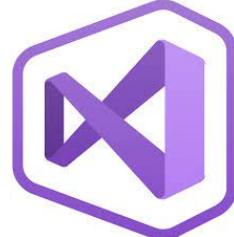


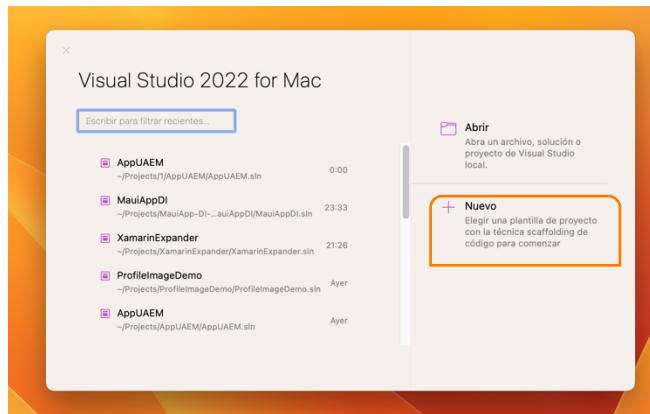
# Workshop .NET MAUIUAEM



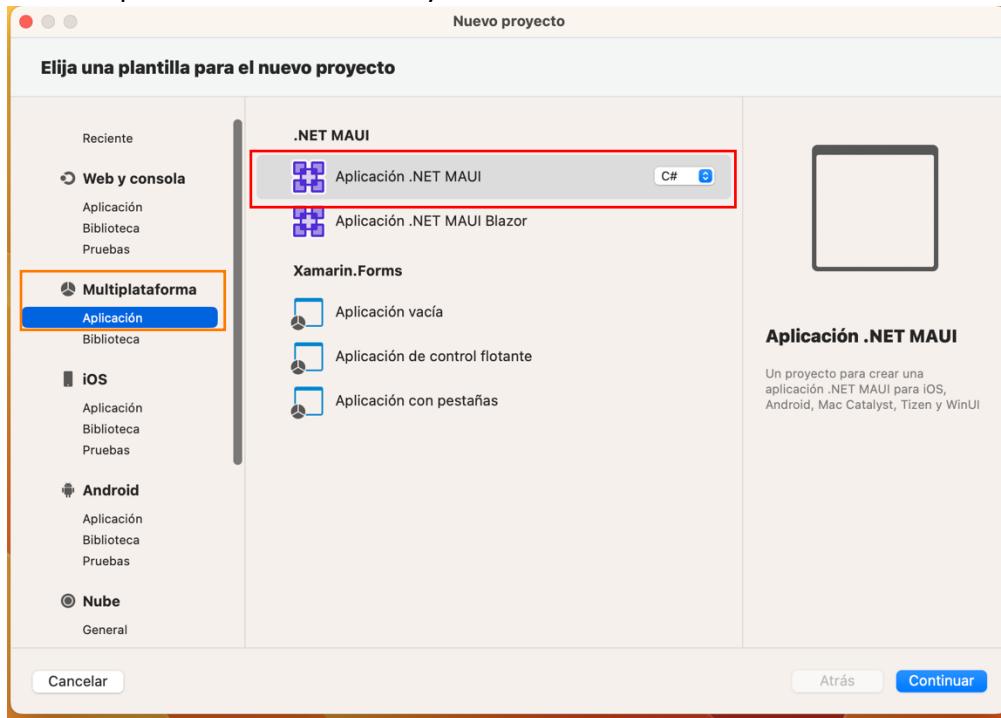
1. Los recursos de este proyecto puedes encontrarlos en el siguiente repositorio de Github [BryanOroxon/AppUAEM \(github.com\)](https://github.com/BryanOroxon/AppUAEM)
2. Abrir Visual Studio 2022 yo utilizare el Visual Studio for Mac pero igual puedes seguir los pasos en Visual Studio 2022 en Windows.



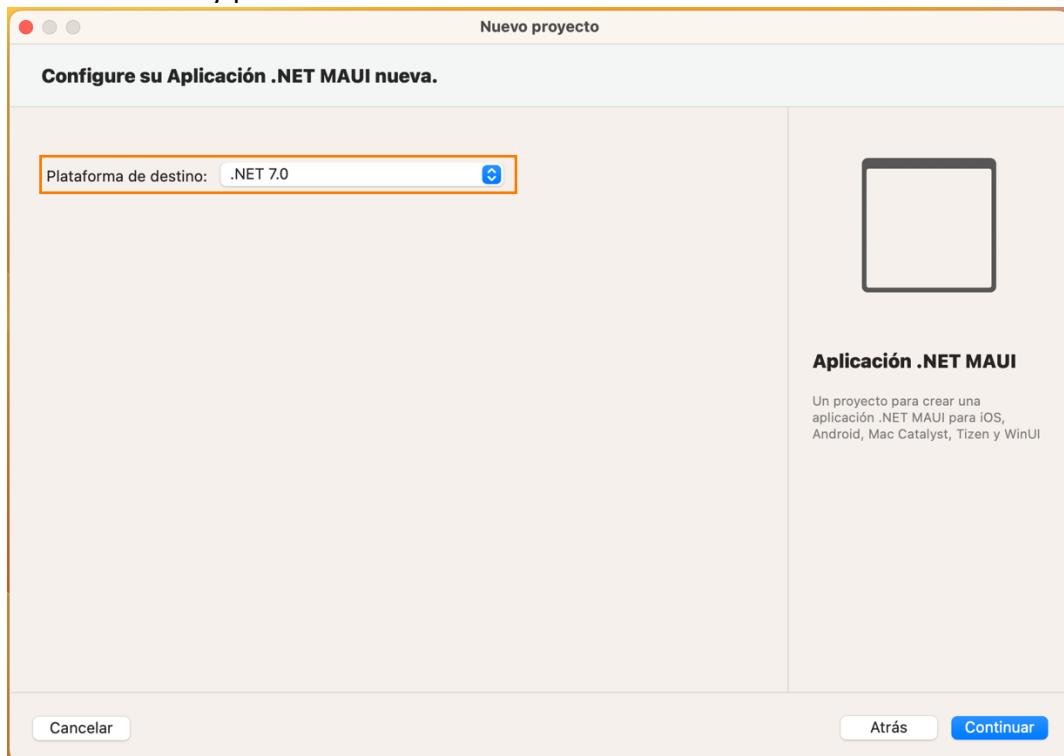
3. Seleccionar “Nuevo” para crear un proyecto Nuevo



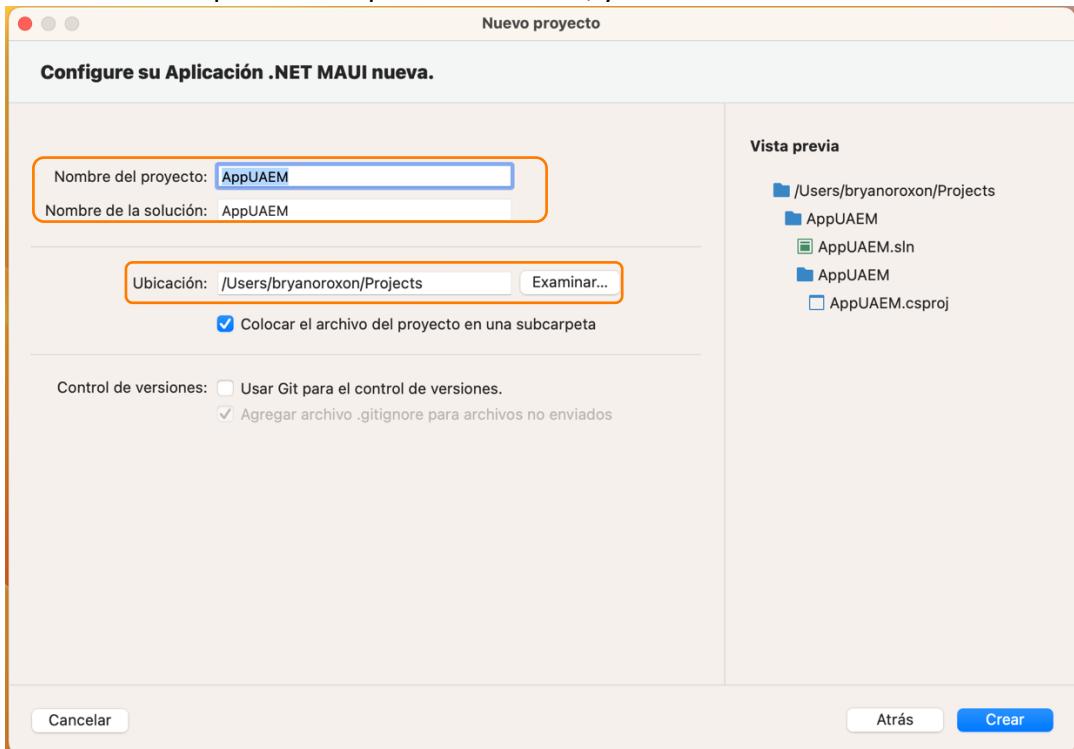
4. En las Plantillas Seleccione Multiplataforma y Aplicación, luego en .NET MAUI Aplicación .Net MAUI C# y el botón Continuar.



