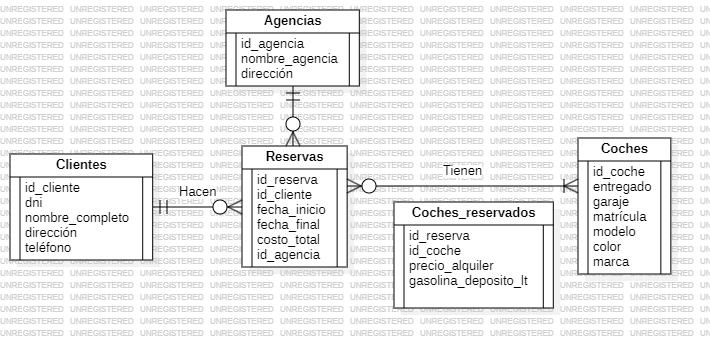
El sistema tiene como fin gestionar las reservas de coches, permitiendo ver los datos actuales en la base de datos y modificarlos.

**Funcionalidad:**

* Inserción de registros.
* Actualización de registros.
* Eliminación de registros.
* Consulta de registros.

## Diseño de la base de datos

Se utilizó el siguiente esquema de entidad relación en la base de datos:



En este esquema de entidad relación tenemos lo mismo que en el proyecto del primer bimestre, con 4 tablas en total, la relación entre las tablas y sus distintos atributos.

La base de datos está alojada en SQL server.

## Desarrollo del sistema

### 4.1 Conexión a la base de datos.

La base de datos se trabaja en Local, en el DBMS Sql Server.

Texto

Descripción generada automáticamente

Esta se realiza dentro de la función **public MainForm()** para que al iniciar el Formulario Main, automáticamente de conecte al servidor, le damos los valores del Data Source, el nombre de la base de datos y que permita le seguridad integrada.

### 4.2 Flujo del programa

El programa inicia con una pantalla de Login, lo que permite añadir seguridad, este es un Form llamado LoginForm.cs.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Luego el botón oculta este Form y abre el MainForm.cs.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Este sistema se maneja con “Tabs”, los cuales permiten moverse entre “Registros”, “Actualizar datos”, “Insertar datos” y “Eliminar datos”.

### 4.3 Navegación por el sistema - Tabs

El sistema permite moverse entre distintas vistas por medio de los Tabs, aquí tenemos los ejemplos de cada Tabs.

* Tab - Registros.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Tab - Actualizar datos.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

* Tab – Insertar datos.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* Tab Eliminar datos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

*Cada uno de estos Tabs tienen otros Tabs para moverse entre las distintas tablas de la base de datos, esto nos permite un flujo claro de como el programa se debe utilizar*

### 4.4 Funciones de la ventana Registros

La ventana Registro tiene como función principal permitir la visualización de los datos registrados en la base de datos de todas las tablas.

Utilizando un Dropbox que tiene como opciones: Agencias, Clientes, Coches y Reservas, se realiza la consulta a la base de datos y el resultado se muestra en un **DataGridView.**

Para realizar una consulta a la base de datos y ver el resultado basta con seleccionar una opción en el Dropbox.

1. Consultar las Agencias.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

1. Consultar los clientes.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

1. Consultar los coches.

Tabla

Descripción generada automáticamente

1. Consultar las reservas

Tabla

Descripción generada automáticamente

**Ejecutar consultas.**

Para ejecutar las consultas utilizamos la siguiente función:

Texto

Descripción generada automáticamente

Vamos a extraer el valor del dropBox y en base a eso guardaremos la query en una variable que luego ejecutaremos y por último mostraremos en el DataGridView.

