



CONSEJO DEPARTAMENTAL DE AYACUCHO
CAPÍTULO DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Acceso de Aplicaciones Móviles (.NET MAUI) a API's sobre Azure

DR. ENRIQUE AGUILAR VARGAS



Agenda


- Contexto de la Base de Datos (SQL Server) para Azure
- Contexto de la API (.NET - Visual Studio 2022)
- Despliegue de la API en Azure AppServices
- Aplicación para Android/iOS con .NET MAUI
- Pruebas de Ejecución en Android
- Pruebas y Ejecución en iOS

An abstract digital cityscape composed of glowing blue cubes and rectangular structures. The surfaces of these structures are covered in a dense pattern of small, light blue dots, resembling a digital grid or data points. Several bright blue beams of light emanate from the structures, and small red and green dots are scattered throughout the scene, adding to the digital aesthetic. The background is a gradient of dark blue and purple.

Contexto de la Base de Datos (SQL Server) para Azure

DEMO DE CREACIÓN DE BASE DE
DATOS EN AZURE.

Contexto de la Base de Datos (SQL Server) para Azure

 Connect to Server ✕

SQL Server

Server type:

Database Engine

Server name:

enrique.database.windows.net

Authentication:

SQL Server Authentication

Login:

enrique

Password:

☒ Remember password

Connect

Cancel

Help

Options >>

[illegible]

```
public class Datos
{
    2 references
    public int Id { get; set; }
    2 references
    public string? Nombre { get; set; }
    2 references
    public string? Apellido { get; set; }
    2 references
    public string? Conferencia { get; set; }
    2 references
    public string? Fecha { get; set; }
    2 references
    public string? Imagen { get; set; }
    2 references
    public string? ImagenFondo { get; set; }
    2 references
    public string? Pais { get; set; }
    2 references
    public string? Semblanza { get; set; }
    2 references
    public string? Latitud { get; set; }
    2 references
    public string? Longitud { get; set; }
}
```

Contexto de la API (.NET – Visual Studio – C#)

- Generamos la clase con la estructura de la tabla de la base de datos.

Contexto de la API (.NET – Visual Studio – C#)

```
using System.Data;
using System.Data.SqlClient;
namespace APIExpo
{
    2 references
    public class Consultas
    {
        public List<Datos>? ListadeConferencistas;
        1 reference
        public List<Datos> ConsultaRegistro(int ID)
        {
            var dt = new DataTable();
            var connect = new SqlConnection
                ("Server=tcp:enrique.database.windows.net,1433;Initial Catalog=Informacion;Persist Security Info=True;User ID=api;Password=api;MultipleActiveResultSets=true");
            var cmd = new SqlCommand("SELECT * From Datos WHERE Id= '" + ID + "'", connect);
            try
            {
                ListadeConferencistas = new List<Datos>();
                connect.Open();
                var da = new SqlDataAdapter(cmd);
                da.Fill(dt);
                connect.Close();
            }
            catch { }
        }
    }
}
```

Iniciamos el código de conexión de la aplicación

Contexto de la API (.NET – Visual Studio – C#)

```
using System.Data;
using System.Data.SqlClient;
namespace APIExpo
{
    2 references
    public class Consultas
    {
        public List<Datos>? ListadeConferencistas;
        1 reference
        public List<Datos> ConsultaRegistro(int ID)
        {
            var dt = new DataTable();
            var connect = new SqlConnection
                ("Server=tcp:enrique.database.windows.net,1433;Initial Catalog=Informacion;Persist Security Info=True;User ID=api;Password=api;MultipleActiveResultSets=true");
            var cmd = new SqlCommand("SELECT * From Datos WHERE Id= '" + ID + "'", connect);

            try
            {
                ListadeConferencistas = new List<Datos>();
                connect.Open();
                var da = new SqlDataAdapter(cmd);
                da.Fill(dt);
                connect.Close();
            }
        }
    }
}
```

