

SHELL

IN .NET MAUI

PREVIEW 12

Natalia Coronado Romero 2ºDAM

ÍNDICE

Introducción.....	3
Objetivo de la memoria.....	3
Material utilizado.....	3
GitHub.....	3
Desarrollo.....	4
Interfaz Visual.....	4
-MainPage.xaml.....	4
-SeleccionCurso.xaml.....	7
-FormaDePago.xaml.....	9
Código Interno.....	11
-AppShell.xaml.cs.....	11
-MainPage.xaml.cs.....	12
-SeleccionCurso.xaml.cs.....	13
-FormaDePago.xaml.cs.....	14
Problemas encontrados y sugerencias.....	15
Conclusión.....	15
Bibliografía/webgrafía.....	16
Video del funcionamiento.....	16

Introducción

Proyecto de una aplicación con MAUI en Visual Studio utilizando la navegación ShellNavigation para desplazarse por las páginas.

Objetivo de la memoria

El objetivo de esta práctica es diseñar una aplicación para poder pasar parámetros y valores entre páginas y utilizarlos en el código gracias al ShellNavigation.

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, OBS Studio

GitHub

Es el enlace de github al repositorio de la asignatura:

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

Desarrollo

Interfaz Visual

-MainPage.xaml

Esta es la vista previa, al ejecutar la aplicación, de la página principal. En esta página se muestran todos los datos que se reciben de la navegación y se utilizan para mostrar un resultado final gracias a estos.

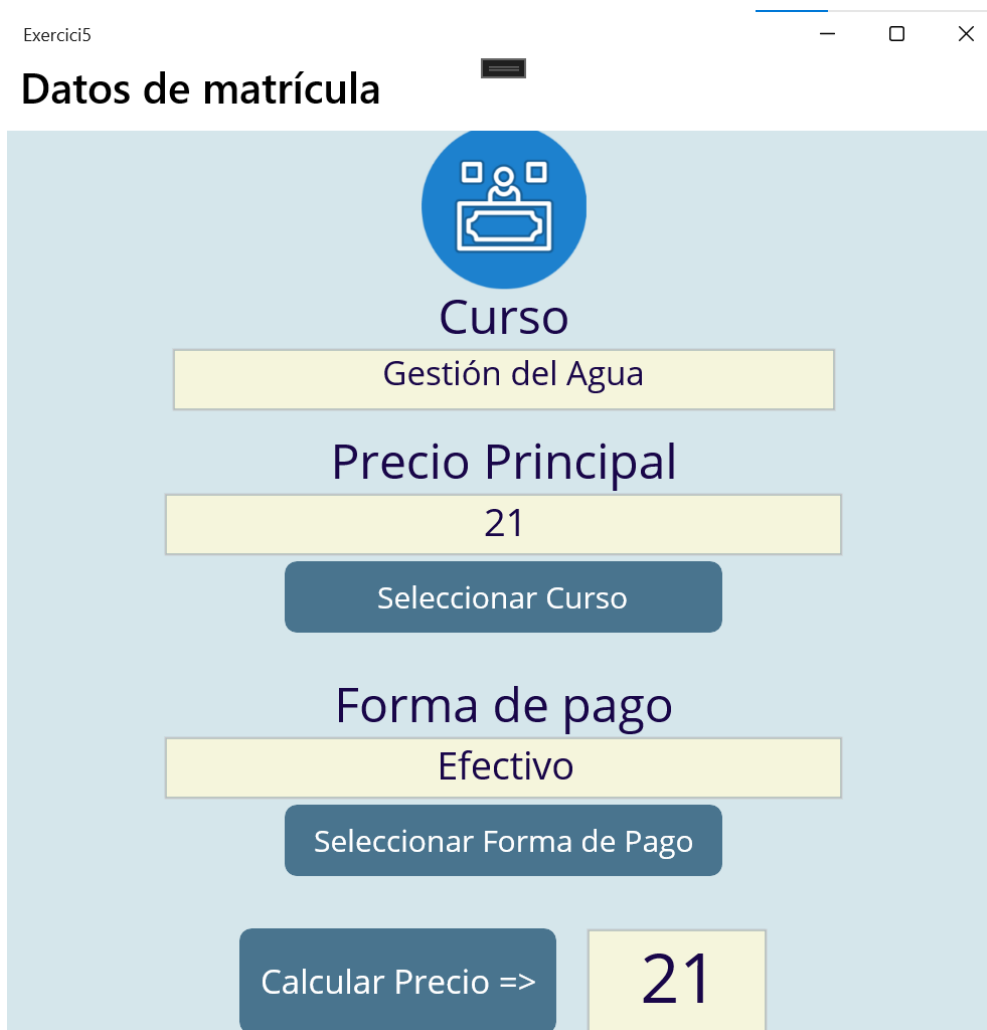


Figura 1: Vista previa de la página principal de la aplicación

Para entrar en más detalle sobre la parte visual del jugador, el código contiene una imagen, labels y botones. Hay diferencias en los label, uno indica información de lo que se mostrará, y el otro muestra los datos recibidos por el usuario mediante binding. Los botones nos llevan a las otras páginas cuando pulsamos sobre ellos.