Texto

Descripción generada automáticamente

Esta es la función que nos da el DropBox para ejecutar, cuando se selecciona una opción en el Dropbox

Estás son las consultas para cada una de las opciones:

* Clientes:

Texto

Descripción generada automáticamente

* Reservas:

Texto

Descripción generada automáticamente

* Agencias:

Texto

Descripción generada automáticamente

* Reservas:

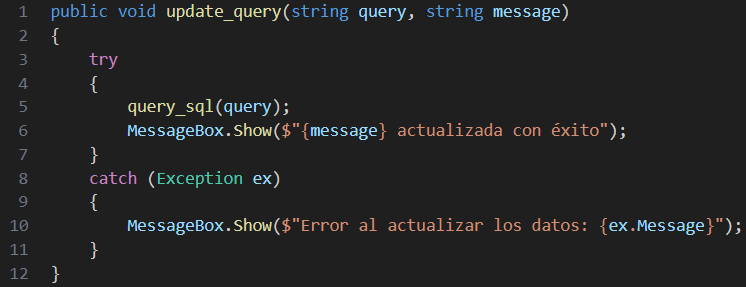
Texto

Descripción generada automáticamente

### 4.5 Funciones de la ventana Actualizar datos

En esta ventana tenemos otro Tab de 4 opciones, una para cada tabla, cada una de estas tiene una interfaz distinta donde pondremos distintos datos para realizar la modificación.

La función update\_query es utilizada para ejecutar las consultas SQL para actualizar los datos. Recibe los parámetros “query” y “message”.



1. Actualizar Agencia.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

En actualizar datos tenemos los campos vistos en la pantalla, usamos el campo “antiguo nombre de la agencia” para poder identificar la agencia que se va a actualizar.

Este es el código para la actualización:

Texto

Descripción generada automáticamente

Los nombres de los text\_box son lo más descriptivos posibles, tienen primero el nombre del tab ej.: “update”, ”insert”, ”delete” y el nombre de la tabla “agencias”, ”clientes”, ”reservas”, “vehículos” y por último el campo que hace referencia.

1. Actualizar clientes.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para actualizar los datos se usa el DNI del cliente, porque los nombres se pueden repetir, luego se añaden los campos nuevos, no se puede cambiar el DNI porque su caso sería extraño, una vez llenados los datos se ejecuta la función **button\_update\_client\_Click.**

**Texto

Descripción generada automáticamente**

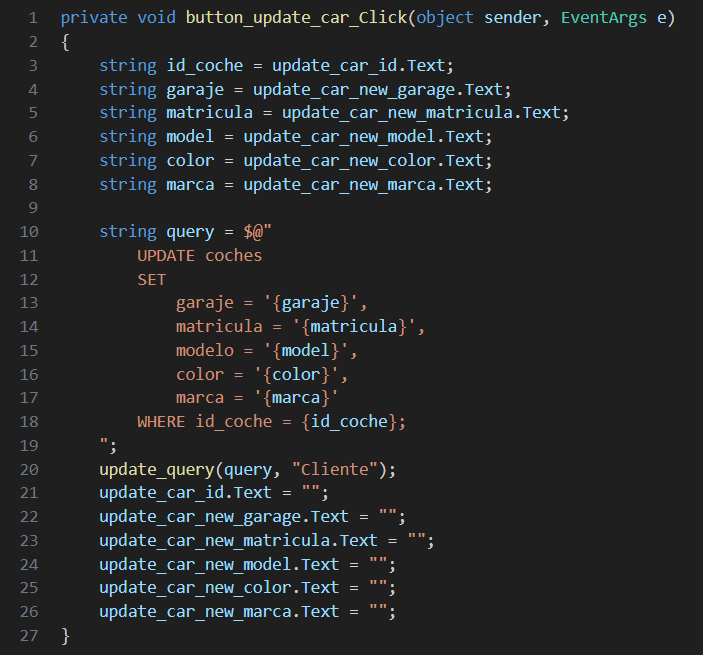
Se utiliza la misma función de **update\_query** para ejecutar la sentencia SQL y para obtener el id del “avalador” realizamos una subconsulta utilizando el dni del avalador, de esta manera no se necesita conocer el id del avalador, sino solo el DNI.

1. Actualizar coches.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para actualizar la tabla de coches se utiliza el id de coche para que se pueda modificar la matrícula del vehículo, después de llenar todos los campos se ejecuta la función **button\_update\_car\_Click.**



1. Actualizar reservas

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Para actualizar las reservas se utiliza como identificador el id, el cuál es usuario puede obtener de la parte de registros, con el fin de poder modificar todos los campos de la reserva que se desea actualizar, se utiliza la función **button\_update\_reservas\_Click.**

