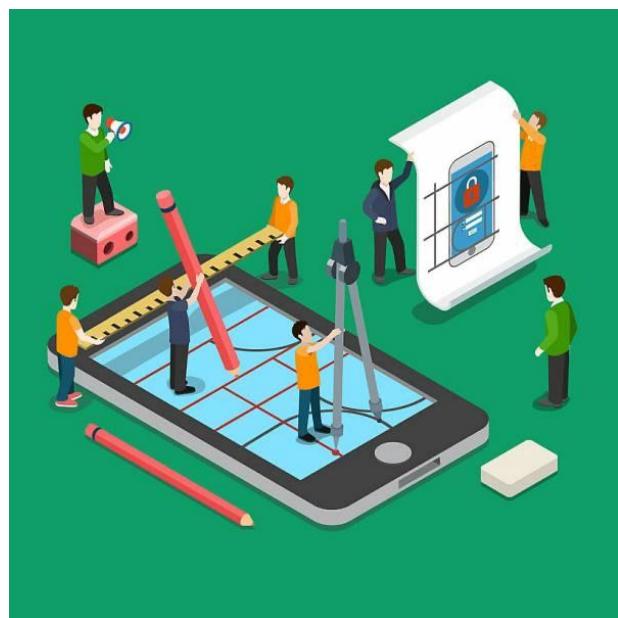




## DESENROTLLAMENT D'INTERFÍCIES 2N DAM

### SA3: Generació d'interfícies gràfiques d'usuari



Professor: Iván Martos



En aquests apunts anem a veure com interactuar i realitzar connexions entre Visual Studio i MySQL o Maria DB per a introduir dades o bé per a realitzar consultes a la base de dades i mostrar els resultats per la interfície de Windows Forms.

En primer lloc, farem ús de la plataforma XAMPP (integra PHP, MySQL o MariaDB, servidor web Apache...). La pàgina oficial de XAMPP és: <https://www.apachefriends.org/es/index.html>

**XAMPP Apache + MariaDB + PHP + Perl**

**¿Qué es XAMPP?**

XAMPP es el entorno más popular de desarrollo con PHP

XAMPP es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

**Descargar**  
Pulsa aquí para otras versiones

**XAMPP para Windows**  
8.1.12 (PHP 8.1.12)

**XAMPP para Linux**  
8.1.12 (PHP 8.1.12)

**XAMPP para OS X**  
8.1.12 (PHP 8.1.12)

En aquest xicotet tutorial crearem una base de dades anomenada provees i disposarà d'una taula de cotxes amb dos camps bàsics (idCotxe i marca). L'identificador dels cotxes serà un enter i serà auto incremental.

✓ Los cambios en la Tabla cotxes se hicieron exitosamente.

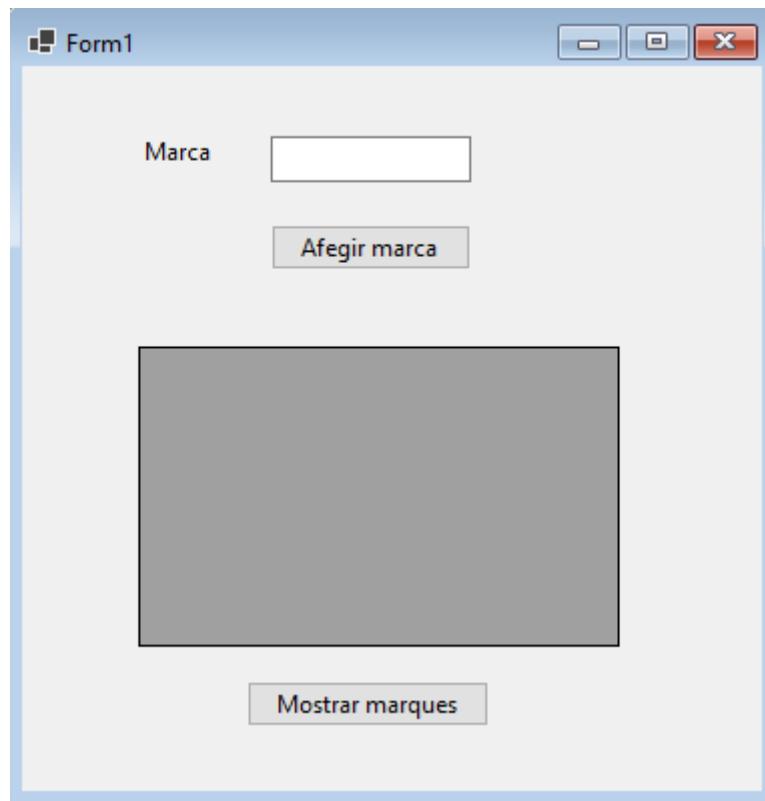
```
ALTER TABLE `cotxes` CHANGE `idCotxe` `idCotxe` INT(11) NOT NULL AUTO_INCREMENT, add PRIMARY KEY (`idCotxe`);
```

