

Manual técnico SQLiteConnect

Función

Esta es una librería DLL desarrollada en C# para poder hacer las conexiones a una base de datos SQLite facilmente, ya que nos ofrece métodos bastantes sencillos de llamar sin utilizar queries tan específicos. Esta librería trabaja a base de la DLL System.Data.SQLite, ya que está es la que nos permite realizar las conexiones y las consultas a la propia base de datos. SQLiteConnect nos sirve para poder hacer inserciones, actualizaciones, eliminaciones y consultas a la base de datos y nos brinda una forma fácil de adaptar cualquier DataGridView y poder desplegar los datos de alguna consulta.

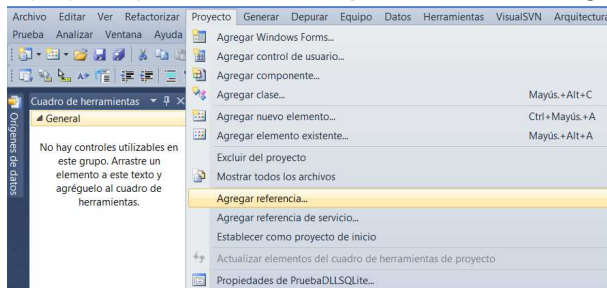
Contenido

Para poder utilizar está librería necesitamos tener en el mismo directorio el DLL System.Data.SQLite, por está razón para poder usarla en cualquier proyecto de Visual Studio solo necesitamos agregar la SQLiteConnect como referencia.

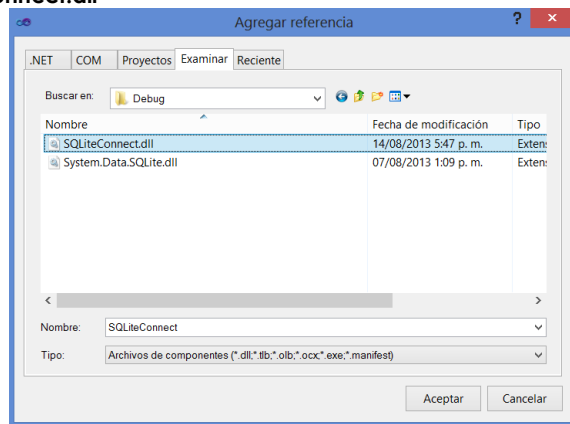
Instrucciones de uso

1. Agregar referencia

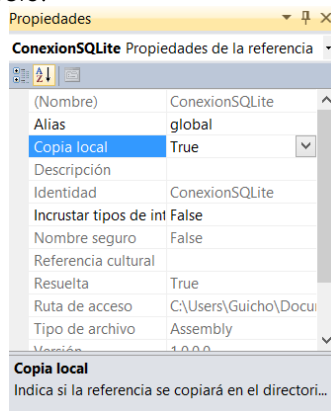
Debemos ir a nuestro proyecto y en el menú de **Proyecto** darle click en **Agregar referencia**.



En la pestaña **Examinar** buscamos la librería, donde la hayamos almacenado, la buscamos con el nombre de **SQLiteConnect.dll**



Ya que la hemos agregado, en el **Explorador de soluciones** en el menú de **Referencias**, seleccionar la librería y en las **Propiedades** verificamos que la opción de **Copia local** tenga **True**. De esta manera nuestra DLL se copiará en nuestro proyecto.



2. Agregar librería a nuestras clases

Ahora, en las clases o en los Forms donde utilizaremos la librería debemos llamarla en nuestra área de librerías. La llamamos con la instrucción **using SQLiteConnect;**

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.Collections;
using SQLiteConnect;
```

3. Instanciar la clase

SQLiteConnect tiene una clase, para poder empezar a usar esta clase donde se encuentran todas las funciones debemos de instanciarla y decirle la ruta exacta donde se encuentra nuestra base de datos.

```
DBConnect db = new DBConnect("campeonato.sqlite");
```

De ahora en adelante podemos llamar a las funciones de la clase DBConnect, donde podremos consultar, insertar, actualizar o eliminar datos.

4. Inserciones (insertar(tabla, datos))

Para poder realizar inserciones debemos de apoyarnos en el método **insertar()**. Este método requiere de dos parámetros, el primero es el nombre de la tabla a donde queremos insertar datos, esta variable es de tipo **string**. El segundo parámetro es un diccionario de datos, en él debemos de especificar el campo y el valor de este que queremos insertar en la base de datos, este es de tipo **Dictionary<string,string>**.

```
Dictionary<string, string> diccionario = new Dictionary<string, string>();
diccionario.Add("nombre", txt_nombre.Text);
db.insertar("rol", diccionario);
```

5. Actualizaciones (actualizar(tabla, datos, condición))

Este es un método parecido al de inserción, pero necesitamos de tres parámetros para poder llamarlo, el nombre de la tabla, un diccionario de datos donde especificamos que campos vamos a afectar y cuáles serán sus valores y la condición que se tiene que cumplir para que se realice el update.

```
Dictionary<string, string> diccionario = new Dictionary<string, string>();
diccionario.Add("nombre", txt_nombre.Text);
db.actualizar("rol", diccionario, "idrol=1");
```

6. Eliminación (eliminar(tabla,condición))

Este método solo necesita de dos parámetros, la tabla y la condición que se tiene que cumplir para poder realizar el Delete.

```
db.eliminar("rol", "idrol=3");
```

7. Consulta (consultar(query))

Este método nos devuelve una **ArrayList**, y dentro de este encontramos cada uno de los registros devueltos por la consulta dentro de un **Dictionary<string,string>**. Esta consulta generalmente es usada para cuando utilizaremos los datos devueltos de la consulta fuera de un DataGridView. Los keys de los Dictionary estarán relacionados con el nombre de los campos que hayamos consultados, de esta manera será más fácil la forma de buscar algún valor específico dentro del diccionario.

```
ArrayList array = db.consultar("select nombre from usuario");  
foreach (Dictionary<string, string> dict in array)  
{  
    Console.WriteLine(dict["nombre"]);  
}
```

8. Consulta a DataGridView (consultarGrid(query))

Este método nos es útil cuando queremos desplegar los datos de alguna consulta directamente a un **DataGridView**, ya que nos devuelve directamente un objeto tipo DataGridView. Solo debemos de mandar la consulta que deseamos y el devolverá los datos correspondientes. Para poder desplegar los datos solo debemos de igualar los DataSource de nuestro DataGridView y el objeto devuelto por este método.

```
Grid_Rol.DataSource = db.consultarGrid("select * from rol").DataSource;
```