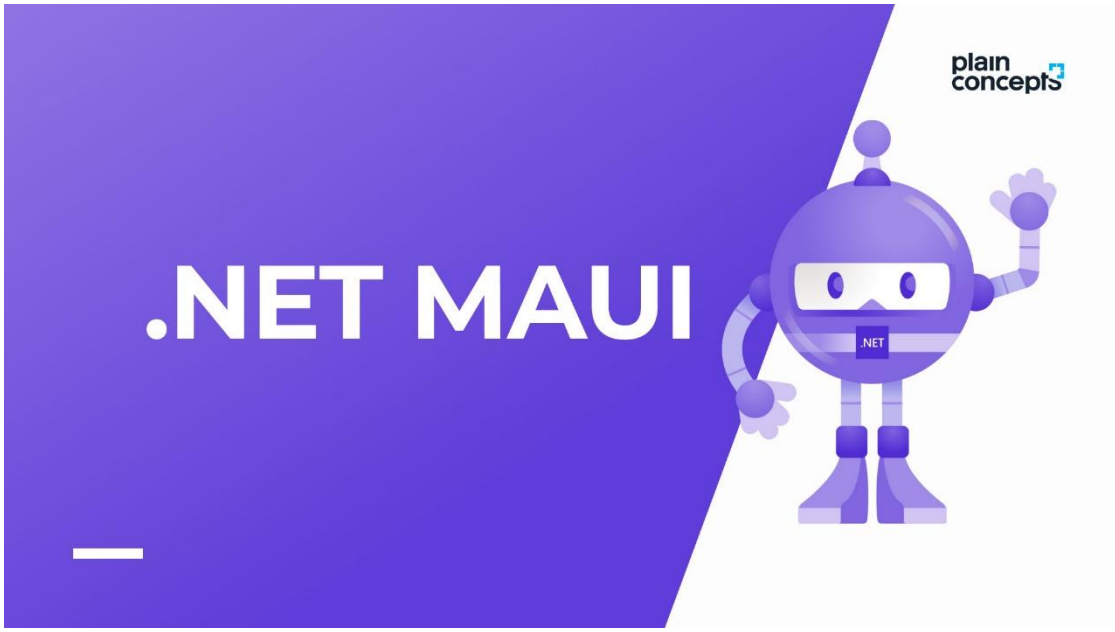


SQLITE

NET MAUI

Piero Zavala Chira | Desenvolupament d'interfícies | 06/02/2025

Contenido

Introducción

PROPÓSITO DEL PROYECTO

Este proyecto tiene como objetivo desarrollar una aplicación en .NET MAUI que implemente un CRUD (Crear, Leer, Actualizar y Eliminar) sobre una base de datos SQLite (empresa.db). La aplicación permitirá gestionar una lista de trabajadores mediante una interfaz gráfica amigable.

TECNOLOGÍAS UTILIZADAS

- .NET MAUI
- SQLite
- C#
- MVVM (Model-View-ViewModel)
- XAML para el diseño de la interfaz
- Procesador 12th Gen Intel(R) Core(TM) i5-12400F 2.50 GHz
- RAM instalada 16,0 GB (15,8 GB usable)
- Tipo de sistema Sistema operativo de 64 bits, procesador basado en x64

ESTRUCTURA DEL DOCUMENTO

1. Introducción
2. Organización de la estructura
3. Diseño de la solución
4. Desarrollo técnico
5. Funcionalidades avanzadas
6. Resultados y pruebas
7. Conclusión
8. Repositorio y bibliografía

Organización de la Estructura

PRIMERA PARTE: INTRODUCCIÓN Y CONTEXTUALIZACIÓN

Se presenta la justificación del proyecto y su utilidad en la gestión de trabajadores.

SEGUNDA PARTE: DISEÑO CONCEPTUAL DE LA INTERFAZ Y FUNCIONALIDADES

Se detallan los elementos UI y cómo interactúan con la base de datos.

TERCERA PARTE: DESARROLLO TÉCNICO

Se presenta el código fuente relevante y explicaciones técnicas sobre su implementación.

CUARTA PARTE: RESULTADOS Y PRUEBAS

Se analizan los resultados obtenidos y se documentan pruebas funcionales.

Diseño de la Solución

DIAGRAMA O BOCETO INICIAL

Se plantea un wireframe donde se visualiza la disposición de los elementos:

- Entry para ingresar el nombre y apellido del trabajador.
- CollectionView para visualizar la lista de trabajadores.
- Botones para agregar, actualizar y eliminar trabajadores.

