

**Natalia Coronado Romero 1ºDAM**

## ÍNDICE

Introducción.....	3
Objetivo de la memoria.....	3
Material utilizado.....	3
GitHub.....	3
Desarrollo.....	4
Interfaz Visual.....	4
-Vista de la aplicación.....	4
-Archivo XAML.....	5
Código del CS.....	7
Video del funcionamiento.....	10

## Introducción

Proyecto de una aplicación con MAUI en Visual Studio para crear un CRUD de una base de datos..

## Objetivo de la memoria

El objetivo de esta práctica es diseñar una aplicación que se sincronice una base de datos y hacer modificaciones en él y se actualicen en tiempo real.

## Material utilizado

**Marca y Modelo del Procesador:** Intel Pentium CPU G4400 @ 3.30GHz

**Tipo Memoria RAM (memoria y slots):** 4GB RAM, 2400 MHz

**Tipo de dispositivo de almacenamiento**

**capacidad (GiB):** HDD 1TB

**Programas utilizados:** Spectacle, Visual Studio 2022

## GitHub

Es es el enlace de github al repositorio de la asignatura:

<https://github.com/MalvaLego/Desarrollo-de-Interfaces.git>

# Desarrollo

## Interfaz Visual

### -Vista de la aplicación

Esta es la vista previa de como se vería la aplicación para el usuario.



The screenshot shows a web application window with a title bar containing standard window controls (minimize, maximize, close). The application has a purple header bar with the text "Home". Below the header, the main content area has a light gray background. At the top of this area is the word "Empleados" in a large, bold, black font. Below this, there are two input fields. The first is labeled "Nombre:" and contains the text "datos". The second is labeled "Apellido:" and contains the number "2". Below these input fields are three buttons: a green button labeled "Añadir", an orange button labeled "Actualizar", and a red button labeled "Eliminar". At the bottom of the form, there is a light green rectangular box containing two lines of text: "datos 1" and "datos 2".

*Figura 1: Vista de la interfaz de la aplicación en ejecución*

## -Archivo XAML

Este es el código del archivo MainPage.xaml donde se encuentran los elementos visuales y sus características.

```
1  <?xml version="1.0" encoding="utf-8" ?>
2  <ContentPage xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
3  xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4  x:Class="SQLite03.MainPage"
5  BackgroundColor="#ECEFF1"
6  >
7  <ScrollView>
8  <VerticalStackLayout
9  Spacing="20"
10 Padding="30,0"
11 >
12 <Label FontSize="Header"
13 Text="Empleados"
14 HorizontalOptions="Center"
15 FontAttributes="Bold"
16 TextColor="#333"
17 />
18
19 <Label FontSize="22"
20 Text="Nombre:"
21 Margin="0,0,-12"
22 TextColor="#333"
23 x:Name="tvNombre"
24 />
25 <Entry BackgroundColor="White"
26 x:Name="etNombre"
27 >
28 </Entry>
29
30 <Label FontSize="20"
31 Text="Apellido:"
32 Margin="0,0,-12"
33 x:Name="tvApellido"
34 />
35 <Entry BackgroundColor="White"
36 x:Name="etApellido"
37 >
38 </Entry>
```

Figura 2: Código de los dos Entry junto con el texto en la parte de arriba

```
40
41
42 <HorizontalStackLayout HorizontalOptions="Center" Margin="0,20,0,0">
43 <Button x:Name="btnAnyadir"
44 Clicked="Anyadir"
45 Text="Añadir"
46 Margin="0,0,30,0"
47 WidthRequest="150"
48 BackgroundColor="#388E3C"
49 >
50 </Button>
51 <Button x:Name="btnActualizar"
52 Clicked="Actualizar"
53 Text="Actualizar"
54 Margin="0,0,30,0"
55 WidthRequest="150"
56 BackgroundColor="#FF9800"
57 >
58 </Button>
59 <Button x:Name="btnEliminar"
60 Clicked="Eliminar"
61 Text="Eliminar"
62 WidthRequest="150"
63 BackgroundColor="#D32F2F"
64 >
65 </Button>
66 </HorizontalStackLayout>
```