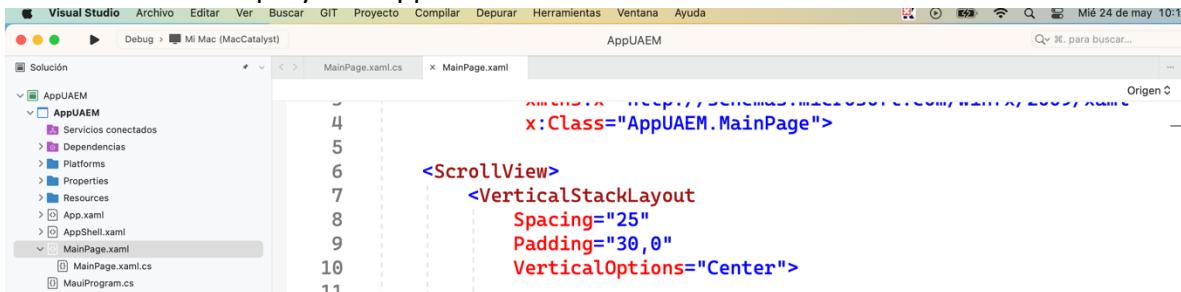
5. En Configura su Aplicación .NET MAUI nueva en plataforma de destino, Seleccione .Net 7.0 y presione el botón continuar



6. El nombre del proyecto y el nombre de la solución será “AppUAEM”, elija una ruta corta de preferencia para la ubicación, y de clic en el botón Crear

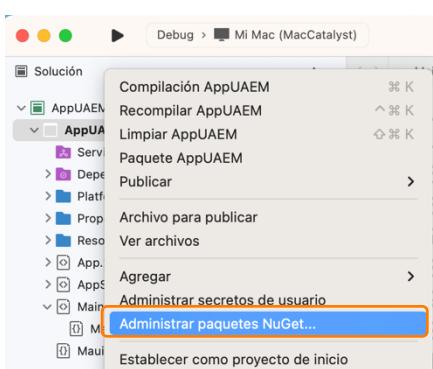


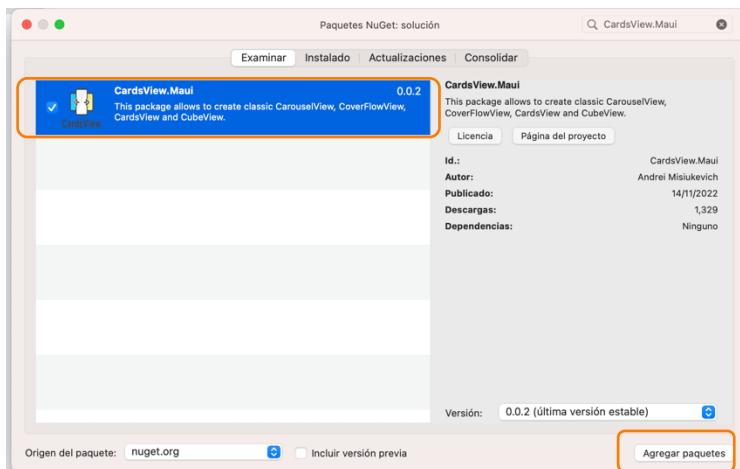
## 7. Se creará el proyecto AppUAEM



8. Con el proyecto creado necesitamos agregar 2 Nugets a nuestro proyecto, para agregarlo de clic derecho sobre la solución

Y busque la opción Administrar paquetes Nuget

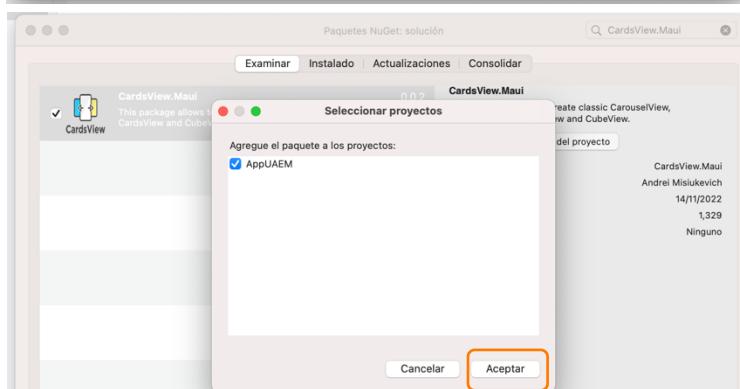




9. Se mostrará la ventana de Agregar Paquetes Nuget Asegúrese de estar en la Pestaña Examinar (Browse) y en el Cuadro de búsqueda escriba **CardView.Maui**

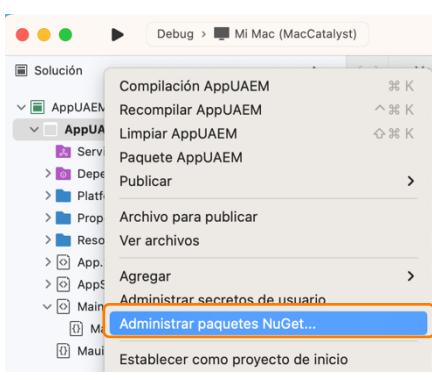
Se usará CardsView.Maui versión 0.0.2

De clic en Agregar paquetes.



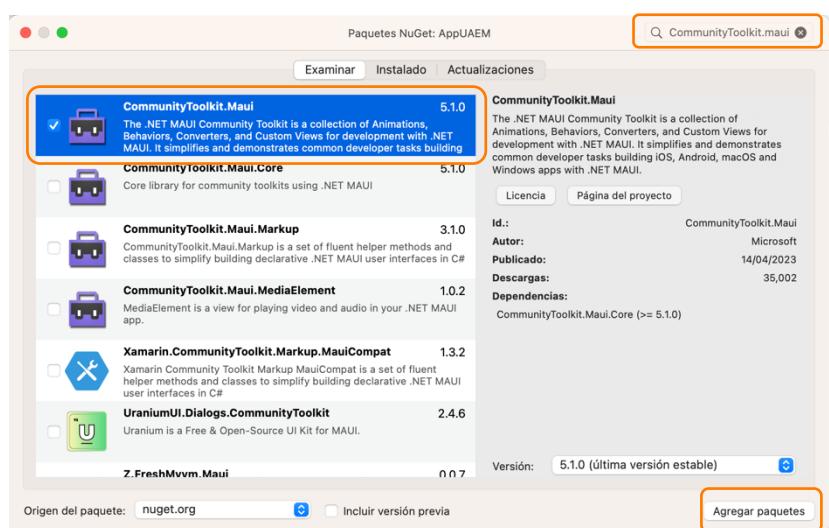
10. Se pedirá que acepte agregar el paquete