EXPLICACIÓN DEL DISEÑO

- Se usa StackLayout y Grid para estructurar la interfaz.
- Se garantiza la adaptabilidad a diferentes tamaños de pantalla.
- Uso de colores neutros para mejorar la legibilidad.

Desarrollo Técnico

CÓDIGO ESTRUCTURADO

Objeto Trabajador

```
namespace SQLite03
{

    public class Trabajador : INotifyPropertyChanged
    {
        private int _id;
        private string _nombre;
        private string _apellidos;
        public event PropertyChangedEventHandler PropertyChanged;

        public int Id
        {
            get { return _id; }
            set
            {
                _id = value;
                OnPropertyChanged();
            }
        }

        public string Nombre
        {
            get { return _nombre; }
            set
            {
                _nombre = value;
                OnPropertyChanged();
            }
        }

        public string Apellidos
        {
            get { return _apellidos; }
            set
            {
                _apellidos = value;
                OnPropertyChanged();
            }
        }


        protected void OnPropertyChanged([CallerMemberName] string name = null)
        {
            PropertyChanged?.Invoke(this, new PropertyChangedEventArgs(name));
        }

}
```

MainPage.xaml

```xaml
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="SQLite03.MainPage">

    <ScrollView>
        <VerticalStackLayout
            Spacing="25"
            Padding="30,0"
            VerticalOptions="Start">
            <Label FontSize="Header"
                   Text="Empleados" />
            <!--Entry Nombre-->
            <Entry Placeholder="Nombre"
                   Text="{Binding Nombre}"
                   TextChanged="EntryNombre_TextChanged" />
            <!--Entry Apellidos-->
            <Entry Placeholder="Apellidos"
                   Text="{Binding Apellidos}"
                   TextChanged="EntryApellidos_TextChanged" />
            <!--Botón Insertar-->
            <Button Text="Insertar"
                    Clicked="ButtonInsertar_Clicked" />


            <CollectionView ItemsSource="{Binding OcTrabajadores}"
                            SelectionMode="Single"
                            SelectedItem="{Binding SelectedTrabajador}">

                <CollectionView.ItemTemplate>
                    <DataTemplate>
                        <StackLayout Orientation="Horizontal"
                                     Padding="10">

                            <!-- Muestra nombre -->
                            <Label Text="{Binding Nombre}"
                                   FontSize="Medium"
                                   VerticalOptions="Center" />
                            <!-- Muestra apellidos  -->
                            <Label Text="{Binding Apellidos}"
                                   FontSize="Medium"
                                   VerticalOptions="Center"
                                   Margin="10,0,0,0" />

                        </StackLayout>
                    </DataTemplate>
                </CollectionView.ItemTemplate>
            </CollectionView>
            <!--Boton Eliminar-->
            <Button Text="Eliminar"
                   Clicked="ButtonEliminar_Clicked"
                    IsEnabled="{Binding SelectedTrabajador}"/>
            <!--Boton Actualizar-->
            <Button Text="Actualizar"
                    Clicked="ButtonActualizar_Clicked"
                    IsEnabled="{Binding SelectedTrabajador}"/>
        </VerticalStackLayout>
    </ScrollView>

</ContentPage>
```

