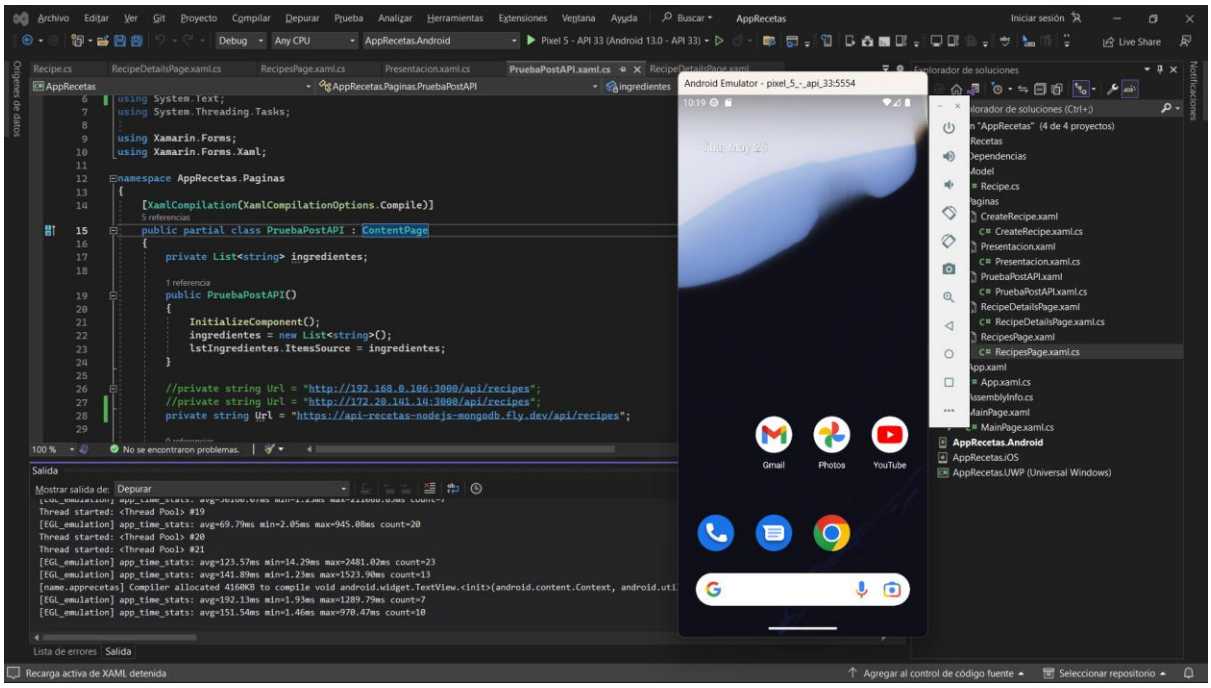
UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE SOFTWARE
ASIGNATURA: APLICACIONES MOVILES NIVEL: 07

Nombres completos:	Roberth Gabriel Lima Carvajal
Fecha:	25/05/2023
Tema:	Recetas (Maestro-Detalle)
Objetivo de esta actividad:	Desarrollar una aplicación móvil con la ayuda de Visual Studio 2022 usando Xamarin.Forms para la creación de una lista de recetas con sus funciones básicas, como agregar, eliminar o editar las recetas, en la cual se pueden agregar varios ingredientes y entender de mejor forma cómo funciona esta solución multiplataforma

INDICACIONES:

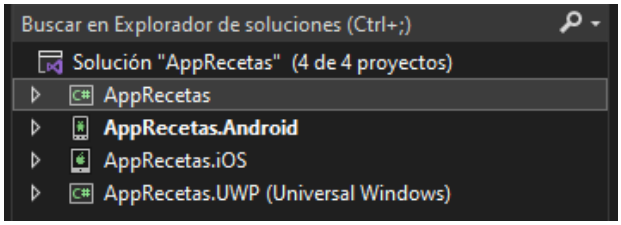
Para la realización de esta actividad se debe de tener el IDE de Visual Studio 2022, e instalado una máquina virtual móvil, para la ejecución del código desarrollado.