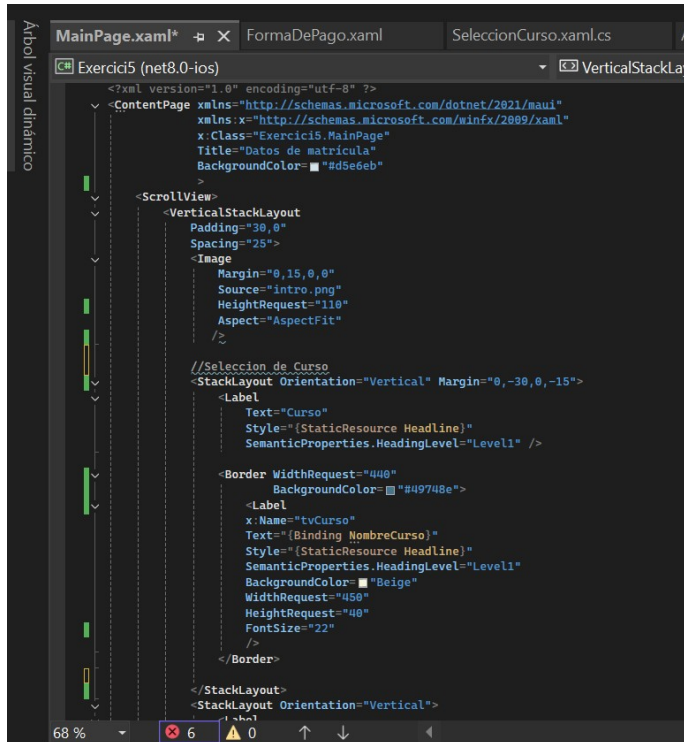


Figura 2: Primera parte del código visual del MainPage: código por predetermined, una imagen y dos label haciendo uno de titulo y otro conteniendo información por bending

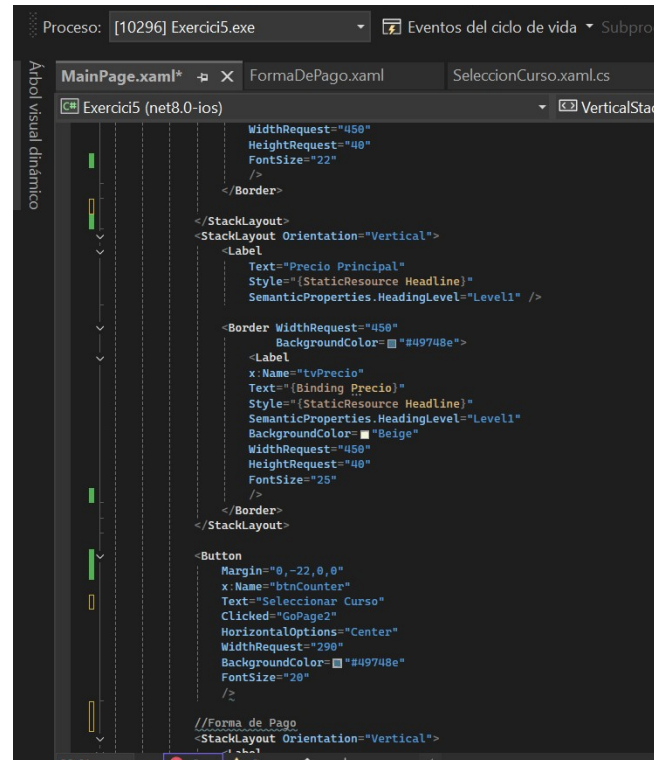
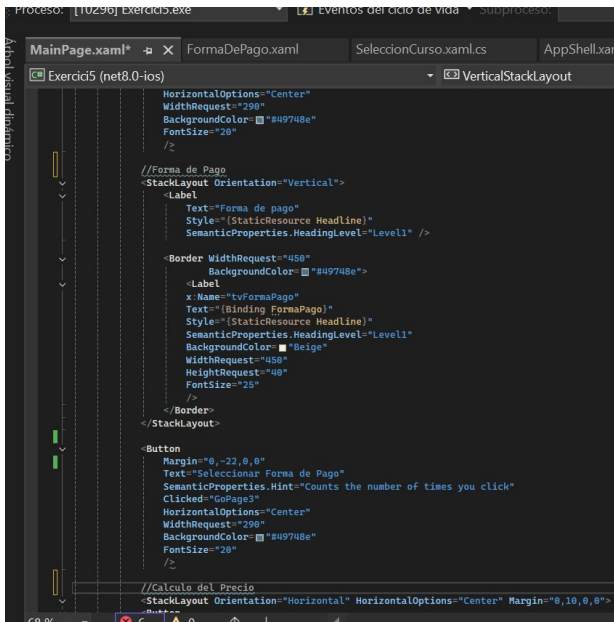


Figura 3: Segunda parte del código visual del MainPage: dos label haciendo uno de titulo y otro conteniendo información por bending y un botón

