

MEMORIA DE FIREBASE

Como elegir sistema operativo



Mario Fernández Alfonso

INDICE

- [1. Introducción](#)
- [2. Objetivos](#)
- [3. Desarrollo](#)
- [4. Errores](#)
- [5. Aplicación en Funcionamiento](#)
- [4. Conclusión](#)
- [6. Problemas Encontrados](#)
- [7. Sugerencias](#)

1. Introducción

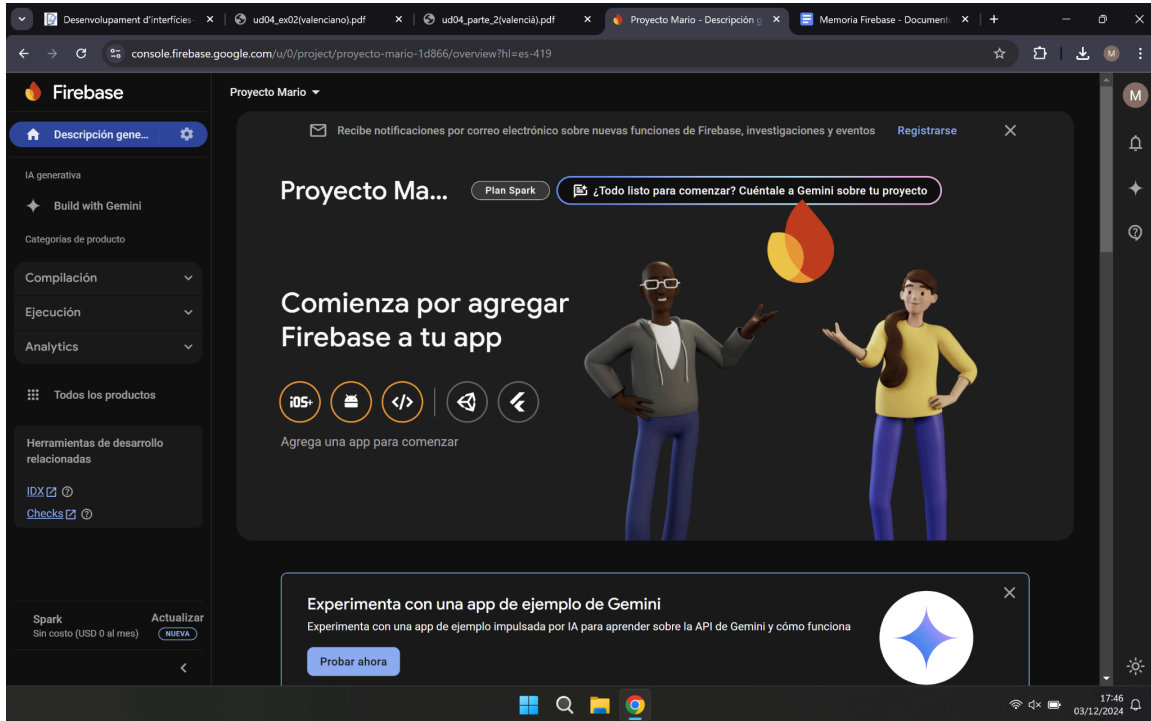
Este proyecto consiste en desarrollar una aplicación en .NET MAUI que se conecte con Firebase para gestionar una lista de tareas. La interfaz permitirá realizar operaciones básicas como mostrar todas las tareas, añadir nuevas, modificar las existentes y eliminarlas. Además, aseguraremos que los datos locales de la aplicación estén siempre sincronizados con los datos almacenados en Firebase. El objetivo es aprender a trabajar con bases de datos en tiempo real y a gestionar una interfaz interactiva en una aplicación multiplataforma.