[ Editar en línea ] [ Editar ] [ Crear código PHP ]

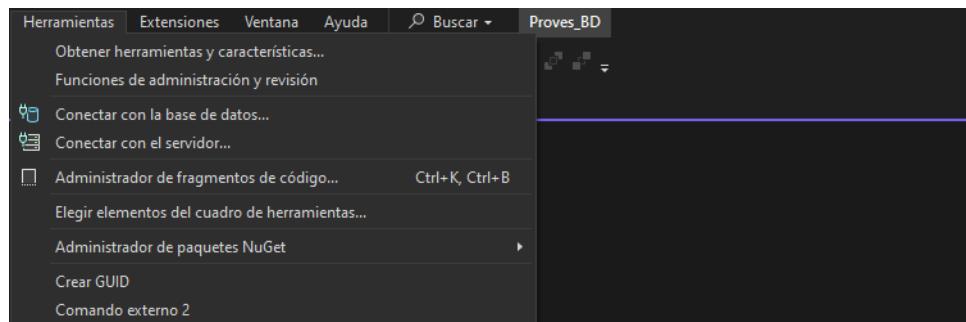
Estructura de tabla Vista de relaciones

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
<input type="checkbox"/>	1 idCotxe	int(11)			No	Ninguna		AUTO_INCREMENT	Cambiar  Eliminar  Más
<input type="checkbox"/>	2 marca	varchar(50)	utf8mb4_general_ci		No	Ninguna			Cambiar  Eliminar  Más

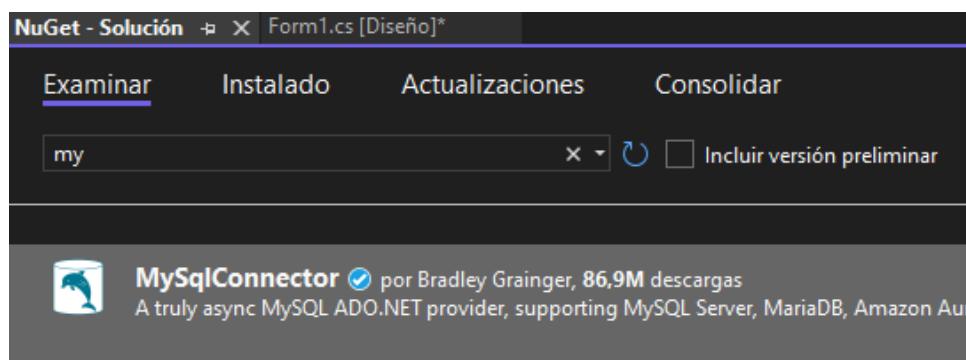
A continuació regressem a Visual Studio i creem un nou projecte de Windows Forms, podeu anomenar-lo com vullgau. El projecte tindrà la següent interfície::