El único botón que no se desplaza entre páginas es el de la parte de Calcular el Precio que, como su propio nombre indica, calculará el precio total de la información insertada.



```
HorizontalOptions="Center"
WidthRequest="290"
BackgroundColor="#49748e"
FontSize="28"
/>

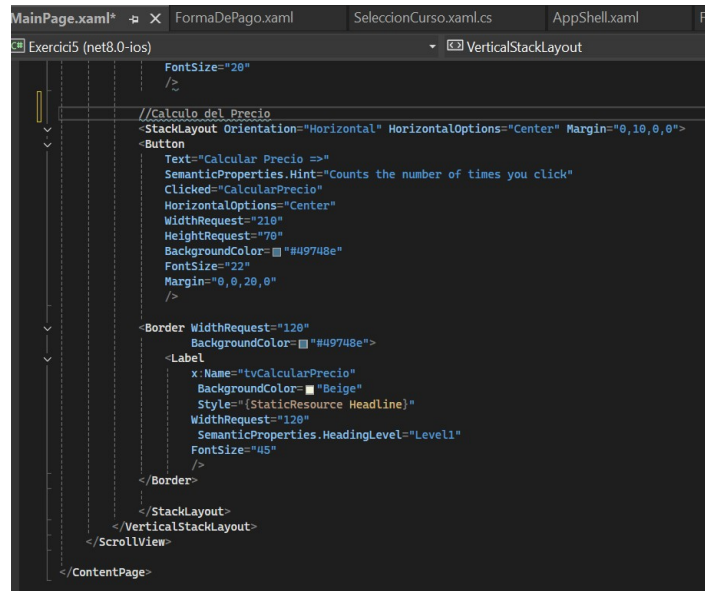
//Forma de Pago
<StackLayout Orientation="Vertical">
  <Label
    Text="Forma de pago"
    Style="{StaticResource Headline}"
    SemanticProperties.HeadingLevel="Level1" />

  <Border WidthRequest="450"
    BackgroundColor="#49748e">
    <Label
      x:Name="tvFormaPago"
      Text="{Binding FormaPago}"
      Style="{StaticResource Headline}"
      SemanticProperties.HeadingLevel="Level1"
      BackgroundColor="Beige"
      WidthRequest="450"
      HeightRequest="40"
      FontSize="25"
      />
    />
  />

  <Button
    Margin="0,-22,0,0"
    Text="Seleccionar Forma de Pago"
    SemanticProperties.Hint="Counts the number of times you click"
    Clicked="GoPago"
    HorizontalOptions="Center"
    WidthRequest="290"
    BackgroundColor="#49748e"
    FontSize="28"
    />

//Calculo del Precio
<StackLayout Orientation="Horizontal" HorizontalOptions="Center" Margin="0,10,0,0">
```

Figura 4: Tercera parte del código visual del MainPage: Forma de Pago: dos label haciendo uno de título y otro conteniendo información por binding y un botón



```
FontSize="28"
/>

//Calculo del Precio
<StackLayout Orientation="Horizontal" HorizontalOptions="Center" Margin="0,10,0,0">
  <Button
    Text="Calcular Precio =>"
    SemanticProperties.Hint="Counts the number of times you click"
    Clicked="CalcularPrecio"
    HorizontalOptions="Center"
    WidthRequest="210"
    HeightRequest="70"
    BackgroundColor="#49748e"
    FontSize="22"
    Margin="0,0,20,0"
    />

  <Border WidthRequest="120"
    BackgroundColor="#49748e">
    <Label
      x:Name="tvCalcularPrecio"
      BackgroundColor="Beige"
      Style="{StaticResource Headline}"
      WidthRequest="120"
      SemanticProperties.HeadingLevel="Level1"
      FontSize="45"
      />
    />
  />
</StackLayout>
</VerticalStackLayout>
</ScrollView>
</ContentPage>
```

Figura 5: Última parte del código visual del MainPage: botón para calcular el precio un label para mostrarlo

-SeleccionCurso.xaml

Al pulsar en Selección de Curso nos envía a otra página donde aparecerán dos imágenes que representan los dos cursos disponibles para seleccionar, el de Gestión del Agua y DAM. Al elegir uno o otro curso, vuelve a la página principal devolviendo un valor diferente cada uno.