Iniciamos el código de conexión de la aplicación

Contexto de la API (.NET – Visual Studio – C#)

```
for (int i = 0; i < dt.Rows.Count; i++)
{
    Datos cliente = new Datos();
    cliente.Id = int.Parse((dt.Rows[i]["Id"]).ToString());
    cliente.Nombre = ((dt.Rows[i]["Nombre"]).ToString());
    cliente.Apellido = ((dt.Rows[i]["Apellido"]).ToString());
    cliente.Conferencia = ((dt.Rows[i]["Conferencia"]).ToString());
    cliente.Fecha = ((dt.Rows[i]["Fecha"]).ToString());
    cliente.Imagen = ((dt.Rows[i]["Imagen"]).ToString());
    cliente.ImagenFondo = ((dt.Rows[i]["ImagenFondo"]).ToString());
    cliente.Pais = ((dt.Rows[i]["Pais"]).ToString());
    cliente.Semblanza = ((dt.Rows[i]["Semblanza"]).ToString());
    cliente.Latitud = ((dt.Rows[i]["Latitud"]).ToString());
    cliente.Longitud = ((dt.Rows[i]["Longitud"]).ToString());
    ListadeConferencistas.Add(cliente);
}
return ListadeConferencistas;
}
catch (Exception ex)
{
    connect.Close();
    return ListadeConferencistas;
}
}
```

1 reference

Extraemos la información de la consulta

Contexto de la API (.NET – Visual Studio – C#)

```
using Microsoft.AspNetCore.Mvc;
namespace APIExpo.Controllers
{
    [ApiController]
    [Route("[controller]")]
    0 references
    public class PrincipalController : ControllerBase
    {
        [HttpGet("ConsultarRegistro")]
        0 references
        public JsonResult Consultar(int ID)
        {
            var Consulta = new Consultas();
            var Lista = Consulta.ConsultaRegistro(ID);
            return new JsonResult(Lista);
        }
    }
}
```

Generamos el controlador de la API

Despliegue de la API en Azure AppServices

Demo del despliegue en Azure App Services

Aplicación para Android/iOS (.NET MAUI)

UI

```
<?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"
              xmlns:maps="clr-namespace:Microsoft.Maui.Controls.Maps;assembly=Microsoft.Maui.Controls.Maps"
              xmlns:local="clr-namespace:MauiAppExpo1.ViewModels"
              x:Class="MauiAppExpo1.MainPage"
              Title="">

    <ContentPage.BindingContext>
        <local:Modelo/>
    </ContentPage.BindingContext>

    <Grid>
        <!-- ImageView as background -->
        <Image Source="{Binding ImagenFondo}" Aspect="AspectFill" Grid.RowSpan="4"/>

        <Grid.RowDefinitions>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="Auto"/>
            <RowDefinition Height="*/>
            <RowDefinition Height="30*/>
            <!-- 30% of the screen height -->
        </Grid.RowDefinitions>
```


Aplicación para Android/iOS (.NET MAUI)

UI

```
<!-- TextBox and Button centered -->
<StackLayout Grid.Row="0" HorizontalOptions="Center">
    <StackLayout Orientation="Horizontal" Padding="10" HorizontalOptions="Center">
        <Entry x:Name="entryID" Placeholder="ID" WidthRequest="200" TextColor="Gray" Text="{Binding ID}"/>
        <Button Text="Consultar" Clicked="OnConsultarClicked"/>
    </StackLayout>

    <!-- Image with border -->
    <Frame BorderColor="White" CornerRadius="10" Padding="5" WidthRequest="210" HeightRequest="210" HorizontalOptions="Center">
        <Image Source="{Binding Imagen}" WidthRequest="200" HeightRequest="200" Aspect="AspectFill"/>
    </Frame>
</StackLayout>
```

Aplicación para Android/iOS (.NET MAUI)

