MEMORIA

DESARROLLO DE INTERFACES

Alejandro Roberto Chiralt
2ºDAM

INDICE

INTRODUCCIÓN:

En este proyecto se diseñará una aplicación utilizando .NET MAUI que permitirá simular el proceso de matrícula para un curso. La aplicación contará con tres páginas interactivas donde los usuarios podrán seleccionar un curso, elegir una forma de pago y calcular el precio final. Este tipo de aplicación no solo es útil para aprender a manejar las interfaces gráficas en .NET MAUI, sino también para entender cómo gestionar la navegación entre pantallas, la interacción con botones y el manejo de datos.

OBJETIVOS:

- **Desarrollar una aplicación funcional en .NET MAUI** que permita al usuario interactuar con tres páginas diferentes: selección de curso, forma de pago y cálculo del precio final.
- **Implementar una interfaz de usuario intuitiva** que permita la navegación fácil y la selección de opciones por parte del usuario.
- **Aplicar lógica de negocio sencilla** para calcular el precio final de un curso en función de la forma de pago seleccionada.
- **Familiarizarse con los controles y la navegación en .NET MAUI**, así como la actualización dinámica de elementos en la interfaz.

TAREAS A REALIZAR:

- **Página de Datos de matrícula:**

  Crear una página que muestre la información básica del curso (nombre, precio inicial y forma de pago) a través de Labels y botones.
  Incluir botones que permitan seleccionar el curso y la forma de pago.
  Incluir un botón para calcular el precio, que debe estar deshabilitado hasta que se haya seleccionado un curso y una forma de pago.

- **Página de selección de curso:**

Crear una página con dos imágenes que actúen como botones para la selección de un curso.
Asignar un precio a cada curso y regresar la información a la página de datos de matrícula.

- **Página de selección de forma de pago:**

  Diseñar una página con dos imágenes que representen las opciones de pago (efectivo y tarjeta de crédito).
  Regresar la información de la forma de pago seleccionada a la página de datos de matrícula.

- **Implementación de la lógica de cálculo de precio:**

  El precio será calculado en función del método de pago seleccionado. Si es en efectivo, se muestra el precio sin cambios. Si es con tarjeta de crédito, se aplicará un descuento del 10%.

DESARROLLO :

- **Página de Datos de matrícula**

  Se creará una página con el título "Datos de matrícula". Esta página incluirá tres Labels para mostrar los detalles del curso seleccionado: "Curso", "Precio inicial" y "Forma de pago", todos con espacios vacíos que se actualizarán conforme se elijan las opciones. Debajo de estos Labels, se colocarán dos botones: "Seleccionar curso" y "Seleccionar forma de pago". Además, se colocará un botón "Calcular precio" que inicialmente estará deshabilitado hasta que se hayan rellenado los tres campos anteriores. Al pulsar este botón, se mostrará el precio final en un Label adicional.

- **Página de selección de curso**

  Al pulsar el botón "Seleccionar curso", la aplicación navegará a una nueva página titulada "Curso". En esta página se mostrarán dos imágenes que representen los cursos disponibles. Cada imagen actuará como un botón, y al pulsarla, el curso seleccionado será enviado de vuelta a la página de Datos de matrícula. Además, el precio correspondiente al curso seleccionado se actualizará en el campo "Precio inicial".

- **Página de selección de forma de pago**

Al seleccionar el botón "Seleccionar forma de pago", se abrirá una nueva página con el título "Forma de pago". Esta página mostrará dos imágenes: una representando el pago en efectivo y otra el pago con tarjeta. Al seleccionar una de las opciones, la forma de pago seleccionada se enviará a la página de Datos de matrícula, donde se actualizará el campo correspondiente.

- **Cálculo del precio final**

  Cuando los tres campos (Curso, Precio inicial, y Forma de pago) estén completos, el botón "Calcular precio" se habilitará. Al presionarlo, se calculará el precio final dependiendo de la forma de pago seleccionada. Si el pago es en efectivo, el precio inicial no sufrirá cambios; si es con tarjeta, se aplicará un descuento del 10%. El resultado será mostrado en un Label debajo del botón.

  Este proyecto proporciona una forma sencilla pero completa de aprender a manejar las interfaces de usuario, la navegación y la lógica de negocio dentro del entorno de desarrollo .NET MAUI, lo que facilita la creación de aplicaciones multiplataforma.

