**CORPORACION UNIVERSITARIA MINUTO DE DIOS**

**ESPECIALIZACION EN DESARROLLO DE SOFTWARE**

**PLATAFORMAS DE DESARROLLO DE SOFTWARE**

**ACTIVIDAD ALTERNATIVA SEMANA 5**

**DOCENTE: JONIER HERNANDO PORRAS DUQUE**

**ESTUDIANTES:**

**ALEJANDRO BOTERO GALINDO**

**JHOHAN ESTIWAR PALACIOS MENA**

Contenido

[Aplicación C# de Registro de citas médicas y Consulta especifica 3](#_Toc137501402)

[Creación del proyecto 3](#_Toc137501403)

[Otras clases 3](#_Toc137501404)

[Desarrollo del diseño Form1.cs [Diseño] 4](#_Toc137501405)

[Desarrollo de la clase ModelCitaMedica 5](#_Toc137501406)

[Desarrollo de la clase AdministratorSQL 6](#_Toc137501407)

[Creación de base de datos y tabla en MySQL Workbench 9](#_Toc137501408)

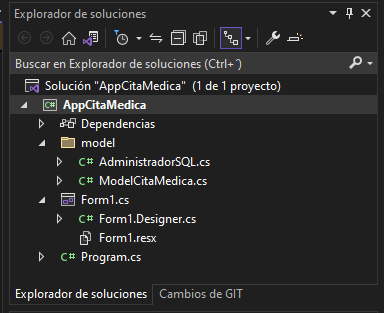
[Desarrollo del código del formulario 9](#_Toc137501409)

[Capturas del programa en funcionamiento 10](#_Toc137501410)

# Aplicación C# de Registro de citas médicas y Consulta especifica

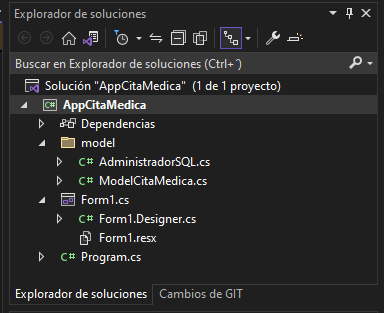
## Creación del proyecto

Para realizar la actividad correspondiente a la semana 5, se procede a crear en Visual Studio un proyecto en C#. Creando una carpeta llamada **model** que contendrá las clases AdministradorSQL y ModelCitaMedica como se puede validar en la siguiente Imagen:



## Otras clases

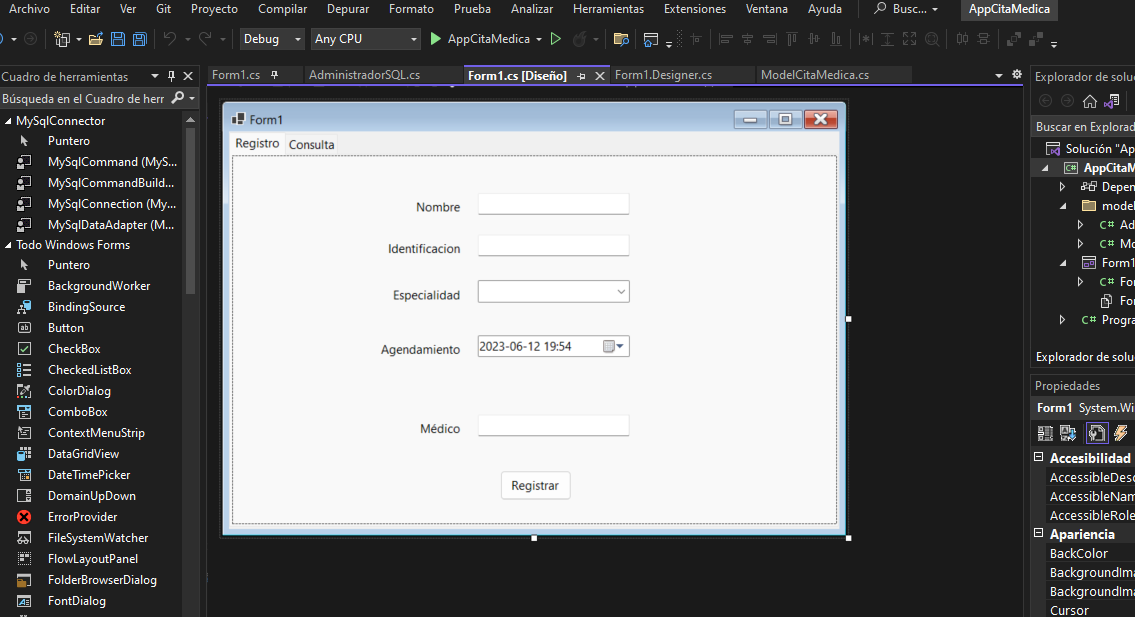
Debido a la naturaleza del proyecto el mismo entorno facilita la clase formulario para proceder a diseñar y codificar el formulario