UI

```
<!-- Labels with watermark style in a Frame for rounded corners -->
<Frame BackgroundColor="White" Opacity="0.7" CornerRadius="10" Padding="10" HorizontalOptions="Center">
  <StackLayout HorizontalOptions="Center" VerticalOptions="Start">
    <Label Text="{Binding Nombre}"
      FontAttributes="Bold"
      FontSize="Medium"
      TextColor="Black"
      HorizontalTextAlignment="Center"
      LineBreakMode="WordWrap"/>
    <Label Text="{Binding Apellido}"
      FontAttributes="Bold"
      FontSize="Medium"
      TextColor="Black"
      HorizontalTextAlignment="Center"
      LineBreakMode="WordWrap"/>
    <Label Text="{Binding Conferencia}"
      FontAttributes="Bold"
      FontSize="Medium"
      TextColor="Black"
      HorizontalTextAlignment="Center"
      LineBreakMode="WordWrap"/>
    <Label Text="{Binding Fecha}"
      FontAttributes="Bold"
      FontSize="Medium"
      TextColor="Black"
      HorizontalTextAlignment="Center"
      LineBreakMode="WordWrap"/>
    <Label Text="{Binding Pais}"
      FontAttributes="Bold"
      FontSize="Medium"
      TextColor="Black"
      HorizontalTextAlignment="Center"
      LineBreakMode="WordWrap"/>
    <Label Text="{Binding Semblanza}"
      FontAttributes="Bold"
      FontSize="Medium"
      TextColor="Black"
      HorizontalTextAlignment="Center"
      LineBreakMode="WordWrap"/>
  </StackLayout>
</Frame>
</Grid>
```

Aplicación para Android/iOS (.NET MAUI)

UI

```
        <!-- Map at the bottom -->  
        <maps:Map x:Name="map" Grid.Row="3" VerticalOptions="FillAndExpand" Margin="10"/>  
    </Grid>  
</ContentPage>
```


Aplicación para Android/iOS (.NET MAUI)

Model

```
using Microsoft.Maui.Maps;
using System.ComponentModel;
using System.Runtime.CompilerServices;
using System.Text.Json;
using System.Text.Json.Serialization;
using Maps = Microsoft.Maui.Controls.Maps.Map;

namespace MauiAppExpo1.ViewModels
{
    3 references
    public class Modelo : INotifyPropertyChanged
    {
        private string _imagen;
        private string _imagenFondo;
        private string _nombre;
        private string _apellido;
        private string _conferencia;
        private string _fecha;
        private string _pais;
        private string _semblanza;
        private double _latitud;
        private double _longitud;
        private string _id;
        private Maps _map;
        public event PropertyChangedEventHandler PropertyChanged;
        0 references
        public Modelo()
        {
            // Inicializa las propiedades con valores predeterminados o vacíos
            Imagen = "default_image.jpg"; // Ruta a una imagen predeterminada
            ImagenFondo = "default_background.jpg"; // Ruta a una imagen de fondo predeterminada
        }
    }
}
```

Aplicación para Android/iOS (.NET MAUI)

Model

```
1 reference
public string Imagen
{
    get => _imagen;
    set => SetProperty(ref _imagen, value);
}

2 references
public string ImagenFondo
{
    get => _imagenFondo;
    set => SetProperty(ref _imagenFondo, value);
}

1 reference
public string Nombre
{
    get => _nombre;
    set => SetProperty(ref _nombre, value);
}

1 reference
public string Apellido
{
    get => _apellido;
    set => SetProperty(ref _apellido, value);
}

1 reference
public string Conferencia
{
    get => _conferencia;
    set => SetProperty(ref _conferencia, value);
}

1 reference
public string Fecha
{
    get => _fecha;
    set => SetProperty(ref _fecha, value);
}
```

```
1 reference
public string Pais
{
    get => _pais;
    set => SetProperty(ref _pais, value);
}

1 reference
public string Semblanza
{
    get => _semblanza;
    set => SetProperty(ref _semblanza, value);
}

2 references
public double Latitud
{
    get => _latitud;
    set => SetProperty(ref _latitud, value);
}

2 references
public double Longitud
{
    get => _longitud;
    set => SetProperty(ref _longitud, value);
}

2 references
public string ID
{
    get => _id;
    set => SetProperty(ref _id, value);
}

3 references
public Maps Mapa
{
    get => _map;
    set => SetProperty(ref _map, value);
}
```

Aplicación para Android/iOS (.NET MAUI)