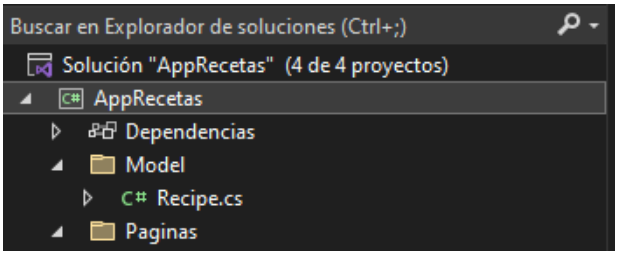


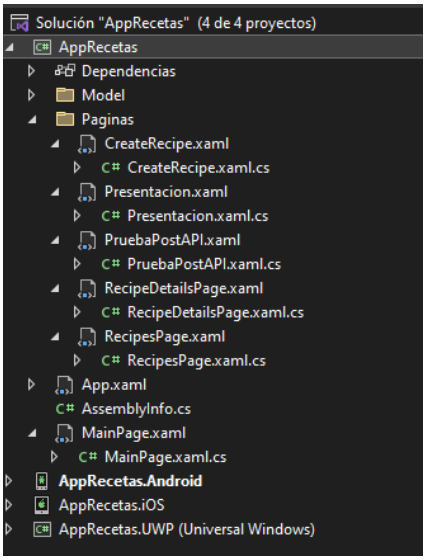
Una vez tenido todo esto, deberemos de programar una aplicación móvil usando Xamarin.Forms que funcione como una lista de recetas.

DESARROLLO:

1. Crear un Proyecto Xamarin.Forms, y que esté cargado y listo para empezar a escribir líneas de código (En este caso se llama AppRecetas):







2. Código dentro de cada archivo:
   - App.xaml.css

```
using System;
using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace AppRecetas
{
    7 referencias
    public partial class App : Application
    {
        3 referencias
        public App()
        {
            InitializeComponent();

            MainPage = new NavigationPage(new AppRecetas.Paginas.Presentacion());

            //MainPage = new MainPage();
        }

        0 referencias
        protected override void OnStart()
        {
        }

        0 referencias
        protected override void OnSleep()
        {
        }

        0 referencias
        protected override void OnResume()
        {
        }
    }
}
```

- Presentación.xaml.css

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace AppRecetas.Paginas
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    5 referencias
    public partial class Presentacion : ContentPage
    {
        1 referencia
        public Presentacion ()
        {
            InitializeComponent ();
        }

        0 referencias
        private void Button_Clicked(object sender, EventArgs e)
        {
            //Abrir una nueva pagina, misma que se coloca en la cima de la pila de paginas
            Navigation.PushAsync(new RecipesPage());
        }

    }
}
```

- Presentación.xaml

```
HorizontalOptions
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="AppRecetas.Paginas.Presentacion">
    <ContentPage.Content>
        <StackLayout>
            <Label Text="Bienvenido a Xamarin.Forms!"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
            <Button Text="Abrir Pagina de Recetas" Clicked="Button_Clicked"/>
            <Label Text="Proyecto de Creación de lista de Recetas"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
            <Label Text="Estudiante: Roberth Lima"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
            <Label Text="Universidad Técnica del Norte"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
            <Label Text="Carrera: Software"
                VerticalOptions="CenterAndExpand"
                HorizontalOptions="CenterAndExpand" />
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

- PruebaPostAPI.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Security;
using System.Text;
using System.Threading.Tasks;

using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace AppRecetas.Paginas
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    5 referencias
    public partial class PruebaPostAPI : ContentPage
    {
        private List<string> ingredientes;

        1 referencia
        public PruebaPostAPI()
        {
            InitializeComponent();
            ingredientes = new List<string>();
            lstIngredientes.ItemsSource = ingredientes;
        }

        //private string Url = "http://192.168.0.106:3000/api/recipes";
        //private string Url = "http://172.20.141.14:3000/api/recipes";
        private string Url = "https://api-recetas-nodejs-mongodb.fly.dev/api/recipes";

        0 referencias
        private void Button_AgregarIngrediente_Clicked(object sender, EventArgs e)
        {
            if (!string.IsNullOrWhiteSpace(txtNuevoIngrediente.Text))
            {
                ingredientes.Add(txtNuevoIngrediente.Text);
                lstIngredientes.ItemsSource = null;
                lstIngredientes.ItemsSource = ingredientes;
                txtNuevoIngrediente.Text = string.Empty;
            }
        }

        0 referencias
        private async void Button_Clicked(object sender, EventArgs e)
        {
            using (var wc = new WebClient())
            {
                ServicePointManager.ServerCertificateValidationCallback = new RemoteCertificateValidationCallback(delegate { return true; });


                wc.Headers.Add("Content-Type", "application/json");
                var datos = new Model.Recipe
                {
                    title = txtTitulo.Text,
                    instructions = txtInstrucciones.Text,
                    ingredients = new List<string>(ingredientes) // Crear una nueva lista con los ingredientes
                };

                var json = Newtonsoft.Json.JsonConvert.SerializeObject(datos);
                await wc.UploadStringTaskAsync(Url, "POST", json);

                await DisplayAlert("Éxito", "La receta se ha creado con éxito.", "OK");