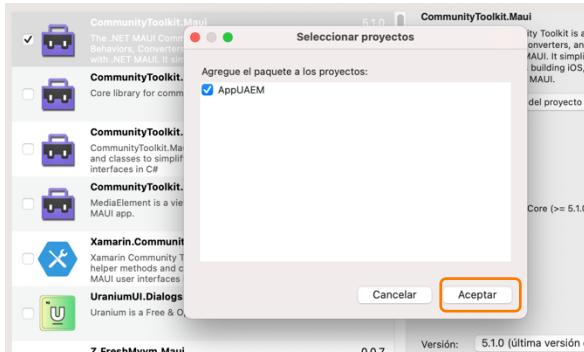
De clic en el botón Aceptar



11. Repetiremos el proceso dar clic derecho sobre la solución y seleccionar **Administrar paquetes NuGet**

12. Ahora busque el **CommunityToolkit.Maui**, la versión 5.1.0 y de clic en Agregar paquetes

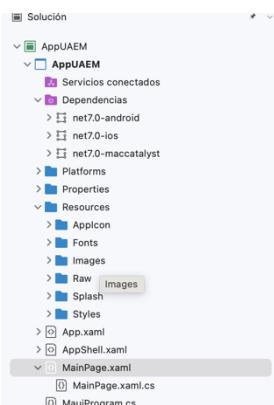




13. De clic en el botón Aceptar para agregar los paquetes.

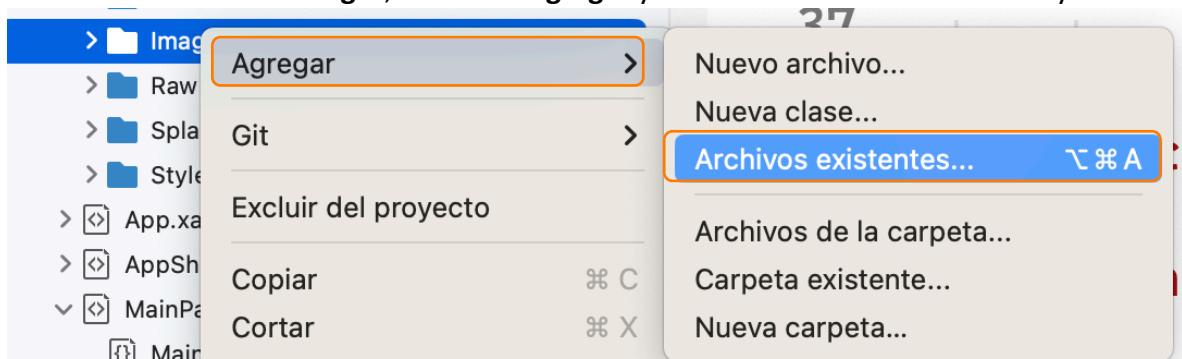


14. Acepte la licencia de uso al dar clic en el botón Aceptar.



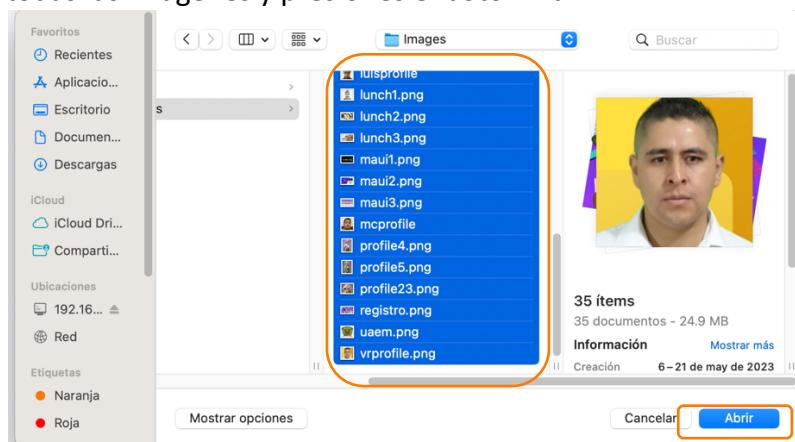
15. Ahora agregue los recursos que se deben descargar del sitio de Github.  
Seleccione la Carpeta **Resources** y expándala.

16. De clic derecho sobre **Images**, selección **Agregar** y selección **contenido Existente** y

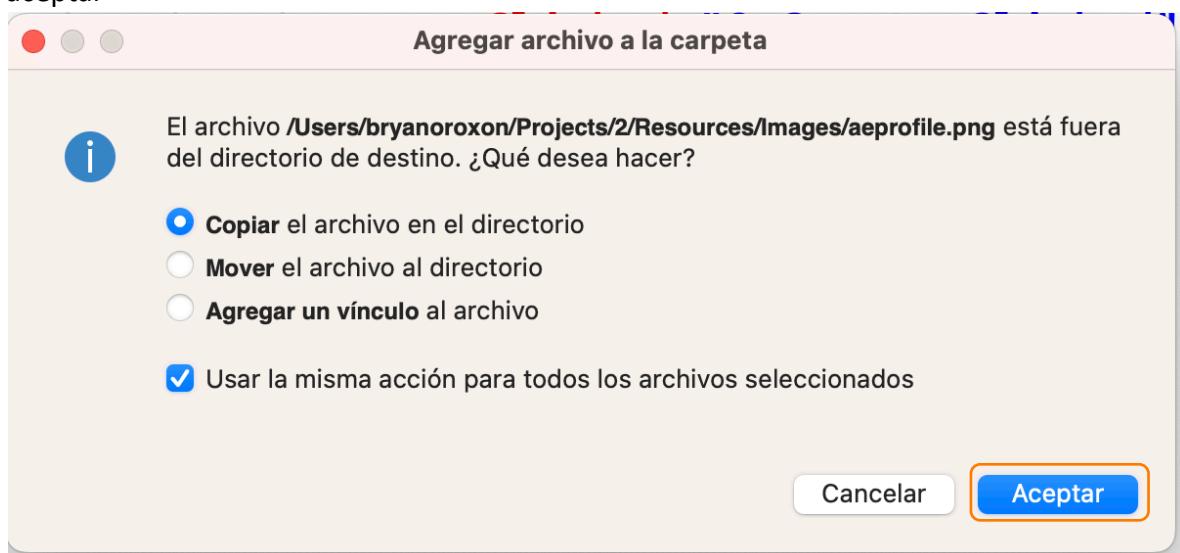


17. Busque en folder donde haya descargado los recursos abra la carpeta **Resources** y abra **Images** Seleccione la primera imagen y presione la tecla **Shift** y seleccione

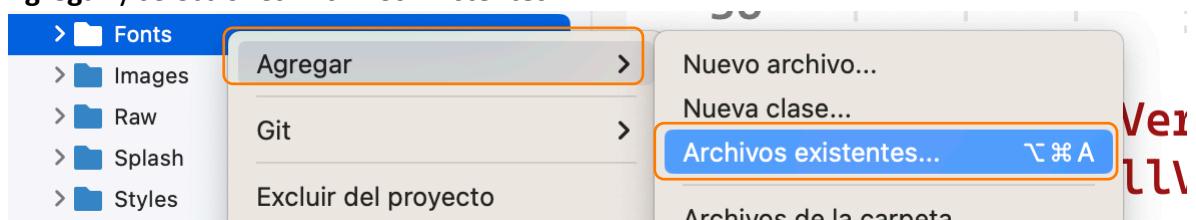
todas las imágenes y presiones el botón **Abrir**.



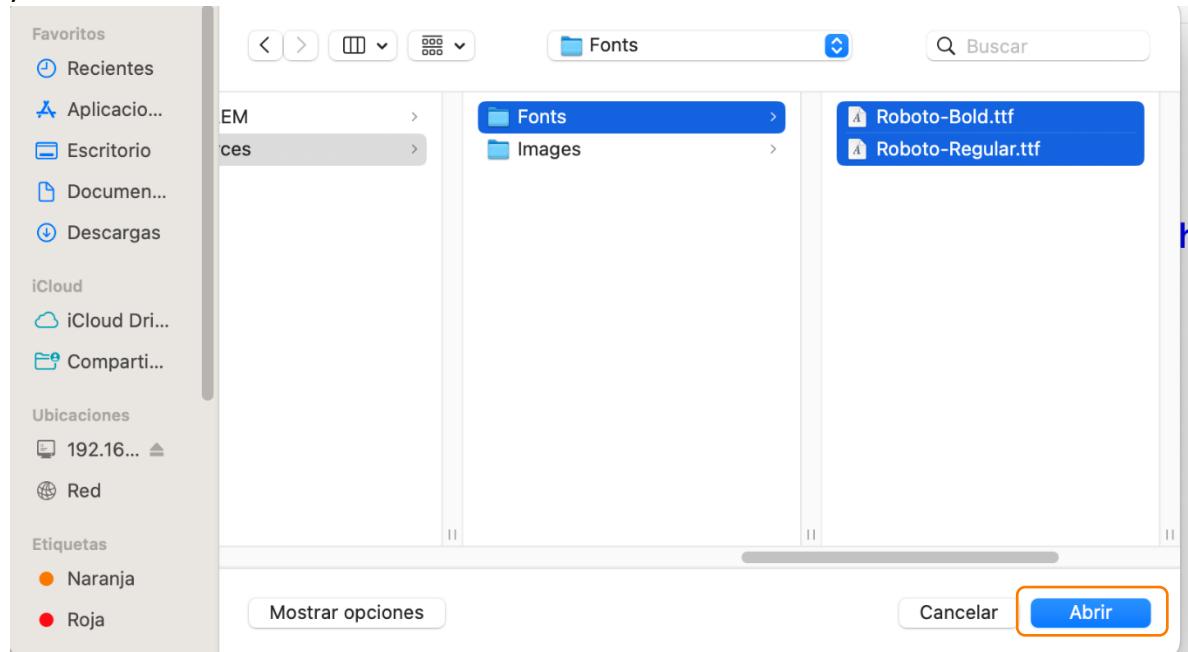
18. Se le preguntara si quiere Copiar el archivo en el directorio los recursos de clic en aceptar



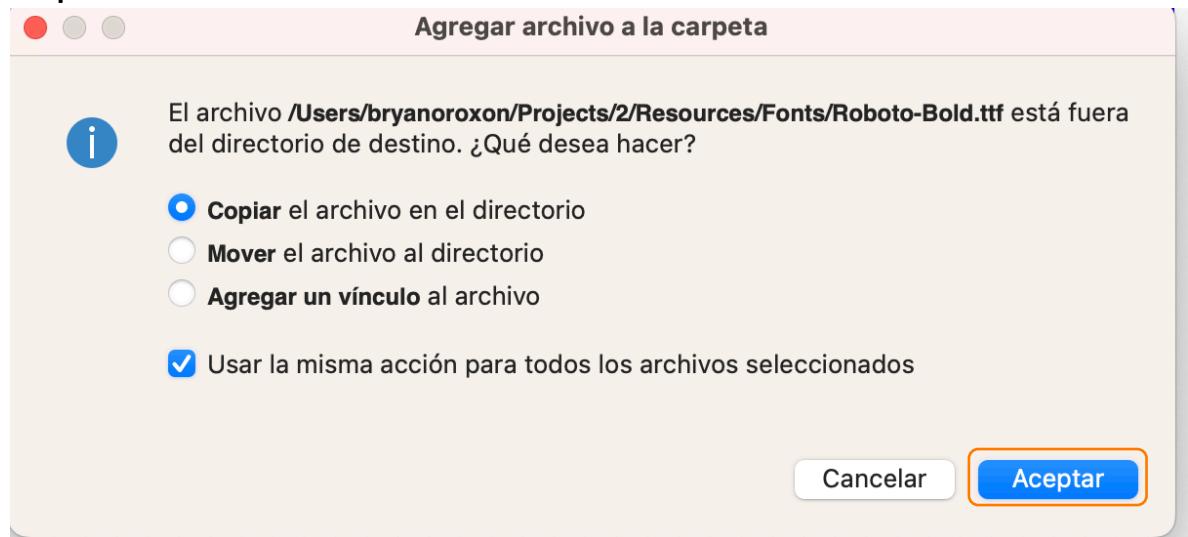
19. Repetiremos estos pasos para Agregar las fuentes De clic derecho sobre **Fonts** Agregar y selecciones **Archivos Existentes**.