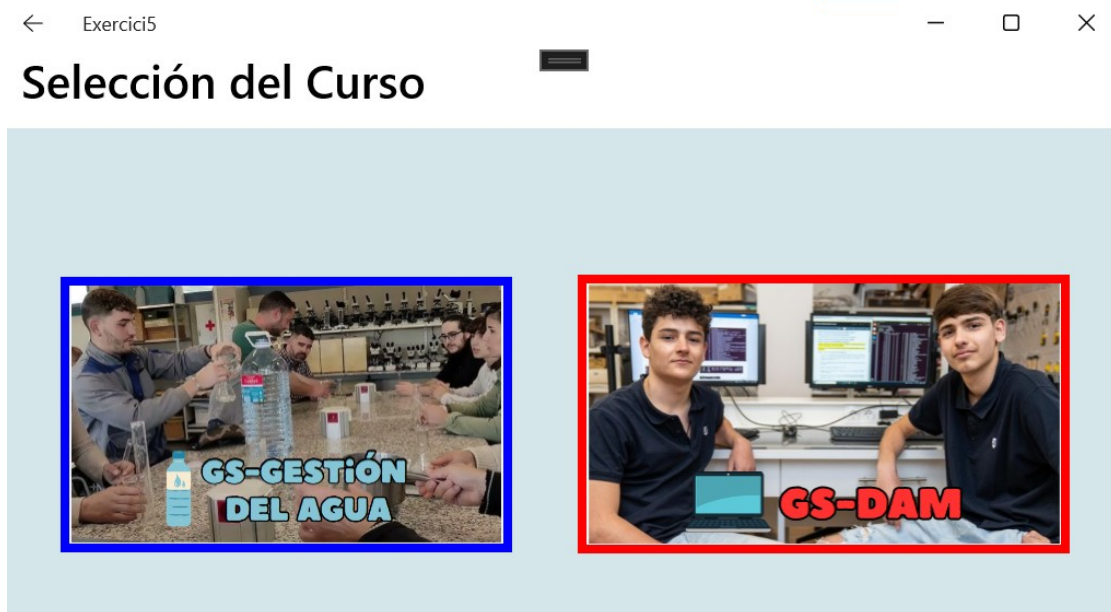


Figura 6: Vista previa de la página “SeleccionCurso” de la aplicación

```
MainWindow.xaml* App.xaml.cs App.xaml FormaDePago.xaml SeleccionCurso.xaml* MauiProgram.cs
Ejercici5 (net8.0-ios) VerticalStackLayout VerticalStackLayout
<?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="Ejercici5.SeleccionCurso"
  BackgroundColor="#d5e6eb"
  Title="Selección del Curso">
  <VerticalStackLayout>
    <StackLayout Orientation="Horizontal" HorizontalOptions="CenterAndExpand" Margin="0,70,0,0">
      <ImageButton Source="agua.png"
        Clicked="GSAgua"
        HorizontalOptions="Center"
        WidthRequest="275"
        HeightRequest="168"
        Margin="0,0,20,0"
        BorderColor="Blue"
        BorderWidth="5"
      />
      <ImageButton Source="dam.png"
        Clicked="GSDAM"
        HorizontalOptions="Center"
        WidthRequest="300"
        HeightRequest="170"
        Margin="20,0,0,0"
        BorderColor="Red"
        BorderWidth="5"
      />
    </StackLayout>
  </VerticalStackLayout>
</ContentPage>
```

Figura 7: Código Visual de 'SeleccionCurso': dos imágenes de botón para elegir

-FormaDePago.xaml

Al pulsar el otro botón llamado ‘Seleccionar Forma de Pago’, se mostrará la última página Método de Pago dos imágenes de manera vertical para seleccionar la forma de pago del usuario y volverá otra vez a la página principal. Según pulse la imagen del dinero será en efectivo o si por el contrario elige la otra imagen será tarjeta.

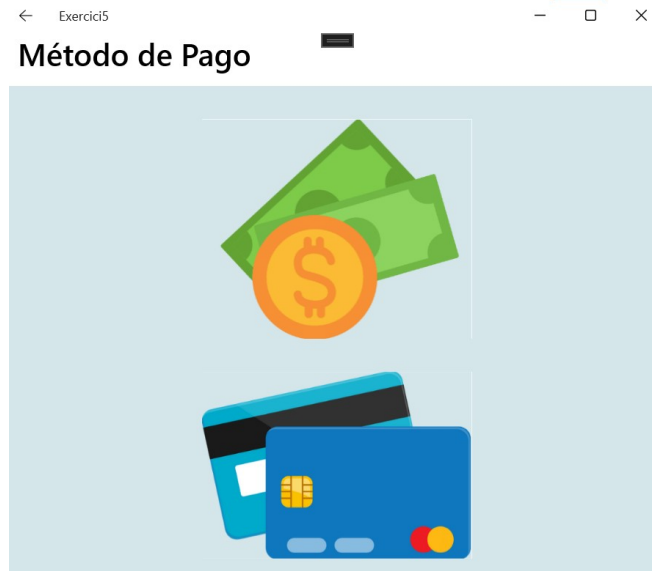


Figura 8: Vista previa de la página “FormaDePago” de la aplicación

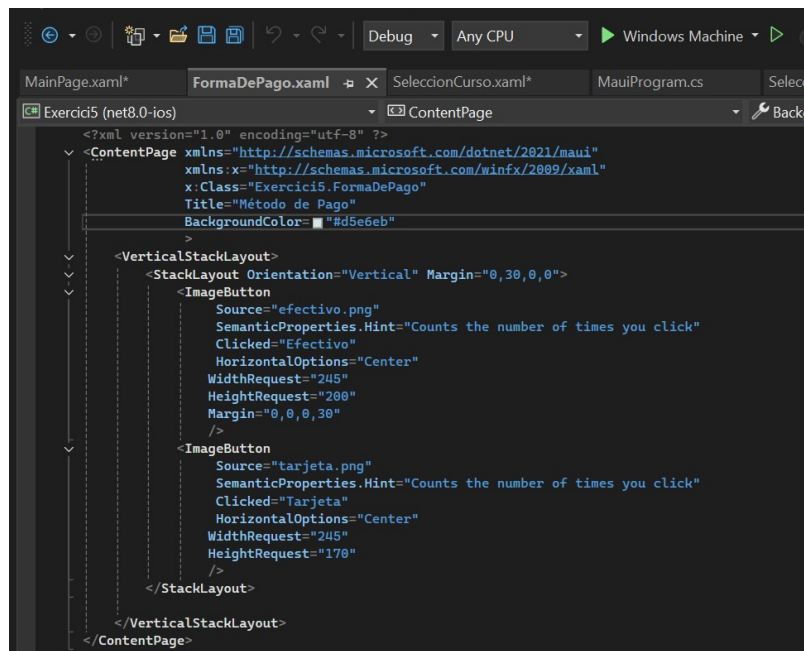


Figura 9: Código Visual de 'FormaDePago': dos imágenes de botón para elegir dependiendo de lo que elija el usuario