                //lblDatos.Text = "DATOS INSERTADOS CORRECTAMENTE";

                /*
                txtTitulo.Text = "";
                txtInstrucciones.Text = "";
                */
                await Navigation.PopAsync();
            }
        }

        0 referencias
        private void ListView_ItemSelected(object sender, SelectedItemChangedEventArgs e)
        {
            if (e.SelectedItem != null)
            {
                ingredientes.Remove(e.SelectedItem.ToString());
                lstIngredientes.ItemsSource = null;
                lstIngredientes.ItemsSource = ingredientes;
            }
        }
    }
}
```

- PruebaPostAPI.xaml

```xml
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="AppRecetas.Paginas.PruebaPostAPI">
    <ContentPage.Content>
        <StackLayout>
            <Label Text="Welcome to Xamarin.Forms!"
               VerticalOptions="CenterAndExpand"
               HorizontalOptions="CenterAndExpand" />
            <Label x:Name="lblDatos" Text="Id"/>
            <Entry Placeholder="Titulo" x:Name="txtTitulo"/>
            <Entry Placeholder="Intrucciones" x:Name="txtInstrucciones"/>
            <StackLayout>
                <Entry Placeholder="Nuevo Ingrediente" x:Name="txtNuevoIngrediente"/>
                <Button Text="Agregar Ingrediente" Clicked="Button_AgregarIngrediente_Clicked" />
                <ListView x:Name="lstIngredientes" ItemSelected="ListView_ItemSelected">
                    <ListView.ItemTemplate>
                        <DataTemplate>
                            <TextCell Text="{Binding}" />
                        </DataTemplate>
                    </ListView.ItemTemplate>
                </ListView>
            </StackLayout>
            <Button Text="Agregar Receta" Clicked="Button_Clicked" />
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

- RecipeDetailsPage.xaml.cs

```
AppRecetas.Paginas.RecipeDetailsPage                editarReceta(object sender, EventArgs e)
using System.Text;
using System.Threading.Tasks;
using System.Collections.ObjectModel;


using Xamarin.Forms;
using Xamarin.Forms.Xaml;
using System.Net.Security;

namespace AppRecetas.Paginas
{
    [XamlCompilation(XamlCompilationOptions.Compile)]
    5 referencias
    public partial class RecipeDetailsPage : ContentPage
    {
        //private string Url = "http://192.168.0.106:3000/api/recipes";
        //private string Url = "http://172.20.141.14:3000/api/recipes";
        private string Url = "http://api-recetas-nodejs-mongodb.fly.dev/api/recipes";

        private List<string> ingredientes;

        //private string recipeId;

        1 referencia
        public RecipeDetailsPage(Model.Recipe recipe, string id)
        {
            InitializeComponent();

            txtTitulo.Text = recipe.title;
            txtInstrucciones.Text = recipe.instructions;

            txtID.Text = recipe.id.ToString();
            //recipeId = recipe.id.ToString();
            ingredientes = new List<string>();

            if (recipe.ingredients != null)
            {
                foreach (var ingrediente in recipe.ingredients)
                {
                    ingredientes.Add(ingrediente);
                }
            }

            listIngredientes.ItemsSource = ingredientes; // Asigna la lista de ingredientes al ListView
```

```
0 referencias
private async void ShowIngredients(object sender, EventArgs e)
{
    await DisplayAlert("Ingredientes", "Aquí se mostrarían los ingredientes", "OK");
}

0 referencias
private async void Button_Clicked(object sender, EventArgs e)
{
    bool answer = await DisplayAlert("Confirmación", "¿Estás seguro de que deseas eliminar esta receta?", "Sí", "No");

    if (answer)
    {
        using (var wc = new WebClient())
        {
            wc.Headers.Add("Content-Type", "application/json");
            ServicePointManager.ServerCertificateValidationCallback = new RemoteCertificateValidationCallback(delegate { return true; });
            wc.UploadString(Url + "/" + txtID.Text, "DELETE", "");
            lblVerificar.Text = "DATOS BORRADOS CON ÉXITO";

            await Navigation.PopAsync();
        }
    }
}
```

```
0 referencias
private async void editarReceta(object sender, EventArgs e)
{
    using (var wc = new WebClient())
    {
        wc.Headers.Add("Content-Type", "application/json");
        ServicePointManager.ServerCertificateValidationCallback = new RemoteCertificateValidationCallback(delegate { return true; });
        var datos = new Model.Recipe
        {
            id = int.Parse(txtID.Text),
            title = txtTitulo.Text,
            instructions = txtInstrucciones.Text,
            ingredients = new List<string>()
        };

        foreach (var item in listIngredientes.ItemsSource)
        {
            if (item is string ingrediente)
            {
                datos.ingredients.Add(ingrediente);
            }
        }

        var json = Newtonsoft.Json.JsonConvert.SerializeObject(datos);
        wc.UploadString(Url + "/" + txtID.Text, "PUT", json);

        // Mostrar ventana emergente de éxito
        await DisplayAlert("Éxito", "La receta se ha editado con éxito.", "Aceptar");
    }
}


0 referencias
private async void AgregarIngrediente_Clicked(object sender, EventArgs e)
{
    string nuevoIngrediente = await DisplayPromptAsync("Nuevo Ingrediente", "Ingrese el nuevo ingrediente", "Agregar", "Cancelar", "");

    if (!string.IsNullOrWhiteSpace(nuevoIngrediente))
    {
        ingredientes.Add(nuevoIngrediente);
        listIngredientes.ItemsSource = null;
        listIngredientes.ItemsSource = ingredientes;
    }
}
}
```