ACTIVIDADES :

Este código está escrito en XAML para una aplicación en .NET MAUI, y define una página llamada Pagina1 que contiene una interfaz gráfica para la gestión de datos de matrícula. Aquí tienes un resumen de lo que hace cada parte:

1. **Encabezado de la página:**
   El Label con el texto "Datos de Matrícula" actúa como título principal de la página, centrado y estilizado con un tamaño de fuente grande y negrita.
2. **Secciones informativas:**
   - Cada sección (Curso, Precio Inicial, Forma de Pago) está contenida en un Frame con esquinas redondeadas y fondo blanco. Dentro de cada Frame, un HorizontalStackLayout organiza los elementos horizontalmente:
     - Un Label con el nombre de la propiedad.
     - Un segundo Label que muestra datos dinámicos a través de un enlace de Binding a propiedades del modelo (NombreCurso, PrecioInicial, FormaPago).

3. **Botones:**
    - Dos botones (Seleccionar Curso y Seleccionar Pago) están organizados horizontalmente y tienen estilos personalizados (color de fondo, texto blanco, esquinas redondeadas). Al hacer clic, se ejecutan eventos (btnSeleccionarCurso_Clicked y btnSeleccionarPago_Clicked).
    - Un tercer botón ("Calcular Precio") está debajo, con un estilo similar, y al hacer clic se ejecuta el método btnCalcularPrecio_Clicked.
4. **Resultado del cálculo:**
    - Un Label muestra el resultado del cálculo de precio, enlazado mediante Binding a la propiedad btnCalcularPrecio_Clicked.

Toda la página tiene un diseño simple y organizado verticalmente con la ayuda de VerticalStackLayout, y la estética se mantiene consistente con un esquema de colores y espaciados ajustados.

Pagina1.xaml

```
<?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="EJER_UD02v5.Pagina1"
             Title="Datos de Matrícula"
             BackgroundColor="#f5f5f5">
    <VerticalStackLayout HorizontalOptions="Center" Padding="20">
        <!-- Encabezado -->
        <Label Text="Datos de Matrícula"
               FontSize="24"
               FontAttributes="Bold"
               TextColor="#2C3E50"
               HorizontalOptions="Center"
               Margin="10,20"/>
        <!-- Sección de curso -->
        <Frame CornerRadius="10" BackgroundColor="White" Padding="10" Margin="0,10">
            <HorizontalStackLayout Spacing="10">
                <Label Text="Curso:"
                       VerticalOptions="Center"
                       FontAttributes="Bold"
                       TextColor="#2C3E50"/>
                <Label x:Name="tvNombreCurso"
                       Text="{Binding NombreCurso}"
                       VerticalOptions="Center"
                       TextColor="#34495E"/>
            </HorizontalStackLayout>
        </Frame>
        <!-- Sección de precio inicial -->
        <Frame CornerRadius="10" BackgroundColor="White" Padding="10" Margin="0,10">
            <HorizontalStackLayout Spacing="10">
                <Label Text="Precio Inicial:"
                       VerticalOptions="Center"
                       FontAttributes="Bold"
                       TextColor="#2C3E50"/>
                <Label x:Name="tvPrecioInicial"
                       Text="{Binding PrecioInicial}"
                       VerticalOptions="Center"
```

```
                    TextColor="#34495E"/>
            </HorizontalStackLayout>
        </Frame>
        <!-- Sección de forma de pago -->
        <Frame CornerRadius="10" BackgroundColor="White" Padding="10" Margin="0,10">
            <HorizontalStackLayout Spacing="10">
                <Label Text="Forma de Pago:"
                       VerticalOptions="Center"
                       FontAttributes="Bold"
                       TextColor="#2C3E50"/>
                <Label x:Name="tvFormaPago"
                       Text="{Binding FormaPago}"
                       VerticalOptions="Center"
                       TextColor="#34495E"/>
            </HorizontalStackLayout>
        </Frame>
        <!-- Botones -->
        <HorizontalStackLayout Spacing="20" Margin="10">
            <Button x:Name="btnSeleccionarCurso"
                    Text="Seleccionar Curso"
                    TextColor="White"
                    BackgroundColor="#1ABC9C"
                    CornerRadius="20"
                    Padding="10"
                    Clicked="btnSeleccionarCurso_Clicked"/>
            <Button x:Name="btnSeleccionarPago"
                    Text="Seleccionar Pago"
                    TextColor="White"
                    BackgroundColor="#3498DB"
                    CornerRadius="20"
                    Padding="10"
                    Clicked="btnSeleccionarPago_Clicked"/>
        </HorizontalStackLayout>
        <Button x:Name="btnCalcularPrecio"
                Text="Calcular Precio"
                TextColor="White"
                BackgroundColor="#E74C3C"
                CornerRadius="20"
                Padding="10"
                Margin="0,10"
                Clicked="btnCalcularPrecio_Clicked"/>
        <!-- Resultado del cálculo -->
        <Label x:Name="tvCalcularPrecio"
               Text="{Binding btnCalcularPrecio_Clicked}"
               FontSize="18"
               TextColor="#E74C3C"
               HorizontalOptions="Center"
               Margin="10"/>
    </VerticalStackLayout>
</ContentPage>
```