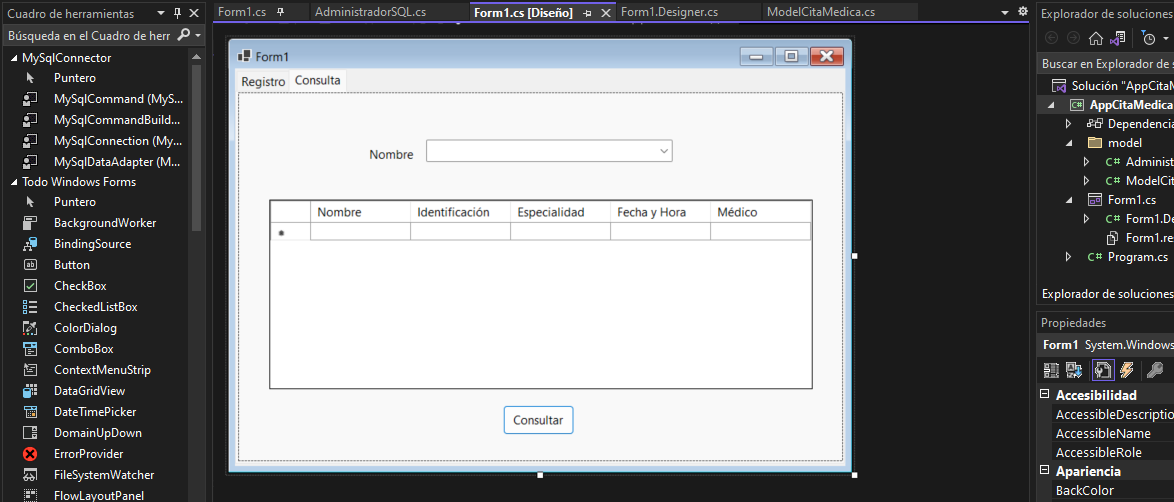


## Desarrollo del diseño Form1.cs [Diseño]

Para el desarrollo se usaron los elementos que el IDE proporciona como son:

1 TabControl, , 6 label, 3 TextBox, 2 Combo Box, 2 Button y 1 DataGrid.

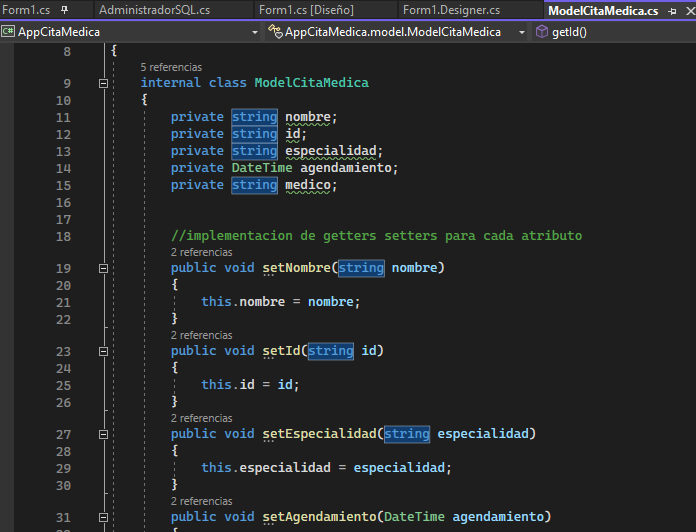




## Desarrollo de la clase ModelCitaMedica

Dentro de la carpeta model se encuentra la clase ModelCitaMedica.cs, en la cual se definen las variables donde se almacenarán los datos provistos por el usuario. Estas variables son: **nombre de tipo String, id de tipo string, especialidad de tipo string, agendamiento de tipo DateTime y medico de tipo String.**

También se implementan los **getter y setter** para cada una de estas variables.