- RecipeDetailsPage.xaml

```xml
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="AppRecetas.Paginas.RecipeDetailsPage"
             Title="Detalles de la receta">
    <StackLayout Padding="20" Spacing="10">
        <Label x:Name="lblVerificar"/>

        <!-- Mostrar ID -->
        <StackLayout Orientation="Horizontal" VerticalOptions="Center">
            <Label Text="ID:" FontSize="18" HorizontalOptions="EndAndExpand" VerticalOptions="Center"/>
            <Label x:Name="txtID" FontSize="18" HorizontalOptions="StartAndExpand" VerticalOptions="Center"/>
        </StackLayout>

        <!-- Mostrar Receta -->
        <Label Text="Receta:" FontSize="18"/>
        <Entry Placeholder="Receta" x:Name="txtTitulo" FontSize="18" />

        <!-- Mostrar Instrucciones -->
        <Label Text="Instrucciones:" FontSize="18"/>
        <Editor Placeholder="Instrucciones" x:Name="txtInstrucciones" FontSize="18" HeightRequest="120" />

        <!-- Mostrar Ingredientes -->
        <Label Text="Ingredientes:" FontSize="18" />

        <StackLayout Orientation="Horizontal">
            <ListView x:Name="listIngredientes" VerticalOptions="FillAndExpand" SeparatorVisibility="None">
                <ListView.ItemTemplate>
                    <DataTemplate>
                        <ViewCell>
                            <StackLayout Padding="10">
                                <Entry Text="{Binding}" FontSize="18" />
                            </StackLayout>
                        </ViewCell>
                    </DataTemplate>
                </ListView.ItemTemplate>
            </ListView>

            <Button Text="+" FontSize="24" WidthRequest="40" HeightRequest="40" CornerRadius="20" BackgroundColor="#536DFE" TextColor="White" Clicked="AgregarIngrediente_Clicked"/>
        </StackLayout>

        <!-- Botones -->
        <StackLayout Orientation="Horizontal" HorizontalOptions="FillAndExpand" Spacing="10">
            <Button Text="Eliminar Receta" BackgroundColor="#FF4081" TextColor="White" FontSize="18" Clicked="Button_Clicked" />
            <Button Text="Editar Receta" BackgroundColor="#536DFE" TextColor="White" FontSize="18" Clicked="editarReceta" />
        </StackLayout>
    </StackLayout>
</ContentPage>
```

- RecipesPage.xaml.cs

```cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net;
using System.Net.Security;
using System.Text;
using System.Threading.Tasks;

using Xamarin.Forms;
using Xamarin.Forms.Xaml;

namespace AppRecetas.Paginas
{
    //[XamlCompilation(XamlCompilationOptions.Compile)]
    5 referencias
    public partial class RecipesPage : ContentPage
    {
        5 referencias
        public List<Model.Recipe> Recetas { get; set; }
        1 referencia
        public bool RecetasVacias => Recetas == null || Recetas.Count == 0;
        0 referencias
        public bool RecetasVisibles => !RecetasVacias;
        1 referencia
        public RecipesPage()
        {
            InitializeComponent();
            BindingContext = this;

            CargarRecetas(); // Llama al método para cargar las recetas automáticamente al abrir la página
        }

        0 referencias
        protected override void OnAppearing()
        {
            base.OnAppearing();

            // Vuelve a cargar las recetas cada vez que la página se muestre
            CargarRecetas();
        }

        //private string Url = "http://192.168.0.106:3000/api/recipes";
        //private string Url = "http://172.20.141.14:3000/api/recipes";
        private string Url = "https://api-recetas-nodejs-mongodb.fly.dev/api/recipes";
```