Código Interno

-AppShell.xaml.cs

Como en esta práctica estamos utilizando NavigationShell para desplazarse entre distintas páginas, se necesita modificar el fichero AppShell.xaml.cs. Por predeterminado solo se encuentra la ruta del MainPage, por lo que se añaden las demás páginas necesarias con el mismo nombre que las hemos creado que són SeleccionCurso y FormaDePago.

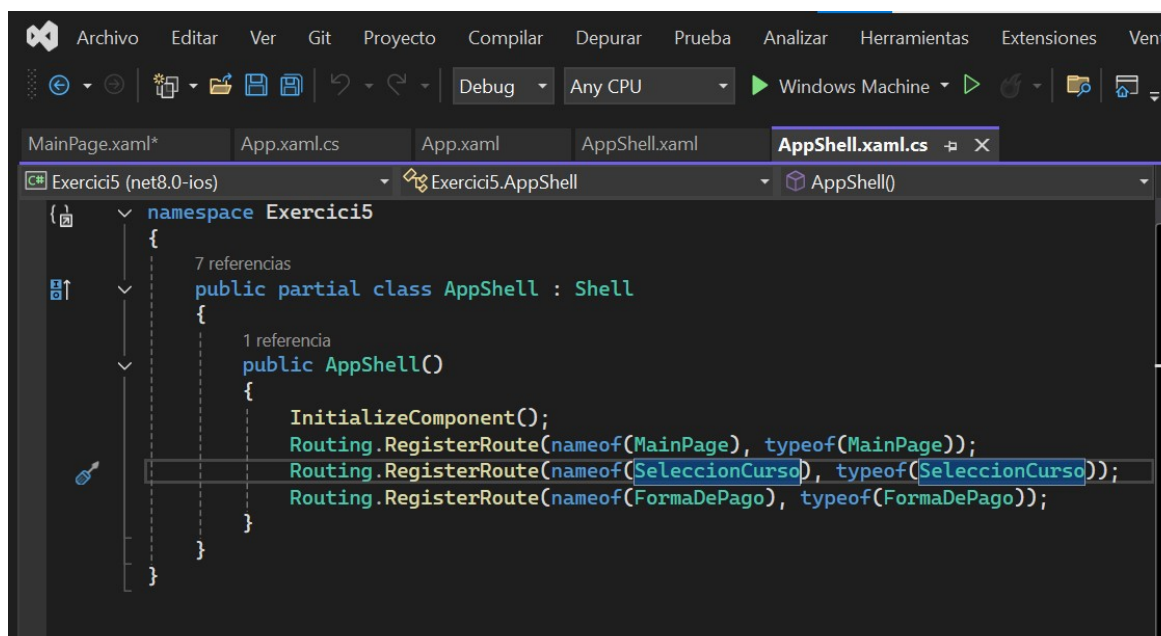


Figura 10: Código del Appshell con las rutas para el NavigationShell

-MainPage.xaml.cs

Este es el principal por lo que recibe todos los parámetros de las otras páginas usando QueryProperty. Los valores están vinculados al diseño gráfico de la página a través de binding para que, ante cualquier cambio, se actualicen automáticamente.