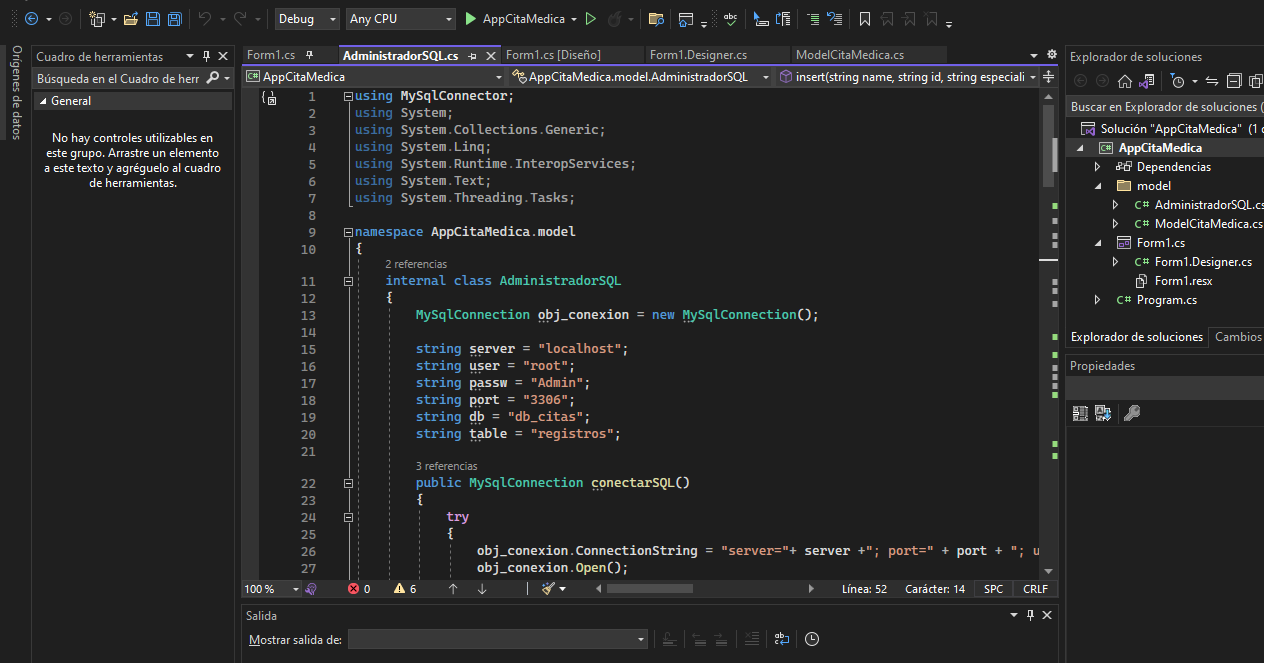


## Desarrollo de la clase AdministratorSQL

Esta clase perteneciente a la carpeta model, se desarrolla con el fin de definir los atributos o variables que son necesarios para realizar la conexión con la base de datos en MySQL Workbench.