Model

```
public async Task ConsultarRegistroAsync()
{
    if (string.IsNullOrEmpty(ID))
        return;
    string url = $"https://apiexpo.azurewebsites.net/Principal/ConsultarRegistro?ID={ID}";
    using (HttpClient client = new HttpClient())
    {
        try
        {
            var response = await client.GetStringAsync(url);
            var registros = JsonSerializer.Deserialize<List<Registro>>(response);

            if (registros != null && registros.Count > 0)
            {
                var registro = registros[0];

                Nombre = registro.Nombre;
                Apellido = registro.Apellido;
                Conferencia = registro.Conferencia;
                Fecha = registro.Fecha;
                Imagen = registro.Imagen;
                ImagenFondo = registro.ImagenFondo;
                Pais = registro.Pais;
                Semblanza = registro.Semblanza;
                Latitud = double.Parse(registro.Latitud);
                Longitud = double.Parse(registro.Longitud);
                Location location = new Location(Latitud, Longitud);
                MapSpan mapSpan = new MapSpan(location, 0.01, 0.01);
                Mapa.MoveToRegion(mapSpan);
                Mapa.MapType = MapType.Satellite;
            }
        }
        catch (JsonException jsonEx)
        {
            // Captura y muestra el JSON recibido en caso de error de deserialización
            Console.WriteLine($"Error deserializando JSON: {jsonEx.Message}");
            Console.WriteLine(await client.GetStringAsync(url));
        }
        catch (Exception ex)
        {
            // Manejo de otros errores
            Console.WriteLine($"Error: {ex.Message}");
        }
    }
}
```


Aplicación para Android/iOS (.NET MAUI)

Model

```
public async Task ConsultarRegistroAsync()
{
    if (string.IsNullOrWhiteSpace(ID))
        return;
    string url = $"https://apiexpo.azurewebsites.net/Principal/ConsultarRegistro?ID={ID}";
    using (HttpClient client = new HttpClient())
    {
        try
        {
            var response = await client.GetStringAsync(url);
            var registros = JsonSerializer.Deserialize<List<Registro>>(response);

            if (registros != null && registros.Count > 0)
            {
                var registro = registros[0];

                Nombre = registro.Nombre;
                Apellido = registro.Apellido;
                Conferencia = registro.Conferencia;
                Fecha = registro.Fecha;
                Imagen = registro.Imagen;
                ImagenFondo = registro.ImagenFondo;
                Pais = registro.Pais;
                Semblanza = registro.Semblanza;
                Latitud = double.Parse(registro.Latitud);
                Longitud = double.Parse(registro.Longitud);
                Location location = new Location(Latitud, Longitud);
                MapSpan mapSpan = new MapSpan(location, 0.01, 0.01);
                Mapa.MoveToRegion(mapSpan);
                Mapa.MapType = MapType.Satellite;
            }
        }
        catch (JsonException jsonEx)
        {
            // Captura y muestra el JSON recibido en caso de error de deserialización
            Console.WriteLine($"Error deserializando JSON: {jsonEx.Message}");
            Console.WriteLine(await client.GetStringAsync(url));
        }
        catch (Exception ex)
        {
            // Manejo de otros errores
            Console.WriteLine($"Error: {ex.Message}");
        }
    }
}
```

Aplicación para Android/iOS (.NET MAUI)

Model

```
public class Registro
{
    [JsonPropertyName("nombre")]
    1 reference
    public string Nombre { get; set; }

    [JsonPropertyName("apellido")]
    1 reference
    public string Apellido { get; set; }

    [JsonPropertyName("conferencia")]
    1 reference
    public string Conferencia { get; set; }

    [JsonPropertyName("fecha")]
    1 reference
    public string Fecha { get; set; }

    [JsonPropertyName("imagen")]
    1 reference
    public string Imagen { get; set; }

    [JsonPropertyName("imagenFondo")]
    1 reference
    public string ImagenFondo { get; set; }

    [JsonPropertyName("pais")]
    1 reference
    public string Pais { get; set; }

    [JsonPropertyName("semblanza")]
    1 reference
    public string Semblanza { get; set; }

    [JsonPropertyName("latitud")]
    1 reference
    public string Latitud { get; set; }

    [JsonPropertyName("longitud")]
    1 reference
    public string Longitud { get; set; }
}
```

Aplicación para Android/iOS (.NET MAUI)