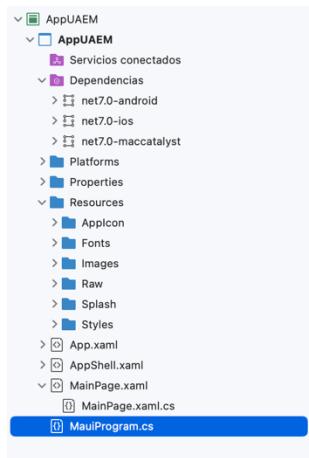
20. Seleccione en la carpeta Resources, la carpeta Fonts y seleccione todas las fuentes y de clic en Abrir.



21. Se le preguntara si quiere Copiar el archivo en el directorio de clic en el botón Aceptar.



22. Ahora modificaremos el Archivo MauiProgram.cs lo buscaremos en el explorador de Solución de doble clic sobre él para que se abra.



```
1  using Microsoft.Extensions.Logging;
2
3  namespace AppUAEM;
4
5  public static class MauiProgram
6  {
7      public static MauiApp CreateMauiApp()
8      {
9          var builder = MauiApp.CreateBuilder();
10         builder
11             .UseMauiApp<App>()
12             .ConfigureFonts(fonts =>
13             {
14                 fonts.AddFont("OpenSans-Regular.ttf", "OpenSansRegular");
15                 fonts.AddFont("OpenSans-Semibold.ttf", "OpenSansSemibold");
16             });
17     }
18 }
19
20
```

23. En la línea 11 del código agregue las siguientes dos líneas

```
.UseCardsView()
.UseMauiCommunityToolkit()
```

```
11
12
13     builder
14         .UseCardsView()
15             .UseMauiCommunityToolkit()
16             .UseMauiApp<App>()
17             .ConfigureFonts(fonts =>
18             {
19                 fonts.AddFont("OpenSans-Regular.ttf", "OpenSansRegular");
20                 fonts.AddFont("OpenSans-Semibold.ttf", "OpenSansSemibold");
21             });
22
23
```

24. Al hacerlo se deben agregar los usings pero si no lo hace puedes agregar en la línea 2 del código los siguientes Usings.

```
using PanCardView;
using CommunityToolkit.Maui;
```

```
1  using Microsoft.Extensions.Logging;
2  using PanCardView;
3  using CommunityToolkit.Maui;
4
```

25. En la línea de código 20 agregamos los fonts con estas líneas de código

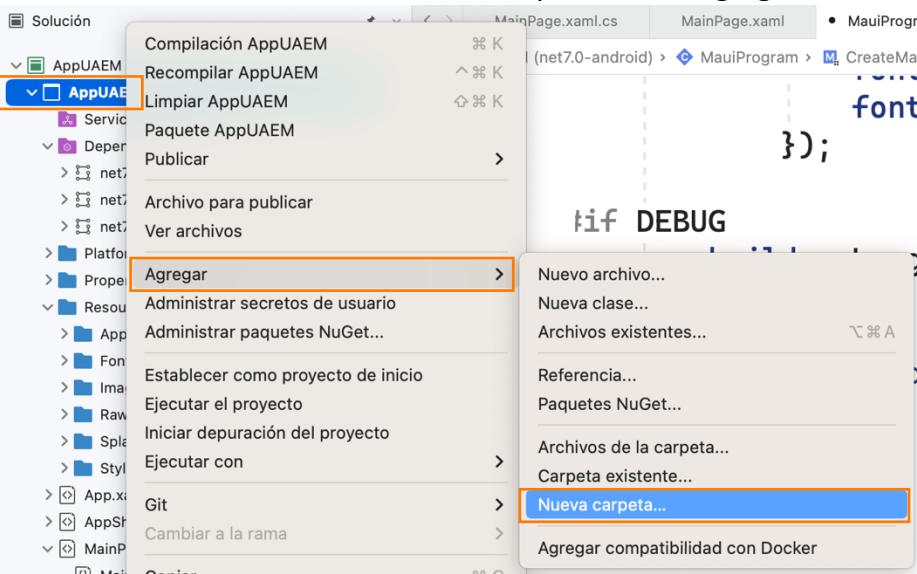
```
fonts.AddFont("Roboto-Bold.ttf", "RobotoBold");
fonts.AddFont("Roboto-Regular.ttf", "RobotoRegular");
```

```

17
18     fonts.AddFont("OpenSans-Regular.ttf", "OpenSansRegular");
19     fonts.AddFont("OpenSans-Semibold.ttf", "OpenSansSemibold");
20     fonts.AddFont("Roboto-Bold.ttf", "RobotoBold");
21     fonts.AddFont("Roboto-Regular.ttf", "RobotoRegular");
22 }
23

```

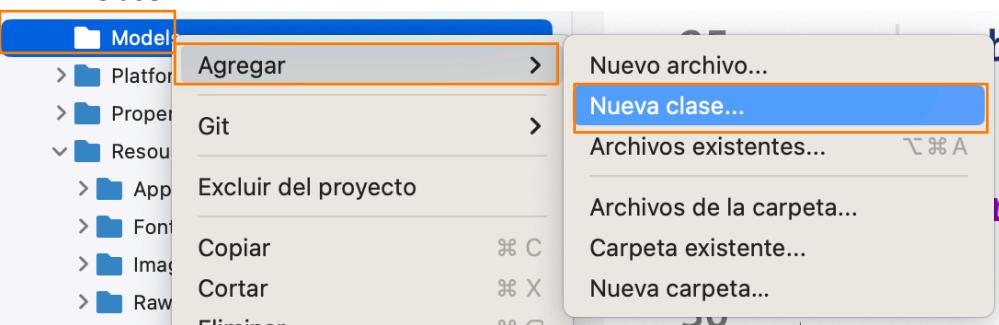
26. Muy bien ahora presione las teclas Command + S, (Ctrl+S) para guardar el archivo  
 27. Dar clic derecho sobre la solución y dar clic en **Agregar**, Seleccione **Nueva Carpeta**



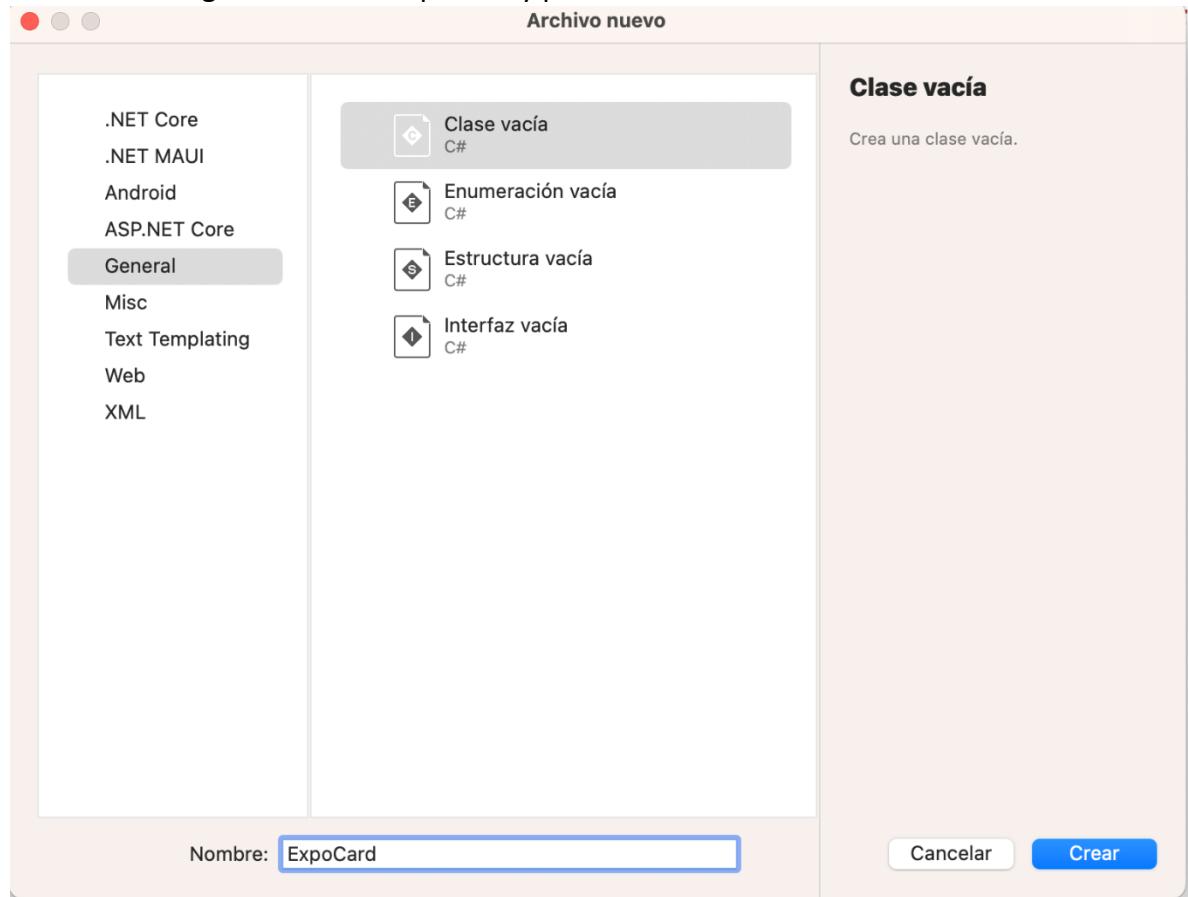
28. En la ventana nueva escriba Models y de clic en Aceptar.