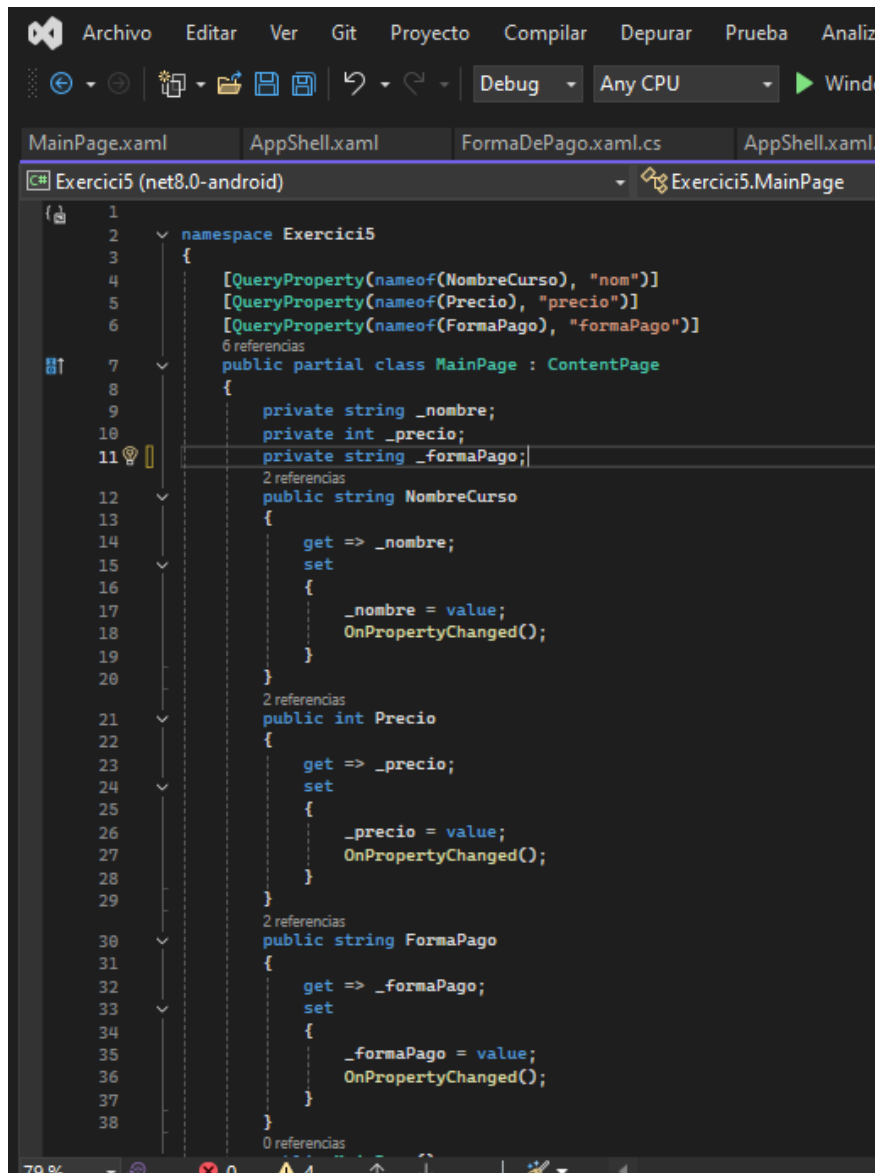
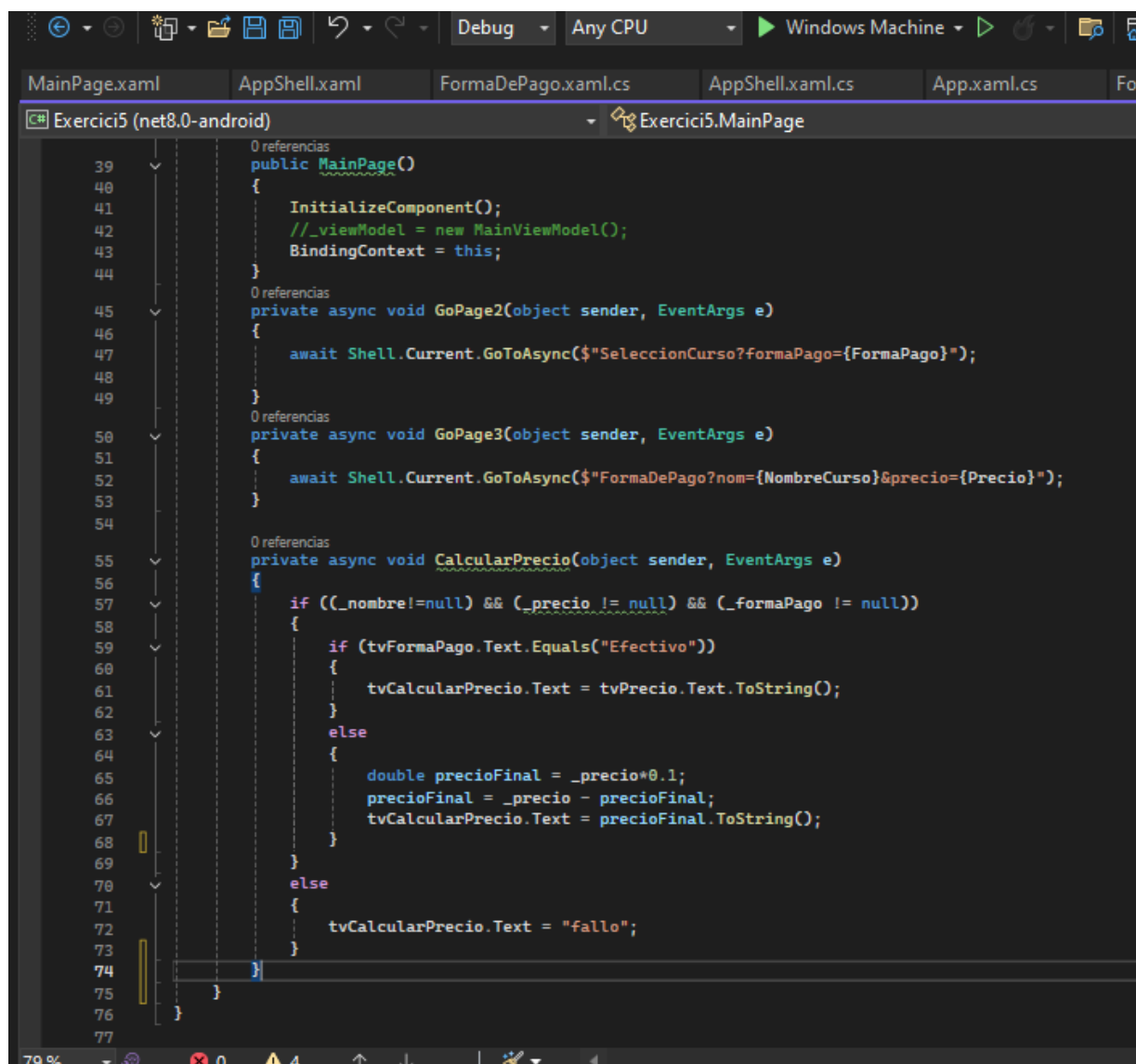


Figura 11: Primera parte del código del MainPage: valores inicializados por bending y queryproperty

Cada botón que se encuentra en la página tiene su propia función de código. Los dos primeros botones obviamente se utilizan para desplazarse a las otras dos páginas para asignar un valor para FormaPago y NombreCurso y Precio utilizando el NavigationShell.