Main Page

```
using MauiAppExpo1.ViewModels;
using Microsoft.Maui.Devices;
using CommunityToolkit.Maui.Core;
using CommunityToolkit.Maui.Extensions;
using CommunityToolkit.Maui.Core.Platform;

namespace MauiAppExpo1
{
    4 references
    public partial class MainPage : ContentPage
    {
        private Modelo _viewModel;

        0 references
        public MainPage()
        {
            InitializeComponent();
            _viewModel = (Modelo)BindingContext;
            _viewModel.Mapa = map; // Asignar el control del mapa al ViewModel
        }

        0 references
        private async void OnConsultarClickado(object sender, EventArgs e)
        {
            HideKeyboard();
            await _viewModel.ConsultarRegistroAsync();
        }

        1 reference
        private async void HideKeyboard()
        {
            if (KeyboardExtensions.IsSoftKeyboardShowing(entryID))
            {
                await KeyboardExtensions.HideKeyboardAsync(entryID);
            }
        }
    }
}
```


Aplicación para Android/iOS (.NET MAUI)

Main Page

```
using MauiAppExpo1.ViewModels;
using Microsoft.Maui.Devices;
using CommunityToolkit.Maui.Core;
using CommunityToolkit.Maui.Extensions;
using CommunityToolkit.Maui.Core.Platform;

namespace MauiAppExpo1
{
    4 references
    public partial class MainPage : ContentPage
    {
        private Modelo _viewModel;

        0 references
        public MainPage()
        {
            InitializeComponent();
            _viewModel = (Modelo)BindingContext;
            _viewModel.Mapa = map; // Asignar el control del mapa al ViewModel
        }

        0 references
        private async void OnConsultarClickado(object sender, EventArgs e)
        {
            HideKeyboard();
            await _viewModel.ConsultarRegistroAsync();
        }

        1 reference
        private async void HideKeyboard()
        {
            if (KeyboardExtensions.IsSoftKeyboardShowing(entryID))
            {
                await KeyboardExtensions.HideKeyboardAsync(entryID);
            }
        }
    }
}
```

Aplicación para Android/iOS (.NET MAUI)

(iOS) Info.plist

```
?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>LSRequiresIPhoneOS</key>
  <true/>
  <key>UIDeviceFamily</key>
  <array>
    <integer>1</integer>
    <integer>2</integer>
  </array>
  <key>GMSApiKey</key>
  <string>AIzaSyD55DWJoDnohbZLzslTGK-AH77-9YqcQEw</string>
  <key>NSLocationAlwaysAndWhenInUseUsageDescription</key>
  <string>Can we use your location at all times?</string>
  <key>NSLocationWhenInUseUsageDescription</key>
  <string>Can we use your location when your app is being used?</string>
  <key>UIRequiredDeviceCapabilities</key>
  <array>
    <string>arm64</string>
  </array>
  <key>UISupportedInterfaceOrientations</key>
  <array>
    <string>UIInterfaceOrientationPortrait</string>
    <string>UIInterfaceOrientationLandscapeLeft</string>
    <string>UIInterfaceOrientationLandscapeRight</string>
  </array>
  <key>UISupportedInterfaceOrientations~ipad</key>
  <array>
    <string>UIInterfaceOrientationPortrait</string>
    <string>UIInterfaceOrientationPortraitUpsideDown</string>
    <string>UIInterfaceOrientationLandscapeLeft</string>
    <string>UIInterfaceOrientationLandscapeRight</string>
  </array>
  <key>XSAAppIconAssets</key>
  <string>Assets.xcassets/appicon.appiconset</string>
  <key>CFBundleIdentifier</key>
  <string>com.enriqueaguilar.PruebaMaui</string>
</dict>
</plist>
```