29. Seleccione la carpeta Models y de clic derecho seleccione **Agregar** y Agregar **Nueva Clase**



30. Se abrirá la ventana de archivo nuevo, del lado izquierdo seleccione **General** y **Clase Vacía** luego En nombre ExpoCard y presione el botón **Crear**

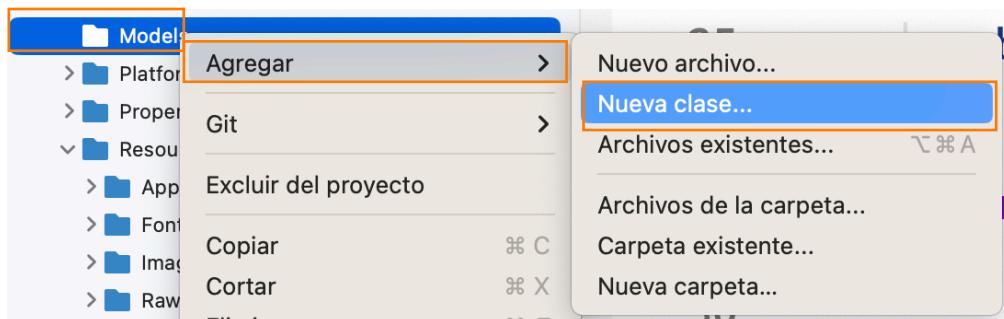


31. Reemplaza todo el código con el siguiente código  
using System;

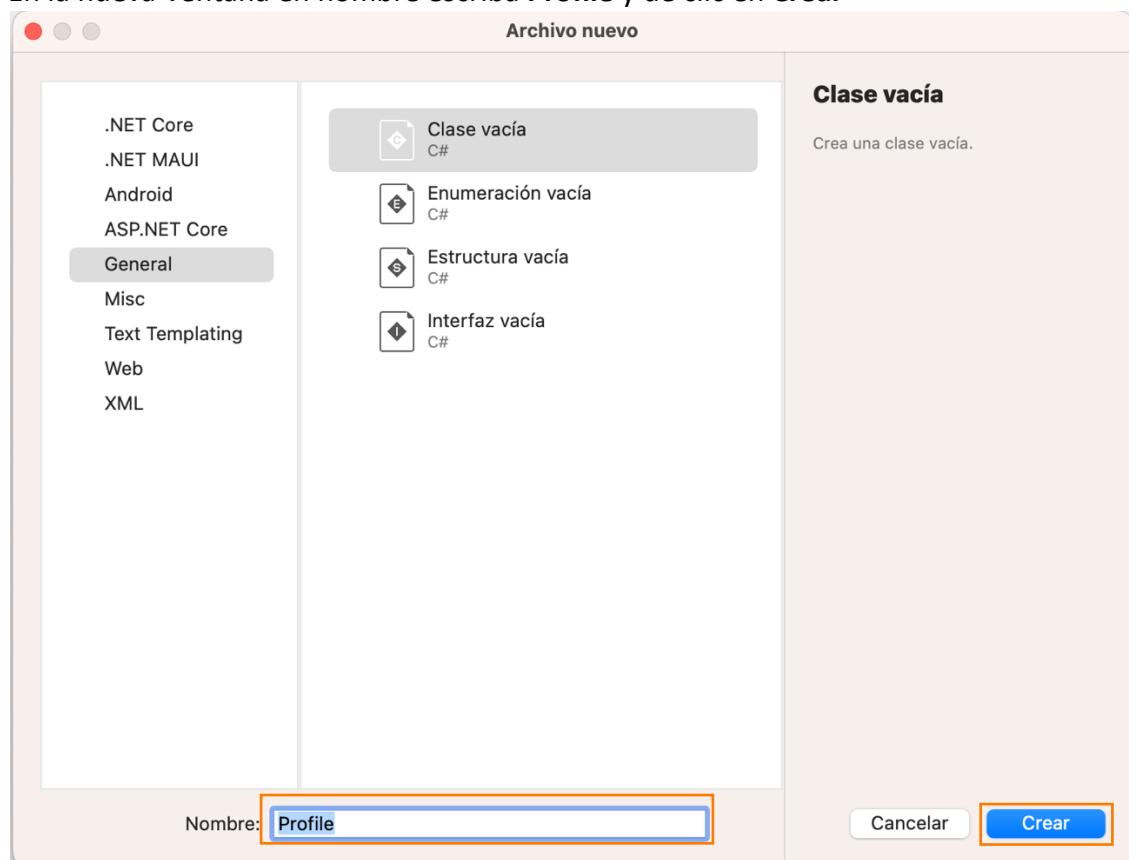
```
namespace AppUAEM.Models
{
    public class ExpoCard
    {
        public string ImagePath { get; set; }
        public string Subject { get; set; }
        public string Hour { get; set; }
        public List<string> Speaker { get; set; }
    }
}
```

32. Presione las teclas Command + S, (Ctrl+S) para guardar el archivo

33. Seleccione la carpeta Models y de clic derecho seleccione **Agregar** y Agregar **Nueva Clase**



34. En la nueva Ventana en nombre escriba **Profile** y de clic en **Crear**

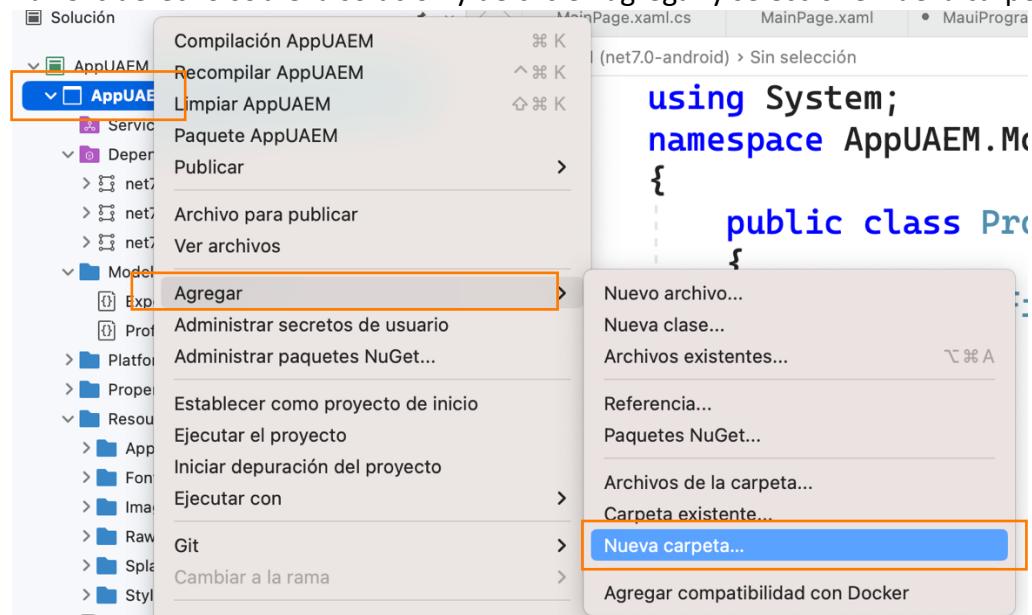


35. Reemplaza todo el código con el siguiente código:  
using System;

```
namespace AppUAEM.Models
{
    public class Profile
    {
        public string ProfileImage { get; set; }
        public string Name { get; set; }
        public string Title { get; set; }
    }
}
```

36. Presione las teclas Command + S, (Ctrl+S) para guardar el archivo

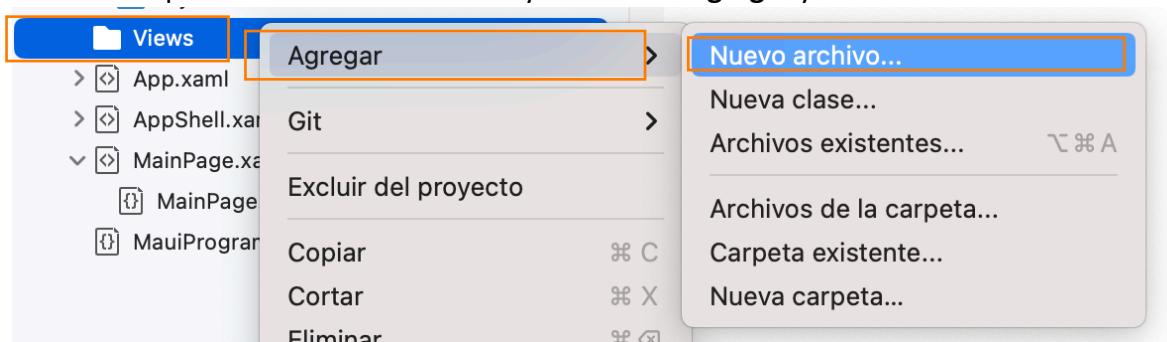
37. Dar Clic derecho sobre la solución y de clic en agregar y seleccione Nueva carpeta