2. Objetivos

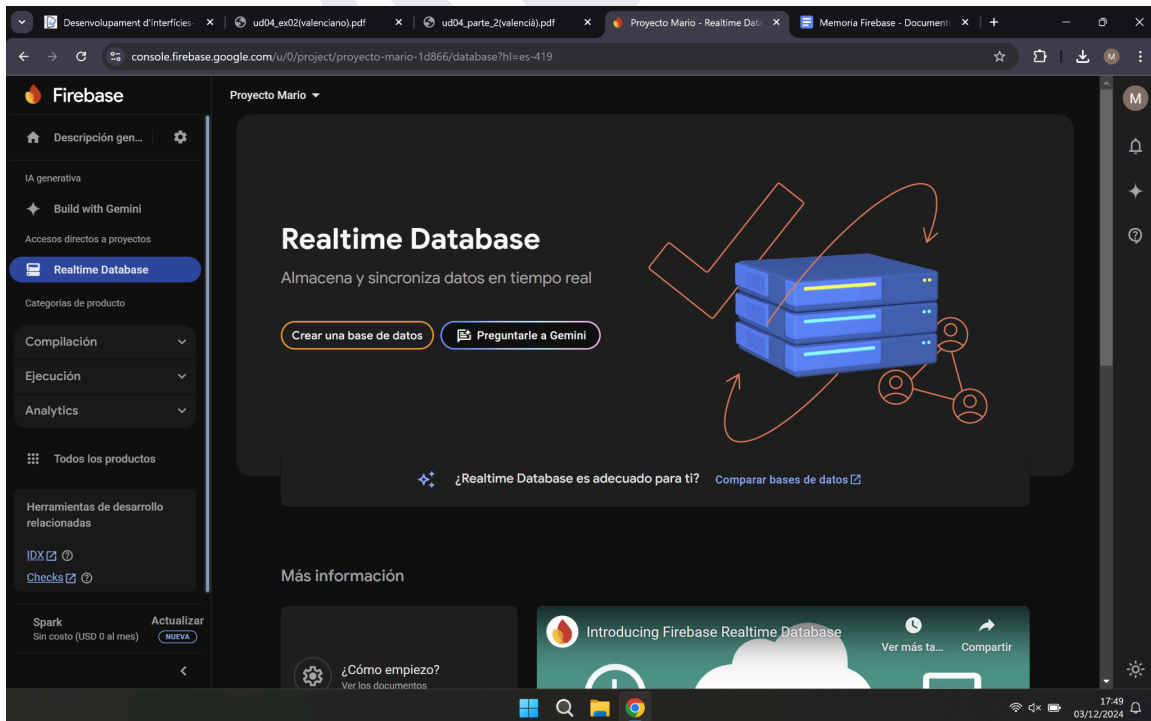
El objetivo principal es usar Firebase y conectarla mediante Visual Studio 2022 y de esta forma tener una base de datos conectada desde la aplicación que vamos a realizar.

3. Desarrollo

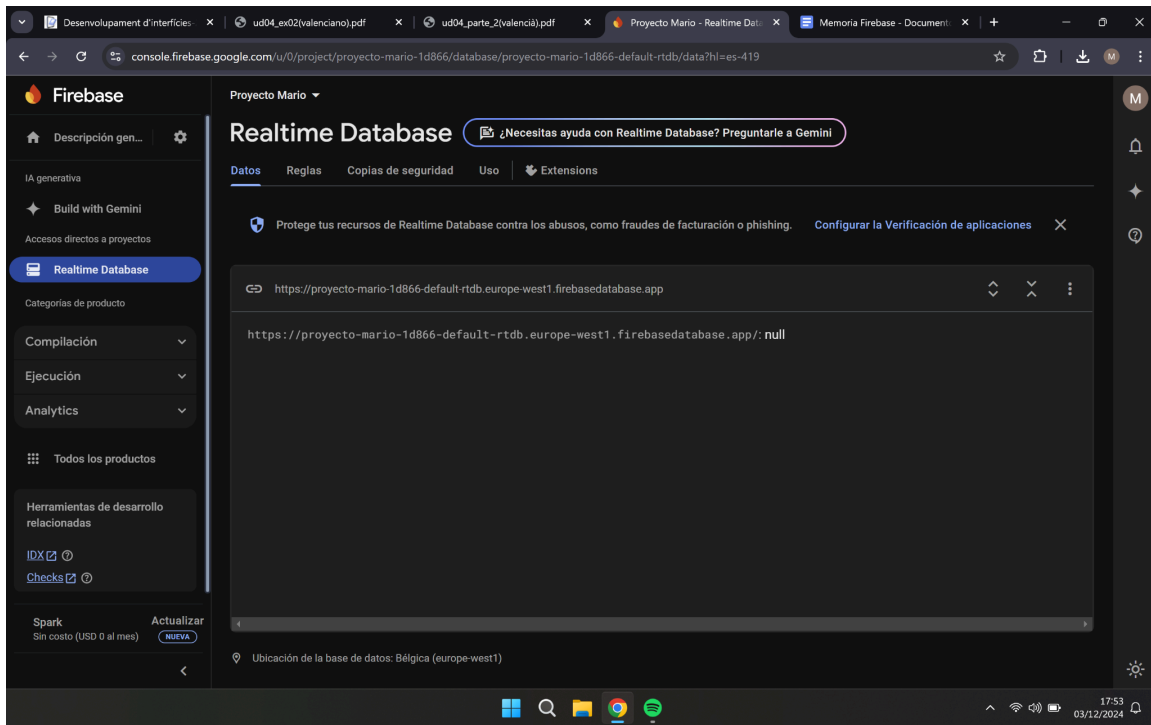
Primero de todo comenzamos creando el proyecto de Firebase, ya creamos el proyecto y comenzamos el proceso siguiente.