Figura 3: Código de los tres botones de la aplicación

```

70 <CollectionView ItemsSource="{Binding OcTrabajadores}"
71 SelectionMode="Single"
72 SelectedItem="{Binding TrabajadorSelected, Mode=TwoWay}"
73 WidthRequest="500"
74 >
75 <CollectionView.ItemTemplate>
76 <DataTemplate>
77 <StackLayout Orientation="Horizontal"
78 Padding="10"
79 WidthRequest="500"
80 BackgroundColor="#d1ede4"
81 >
82 <!-- Muestra nombre -->
83 <Label Text="{Binding Nombre}"
84 FontSize="Medium"
85 VerticalOptions="Center"
86 />
87 <!-- Muestra apellidos -->
88 <Label Text="{Binding Apellidos}"
89 FontSize="Medium"
90 VerticalOptions="Center"
91 Margin="10,0,0,0"
92 />
93 <VisualStateManager.VisualStateGroups>
94 <VisualStateGroup Name="CommonStates">
95 <VisualState Name="Normal"></VisualState>
96 <VisualState Name="Selected">
97 <VisualState.Setters>
98 <Setter Property="BackgroundColor" Value="#ff5c5c"></Setter>
99 </VisualState.Setters>
100 </VisualState>
101 </VisualStateGroup>
102 </VisualStateManager.VisualStateGroups>
103 </StackLayout>
104 </DataTemplate>
105 </CollectionView.ItemTemplate>
106 </CollectionView>
107 </VerticalStackLayout>
108 </ScrollView>
109 </ContentPage>
110

```

Figura 4: Este es el final del código que incluye el CollectionView de la base de datos. Dentro de este están los dos label para mostrar el nombre y apellidos actualizados con binding

## Código del CS

Aquí se encuentra el código para que pueda funcionar la aplicación y se puede ejecutar, añadiendo también acciones que se quieran realizar en ella.



```
1  using System.Collections.ObjectModel;
2  using System.Data.SQLite;
3
4  namespace SQLite03
5  {
6      public partial class MainPage : ContentPage
7      {
8          private List<Trabajador> _trabajadorList= new List<Trabajador>();
9          private ObservableCollection<Trabajador> _ocTrabajadores;
10         public ObservableCollection<Trabajador> OcTrabajadores
11         {
12             get { return _ocTrabajadores; }
13             set
14             {
15                 _ocTrabajadores = value;
16                 OnPropertyChanged();
17             }
18         }
19         private Trabajador _trabajadorSelected;
20         public Trabajador TrabajadorSelected
21         {
22             get { return _trabajadorSelected; }
23             set
24             {
25                 _trabajadorSelected = value;
26                 OnPropertyChanged();
27             }
28         }
29     }
30 }
```

Figura 5: El principio del código crea variables para que se vayan actualizando todo el rato datos entre la base de datos, la lógica interna y la interfaz gráfica. También inicializa una Lista para servir de intermediaria más adelante.

```

29 0 referencias
30 public MainPage()
31 {
32     InitializeComponent();
33     // Lista de los trabajadores
34     OcTrabajadores = new ObservableCollection<Trabajador>();
35
36     // Conexión con la base de datos
37     string rutaDirectorioApp = System.AppContext.BaseDirectory;
38     DirectoryInfo directorioApp = new DirectoryInfo(rutaDirectorioApp);
39     directorioApp = directorioApp.Parent.Parent.Parent.Parent.Parent.Parent;
40     string databasePath = Path.Combine(directorioApp.FullName, "empresa.db");
41     string connectionString = $"Data Source={databasePath};Version=3;";
42
43     using (SQLiteConnection connection = new SQLiteConnection(connectionString))
44     {
45         connection.Open();
46
47         CrearTablaTrabajador(connection);
48         // InsertarDatosEjemplo(connection);
49
50         connection.Close();
51     }
52     BindingContext = this;
53
54 1 referencia
55 private void CrearTablaTrabajador(SQLiteConnection connection)
56 {
57     // Creamos la tabla Trabajador en caso de que no exista
58     // Su clave principal es un autonumérico
59     string queryCrearTablaTrabajador = "CREATE TABLE IF NOT EXISTS Trabajador (" +
60         "id INTEGER PRIMARY KEY AUTOINCREMENT, " +
61         "nombre TEXT, " +
62         "apellidos TEXT)";
63     EjecutarNonQuery(connection, queryCrearTablaTrabajador);
64
65 2 referencias
66 private void EjecutarNonQuery(SQLiteConnection connection, string query)
67 {
68     // Este método ejecuta órdenes SQL que no devuelven consultas (Non-query command)
69
70     using (SQLiteCommand command = new SQLiteCommand(query, connection))
71     {
72         command.ExecuteNonQuery();
73     }
74 }

```

Figura 6: En el MainPage del código, aparte de las líneas de conceptos anteriormente dados, se crea la base de datos y se sincroniza. Dentro de esta, llama a dos funciones las cuales una crea una tabla en la base de datos y la otra ejecuta el comando sql puesto para la creación de la tabla.