Este código C# es la implementación de la lógica detrás de la página XAML que describiste anteriormente, en una aplicación .NET MAUI. Aquí te doy una explicación de cada parte:

1. **Atributos QueryProperty:**
   Las líneas [QueryProperty] permiten pasar datos entre páginas dentro de la aplicación usando parámetros de navegación. Estos atributos indican que se pueden recibir tres propiedades: PrecioInicial, FormaPago, y NombreCurso a través de la navegación en Shell.

2. **Variables estáticas y propiedades:**
   - Las variables _precioInicial, _formaPago, y _nombreCurso son estáticas para mantener su valor compartido a lo largo de las instancias de la clase.
   - Cada variable tiene una propiedad pública correspondiente (PrecioInicial, FormaPago, NombreCurso). Cada propiedad invoca OnPropertyChanged() al cambiar su valor, lo que indica a la interfaz que debe actualizarse para reflejar los nuevos datos. Este mecanismo es clave en el enlace de datos (data binding).

3. **Método OnDatosDevueltos():**
   Muestra una alerta con los datos actuales de NombreCurso, PrecioInicial y FormaPago. Es útil para confirmar que los datos están almacenados correctamente.

4. **Constructor Pagina1():**
   Inicializa la página y establece el contexto de enlace de datos (BindingContext = this).
   Además, se habilita o deshabilita el botón "Calcular Precio" (btnCalcularPrecio) dependiendo de si los valores de NombreCurso y FormaPago están vacíos.

5. **Eventos de botones:**
   - btnSeleccionarCurso_Clicked: Navega a la página Pagina2 cuando el usuario selecciona un curso.
   - btnSeleccionarPago_Clicked: Navega a la página Pagina3 cuando el usuario selecciona una forma de pago.

6. **Cálculo del precio (btnCalcularPrecio_Clicked):**
   - Dependiendo de si la forma de pago es "Efectivo" o "Tarjeta", realiza un cálculo distinto:
     - Si es "Efectivo", simplemente muestra el valor de PrecioInicial.

- Si es "Tarjeta", aplica un descuento del 10% (multiplicando PrecioInicial por 0.9) y muestra el valor resultante.
- El resultado del cálculo se muestra en la etiqueta tvCalcularPrecio.

En resumen, este código maneja la lógica de la página, incluyendo la navegación entre pantallas, la actualización de datos en la interfaz, y el cálculo de precios basado en la forma de pago.