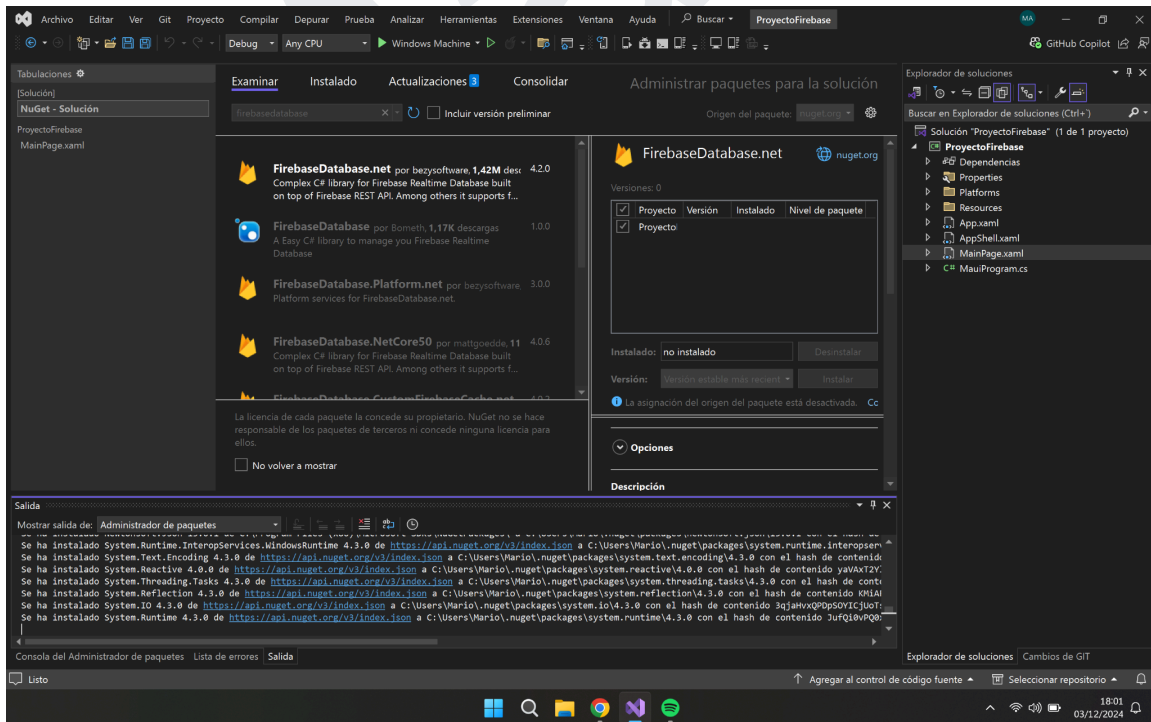
Ahora vamos a crear la base de datos desde el apartado “Realtime Database” accedemos y veremos el siguiente apartado:



Aquí ya vemos cómo hemos creado la base de datos en su totalidad:



Instalamos la extensión mediante el administrador de paquetes nuget.