MainPage.cs

```
using System.Collections.ObjectModel;
using System.Data.SQLite;

namespace SQLite03
{

    public partial class MainPage : ContentPage
    {
        private ObservableCollection<Trabajador> _ocTrabajadores;
        private String _nombre;
        private String _apellidos;
        private Trabajador _selectedTrabajador;
        public Trabajador SelectedTrabajador
        {
            get { return _selectedTrabajador; }
            set
            {
                _selectedTrabajador = value;
                OnPropertyChanged();


            }
        }
        public ObservableCollection<Trabajador> OcTrabajadores


        {
            get { return _ocTrabajadores;}
            set
            {
                _ocTrabajadores = value;
                OnPropertyChanged();
            }
        }
        public MainPage()
        {
            InitializeComponent();
            // Lista de los trabajadores
            OcTrabajadores = new ObservableCollection<Trabajador>();
```

```
            // Creamos la ruta de la base de datos en el directorio de nuestro proyecto
            // No lo haríamos en producción pero en pruebas es más cómodo
            // tener localizada la base de datos y poder examinarla
            // con otros programas para ver los cambios producidos

            // En mi caso, devuelve:
            //
"D:\\Datos\\proyectos_DI2425\\ud04part01ExempleSQLite\\SQLite03\\bin\\Debug\\net8.0-
windows10.0.19041.0\\win10-x64\\AppX\\"
            string rutaDirectorioApp = System.AppContext.BaseDirectory;

            DirectoryInfo directorioApp = new DirectoryInfo(rutaDirectorioApp);

            // El objeto directorio será ahora:
            // D:\Datos\proyectos_DI2425\ud04part01ExempleSQLite\SQLite03
            directorioApp = directorioApp.Parent.Parent.Parent.Parent.Parent.Parent;

            // Creamos la ruta completa del fichero de la base de datos
            // En mi ejemplo:
            // D:\Datos\proyectos_DI2425\ud04part01ExempleSQLite\SQLite03\empresa.db
            string databasePath = Path.Combine(directorioApp.FullName, "empresa.db");

            // Creamos la cadena de conexión
            string connectionString = $"Data Source={databasePath};Version=3;";

            using (SQLiteConnection connection = new SQLiteConnection(connectionString))
            {
                connection.Open();
                // Creamos la tabla de trabajador
                CrearTablaTrabajador(connection);

                // Creamos la consulta y la ejecutamos
                string sql = "SELECT * FROM Trabajador";
                SQLiteCommand command = new SQLiteCommand(sql, connection);
                SQLiteDataReader reader = command.ExecuteReader();

                // Recorremos los registros devueltos del SELECT
                while (reader.Read())
                {
                    int idTrabajador = reader.GetInt32(0);
                    string nombreTrabajador = reader.GetString(1);
                    string apellidosTrabajador = reader.GetString(2);
```

```
                    // Creamos un objeto Trabajador y lo añadimos al Observable Collection
                    Trabajador trabajador = new Trabajador
                    {
                        Id = idTrabajador,
                        Nombre = nombreTrabajador,
                        Apellidos = apellidosTrabajador,
                    };

                    // Añadimos el trabajador al Observable Collection
                    OcTrabajadores.Add(trabajador);
                }

                reader.Close();
                connection.Close();
            }

            BindingContext = this;
        }

        private void CrearTablaTrabajador(SQLiteConnection connection)
        {
            // Creamos la tabla Trabajador en caso de que no exista
            // Su clave principal es un autonumérico
            string queryCrearTablaTrabajador = "CREATE TABLE IF NOT EXISTS Trabajador (" +
                                    "id INTEGER PRIMARY KEY AUTOINCREMENT, " +
                                    "nombre TEXT, " +
                                    "apellidos TEXT)";
            EjecutarNonQuery(connection, queryCrearTablaTrabajador);
        }



        private void EjecutarNonQuery(SQLiteConnection connection, string query)
        {
            // Este método ejecuta órdenes SQL que no devuelven consultas (Non-query command)

            using (SQLiteCommand command = new SQLiteCommand(query, connection))
            {
                command.ExecuteNonQuery();
            }
        }

        private void EntryNombre_TextChanged(object sender,
Microsoft.Maui.Controls.TextChangedEventArgs e)
        {
            _nombre = e.NewTextValue;
        }
```

```
        private void EntryApellidos_TextChanged(object sender,
Microsoft.Maui.Controls.TextChangedEventArgs e)
        {
            _apellidos = e.NewTextValue;
        }

        private void ButtonInsertar_Clicked(object sender, EventArgs e)
        {
            // Creamos la ruta completa del fichero de la base de datos
            string rutaDirectorioApp = System.AppContext.BaseDirectory;
            DirectoryInfo directorioApp = new DirectoryInfo(rutaDirectorioApp);
            directorioApp = directorioApp.Parent.Parent.Parent.Parent.Parent.Parent;
            string databasePath = Path.Combine(directorioApp.FullName, "empresa.db");
            string connectionString = $"Data Source={databasePath};Version=3;";

            using (SQLiteConnection connection = new SQLiteConnection(connectionString))
            {
                connection.Open();

                // Ejecutamos la consulta de inserción
                EjecutarNonQuery(connection, $"INSERT INTO Trabajador (nombre, apellidos) VALUES
('{_nombre}', '{_apellidos}')");

                // Vaciamos la colección actual
                OcTrabajadores.Clear();

                // Volvemos a cargar los trabajadores
                string sql = "SELECT * FROM Trabajador";
                SQLiteCommand command = new SQLiteCommand(sql, connection);
                SQLiteDataReader reader = command.ExecuteReader();

                while (reader.Read())
                {
                    int idTrabajador = reader.GetInt32(0);
                    string nombreTrabajador = reader.GetString(1);
                    string apellidosTrabajador = reader.GetString(2);

                    Trabajador trabajador = new Trabajador
                    {
                        Id = idTrabajador,
                        Nombre = nombreTrabajador,
                        Apellidos = apellidosTrabajador,
                    };

                    OcTrabajadores.Add(trabajador);
                }
```

```
                }

                reader.Close();
                connection.Close();
            }
        }

        private void ButtonEliminar_Clicked(object sender, EventArgs e)
        {
            if (SelectedTrabajador == null)
                return;

            // Creamos la ruta completa del fichero de la base de datos
            string rutaDirectorioApp = System.AppContext.BaseDirectory;
            DirectoryInfo directorioApp = new DirectoryInfo(rutaDirectorioApp);
            directorioApp = directorioApp.Parent.Parent.Parent.Parent.Parent.Parent;
            string databasePath = Path.Combine(directorioApp.FullName, "empresa.db");
            string connectionString = $"Data Source={databasePath};Version=3;";

            using (SQLiteConnection connection = new SQLiteConnection(connectionString))
            {
                connection.Open();

                // Eliminamos el trabajador de la base de datos
                string query = "DELETE FROM Trabajador WHERE id = @id";
                using (SQLiteCommand command = new SQLiteCommand(query, connection))
                {
                    command.Parameters.AddWithValue("@id", SelectedTrabajador.Id);
                    command.ExecuteNonQuery();
                }

                connection.Close();
            }

            // Eliminamos el trabajador de la interfaz
            OcTrabajadores.Remove(SelectedTrabajador);

            // Limpia la selección
            SelectedTrabajador = null;
        }

        private void ListViewTrabajadores_ItemSelected(object sender,
SelectedItemChangedEventArgs e)
        {
            if (e.SelectedItem == null)
                return;
```