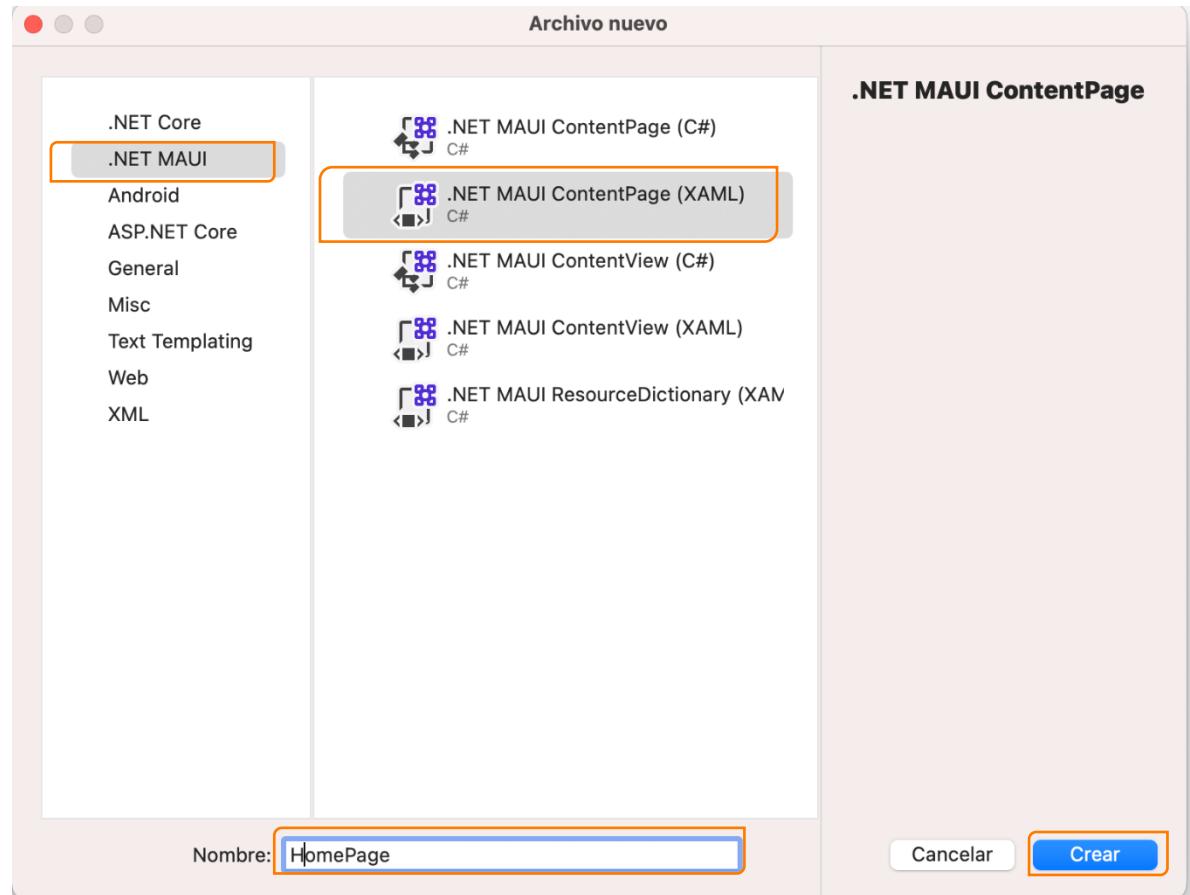
38. En la nueva ventana escriba Views y de clic en agregar.



39. Sobre la Carpeta Views de Clic derecho y seleccione Agregar y Nuevo Archivos



40. En la nueva ventana en las plantillas del lado izquierdo Seleccione **.Net MAUI**, luego selecciones **.NET MAUI ContentPage (XAML)**, en la parte inferior en **Nombre:** dentro del cuadro de texto escriba el nombre **HomePage** y de clic en **Crear**.



41. Se creara la Página HomePage se mostrara el siguiente código.

```
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="AppUAEM.Views.HomePage"
5      Title="HomePage">
6      <VerticalStackLayout>
7          <Label
8              Text="Welcome to .NET MAUI!"
9              VerticalOptions="Center"
10             HorizontalOptions="Center" />
11     </VerticalStackLayout>
12 </ContentPage>
13
```

42. Pero remplazaremos el código con el siguiente Código:

```
<?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:cards="clr-namespace:PanCardView;assembly=PanCardView"
xmlns:proc="clr-namespace:PanCardView.Processors;assembly=PanCardView"
xmlns:toolkit="http://schemas.microsoft.com/dotnet/2022/maui/toolkit"
xmlns:mct="clr-
namespace:CommunityToolkit.Maui.Behaviors;assembly=CommunityToolkit.Maui"
x:Class="AppUAEM.Views.HomePage"
Title="HomePage"
BackgroundColor="{AppThemeBinding Light=#545583, Dark=Black}"
xmlns:ios="clr-
namespace:Microsoft.Maui.Controls.PlatformConfiguration.iOS;assembly=Microsof
t.Maui.Controls"
    ios:Page.UseSafeArea="False">
<ContentPage.Behaviors>
    <mct:StatusBarBehavior StatusBarColor="{AppThemeBinding Light=#545583,
Dark=Black}"/>
</ContentPage.Behaviors>
<ContentPage.Resources>
    <ResourceDictionary>
        <toolkit:StringToListConverter x:Key="StringToListConverter" Separator=", " />
    </ResourceDictionary>
</ContentPage.Resources>
<ScrollView>
<Grid Margin="25,25,25,0" RowDefinitions=".1*,.2*,.52*,90" >
    <VerticalStackLayout Grid.Row="0" VerticalOptions="Center" >
        <Label FontSize="30"
            FontFamily="RobotoBold"
            HorizontalTextAlignment="Center"
            Text="Universidad Autonoma del Estado de México"
            TextColor="White" />
        <Label Margin="0,10,0,0"
            FontFamily="RobotoBold"
            HorizontalTextAlignment="Center"
            FontSize="20"
            Text="Gracias por participar en este evento."
            TextColor="{AppThemeBinding Light=#ACB1BB, Dark=#f0f0f0} " />
    </VerticalStackLayout>
<Grid Grid.Row="1" Margin="15,30,15,0" ColumnDefinitions="*,*">
    <VerticalStackLayout VerticalOptions="Center" >
        <Border BackgroundColor="Purple"
            StrokeShape="RoundRectangle 40,3,1,40" >
            <Border.Stroke>
                <LinearGradientBrush EndPoint="0,1">
                    <GradientStop Color="Orange" Offset="0.1" />
                    <GradientStop Color="OrangeRed" Offset="1.0" />
            </Border.Stroke>
    </VerticalStackLayout>
</Grid>

```

```

        </LinearGradientBrush>
    </Border.Stroke>
    <Border.Shadow>
        <Shadow Brush="{AppThemeBinding Light=White, Dark=Black}"
            Offset="10,20"
            Opacity=".6" />
    </Border.Shadow>
    <Label Text="27 de mayo"
        TextColor="{AppThemeBinding Light=White, Dark=#e7e6ff}"
        FontSize="25"
        HorizontalTextAlignment="Center"
        VerticalTextAlignment="Center"
        FontAttributes="Bold" />
</Border>

</VerticalStackLayout>
<Frame
    Grid.Column="1"
    HeightRequest="110"
    WidthRequest="110"
    CornerRadius="55"
    Background="Transparent"
    HorizontalOptions="Center"
    IsClippedToBounds="True"
    Padding="0"
    BorderColor="#00d2ff"
    Margin="0,0,0">
    <Image
        Aspect="AspectFit"
        Source="uaem"
        SemanticProperties.Description="Logo UAEM!"
        HeightRequest="90"
        WidthRequest="90"
        VerticalOptions="Center"
        HorizontalOptions="Center" />
</Frame>
</Grid>
<CollectionView
    Grid.Row="3"
    ItemsSource="{Binding Profiles}"
    >
    <CollectionView.ItemsLayout >
        <LinearItemsLayout ItemSpacing="10" Orientation="Horizontal" />