El último botón de Calcular Precio emplea los valores anteriores y calcula el precio total de lo que valdría un curso dependiendo lo que elija el usuario, si el método de pago es en efectivo el precio se queda igual, en cambio si es con tarjeta se hace un descuento de un 10%.



```
39 0 referencias
40 public MainPage()
41 {
42     InitializeComponent();
43     //_viewModel = new MainViewModel();
44     BindingContext = this;
45 }
46 0 referencias
47 private async void GoPage2(object sender, EventArgs e)
48 {
49     await Shell.Current.GoToAsync($"SeleccionCurso?formaPago={FormaPago}");
50 }
51 0 referencias
52 private async void GoPage3(object sender, EventArgs e)
53 {
54     await Shell.Current.GoToAsync($"FormaDePago?nom={NombreCurso}&precio={Precio}");
55 }
56 0 referencias
57 private async void CalcularPrecio(object sender, EventArgs e)
58 {
59     if ((_nombre!=null) && (_precio != null) && (_formaPago != null))
60     {
61         if (tvFormaPago.Text.Equals("Efectivo"))
62         {
63             tvCalcularPrecio.Text = tvPrecio.Text.ToString();
64         }
65         else
66         {
67             double precioFinal = _precio*0.1;
68             precioFinal = _precio - precioFinal;
69             tvCalcularPrecio.Text = precioFinal.ToString();
70         }
71     }
72     else
73     {
74         tvCalcularPrecio.Text = "fallo";
75     }
76 }
77 }
```

Figura 12: Segunda parte del código del MainPage: Funciones de los botones, dos desplazándose de página con parámetros y otro calculando el precio total utilizando los valores recibidos

-SeleccionCurso.xaml.cs

Se ha utilizado el QueryProperty en el cual nos encontramos que es el SeleccionCurso para poder hacer la navegación entre datos y así se actualicen automáticamente en la página principal al capturarlos.

Cuando el usuario selecciona un curso de los dos que se encuentran en la página, se asignan los valores correspondientes al nombre y precio del curso y se devuelven esos valores como parámetros al MainPage.

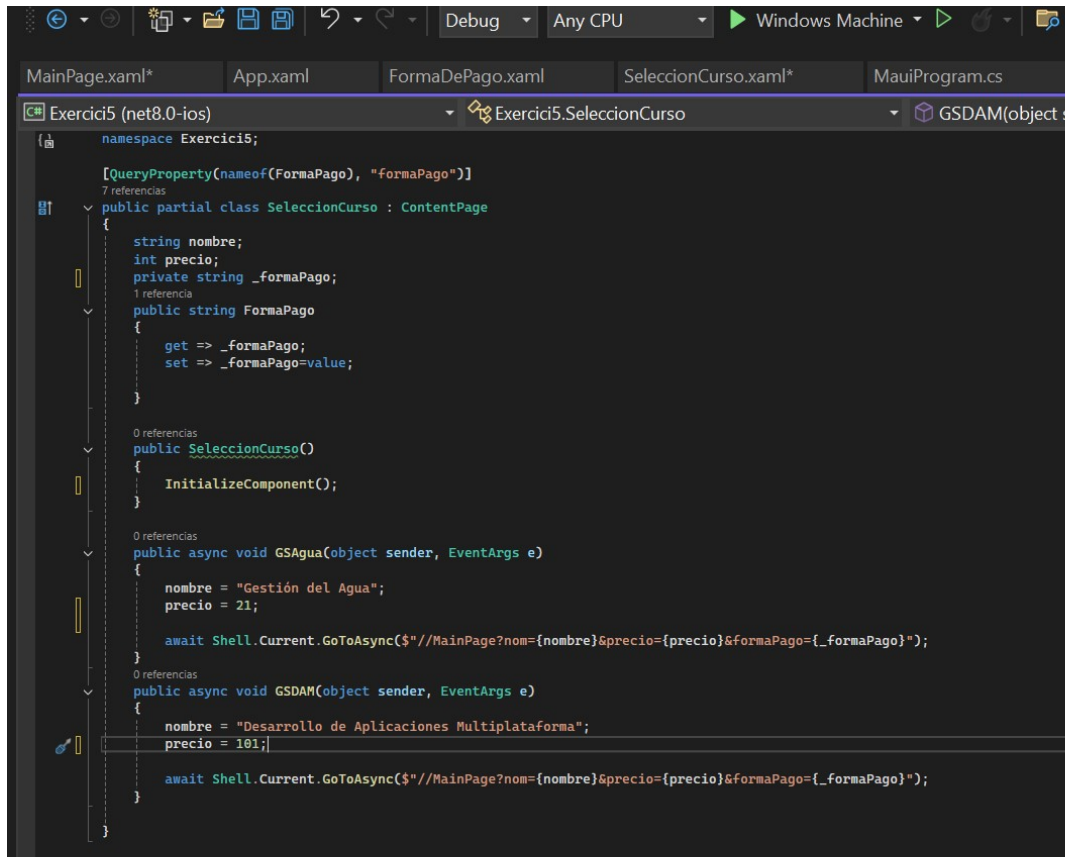


Figura 13: Código de SeleccionCurso: QueryProperty, Binding y dos funciones que representan cada imagen de botón para devolver sus datos al MainPage