(Android) AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
  <application android:allowBackup="true" android:icon="@mipmap/appicon" android:roundIcon="@mipmap/appicon_round"
    <meta-data
      android:name="com.google.android.geo.API_KEY"
      android:value="AIzaSyD55DWJoDnohbZLzslTGK-AH77-9YqcQEw" />
    </application>
  <uses-permission android:name="android.permission.INTERNET" />
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
</manifest>
```

Aplicación para Android/iOS (.NET MAUI)

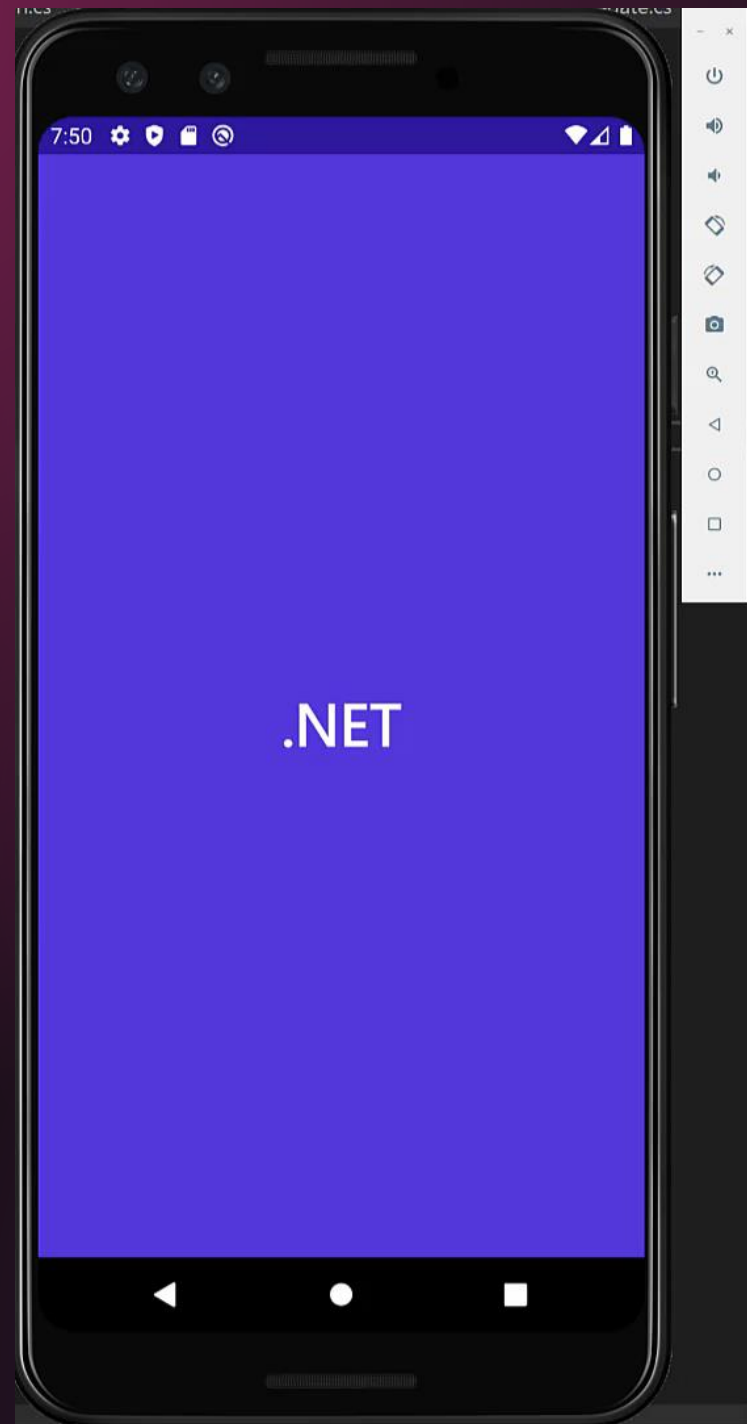
(iOS) Info.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>LSRequiresIPhoneOS</key>
  <true/>
  <key>UIDeviceFamily</key>
  <array>
    <integer>1</integer>
    <integer>2</integer>
  </array>
  <key>GMSApiKey</key>
  <string>AIzaSyD55DWJoDnohbZLzslTGK-AH77-9YqcQEw</string>
  <key>NSLocationAlwaysAndWhenInUseUsageDescription</key>
  <string>Can we use your location at all times?</string>
  <key>NSLocationWhenInUseUsageDescription</key>
  <string>Can we use your location when your app is being used?</string>
  <key>UIRequiredDeviceCapabilities</key>
  <array>
    <string>arm64</string>
  </array>
  <key>UISupportedInterfaceOrientations</key>
  <array>
    <string>UIInterfaceOrientationPortrait</string>
    <string>UIInterfaceOrientationLandscapeLeft</string>
    <string>UIInterfaceOrientationLandscapeRight</string>
  </array>
  <key>UISupportedInterfaceOrientations~ipad</key>
  <array>
    <string>UIInterfaceOrientationPortrait</string>
    <string>UIInterfaceOrientationPortraitUpsideDown</string>
    <string>UIInterfaceOrientationLandscapeLeft</string>
    <string>UIInterfaceOrientationLandscapeRight</string>
  </array>
  <key>XSAppIconAssets</key>
  <string>Assets.xcassets/appicon.appiconset</string>
  <key>CFBundleIdentifier</key>
  <string>com.enriqueaguilar.PruebaMaui</string>
</dict>
</plist>
```