```

```

    </CollectionView.ItemsLayout>
<CollectionView.ItemTemplate>
    <DataTemplate>
        <Grid
            ColumnDefinitions=".35*, .65*"
            HeightRequest="60"
            WidthRequest="160" >
            <RoundRectangle Grid.ColumnSpan="2"
                CornerRadius="45"
                Fill="{AppThemeBinding Light=#323248, Dark=#313131}" />
            <Frame
                Grid.RowSpan="2"
                HeightRequest="50"
                WidthRequest="50"
                CornerRadius="25"
                HorizontalOptions="Center"
                IsClippedToBounds="True"
                Padding="0"
                BorderColor="#00d2ff"
                Margin="5,0,0,0" >
                <Image HeightRequest="55" Source="{Binding ProfileImage}"
                    Aspect="AspectFit"
                    VerticalOptions="Center"
                    HorizontalOptions="Center" />
            </Frame>

            <VerticalStackLayout Grid.Column="1"
                VerticalOptions="Center">
                <Label Margin="10,0,0,0"
                    FontFamily="RobotoBold"
                    HorizontalTextAlignment="Start"
                    Text="{Binding Name}"
                    TextColor="{AppThemeBinding Light=White, Dark=#e7e6ff}"
                    VerticalTextAlignment="Center" />
                <Label Grid.Column="1"
                    Margin="10,0,0,0"
                    FontFamily="RobotoBold"
                    FontSize="12"
                    Text="{Binding Title}"
                    TextColor="{AppThemeBinding Light=#099FFF, Dark=#0c81ed}"
                    VerticalOptions="Start"/>
            </VerticalStackLayout>
        </Grid>
    </DataTemplate>
</CollectionView.ItemTemplate>

```

```

        </DataTemplate>

    </CollectionView.ItemTemplate>
</CollectionView>

<cards:CoverflowView Grid.Row="2"
    Margin="-25,0,25,0"

        ItemsSource="{Binding ExpoCards}"
        PositionShiftValue="125">
<x:Arguments>
    <proc:CoverflowProcessor OpacityFactor="0.15" ScaleFactor="0.90"/>
</x:Arguments>
<cards:CoverflowView.ItemTemplate>
<DataTemplate>
    <Border StrokeShape="RoundRectangle 25,25,25,25"
        StrokeThickness="0">
        <Grid RowDefinitions=".5*, .5*>
            <Image
                Grid.RowSpan="2"
                Aspect="AspectFill"
                MaximumHeightRequest="400"
                MaximumWidthRequest="280"
                Source="{Binding ImagePath}" />
            <Grid Grid.Row="1">
                <Rectangle Fill="Black"
                    Stroke="White"
                    Opacity=".8"
                    StrokeThickness="0" />
            <VerticalStackLayout Margin="10,20,0,0" VerticalOptions="Start">
                <Label Margin="15,0,0,0"
                    FontFamily="RobotoBold"
                    FontSize="22"
                    Text="{Binding Subject}"
                    TextColor="White" />
                <Label
                    Margin="15,5,0,0"
                    FontFamily="RobotoRegular"
                    FontSize="20"
                    TextColor="{AppThemeBinding Light=White, Dark=#e7e6ff}"
                    Text="{Binding Hour}"/>
                <Label
                    Margin="15,0,0,0"

```

```

        FontFamily="RobotoBold"
        FontSize="15"
        Text="{Binding Speaker, Converter={StaticResource
ListToStringConverter}}"
        TextColor="#60a5ff" />

    </VerticalStackLayout>
</Grid>
</Grid>
</Border>
</DataTemplate>

</cards:CoverflowView.ItemTemplate>
</cards:CoverflowView>

</Grid>
</ScrollView>
</ContentPage>

```

43. Presione las **teclas Command + S** (Ctrl + S) para guardar el archivo
44. Cuando se crea un archivo XAML realmente se están creando dos archivos, el primero es el **NombreDeArchivo.xaml** el segundo archivo es el **NombreDelArchivo.xaml.cs**, para verlo debemos expandir la flecha al lado izquierdo del archivo **HomePage.xaml** esto mostrara a **HomePage.xaml.cs** de doble clic sobre **HomePage.xaml.cs** para abrirlo.
45. Reemplazaremos el código con el siguiente código:

```

using System.Collections.ObjectModel;
using AppUAEM.Models;

namespace AppUAEM.Views;

public partial class HomePage : ContentPage
{
    public ObservableCollection<Profile> Profiles { get; set; }
    public ObservableCollection<ExpoCard> ExpoCards { get; set; }

    public HomePage()
    {
        InitializeComponent();
        LoadData();
        BindingContext = this;
    }
}

```

```
public void LoadData()
{
    Profiles = new ObservableCollection<Profile>
    {
        new Profile
        {
            Name="Alejandra Puebla",
            ProfileImage="aprofile.png",
            Title="SQL"
        },
        new Profile
        {
            Name="Judith Robles",
            ProfileImage="jrprofile.png",
            Title="Microservicios"
        },
        new Profile
        {
            Name="Victor Roman",
            ProfileImage="vrprofile.png",
            Title="SQL"
        },
        new Profile
        {
            Name="Amin Espinoza",
            ProfileImage="aeprofile.png",
            Title="Dev Ops"
        },
        new Profile
        {
            Name="Humberto Jaimes",
            ProfileImage="hjprofile.png",
            Title="Microservicios"
        },
        new Profile
        {
            Name="Jésus Gil",
            ProfileImage="jgprofile.png",
            Title="Microservicios,SQL"
        },
        new Profile
        {
            Name="Luis Beltrán",

```

```
    ProfileImage="lbprofile.png",
    Title="IA"
},
new Profile
{
    Name="Misael Cazarez",
    ProfileImage="mcprofile.png",
    Title=".NET MAUI"
},
new Profile
{
    Name="Bryan Oroxón",
    ProfileImage="boprofile.png",
    Title=".NET MAUI"
}
};

ExpoCards = new ObservableCollection<ExpoCard>
{
    new ExpoCard
    {
        ImagePath="registro",
        Subject="Registro",
        Hour="08:00 - 09:00",
        Speaker=new List<string>
        {
            "Público asistente"
        }
    },
    new ExpoCard
    {
        ImagePath="keynote1",
        Subject="Keynote",
        Hour="09:00 - 09:30",
        Speaker=new List<string>
        {
            "Dra. Dora Calderon "
        }
    },
    new ExpoCard
    {
        ImagePath="microservicios1",
        Subject="Introducción a los Microservicios",
        Hour="09:30 - 10:50",
        Speaker=new List<string>
```

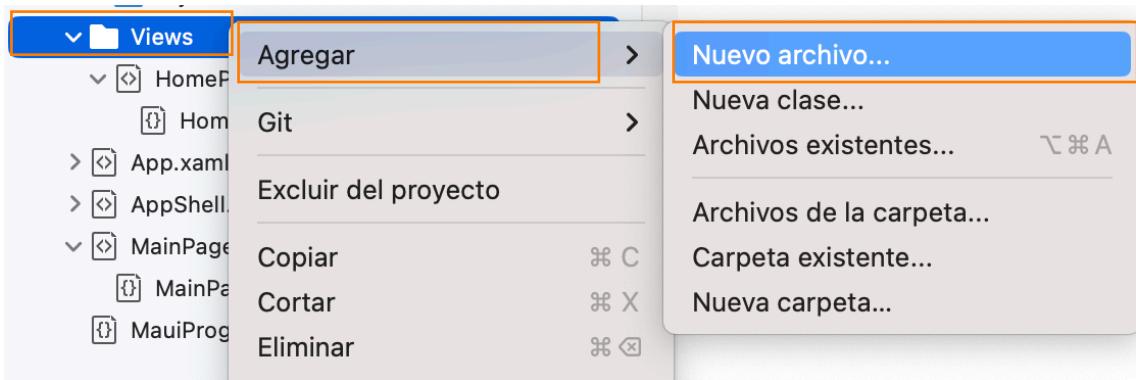
```
{
    " Judith Robles", "Humberto Jaimes", "Jesús Gil"
}
},
new ExpoCard
{
    ImagePath="maui4",
    Subject="Creando tu primera App .NET MAUI",
    Hour="11:00 - 12:20",
    Speaker=new List<string>
    {
        "Misael Cazarez", "Bryan Oroxón"
    }
},
new ExpoCard
{
    ImagePath="lunch2",
    Subject="Break Almuerzo",
    Hour="12:30 - 13:30",
    Speaker=new List<string>
    {
        "Público asistente"
    }
},
new ExpoCard
{
    ImagePath="chatgpt2",
    Subject="Aprendiendo SQL con la IA ChatGPT",
    Hour="13:30 - 14:30",
    Speaker=new List<string>
    {
        "Alejandra Puebla ", "Jesús Gil", "Victor Roman"
    }
},
new ExpoCard
{
    ImagePath="ia1",
    Subject="Charla Magistral de IA",
    Hour="14:30 - 16:00",
    Speaker=new List<string>
    {
        "Luis Beltrán"
    }
},
```

```

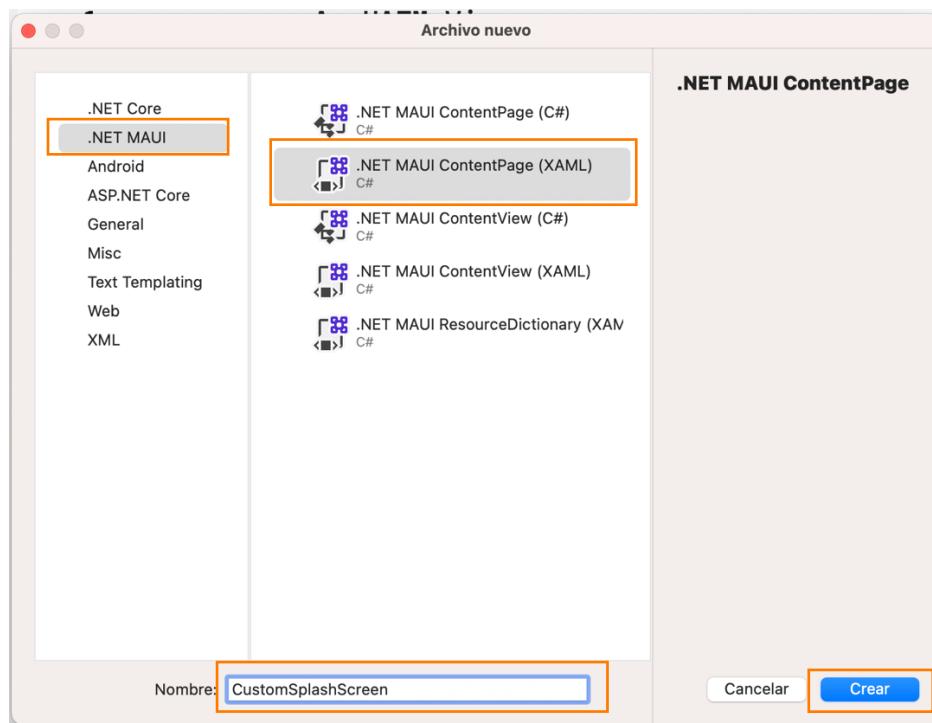
new ExpoCard
{
    ImagePath="cierre1",
    Subject="Cierre del evento",
    Hour="16:00 - 16:30",
    Speaker=new List<string>
    {
        "Alejandra Puebla", "Jesús Gil"
    }
}
};
}
}