Diseño MainPage.xaml.cs

Este código implementa una aplicación en .NET MAUI que interactúa con Firebase Realtime Database para gestionar tareas. Utiliza FirebaseClient para la conexión y ObservableCollection<Tarea> para la lista reactiva de tareas. Las principales funciones son:

- Cargar tareas desde Firebase y mostrarlas en la interfaz.
- Añadir tareas con PostAsync y actualizarlas en la colección local.
- Modificar tareas seleccionadas con PutAsync para actualizar datos en Firebase.
- Eliminar tareas con DeleteAsync y eliminarlas de la lista local.
- Usa SelectionChangedEventArgs para seleccionar y mostrar una tarea en un Entry para su edición.

```
1 using Firebase.Database;
2 using Firebase.Database.Query;
3 using System.Collections.ObjectModel;
4
5 namespace ProyectoFirebase
6 {
7     public partial class MainPage : ContentPage
8     {
9         private FirebaseClient firebase;
10        public ObservableCollection<Tarea> Tareas { get; set; }
11
12        public MainPage()
13        {
14            InitializeComponent();
15
16            firebase = new FirebaseClient("https://proyecto-mario-1d866-default-
17            rtdb.europe-west1.firebaseio.com/");
18            Tareas = new ObservableCollection<Tarea>();
19
20            BindingContext = this;
21
22            CargarTareas();
23        }
24
25        private async void CargarTareas()
26        {
27            try
28            {
29                var firebaseTareas = await firebase
30                    .Child("tareass")
31                    .OnceAsync<Tarea>();
32
33                Tareas.Clear();
34                foreach (var item in firebaseTareas)
35                {
36                    Tareas.Add(new Tarea { Id = item.Key, Descripcion =
37                        item.Object.Descripcion });
38                }
39
40                TareaCollectionView.ItemsSource = Tareas;
41            }
42            catch (Exception ex)
43            {
44                await DisplayAlert("Error", $"No se pudieron cargar las tareas:
45                {ex.Message}", "OK");
46            }
47        }
48    }
49 }
```



Firebase

```
1 private async void OnAñadirClicked(object sender, EventArgs e)
2 {
3     if (!string.IsNullOrEmpty(EntryTarea.Text))
4     {
5         var nuevaTarea = new Tarea { Descripcion = EntryTarea.Text };
6
7         try
8         {
9             var respuesta = await
10 firebase.Child("tareas").PostAsync(nuevaTarea);
11 Tareas.Add(new Tarea { Id = respuesta.Key, Descripcion =
12 nuevaTarea.Descripcion });
13
14 EntryTarea.Text = string.Empty;
15 }
16 catch (Exception ex)
17 {
18     await DisplayAlert("Error", $"No se pudo añadir la tarea:
19 {ex.Message}", "OK");
20 }
21 }
22
23 private async void OnModificarClicked(object sender, EventArgs e)
24 {
25     if (TareaCollectionView.SelectedItem is Tarea tareaSeleccionada &&
26 !string.IsNullOrEmpty(EntryTarea.Text))
27     {
28         try
29         {
30             tareaSeleccionada.Descripcion = EntryTarea.Text;
31
32             await
33 firebase.Child("tareas").Child(tareaSeleccionada.Id).PutAsync(tareaSeleccionada);
34 CargarTareas();
35
36 EntryTarea.Text = string.Empty;
37 }
38 catch (Exception ex)
39 {
40     await DisplayAlert("Error", $"No se pudo modificar la tarea:
41 {ex.Message}", "OK");
42 }
43 }
44
45 private async void OnEliminarClicked(object sender, EventArgs e)
46 {
47     if (TareaCollectionView.SelectedItem is Tarea tareaSeleccionada)
48     {
49         try
50         {
51             await
52 firebase.Child("tareas").Child(tareaSeleccionada.Id).DeleteAsync();
53 Tareas.Remove(tareaSeleccionada);
54 }
55 catch (Exception ex)
56 {
57     await DisplayAlert("Error", $"No se pudo eliminar la tarea:
58 {ex.Message}", "OK");
59 }
60 }
61 }
62
63 private void OnTareaSeleccionada(object sender, SelectionChangedEventArgs
64 e)
65 {
66     if (e.CurrentSelection.FirstOrDefault() is Tarea tareaSeleccionada)
67     {
68         EntryTarea.Text = tareaSeleccionada.Descripcion;
69     }
70 }
```