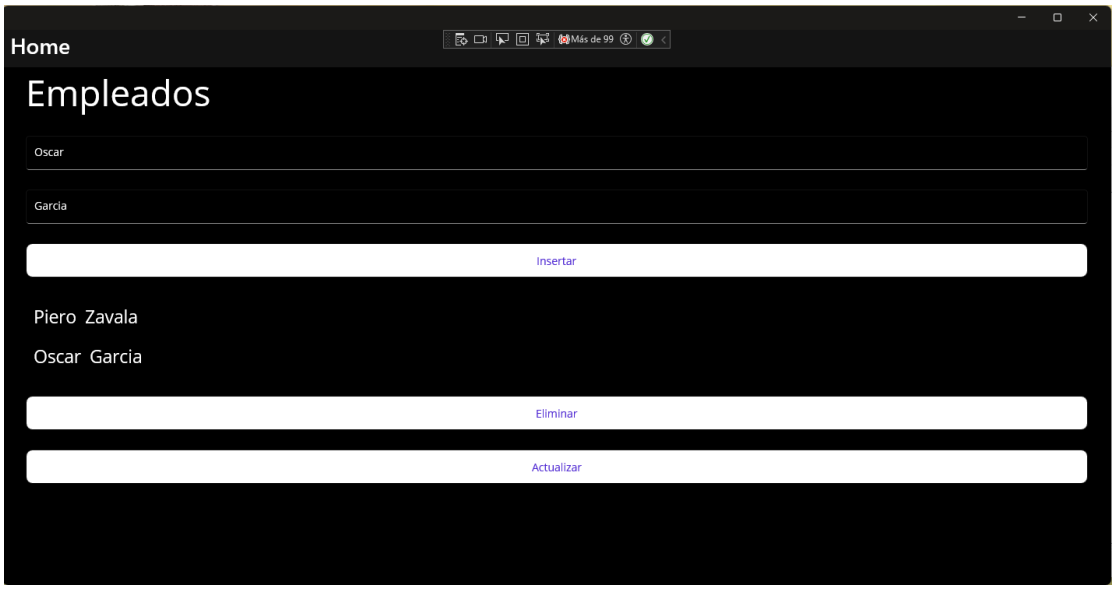
```
            SelectedTrabajador = e.SelectedItem as Trabajador;

            // Asignar los valores del trabajador seleccionado a los entries
            _nombre = SelectedTrabajador.Nombre;
            _apellidos = SelectedTrabajador.Apellidos;
        }

        private void ButtonActualizar_Clicked(object sender, EventArgs e)
        {
            if (SelectedTrabajador == null)
                return;

            // Actualizar los valores del trabajador seleccionado con los valores de los entries
            SelectedTrabajador.Nombre = _nombre;
            SelectedTrabajador.Apellidos = _apellidos;

            // Creamos la ruta completa del fichero de la base de datos
            string rutaDirectorioApp = System.AppContext.BaseDirectory;
            DirectoryInfo directorioApp = new DirectoryInfo(rutaDirectorioApp);
            directorioApp = directorioApp.Parent.Parent.Parent.Parent.Parent.Parent;
            string databasePath = Path.Combine(directorioApp.FullName, "empresa.db");
            string connectionString = $"Data Source={databasePath};Version=3;";

            using (SQLiteConnection connection = new SQLiteConnection(connectionString))
            {
                connection.Open();

                // Actualizamos el registro en la base de datos
                string query = "UPDATE Trabajador SET nombre = @nombre, apellidos = @apellidos
WHERE id = @id";
                using (SQLiteCommand command = new SQLiteCommand(query, connection))
                {
                    command.Parameters.AddWithValue("@nombre", SelectedTrabajador.Nombre);
                    command.Parameters.AddWithValue("@apellidos", SelectedTrabajador.Apellidos);
                    command.Parameters.AddWithValue("@id", SelectedTrabajador.Id);
                    command.ExecuteNonQuery();
                }

                connection.Close();
            }

            // Actualiza la lista en la interfaz
            var trabajador = OcTrabajadores.FirstOrDefault(t => t.Id == SelectedTrabajador.Id);
            if (trabajador != null)
            {
                trabajador.Nombre = SelectedTrabajador.Nombre;
                trabajador.Apellidos = SelectedTrabajador.Apellidos;
```

```
                connection.Close();
            }

            // Actualiza la lista en la interfaz
            var trabajador = OcTrabajadores.FirstOrDefault(t => t.Id == SelectedTrabajador.Id);
            if (trabajador != null)
            {
                trabajador.Nombre = SelectedTrabajador.Nombre;
                trabajador.Apellidos = SelectedTrabajador.Apellidos;

                // Notificar cambios en la colección
                OnPropertyChanged(nameof(OcTrabajadores));
            }
        }

    }
} </CollectionView.ItemTemplate>
            </CollectionView>
            <!--Boton Eliminar-->
            <Button Text="Eliminar"
                    Clicked="ButtonEliminar_Clicked"
                     IsEnabled="{Binding SelectedTrabajador}"/>
            <!--Boton Actualizar-->
            <Button Text="Actualizar"
                    Clicked="ButtonActualizar_Clicked"
                    IsEnabled="{Binding SelectedTrabajador}"/>
        </VerticalStackLayout>
    </ScrollView>

</ContentPage>
```