Pagina1.xaml.cs

```
using Microsoft.Maui.Graphics.Text;
namespace EJER_UD02v5;
[QueryProperty(nameof(PrecioInicial), "precioInicial")]
[QueryProperty(nameof(FormaPago), "formaPago")]
[QueryProperty(nameof(NombreCurso), "nombreCurso")]
public partial class Pagina1 : ContentPage
{
    static double _precioInicial;
    static string _formaPago;
    static string _nombreCurso;

    public string NombreCurso
    {
        get { return _nombreCurso; }
        set
        {
            _nombreCurso = value;
            OnPropertyChanged();

        }
    }
    public double PrecioInicial
    {
        get { return _precioInicial; }
        set
        {
            _precioInicial = value;
            OnPropertyChanged();

        }
    }
    public string FormaPago
    {
        get { return _formaPago; }
        set
        {
            _formaPago = value;

            tvFormaPago.Text = _formaPago;
            OnPropertyChanged();

        }
    }
    private void OnDatosDevueltos()
    {
        // Mostrar una alerta con los datos almacenados
        DisplayAlert("Datos almacenados", $"NombreCurso: {_nombreCurso},
PrecioInicial: {_precioInicial}, FormaPago: {_formaPago}", "OK");
```

```
    }
    public Pagina1()
    {
        InitializeComponent();
        BindingContext = this;
        if ((string.IsNullOrWhiteSpace(_nombreCurso))&&
(string.IsNullOrWhiteSpace(_formaPago)))
        {
            btnCalcularPrecio.IsEnabled = false;
        }
        else
        {
            btnCalcularPrecio.IsEnabled = true;
        }
    }
    private async void btnSeleccionarCurso_Clicked(object sender, EventArgs e)
    {
        await Shell.Current.GoToAsync($"Pagina2");
    }
    private async void btnSeleccionarPago_Clicked(object sender, EventArgs e)
    {
        await Shell.Current.GoToAsync($"Pagina3");
    }

    private void btnCalcularPrecio_Clicked(object sender, EventArgs e)
    {

        if (_formaPago.Equals("Efectivo"))
        {
            string precioEfectivo = _precioInicial.ToString();
            tvCalcularPrecio.Text = precioEfectivo;
            OnPropertyChanged();
        }
        if (_formaPago.Equals("Tarjeta"))
        {
            double calculo = _precioInicial * 0.9;
            string precioTarjeta = calculo.ToString();
            tvCalcularPrecio.Text = precioTarjeta;
            OnPropertyChanged();
        }
    }
}
```

Este código XAML describe la interfaz de una segunda página (Pagina2) en la aplicación .NET MAUI, donde el usuario puede seleccionar un curso. Aquí está el desglose:

1. **Encabezado de la página:**
    - El título "Selecciona tu curso" es mostrado en la parte superior mediante un Label centrado, con un tamaño de fuente grande y en negrita.
2. **Lista de cursos**: Cada curso se representa con un Frame, que tiene un fondo blanco y esquinas redondeadas, para darle una apariencia de tarjeta o caja, con un diseño vertical que agrupa elementos:
    - **Curso Llados:**

- Dentro del primer Frame, hay un HorizontalStackLayout que organiza dos elementos horizontalmente:
    1. Un ImageButton que muestra una imagen (curso_llados.jpeg) representando el curso y que, al hacer clic, activa el evento btnCursoLlados_Clicked.
    2. Un Label que muestra el precio del curso "Precio: 1000$" con estilo en negrita y color gris oscuro.
- **Curso Java:**
    - Similar al curso anterior, otro Frame contiene un ImageButton con la imagen del curso (curso_java.jpeg), y un Label que muestra el precio "Precio: 50$".
    - El evento que se ejecuta cuando el usuario selecciona este curso es btnCursoJava_Clicked.

3. **Interactividad:**
    - Cada curso tiene un botón en forma de imagen (ImageButton) que permite al usuario interactuar con los cursos visualmente. Los eventos btnCursoLlados_Clicked y btnCursoJava_Clicked se activan cuando el usuario selecciona un curso específico.

