# **Unidad 5 - Actividad 1**

## Mira Azure

• Vista del código del modelo de datos (Tarea.cs).

```
MainPage.xaml DatabaseService.cs MainPage.xaml.cs NuGet-Solución Tareax.cs* a x MauiProgram.cs 

Defendor de soluciones 

Defendor de soluciones
```

Código de la clase DatabaseService.cs con operaciones CRUD.

```
        DatabaseService.cs
        →
        X
        MainPage.xaml.cs
        NuGet - Solución

→ 
↑ ToggleCompletadaAsync(Tarea tarea)

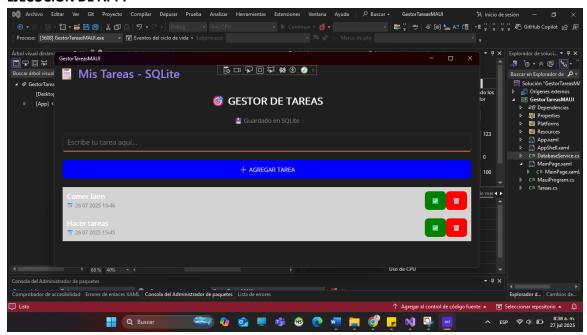
          using System;
using System.Collections.Generic;
using System.Ling;
using System.Text;
using System.Threading.Tasks;
            using SQLite;
                mespace GestorTareasMAUI
                       private SQLiteAsyncConnection _database;
                        public DatabaseService()
                           if (_database is not null)
    return;
                            // Crear la base de datos en la carpeta de la app
var databasePath = Path.Combine(FileSystem.AppOataDirectory, "Tareas.db");
_database = new SQLiteAsyncConnection(databasePath);
                             // Crear la tabla si no existe
await _database.CreateTableAsync<Tarea>();
                           blic async Task<List<Tarea>> GetTareasAsync()
                            await\ Init();\\ \underline{return}\ await\ \underline{database}. Table \\ \underline{Tarea}(). Order \underline{RyDescending}(t \Rightarrow t. FechaCreacion). To List Async();
                          ublic async Task<int> SaveTareaAsync(Tarea tarea)
                            await Init();

    ▷ &⊕ Dependencias
    ▷ ■ Properties
    ▷ ■ Platforms

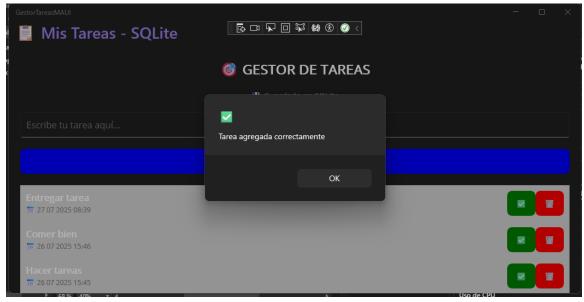
                           // Actualizar tarea existente
return await _database.UpdateAsync(tarea);
                                                                                                                                                                                                    DatabaseService.cs
                           // Insertar nueva tarea
return await _database.InsertAsync(tarea);
                                                                                                                                                                                                             C# MainPage.xaml.cs
                                                                                                                                                                                                     C# MauiProgram.cs
                        await Init();
return await _database.DeleteAsync(tarea);
                       await Init();
tarea.Completada = !tarea.Completada;
return await _database.UpdateAsync(tarea);
```

• Captura de la app en funcionamiento con tareas agregadas y eliminadas.

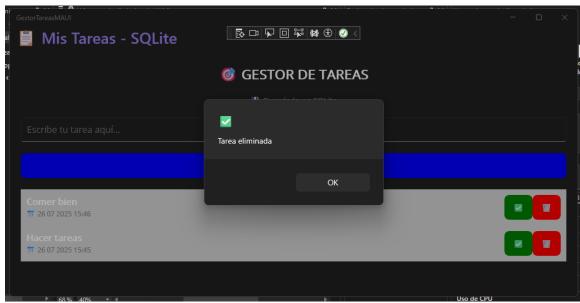
## **EJECUCIÓN DE APP:**



## **FUNCIÓN AGREGAR TAREA:**



## **FUNCIÓN ELIMINAR TAREA:**



## **TACHAR TAREA:**