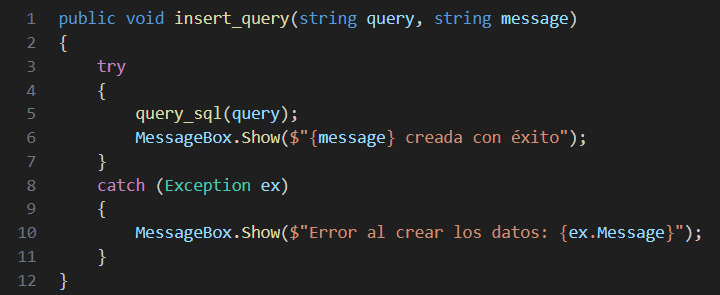
Texto

Descripción generada automáticamente

### Funciones de la ventana Insertar datos

Esta ventana tiene como función insertar nuevos datos dentro de las tablas correspondientes en la base de datos. En esta ventana contamos con las mismas 4 pestañas que tiene “Actualizar datos”, para poder seleccionar que dato vamos a crear.

Para esto tenemos la función **insert\_query.**



A continuación, vamos a revisar cada una de las pestañas con las funciones que tienen:

1. Insertar Agencia

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Esta pestaña solo tiene los campos de nombre de la agencia y dirección que se utilizan para insertar una nueva agencia, se utiliza la función **insert\_agencia\_Click.**

Texto

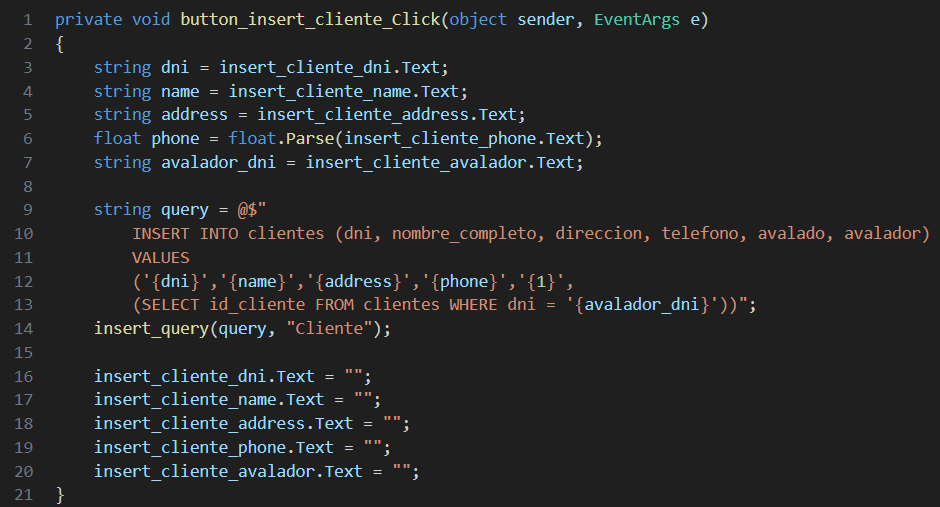
Descripción generada automáticamente

1. Insertar Clientes.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Utilizamos todos los campos necesarios para clientes, el único campo que se utiliza para una **subconsulta** es “DNI Avalador” por el cuál se va a obtener el id del avalador y se insertará el cliente con dicho valor en el campo id avalador, utilizamos la función **button\_insert\_cliente\_Click.**



En el campo avalador se necesita el id del avalador, por lo que se utiliza el dni del avalador para obtener el valor de su id en la línea 13 para que no se necesite mostrar los id de los clientes, sino solo los dni.

1. Insertar Coches.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Utilizamos todos los valores necesarios para crear un nuevo coche en la tabla de Coches, en este caso no utilizamos las subconsultas, y el valor de disponible está en 1 automáticamente, utilizamos la función **button2\_Click\_1** debido a una mala generación automática de la función.

Texto

Descripción generada automáticamente

La función utiliza todos los valores dados y pone en la sentencia SQL, solo el valor de “entregado” se pone en 1 por defecto.

1. Insertar Reservas.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Utilizamos todos los campos para insertar los datos excepto los campos de “DNI cliente” y “Nombre de agencia”, los cuales usamos para hacer una **subconsulta** para obtener el id del cliente y de la agencia y crear la reserva utilizando estos valores, para crear esta reserva utilizamos la función **button\_insert\_reserva\_Click.**

Texto

Descripción generada automáticamente

Podemos ver las **subconsultas** usando” dni\_cliente” y “agencia\_name” en las líneas 12 y 16 para obtener los id de cliente y agencia e incluirlos en los datos para crear una reserva.