Pagina2.xaml

```
<?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="EJER_UD02v5.Pagina2"
             Title="Cursos"
             BackgroundColor="#f5f5f5">
    <VerticalStackLayout Padding="20">
        <Label Text="Selecciona tu curso"
               FontSize="24"
               FontAttributes="Bold"
               TextColor="#2C3E50"
               HorizontalOptions="Center"
               Margin="0,20"/>
        <!-- Curso Llados -->
        <Frame CornerRadius="10" BackgroundColor="White" Padding="10" Margin="0,10">
            <HorizontalStackLayout VerticalOptions="Center" Spacing="10">
                <ImageButton
                    x:Name="btnCursoLlados"
                    Source="curso_llados.jpeg"
                    WidthRequest="100"
                    HeightRequest="100"
                    BackgroundColor="Transparent"
                    Clicked="btnCursoLlados_Clicked" />
                <Label Text="Precio: 1000$"
                       VerticalOptions="Center"
                       TextColor="#34495E"
                       FontAttributes="Bold" />
            </HorizontalStackLayout>
        </Frame>
        <!-- Curso Java -->
        <Frame CornerRadius="10" BackgroundColor="White" Padding="10" Margin="0,10">
            <HorizontalStackLayout VerticalOptions="Center" Spacing="10">
```

```
                <ImageButton
                    x:Name="btnCursoJava"
                    Source="curso_java.jpeg"
                    WidthRequest="100"
                    HeightRequest="100"
                    BackgroundColor="Transparent"
                    Clicked="btnCursoJava_Clicked" />
                <Label Text="Precio: 50$"
                       VerticalOptions="Center"
                       TextColor="#34495E"
                       FontAttributes="Bold" />
            </HorizontalStackLayout>
        </Frame>
    </VerticalStackLayout>
</ContentPage>
```

Este código C# define la lógica de la página Pagina2, que maneja la selección de un curso en la aplicación .NET MAUI. Aquí tienes la explicación de cada parte:

1. **Clase Pagina2**:
    - La clase Pagina2 hereda de ContentPage, lo que la convierte en una página de contenido en la aplicación.
    - En el constructor Pagina2(), se llama a InitializeComponent() para inicializar los elementos de la interfaz definidos en el archivo XAML correspondiente.
2. **Manejo de eventos de selección de curso**:
    - Hay dos métodos que se ejecutan cuando el usuario selecciona uno de los cursos:
    - **btnCursoLlados_Clicked**:
        - Cuando el usuario selecciona el curso "Llados", se asigna el nombre del curso ("Curso Llados") y su precio inicial ("1000").
        - Luego, mediante Shell.Current.GoToAsync(), la aplicación navega a Pagina1 pasando los parámetros del curso seleccionado en la URL (precioInicial=1000&nombreCurso=Curso Llados).
    - **btnCursoJava_Clicked**:
        - De manera similar, cuando el usuario selecciona el curso "Java", se asigna el nombre del curso ("Curso Java") y su precio inicial ("50").
        - La navegación a Pagina1 también incluye los parámetros correspondientes del curso (precioInicial=50&nombreCurso=Curso Java).
3. **Navegación en Shell**:

- La navegación entre páginas se realiza con Shell.Current.GoToAsync(), que permite enviar parámetros a Pagina1 usando una URL de consulta.
- Esto es útil para transferir datos (como el nombre del curso y el precio) de una página a otra sin necesidad de usar variables globales.

Pagina2.xaml.cs

```
namespace EJER_UD02v5;
public partial class Pagina2 : ContentPage
{
      public Pagina2()
      {
        InitializeComponent();
      }
    private async void btnCursoLlados_Clicked(object sender, EventArgs e)
    {
        string nombreCurso="Curso Llados";
        string precioInicial="1000";
        await Shell.Current.GoToAsync("Pagina1?precioInicial=1000&nombreCurso=Curso
Llados");
    }
    private async void btnCursoJava_Clicked(object sender, EventArgs e)
    {
        string nombreCurso= "Curso Java";
        string precioInicial="50";
        await Shell.Current.GoToAsync("Pagina1?precioInicial=50&nombreCurso=Curso
Java");
    }
}
```

Este código XAML define la interfaz gráfica de la página Pagina3, la cual permite al usuario seleccionar la forma de pago en la aplicación .NET MAUI. A continuación, te explico cada parte del código:

1. **Encabezado de la página:**
   - Un Label centrado con el texto "Selecciona tu forma de pago", estilizado con un tamaño de fuente grande y en negrita. Sirve como el título de la página para guiar al usuario.
2. **Opciones de forma de pago**: La página presenta dos opciones de forma de pago: **Efectivo** y **Tarjeta**. Cada opción está organizada en un Frame con un diseño visual atractivo:
   - **Opción Efectivo:**
     - Dentro del Frame hay un HorizontalStackLayout que organiza la opción de forma horizontal.
     - Un ImageButton que muestra una imagen (efectivo.jpeg) representando el pago en efectivo, con un evento de clic asociado (btnEfectivo_Clicked).

