

por favor, dame una documantacion completa de como trabajar una base de datos sqlite con c#

Aquí tienes una documentación completa y estructurada para trabajar con bases de datos SQLite en C#:

1. Introducción a SQLite con C#

SQLite es una base de datos ligera, basada en archivos, que no requiere servidor y es ideal para aplicaciones pequeñas y medianas. En C#, se puede usar a través de diferentes enfoques como ADO.NET, Entity Framework Core, Dapper, entre otros^{[1] [2] [3]}.

2. Instalación y configuración

- Instala el paquete NuGet oficial `Microsoft.Data.Sqlite` para usar SQLite en tu proyecto C#:

```
dotnet add package Microsoft.Data.Sqlite
```

- Incluye la directiva en tu código:

```
using Microsoft.Data.Sqlite;
```

Esto te permitirá usar las clases para conexión, comandos y lectura de datos^{[2] [3]}.

3. Crear y conectar a una base de datos SQLite

SQLite usa archivos para almacenar la base de datos. Puedes crear y conectar a un archivo `.db` así:

```
var connectionString = "Data Source=miBaseDeDatos.db";  
using var connection = new SqliteConnection(connectionString);  
connection.Open();
```

Si el archivo no existe, SQLite lo crea automáticamente al abrir la conexión^{[2] [3]}.

4. Crear tablas

Para crear tablas, usa comandos SQL con SqlCommand:

```
var createTableCmd = connection.CreateCommand();
createTableCmd.CommandText =
@"
CREATE TABLE IF NOT EXISTS Estudiantes (
    Id INTEGER PRIMARY KEY AUTOINCREMENT,
    Nombre TEXT NOT NULL,
    Carrera TEXT,
    Universidad TEXT,
    Estado TEXT
);
";
createTableCmd.ExecuteNonQuery();
```

5. Operaciones CRUD básicas

Insertar datos

```
var insertCmd = connection.CreateCommand();
insertCmd.CommandText =
@"
INSERT INTO Estudiantes (Nombre, Carrera, Universidad, Estado)
VALUES ($nombre, $carrera, $universidad, $estado);
";
insertCmd.Parameters.AddWithValue("$nombre", "Juan Pérez");
insertCmd.Parameters.AddWithValue("$carrera", "Ingeniería");
insertCmd.Parameters.AddWithValue("$universidad", "Universidad X");
insertCmd.Parameters.AddWithValue("$estado", "Activo");
insertCmd.ExecuteNonQuery();
```

Consultar datos

```
var selectCmd = connection.CreateCommand();
selectCmd.CommandText = "SELECT Id, Nombre, Carrera FROM Estudiantes;";
using var reader = selectCmd.ExecuteReader();
while (reader.Read())
{
    var id = reader.GetInt32(0);
    var nombre = reader.GetString(1);
    var carrera = reader.GetString(2);
    Console.WriteLine($"ID: {id}, Nombre: {nombre}, Carrera: {carrera}");
}
```

Actualizar datos

```
var updateCmd = connection.CreateCommand();
updateCmd.CommandText =
@"
UPDATE Estudiantes
SET Estado = $estado
WHERE Id = $id;
";
updateCmd.Parameters.AddWithValue("$estado", "Graduado");
updateCmd.Parameters.AddWithValue("$id", 1);
updateCmd.ExecuteNonQuery();
```

Eliminar datos

```
var deleteCmd = connection.CreateCommand();
deleteCmd.CommandText = "DELETE FROM Estudiantes WHERE Id = $id;";
deleteCmd.Parameters.AddWithValue("$id", 1);
deleteCmd.ExecuteNonQuery();
```

6. Uso de transacciones

Para asegurar integridad en operaciones múltiples:

```
using var transaction = connection.BeginTransaction();

var cmd = connection.CreateCommand();
cmd.Transaction = transaction;
cmd.CommandText = "INSERT INTO Estudiantes (Nombre) VALUES ($nombre);";
cmd.Parameters.AddWithValue("$nombre", "Ana Gómez");
cmd.ExecuteNonQuery();

transaction.Commit();
```

7. Ejemplo completo mínimo

```
using Microsoft.Data.Sqlite;

class Program
{
    static void Main()
    {
        var connectionString = "Data Source=miBaseDeDatos.db";
        using var connection = new SqliteConnection(connectionString);
        connection.Open();

        var createTableCmd = connection.CreateCommand();
        createTableCmd.CommandText =
```

```

@"
CREATE TABLE IF NOT EXISTS Estudiantes (
    Id INTEGER PRIMARY KEY AUTOINCREMENT,
    Nombre TEXT NOT NULL
);
";
createTableCmd.ExecuteNonQuery();

var insertCmd = connection.CreateCommand();
insertCmd.CommandText = "INSERT INTO Estudiantes (Nombre) VALUES ($nombre);";
insertCmd.Parameters.AddWithValue("$nombre", "Carlos");
insertCmd.ExecuteNonQuery();

var selectCmd = connection.CreateCommand();
selectCmd.CommandText = "SELECT Id, Nombre FROM Estudiantes;";
using var reader = selectCmd.ExecuteReader();
while (reader.Read())
{
    Console.WriteLine($"ID: {reader.GetInt32(0)}, Nombre: {reader.GetString(1)}")
}
}
}

```

8. Recursos adicionales

- Documentación oficial de Microsoft para Microsoft.Data.Sqlite con ejemplos: [Microsoft Docs](#)^[2]
- Tutoriales para operaciones avanzadas y manejo de transacciones con ADO.NET: [SQLite Tutorial en C#](#)^[4]
- Video tutorial paso a paso para crear un CRUD en Visual Studio con C# y SQLite: [YouTube - PicaioVideos](#)^[5]

Esta guía cubre desde la instalación, conexión, creación de tablas hasta operaciones CRUD y transacciones para trabajar eficazmente con SQLite en C#.

**

1. <https://learn.microsoft.com/es-es/shows/on-dotnet/how-do-i-use-csharp-and-dotnet-with-sqlite>
2. <https://learn.microsoft.com/es-es/dotnet/standard/data/sqlite/>
3. <https://www.tutkit.com/es/tutoriales-de-texto/8179-guia-para-principiantes-de-c-sobre-bases-de-datos-sqlite>
4. <https://www.sqlitetutorial.net/sqlite-csharp/>
5. <https://www.youtube.com/watch?v=Si-VYB3LExk>