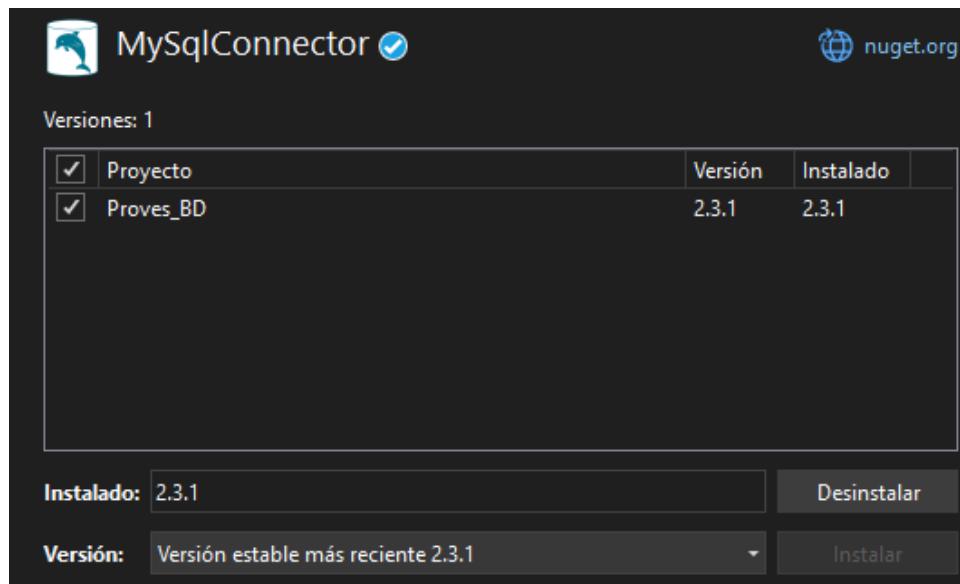


Anem a provar a afegir contingut a la base de dades. Abans de començar, anem a preparar el nostre projecte pera que siga capç de treballar i reconèixer les bases de dades mysql i les seues consultes associades, per això, anem al menú Ferramentes, Administrador de paquets Nuget, Administrar paquets Nuget per a la solució:



En la opció Examinar busquem MySQLConnector, el seleccionem i l'instal·lem:





Ja podem començar a programar la nostra aplicació.

A continuació, anem a crear una nova classe al nostre projecte per a crear la connexió, el seu còdec pot ser el següent:

```
namespace Proves_BD
{
    0 referencias
    internal class conexion
    {
        0 referencias
        public static MySqlConnection conexio()
        {
            string servidor = "localhost";
            string bd = "cotxes";
            string usuario = "root";
            string pass = "";

            string cadenaConexion = "Database=" + bd + "; Data Source=" + servidor + ";"
                + "User Id= " + usuario + ";Password=" + pass;

            MySqlConnection conexionBD;

            try
            {
                conexionBD = new MySqlConnection(cadenaConexion);
                return conexionBD;
            }

            catch (MySqlException e)
            {
                Console.WriteLine("Error en la conexión " + e.Message);
            }
            return null;
        }
    }
}
```

Ja tenim la classe conexio() creada, només em d'anomenar-la quan necessitem obrir una connexió a la base de dades.

A continuació anem a utilitzar el nostre formulari per arreplegar marques de cotxes des del textBox i que es guarden a la base de dades, per això, fem doble clic al botó de Afegir marca i programen el següent còdec:

```

private void button1_Click(object sender, EventArgs e)
{
    // Arrepleguem el text del textBox
    string marcaCotxe = textBox1.Text;
    // Creem la consulta PARAMETRIZADA
    string sql = "INSERT INTO cotxes(marca) VALUES(@marca)";
    // Generem la connexió
    MySqlConnection conexionBD = Conexion.Conexion();
    try
    {
        conexionBD.Open();
        // MySqlComman permetix realitzar operacions sobre la base de dades una vegada està
        // establerta la conexió
        MySqlCommand comando = new MySqlCommand(sql, conexionBD);
        // Definim els paràmetres
        comando.Parameters.AddWithValue("@marca", marcaCotxe);
        // ExecuteNonQuery permetix insertar, actualizar y eliminar dades de la base de dades.
        comando.ExecuteNonQuery();
        textBox1.Text = "";
        MessageBox.Show("Registro guardado correctamente");
    }
    catch (MySqlException ex)
    {
        MessageBox.Show("Error al guardar cliente: " + ex.Message);
    }
    finally
    {
        conexionBD.Close();
    }
}

```

Para finalitzar aquest tutorial, anem a mostrar les dades de la base de dades en el dataGridWiew del nostre form, per això fem doble clic al botó Mostrar marques i programen el següent còdec:

```

private void button2_Click(object sender, EventArgs e)
{
    string consulta = "SELECT * FROM cotxes";
    // Serveix com a pont entre un conjunt de dades i MySQL, permetint recuperar i desar dades.
    // https://dev.mysql.com/doc/dev/connector-
    net/latest/api/data_api/MySQL.Data.MySqlClient.MySqlDataAdapter.html
    MySqlDataAdapter adapter = new MySqlDataAdapter(consulta, Conexion.conexion());
    // Representa una taula de dades en memòria
    DataTable dt = new DataTable();
    adapter.Fill(dt);
    dataGridView1.DataSource = dt;
}

```

Ja deurien mostrar-se les dades de la taula cotxes al dataGridWiew.