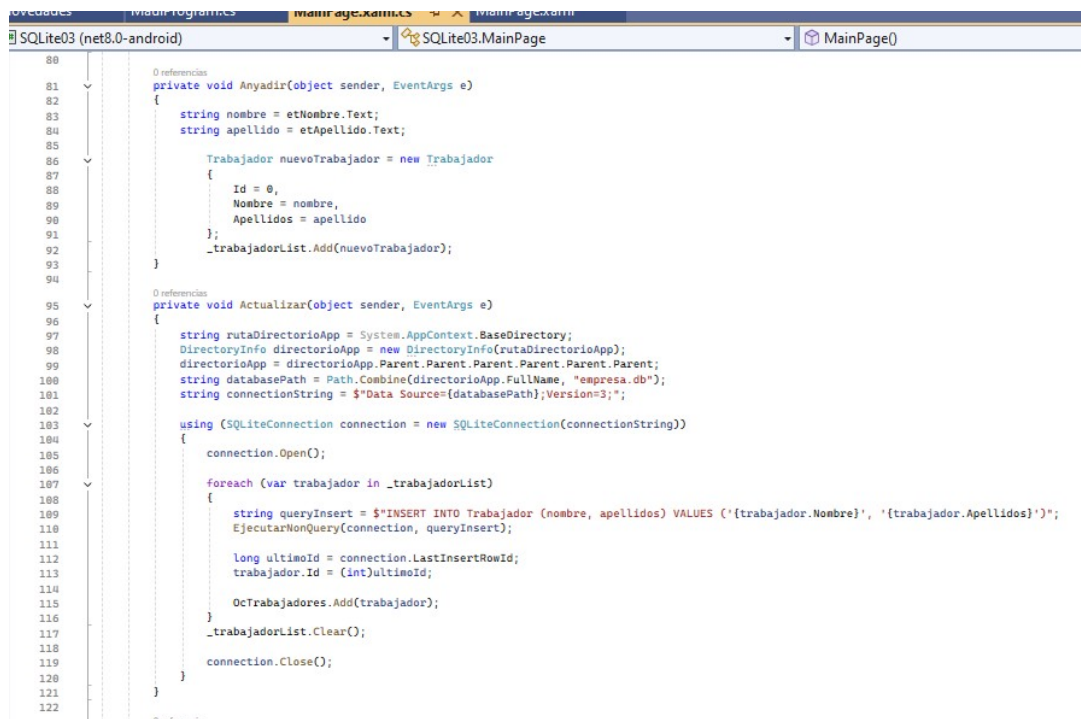


Figura 7: Estas dos funciones de la imagen son llamadas con los botones del xaml. Al añadir un nuevo registro, se crea un nuevo trabajador con los datos de los Entry y se pone en la Lista intermediaria. Al actualizar, se sincroniza la base de datos y se recorre la lista intermediaria de la cual se irá añadiendo cada registro ahora sí a la base de datos.

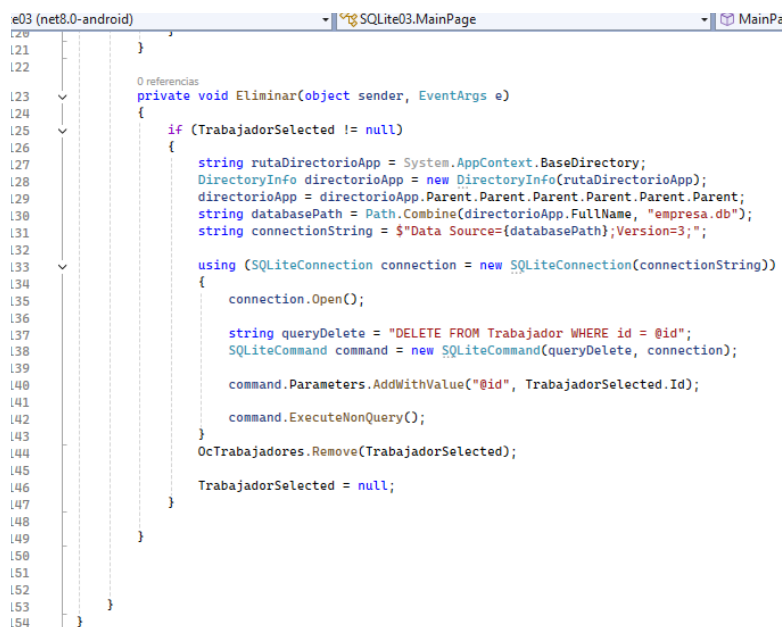


Figura 8: La última función del código es eliminar un registro. Al registro que se seleccione para eliminar, recogerá su Id y la buscara en la base de datos para eliminar ese trabajador.

## Video del funcionamiento