- Al lado de la imagen, un Label muestra el texto "Efectivo", centrado y estilizado en negrita y con un color gris oscuro.
- **Opción Tarjeta:**
  - Similar a la opción de "Efectivo", otro Frame contiene un ImageButton con la imagen de un datafono (datafono.jpeg) que representa el pago con tarjeta, con el evento de clic correspondiente (btnTarjeta_Clicked).
  - El Label a la derecha de la imagen muestra el texto "Tarjeta" con el mismo estilo que la opción anterior.

3. **Diseño de la página:**
   - Los elementos están organizados en un VerticalStackLayout, lo que significa que se colocan uno debajo del otro de manera vertical.
   - Cada opción de pago está dentro de un Frame que tiene bordes redondeados (CornerRadius="10") y un fondo blanco, haciendo que las opciones se destaquen visualmente del fondo gris claro de la página (#f5f5f5).

Pagina3.xaml

```
<?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="EJER_UD02v5.Pagina3"
             Title="Seleccionar Forma de Pago"
             BackgroundColor="#f5f5f5">
    <VerticalStackLayout Padding="20">
        <Label Text="Selecciona tu forma de pago"
               FontSize="24"
               FontAttributes="Bold"
               TextColor="#2C3E50"
               HorizontalOptions="Center"
               Margin="0,20"/>
        <!-- Opción Efectivo -->
        <Frame CornerRadius="10" BackgroundColor="White" Padding="10" Margin="0,10">
            <HorizontalStackLayout VerticalOptions="Center" Spacing="10">
                <ImageButton
                    x:Name="btnEfectivo"
                    Source="efectivo.jpeg"
                    WidthRequest="100"
                    HeightRequest="100"
                    BackgroundColor="Transparent"
                    Clicked="btnEfectivo_Clicked" />
                <Label Text="Efectivo"
                       VerticalOptions="Center"
                       TextColor="#34495E"
                       FontAttributes="Bold" />
            </HorizontalStackLayout>
        </Frame>
        <!-- Opción Tarjeta -->
        <Frame CornerRadius="10" BackgroundColor="White" Padding="10" Margin="0,10">
            <HorizontalStackLayout VerticalOptions="Center" Spacing="10">
                <ImageButton
```

```
                    x:Name="btnTarjeta"
                    Source="datafono.jpeg"
                    WidthRequest="100"
                    HeightRequest="100"
                    BackgroundColor="Transparent"
                    Clicked="btnTarjeta_Clicked" />
                <Label Text="Tarjeta"
                       VerticalOptions="Center"
                       TextColor="#34495E"
                       FontAttributes="Bold" />
            </HorizontalStackLayout>
        </Frame>
    </VerticalStackLayout>
</ContentPage>
```

Este código C# define la lógica de la página Pagina3 en la aplicación .NET MAUI, encargada de manejar la selección de la forma de pago. Aquí te explico cada parte:

1. **Clase Pagina3**:
   - Pagina3 hereda de ContentPage, lo que la convierte en una página de contenido dentro de la aplicación.
   - En el constructor Pagina3(), se llama a InitializeComponent(), que inicializa los elementos de la interfaz definidos en el archivo XAML correspondiente.
2. **Método btnEfectivo_Clicked**:
   - Este método se ejecuta cuando el usuario selecciona la opción de pago en **Efectivo**.
   - Al hacer clic en el botón asociado, se establece la forma de pago como "Efectivo" y se navega a Pagina1 utilizando Shell.Current.GoToAsync().
   - En la URL de navegación, se pasa el parámetro formaPago=Efectivo, que es recibido por Pagina1 para continuar con el proceso.
3. **Método btnTarjeta_Clicked**:
   - De manera similar al anterior, este método se ejecuta cuando el usuario selecciona la opción de pago con **Tarjeta**.
   - Se establece la forma de pago como "Tarjeta", y la navegación a Pagina1 incluye el parámetro formaPago=Tarjeta, que también será usado en Pagina1.
4. **Navegación en Shell**:
   - Shell.Current.GoToAsync() permite que los datos sobre la forma de pago se pasen a la siguiente página (Pagina1) como parámetros en la URL.

- Esta navegación basada en Shell es fundamental para transferir la selección de forma de pago entre diferentes páginas de la aplicación sin depender de variables globales.