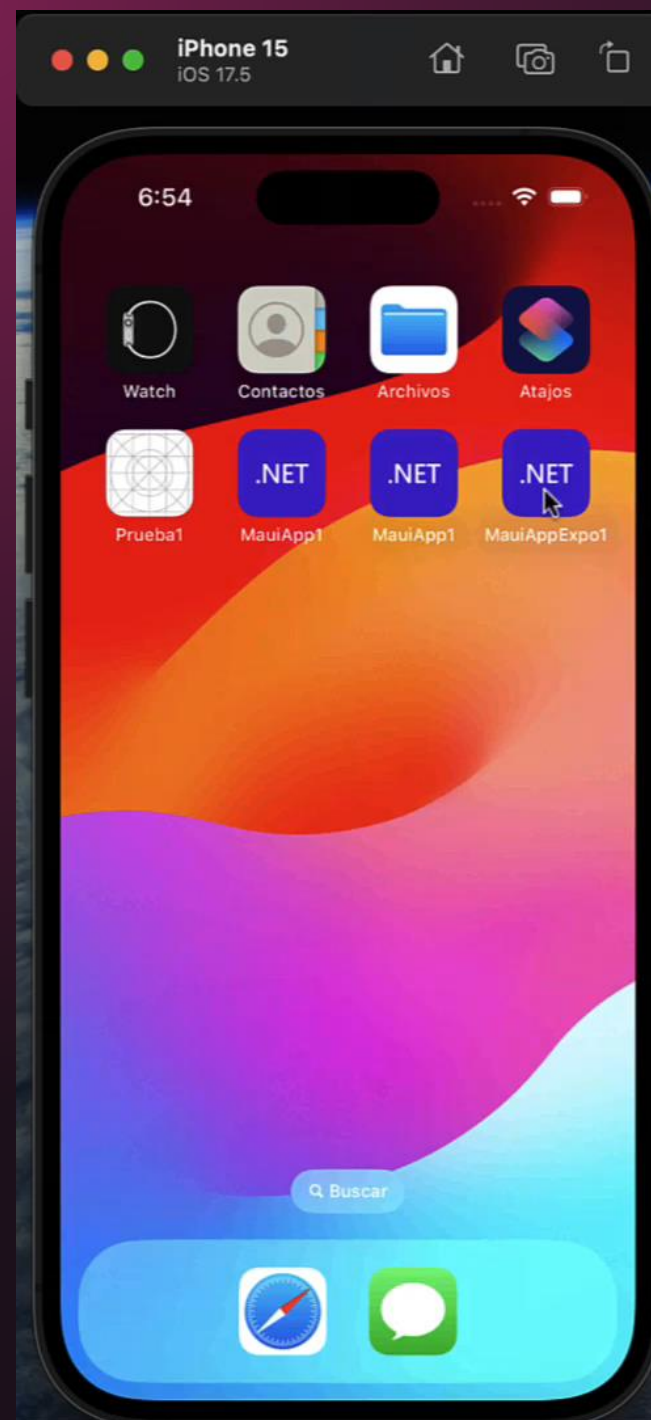
(Android) AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android">
  <application android:allowBackup="true" android:icon="@mipmap/appicon" android:roundIcon="@mipmap/appicon_round">
    <meta-data
      android:name="com.google.android.geo.API_KEY"
      android:value="AIzaSyD55DWJoDnohbZLzslTGK-AH77-9YqcQEw" />
    </application>
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
  </manifest>
```

Pruebas de Ejecución para Android



Pruebas de Ejecución para iOS



Código disponible en:

github.com/enriqueaguilarvargas/MAUIAPIAZURE/

@enriqueaguilar
enriqueaguilar@outlook.com



CONSEJO DEPARTAMENTAL DE AYACUCHO
CAPÍTULO DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Acceso de Aplicaciones Móviles (.NET MAUI) a API's sobre Azure

DR. ENRIQUE AGUILAR VARGAS