-FormaDePago.xaml.cs

Parecido al código anterior se utiliza también el QueryProperty de la página actual FormaDePago para poder hacer la navegación entre datos. Para que los datos anteriores puestos de SeleccionCurso no se borren, se pasan también junto a los nuevos.

Utiliza la misma dinámica con las imágenes de botones devolviendo valores asignados como parámetros al MainPage cada una de las dos funciones.

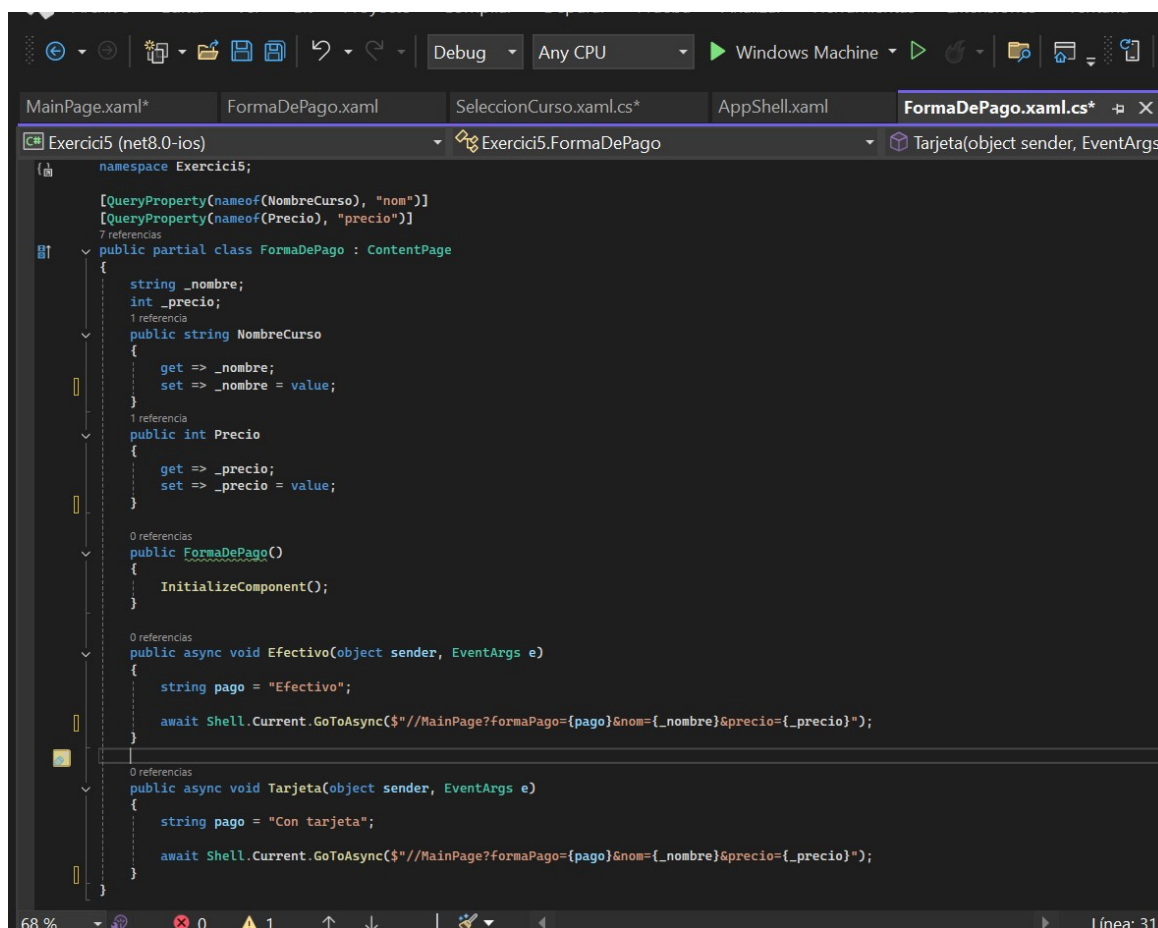


Figura 14: Código de FormaDePago: utilización de los queryproperty y dos imágenes de botón que envía valores según se pulse uno o otro

Problemas encontrados y sugerencias

Ha habido distintos errores al igual que siempre, después de ir probando, hacer otras veces el proyecto y ayudas externas para conseguir información se ha conseguido.

Conclusión

Buena actividad para la implementación de navegación y distintos datos entre páginas utilizando NavigationShell.

Bibliografía/webgrafía

_.Pdfs del profesor de mi profesor.

Video del funcionamiento



Dibujo 1: <https://youtu.be/wEY4vH4OTiA>