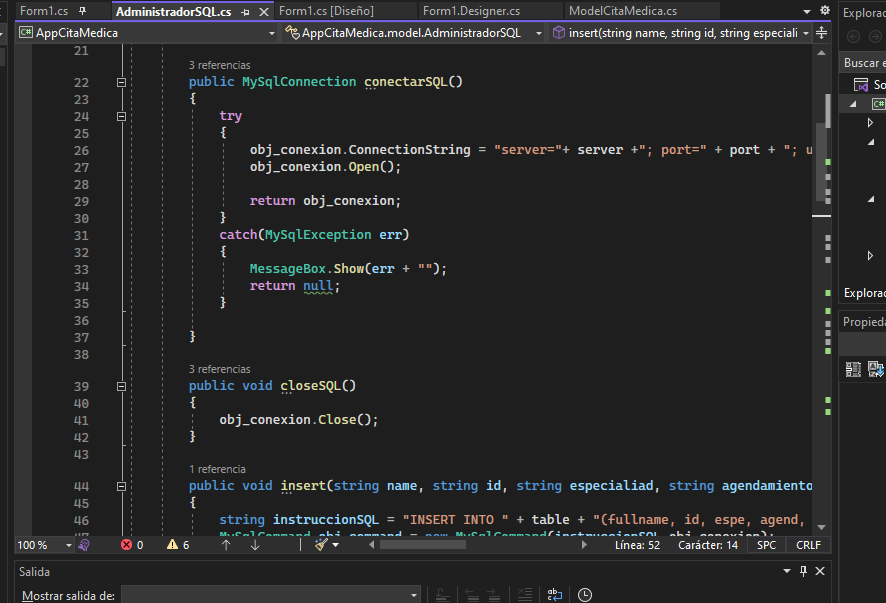
Los métodos desarrollados en esta clase son:

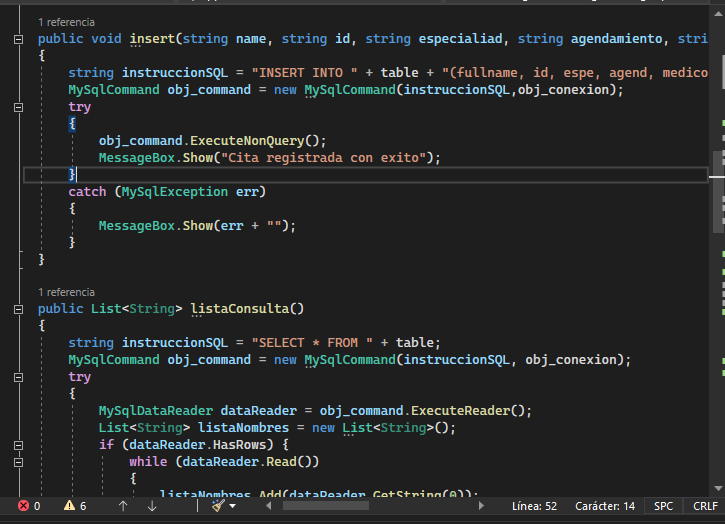
1. conectarSQL, el cual permite realizar la conexión con la base de datos en Workbench MySQL.
2. closeSQL, el cual cierra la conexión establecida con la base de datos
3. insert, se crea una variable **instructionSQL** con el fin de unificar la sentencia SQL y las variables que se deben almacenar en la tabla de la base de datos.
4. ListaConsulta, el cual permite mediante una sentencia SQL, generar en el combobox de la pestaña consulta, un listado de los usuarios previamente registrados. Esto se hace mediante una Lista de tipo String. Esto se implementa dentro de un if ya que es posible que no existan citas registradas.
5. Consulta: se realiza la consulta a la base de datos por medio de una instrucción donde se solicita específicamente el nombre del usuario que tiene la cita (usuarios disponibles en el comboBox). Esto retorna un único registro el cual hemos denominado obj\_cita.