Pagina3.xaml.cs

```
namespace EJER_UD02v5;
public partial class Pagina3 : ContentPage
{
       public Pagina3()
       {

         InitializeComponent();
       }
    private async void btnEfectivo_Clicked(object sender, EventArgs e)
    {
        string formaPago;
        await Shell.Current.GoToAsync("Pagina1?formaPago=Efectivo");
    }
    private async void btnTarjeta_Clicked(object sender, EventArgs e)
    {
        string formaPago;
        await Shell.Current.GoToAsync("Pagina1?formaPago=Tarjeta");
    }
}
```

Este archivo XAML define la estructura de navegación de la aplicación mediante el componente **Shell** en .NET MAUI. El **Shell** proporciona una forma simplificada y eficiente de gestionar la navegación en aplicaciones de múltiples páginas. A continuación, te explico cada parte del código:

1. **Shell:**
   - El elemento <Shell> actúa como el contenedor principal para la navegación en la aplicación.
   - A través de este componente, la aplicación define y organiza su estructura de páginas y la forma en que el usuario navega entre ellas.
2. **Atributos del Shell:**
   - **x:Class**: Especifica el nombre de la clase de la aplicación, en este caso, EJER_UD02v5.AppShell, que conecta este archivo XAML con su código detrás (C#).
   - **xmlns y xmlns:x**: Estos atributos importan los espacios de nombres necesarios para usar elementos XAML y las características de .NET MAUI.
   - **xmlns:local**: Importa el espacio de nombres local, que corresponde a la aplicación actual (EJER_UD02v5), permitiendo acceder a las clases de la aplicación, como Pagina1, Pagina2, y Pagina3.
   - **Shell.FlyoutBehavior="Disabled"**: Deshabilita el menú lateral (flyout) para que la navegación no muestre un panel

lateral con las opciones, forzando una navegación más directa entre las páginas.

3. **Contenido del Shell:**
   - Se definen tres elementos <ShellContent>, uno para cada página de la aplicación:
     - **Title="Pagina1"**: Este título se usa para la página Pagina1, y el contenido de esta página se especifica con el atributo ContentTemplate, que apunta a local:Pagina1, es decir, la clase Pagina1 de la aplicación.
     - **Title="Pagina2"**: Configura la navegación para la segunda página de la aplicación, Pagina2, de manera similar.
     - **Title="Pagina3"**: La tercera página (Pagina3) también se configura del mismo modo.
4. **Navegación en Shell:**
   - El **Shell** permite la navegación entre las diferentes páginas declaradas como ShellContent. Cada uno de estos elementos define una página que será accesible en la aplicación.
   - Aunque no se ha configurado un menú de navegación visible, el usuario puede navegar entre estas páginas a través de comandos o eventos en el código detrás.

AppShell.xaml

```
<?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="Ejer2.Pages.Pagina4">
    <ScrollView>
        <Grid.Background>
            <LinearGradientBrush StartPoint="0,0"
                                 EndPoint="1,0">
                <GradientStopCollection>
                    <GradientStop Offset="0" Color="DarkBlue"></GradientStop>
                    <GradientStop Offset="1" Color="LightCoral"></GradientStop>
                </GradientStopCollection>
            </LinearGradientBrush>
        </Grid.Background>
        <VerticalStackLayout
            Padding="30,0"
            Spacing="25"
            HeightRequest="600"
            WidthRequest="400">
            <Image
                x:Name="FotoDireccion"
                Source="direccion.jpeg"
                WidthRequest="300"
                HeightRequest="300"
                HorizontalOptions="Center">

            </Image>
```

```
            <Label x:Name="Titulo4" Text="Direccion" FontSize="Title"
HorizontalTextAlignment="Center"></Label>
            <Entry x:Name="etDireccion"></Entry>
            <Button x:Name="btnVolverTo3" Text="Volver"
Clicked="btnVolverTo3_Clicked"></Button>
            <Button x:Name="btnSiguienteTo5" Text="Siguiente"
Clicked="btnSiguienteTo5_Clicked"></Button>
        </VerticalStackLayout>
    </ScrollView>
</ContentPage>
```

Este código C# define la clase AppShell, que extiende de Shell y actúa como el punto central de la navegación en la aplicación .NET MAUI. Aquí te explico cada parte:

1. **Clase AppShell:**
   - Hereda de Shell, lo que convierte a esta clase en el manejador de la navegación dentro de la aplicación. El uso de Shell permite una navegación más estructurada y sencilla entre las diferentes páginas o vistas de la app.
2. **Constructor AppShell():**
   - Este constructor inicializa la clase y ejecuta dos operaciones importantes:
     - **InitializeComponent()**: Este método se encarga de cargar e inicializar los componentes XAML asociados a esta clase, configurados en el archivo AppShell.xaml.
     - **Routing.RegisterRoute()**: Aquí se definen las rutas de navegación manual entre las páginas de la aplicación.
3. **Registro de rutas:**
   - **Routing.RegisterRoute("Pagina1", typeof(Pagina1))**: Registra la página Pagina1 bajo la ruta "Pagina1". Esto permite que cualquier componente de la aplicación pueda navegar hacia esta página usando esta ruta.
   - **Routing.RegisterRoute("Pagina2", typeof(Pagina2))**: Similar al anterior, registra la ruta para Pagina2.
   - **Routing.RegisterRoute("Pagina3", typeof(Pagina3))**: También registra la ruta para Pagina3.
4. **Navegación en la aplicación:**
   - El uso de Routing.RegisterRoute() permite una navegación más flexible. Las páginas no solo se pueden acceder por medio del menú o Shell estándar, sino también mediante rutas programáticas o URLs específicas.
   - Esto es útil cuando se requiere navegar entre páginas con parámetros (como se hace en los ejemplos anteriores con GoToAsync("Pagina1?param=value")).

AppShell.xaml.cs

```
namespace EJER_UD02v5
{
    public partial class AppShell : Shell
    {
        public AppShell()
        {
            InitializeComponent();
            Routing.RegisterRoute("Pagina1", typeof(Pagina1));
            Routing.RegisterRoute("Pagina2", typeof(Pagina2));
            Routing.RegisterRoute("Pagina3", typeof(Pagina3));
        }
    }
}
```

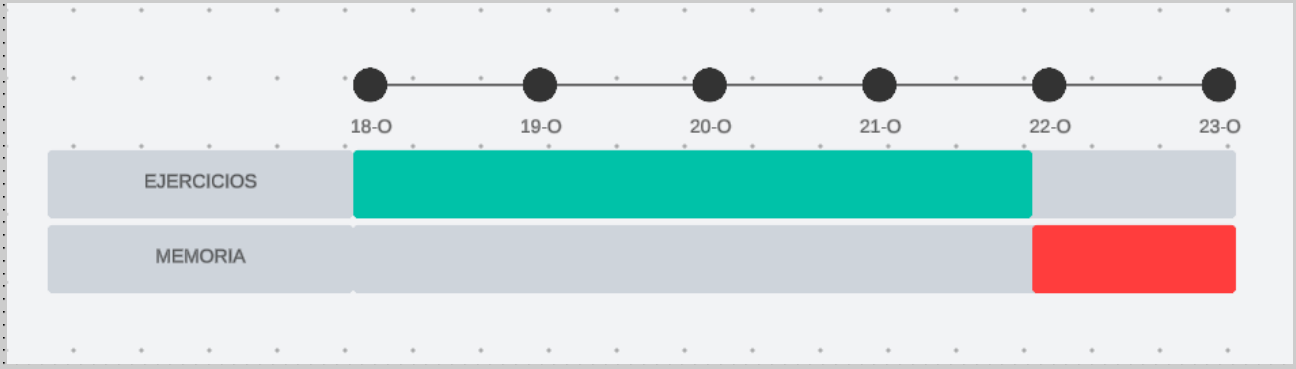
Imagen Pagina1:

Imagen Pagina2:

Imagen Pagina3:

Imagen Pagina1(con datos):

DIAGRAMA DE GANT:



PROBLEMAS:

Los mayores problemas que he tenido ha sido a la hora de comprender como funciona MAUI , y el como almacenar y hacer un setter de los datos.

CONCLUSIÓN:

El entorno de desarrollo de Visual Studio va bastante mal, y no es muy intuitivo.

WEBGRAFIA:

https://learn.microsoft.com/es-es/dotnet/maui/get-started/first-app?view=net-maui-8.0&tabs=vswin&pivots=devices-android