### 4.7 Funciones de la ventana Eliminar datos

Para eliminar datos en la base de datos utilizamos la pestaña “Eliminar datos”, todas las funciones que se ejecutan dentro de esta ventana utilizan la función **delete\_query**.

Texto

Descripción generada automáticamente

A continuación, veremos las pestañas de cada tabla:

1. Eliminar Agencias.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Para eliminar una agencia solo se necesita el campo “nombre de la agencia” para que al utilizar la función **button\_delete\_agencia\_Click** podamos eliminar la agencia deseada.

Texto

Descripción generada automáticamente

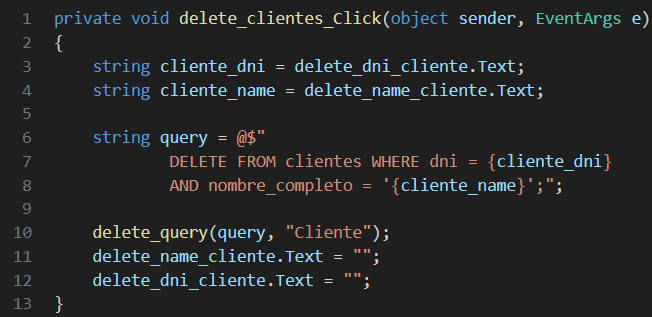
La función obtiene el nombre de la agencia de la interfaz y ejecuta la sentencia SQL para eliminar la agencia de la base de datos.

1. Eliminar Clientes.

Interfaz de usuario gráfica, Aplicación

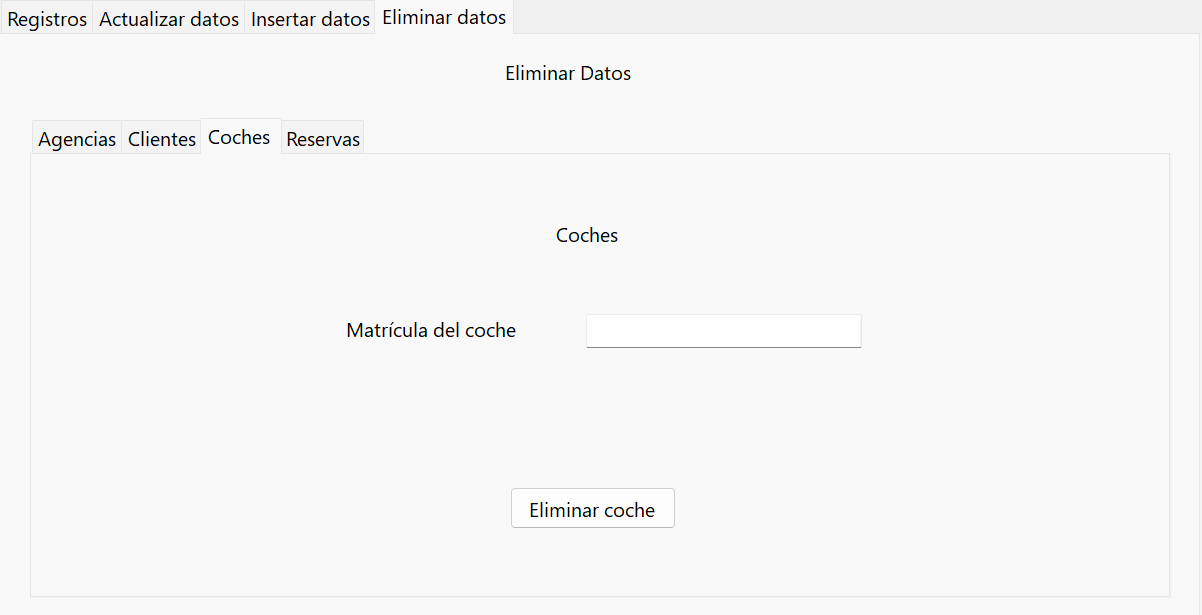
Descripción generada automáticamente

Para eliminar un cliente solo se usa el DNI y el Nombre completo, esto con el fin de tener una confirmación de que sea el cliente indicado y no se copió erróneamente el DNI del cliente que se desea eliminar. Usamos la función **delete\_clientes\_Click.**