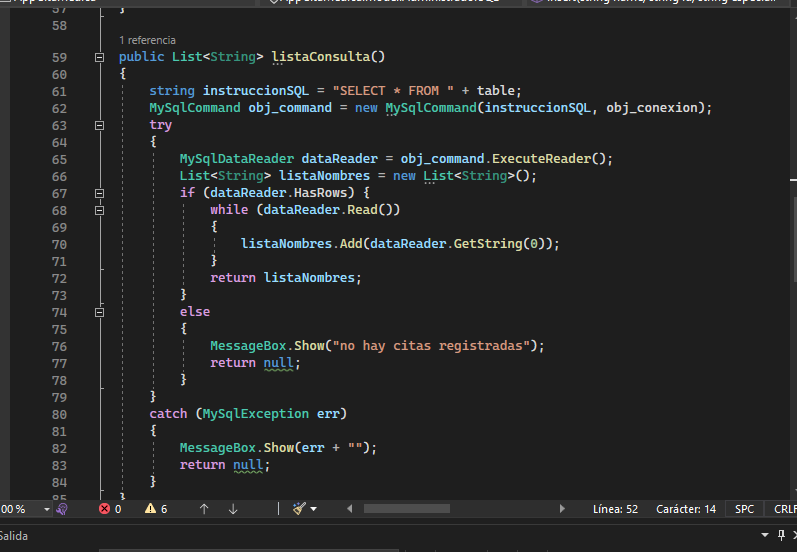


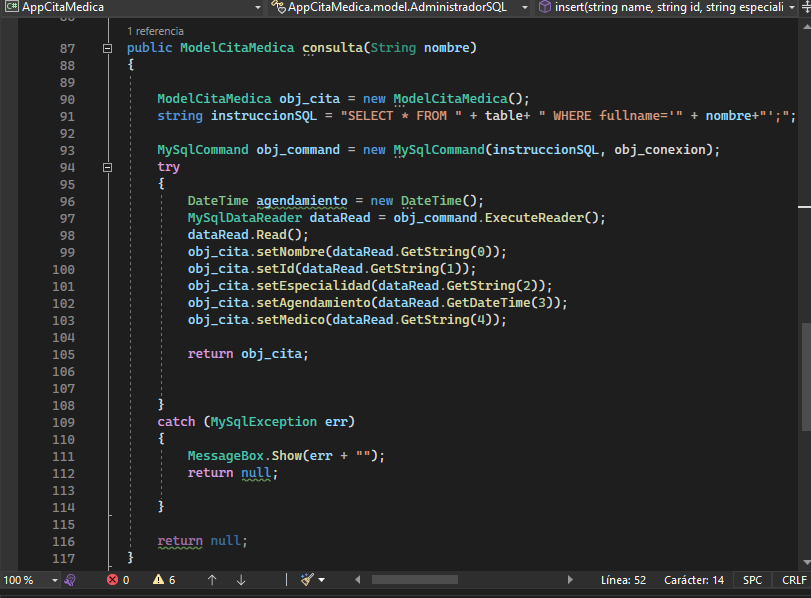
Datos necesarios

para conexion







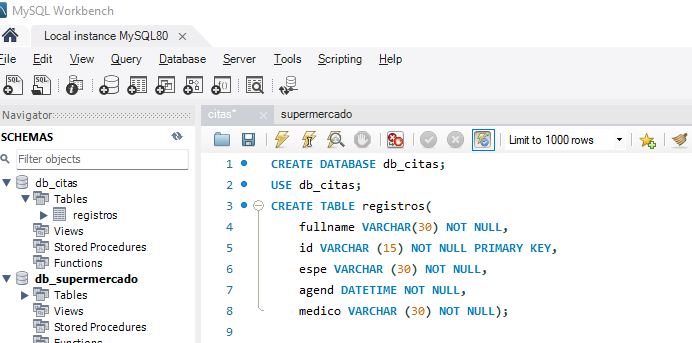


## Creación de base de datos y tabla en MySQL Workbench

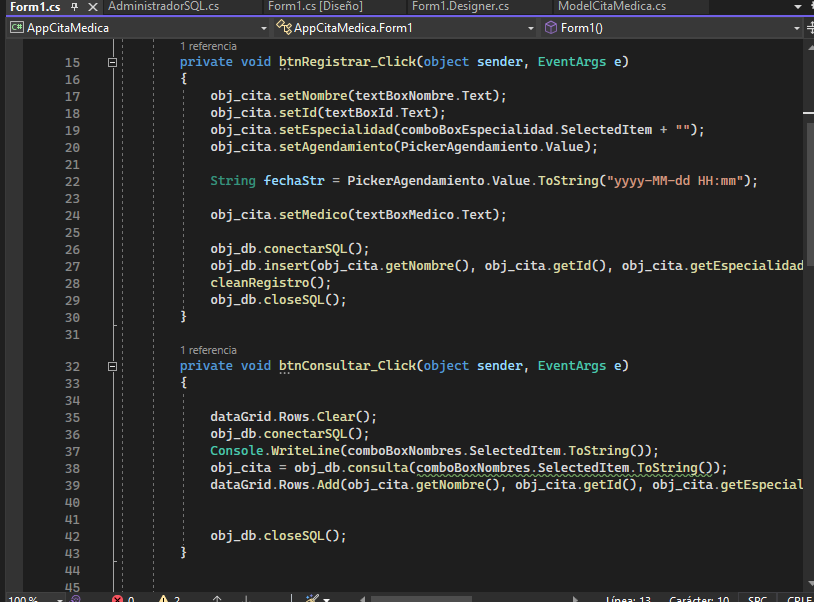
Se procede a crear una base de datos mediante la sentencia SQL **CREATE DATABASE db\_citas;**