```

46. Presione **Command + S (Ctrl +S)** para guardar el archivo.
47. Crearemos un nuevo archivo Xaml, de clic sobre la carpeta **Views** y de clic derecho **Agregar** y seleccione **Nuevo Archivo**



48. Seleccione las **Plantilla .NET MAUI** busque la plantilla **.NET MAUI Content (Xaml)**, en nombre Escriba **CustomSplashScreen** y de clic en **Crear**



49. Reemplace todo el código con este código:

```
<?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="AppUAEM.Views.CustomSplashScreen"
    xmlns:mct="clr-namespace:CommunityToolkit.Maui.Behaviors;assembly=CommunityToolkit.Maui"
    Title="CustomSplashScreen"
    xmlns:ios="clr-namespace:Microsoft.Maui.Controls.PlatformConfiguration.iOS;assembly=Microsof
t.Maui.Controls"
    ios:Page.UseSafeArea="False">
<ContentPage.Behaviors>
    <mct:StatusBarBehavior StatusBarColor="{AppThemeBinding Light=#545583,
Dark=Black}" />
</ContentPage.Behaviors>
<ContentPage.Padding>
    <OnPlatform x:TypeArguments="Thickness">
        <On Platform="Android, iOS" Value="0" />
        <On Platform="Windows" Value="0, 32, 0, 0" />
    </OnPlatform>
</ContentPage.Padding>
<Grid VerticalOptions="FillAndExpand"
    HorizontalOptions="FillAndExpand"
```

```

ColumnDefinitions=".2*,.6*,.2*"
RowDefinitions=".2*,.5*,.1*,.2*"
Background="{AppThemeBinding Light=#545583, Dark=Black}"
<Image Grid.Column="1" Grid.ColumnSpan="2"
      Grid.Row="0" Grid.RowSpan="4"
      Source="grup"
      HorizontalOptions="EndAndExpand"
      VerticalOptions="CenterAndExpand"
      HeightRequest="800"
      Aspect="AspectFill" />

<Image Grid.Column="0" Grid.Row="1" Grid.RowSpan="4"
      Source="dotnet_bot_hg.png"
      x:Name="bots"
      HorizontalOptions="Start"
      VerticalOptions="Start"
      WidthRequest="200"
      Aspect="AspectFit"/>

<ActivityIndicator Grid.Column="1" Grid.Row="2"
                  IsRunning="True"
                  Color="White"/>

<Label Grid.Column="1" Grid.Row="3"
      Text="Loading information..."
      HorizontalOptions="Center"
      TextColor="White"/>
</Grid>
</ContentPage>

```

50. Presione **Command + S (Ctrl + S)**, para guardar el archivo.

51. Abra el archivo CustomSplashScreen.xaml.cs y reemplace el código con este código:

```

namespace AppUAEM.Views;

public partial class CustomSplashScreen : ContentPage
{
    public CustomSplashScreen()
    {
        InitializeComponent();
    }

    protected async override void OnAppearing()

```

```

{
    base.OnAppearing();

    await Task.Delay(1000);

    var anim1 = bots.TranslateTo(450, 0, 3000, Easing.Linear);
    var anim2 = bots.ScaleTo(2, 3000, Easing.Linear);

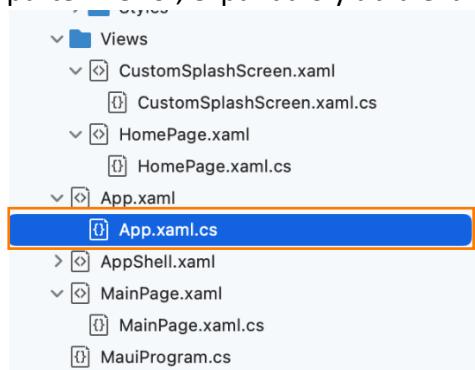
    await Task.WhenAll(anim1, anim2);

    App.Current.MainPage = new HomePage();
}
}

```

52. Presione **Command + S (Ctrl + S)**, y guarde el archive.

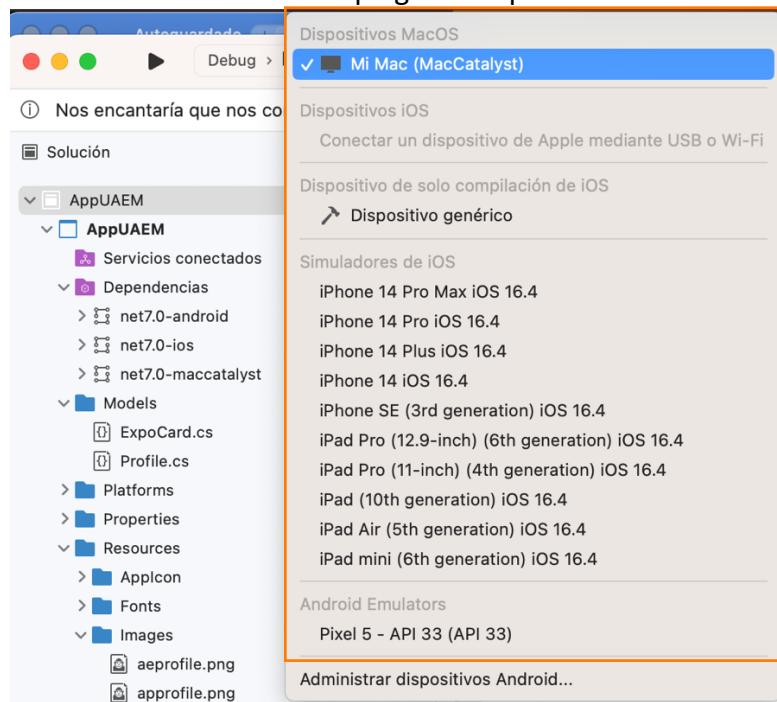
53. En el **explorador de la Solución** ahora busque el archivo **App.xaml** que está en la parte inferior, expandalo y abra el archivo **App.xaml.cs**



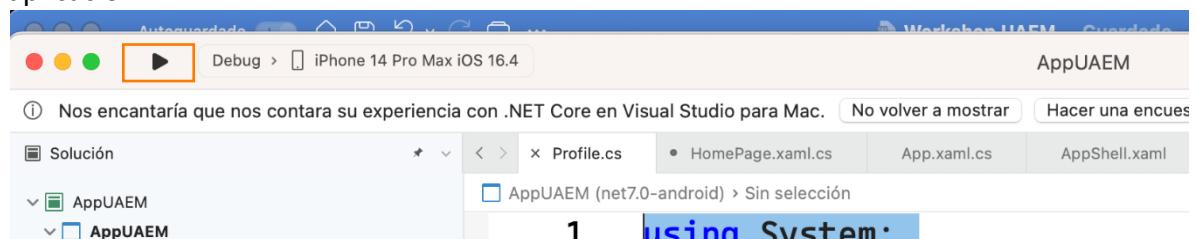
54. En la **línea 9 del código** Reemplace el siguiente código:

**MainPage = new CustomSplashScreen();**

55. Ahora en la barra superior de un clic al lado derecho de Debug y seleccione la Plataforma donde sera desplegada la aplicación



56. Despues de seleccionar la plataforma de clic en icono de Play y se desplegará la aplicación



57. Y descubrirás que hace la aplicación,

58. Gracias por tomarte el tiempo de crear este proyecto.

Esta App fue diseñada y creada por Luis Bryan Orozón .