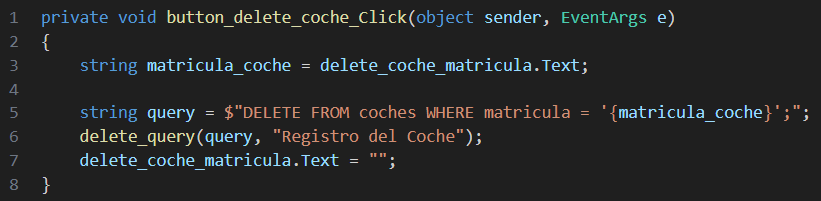


La función ejecuta la sentencia SQL usando los dos valores extraídos de la interfaz, y al final limpia los valores puestos en la interfaz.

1. Eliminar coches.



En la interfaz solo tomamos el valor de “Matrícula de coche” porque es único para cada coche, la razón de no usar el id es para no tener que mostrarlo en la parte de registros con el fin de que sea más fácil la gestión y no depende de un número asignado en orden, sino de uno **Lógico** como el número de matrícula del coche. Usamos la función **button\_delete\_coche\_Click** para la eliminación en la base de datos.



La función toma el valor del número de la matrícula del coche para usarlo en le sentencia SQL y eliminarlo de la base de datos.

1. Eliminación de Reservas.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Al momento de crear la interfaz para eliminar las reservas no se pensaba implementar el id de Reservas para que se usen datos útiles, en este caso se utilizó el DNI cliente, la fecha de inicio y el nombre de la agencia, porque no podría coincidir estos 3 valores en otras Reservas.

Para poder eliminar la reserva se necesita el “id\_cliente” la ”fecha de incio” y “id\_agencia”, para esto realizamos una **subconsulta** dentro de nuestra sentencia SQL. Usamos la función **button3\_Click** porque se generó automáticamente.

Texto

Descripción generada automáticamente

En la sentencia podemos observar las **subconsultas** en la línea 10 y 12 para el “id\_cliente” e “id\_agencia”, utilizando los valores de dni y nombre de la agencia extraídos de la interfaz del usuario

## Prueba del sistema.

Para las pruebas del sistema se realizó una grabación corta donde se hace uso de cada una de las funcionalidades como : “Consultar tabla”, “Insertar datos”, “Actualizar datos” y “Eliminar datos”.

Por medio de este link se puede visualizar la prueba en tiempo real del sistema.

[Presentación Proyecto 2B Fundamentos de base de datos - Enrique Pérez.mp4](https://epnecuador-my.sharepoint.com/:v:/g/personal/luis_perez05_epn_edu_ec/EXeqXGR_vrFCsIFuAm0_DoYBQimsETGcpx-BDYVGg23UEw?nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcml2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=0hS3ZQ)

## Conclusiones

Como conclusiones después del desarrollo, tenemos como principal:

1. Creación del proyecto:

El proyecto tomó tiempo y se tuvo que dedicar bastante a resolver problemas con las consultas SQL y las ejecuciones, pero se logró realizar correctamente

1. Uso del sistema:

Aunque el sistema fue realizado con una finalidad puramente funcional sigue siendo un sistema bastante intuitivo y capaz de permitir una gestión clara de un sistema de ventas o cualquier tipo de sistema comercial de forma aceptable.

1. Oportunidad de mejora:

El sistema tiene muchas oportunidades de mejora, tanto visuales como de interfaz de navegación, también se puede mejorar bastantes operaciones para que el manejador de este sistema sea capaz de usarlos de una manera mucho más rápido, como añadiendo funciones al momento de hacer clic en cada registro y demás.

1. Aprendizaje

Gracias a este proyecto se aprendió bastante el uso de C# y las posibles formas de crear estos sistemas de forma sencilla y rápida.

**Código del proyecto.**

El proyecto realizado se adjunta como un .zip donde contiene todo lo que Visual studio creo para poder funcionar correctamente, se recuerda que la base de datos es distinto del sistema pero se adjunta el schema.sql para crear las tablas y el insert.sql para los datos de prueba.