Codigo MainPage.xaml

La interfaz de usuario en XAML de la aplicación .NET MAUI permite gestionar tareas con un campo de entrada, botones para añadir, modificar y eliminar, y una lista que muestra las tareas almacenadas, todo estilizado con colores y diseño interactivo.

```
1 <ContentPage
2   xmlns="http://schemas.microsoft.com/dotnet/2021/maui"
3   xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
4   x:Class="ProyectoFirebase.MainPage"
5   BackgroundColor="White">
6   <StackLayout Padding="20" Spacing="15">
7       <Frame BackgroundColor="#F0F0F0"
8             BorderColor="#D1D1D1"
9             Padding="10"
10            CornerRadius="5">
11         <Entry x:Name="EntryTarea"
12               Placeholder="Escribe la descripción de la
13 tarea"
14               FontSize="16"
15               TextColor="#333"
16               HorizontalOptions="FillAndExpand" />
17     </Frame>
18     <StackLayout Orientation="Horizontal" Spacing="15"
19       HorizontalOptions="Center">
20         <Button Text="Añadir"
21               BackgroundColor="#0078D4"
22               TextColor="White"
23               Padding="10,5"
24               FontSize="14"
25               Clicked="OnAñadirClickado" />
26         <Button Text="Modificar"
27               BackgroundColor="#FF8C00"
28               TextColor="White"
29               Padding="10,5"
30               FontSize="14"
31               Clicked="OnModificarClickado" />
32         <Button Text="Eliminar"
33               BackgroundColor="#D32F2F"
34               TextColor="White"
35               Padding="10,5"
36               FontSize="14"
37               Clicked="OnEliminarClickado" />
38     </StackLayout>
39     <CollectionView x:Name="TareaCollectionView"
40                   SelectionMode="Single"
41                   SelectionChanged="OnTareaSeleccionada"
42                   BackgroundColor="#F9F9F9"
43                   Margin="0,10,0,0">
44         <CollectionView.ItemTemplate>
45             <DataTemplate>
46                 <Frame BackgroundColor="White"
47                       BorderColor="#D1D1D1"
48                       CornerRadius="5"
49                       Margin="0,5,0,5"
50                       Padding="10"
51                       HasShadow="True">
52                     <Label Text="{Binding Description}"
53                           FontSize="16"
54                           TextColor="#333"
55                           HorizontalOptions="Center" />
56                 </Frame>
57             </DataTemplate>
58         </CollectionView.ItemTemplate>
59     </CollectionView>
60 </StackLayout>
61 </ContentPage>
```


Codigo Tarea.cs

```
Tarea.cs Firebase
1 public class Tarea
2 {
3     public string Id { get; set; } // ID único generado por Firebase
4     public string Descripcion { get; set; } // Descripción de la tarea
5 }
6
```

5. Aplicación en Funcionamiento



4. Conclusión

Este código de XAML crea una interfaz funcional y estilizada en una aplicación .NET MAUI para gestionar tareas de forma sencilla y eficiente, permitiendo a los usuarios agregar, modificar, eliminar y visualizar tareas almacenadas en una base de datos. La combinación de interactividad, diseño atractivo y enlace con la lógica de negocio hace que la aplicación sea intuitiva y fácil de usar.

6. Problemas Encontrados

Problemas con el PutAsync y también junto poder conectar al Firebase, al final lo pude solucionar por un error tonto ya que puse mal la forma de ponerlo.

7. Sugerencias

Por el momento ninguna, todo ok.