Atendiendo las instrucciones del docente, se procede con la sentencia **USE db\_citas,** esto con el fin de que el entorno trabaje por defecto con la nueva base de datos en las siguientes instrucciones.

Posteriormente se procede a crear la tabla llamada registros con la siguiente sentencia:

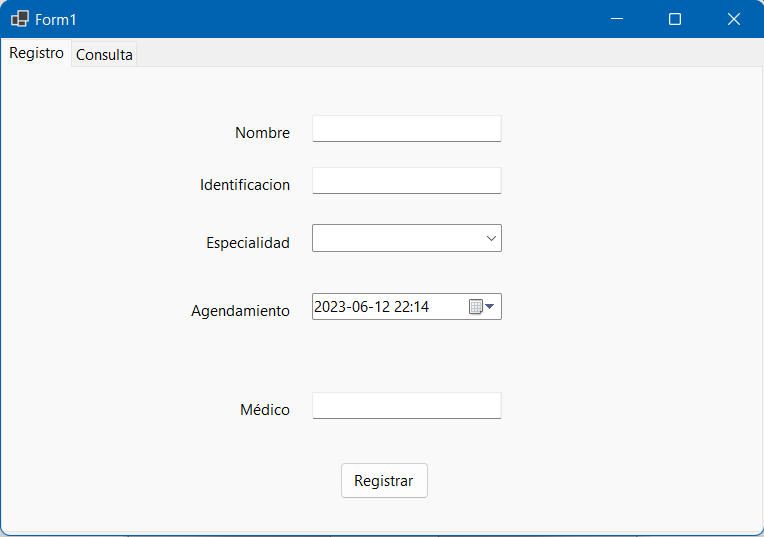


## Desarrollo del código del formulario

se asignan acciones a los botones registrar y consultar. 

## Capturas del programa en funcionamiento

Una vez se corre el programa, en pantalla se presenta el formulario para diligenciar con sus respectivos campos. Se tienen dos pestañas, una para registro y otra para consulta



Nos encontramos con la interfaz correspondiente al módulo registrar, el cual nos permite agendar una cita médica con los datos nombre, identificación, la especialidad que busca, la fecha y hora de la cita y por último el medico que desea.

Para este ejemplo realizaremos el siguiente registro:

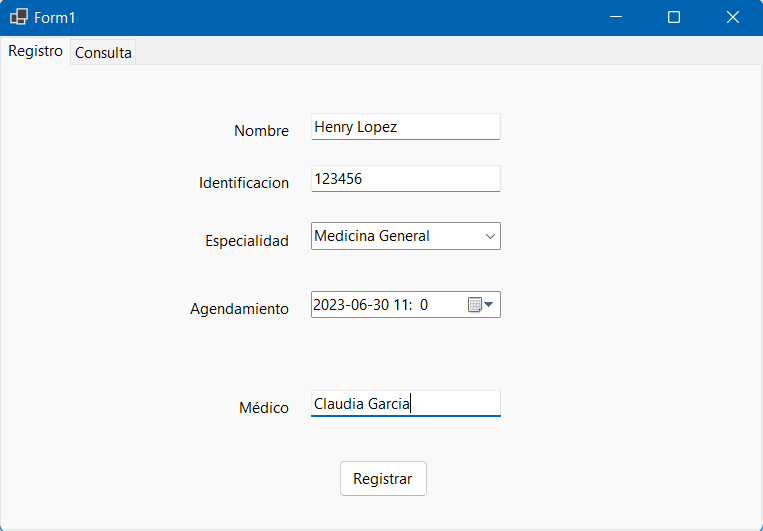
Nombre: Henry López

Identificación: 12345678

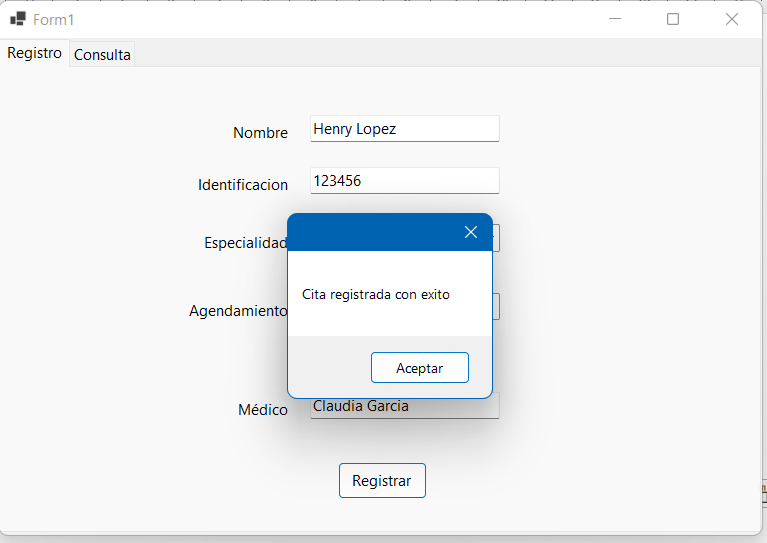
Especialidad: Medicina General

Agendamiento: 30 de junio de 2023 a las 11: 00 de la mañana

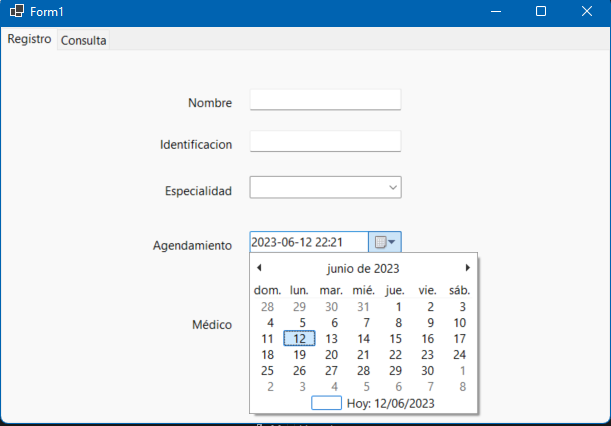
Medico: Claudia García



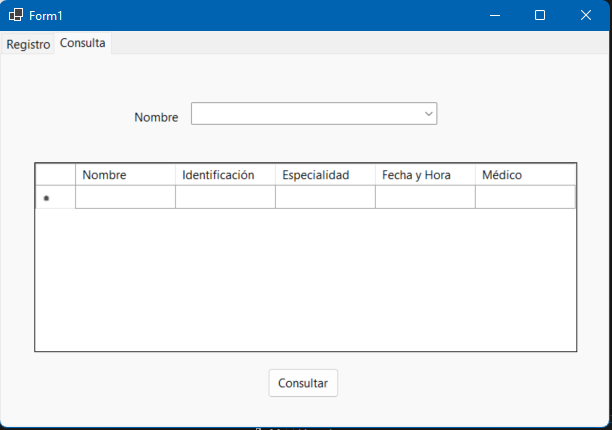
Se procede a oprimir el botón registrar y saldrá un mensaje indicando que el registro ha sido almacenado.



El campo agendamiento al ser un DateTimePicker, despliega el calendario para hacer la selección de la fecha de una manera más intuitiva:



Cuando existan registros en la base de datos, se podrán consultar accediendo a la segunda pestaña del módulo.



En la lista desplegable se podrán encontrar todos los registros existentes, al elegir uno podremos filtrar los datos almacenados y establecer cuáles son los que cumplen con el criterio indicado. Organizados en un grid