PASOS DE IMPLEMENTACIÓN

1. Definir el modelo Trabajador.cs.
2. Configurar SQLite en MauiProgram.cs.
3. Crear la interfaz en MainPage.xaml.
4. Implementar métodos CRUD en MainPage.xaml.cs.

CAPTURAS DE PANTALLA



Funcionalidades Avanzadas

- Validación de entrada en los Entry.
- Navegación entre pantallas si es requerido.
- Mejora de accesibilidad para lectores de pantalla.

Resultados y Pruebas

RESULTADOS OBTENIDOS

- Implementación completa del CRUD.
- Interfaz funcional y sencilla de usar.

PRUEBAS REALIZADAS

- Inserción, actualización y eliminación de trabajadores.
- Verificación del almacenamiento en la base de datos.

ERRORES Y CORRECCIONES

- Manejo de errores en la base de datos.
- Validaciones para evitar datos vacíos.

Conclusión

VALORACIÓN PERSONAL

Este proyecto permitió aplicar conocimientos en MAUI y bases de datos SQLite, consolidando buenas prácticas en el desarrollo de interfaces.

IMPACTO DEL PROYECTO

La aplicación facilita la gestión de trabajadores de manera intuitiva y eficiente.

Repositorio y Bibliografía

REPOSITORIO GITHUB

Enlace al código fuente: Github

FUENTES DE INFORMACIÓN

- Documentación oficial de .NET MAUI.
- Blogs y tutoriales sobre desarrollo en C# y SQLite.