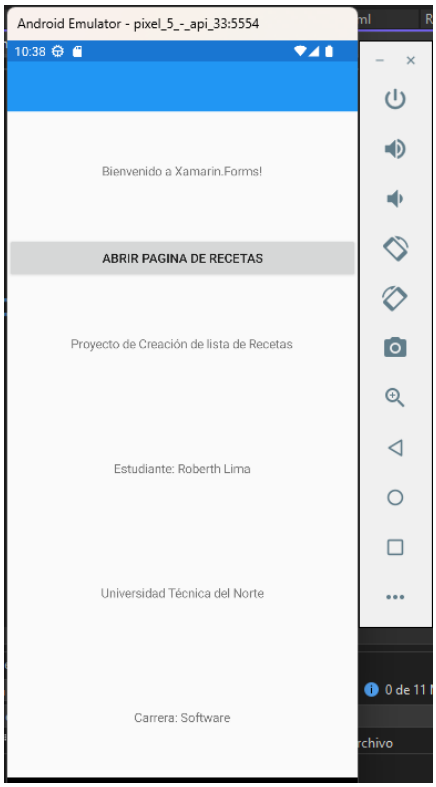
```
2 referencias
private async void CargarRecetas()
{
    using (var wc = new WebClient())
    {
        ServicePointManager.ServerCertificateValidationCallback = new RemoteCertificateValidationCallback(delegate { return true; });

        wc.Headers.Add("Content-Type", "application/json");
        var json = await wc.DownloadStringTaskAsync(new Uri(Url));
        Recetas = Newtonsoft.Json.JsonConvert.DeserializeObject<List<Model.Recipe>>(json);
    }

    recipesTable.Root.Clear(); // Limpiar las filas existentes

    var headerSection = new TableSection();
    var headerCell = new ViewCell();

    var headerGrid = new Grid();
    headerGrid.ColumnDefinitions.Add(new ColumnDefinition { Width = new GridLength(1, GridUnitType.Star) });
    headerGrid.ColumnDefinitions.Add(new ColumnDefinition { Width = new GridLength(1, GridUnitType.Star) });

    var headerLabel1 = new Label { Text = "Receta", FontAttributes = FontAttributes.Bold, HorizontalOptions = LayoutOptions.Center, VerticalOptions = LayoutOptions.Center };
    var headerLabel2 = new Label { Text = "Instrucciones", FontAttributes = FontAttributes.Bold, HorizontalOptions = LayoutOptions.Center, VerticalOptions = LayoutOptions.Center };

    headerGrid.Children.Add(headerLabel1, 0, 0);
    headerGrid.Children.Add(headerLabel2, 1, 0);

    headerCell.View = headerGrid;
    headerSection.Add(headerCell);
    recipesTable.Root.Add(headerSection);

    if (Recetas != null)
    {
        foreach (var receta in Recetas)
        {
            var titleCell = new TextCell { Text = receta.title };
            var instructionsCell = new TextCell { Text = receta.instructions };

            var tableSection = new TableSection();
            var viewCell = new ViewCell();

            var grid = new Grid();
            grid.ColumnDefinitions.Add(new ColumnDefinition { Width = new GridLength(1, GridUnitType.Star) });
            grid.ColumnDefinitions.Add(new ColumnDefinition { Width = new GridLength(1, GridUnitType.Star) });

            var titleLabel = new Label { Text = receta.title, FontAttributes = FontAttributes.Bold, HorizontalOptions = LayoutOptions.Center, VerticalOptions = LayoutOptions.Center };
            var instructionsLabel = new Label { Text = receta.instructions, FontAttributes = FontAttributes.Bold, HorizontalOptions = LayoutOptions.Center, VerticalOptions = LayoutOptions.Center };

            grid.Children.Add(titleLabel, 0, 0);
            grid.Children.Add(instructionsLabel, 1, 0);

            viewCell.View = grid;
            tableSection.Add(viewCell);
            recipesTable.Root.Add(tableSection);
```

```
            recipesTable.Root.Add(tableSection);

            // Agregar el evento ItemTapped a la celda de la receta
            viewCell.Tapped += async (s, e) =>
            {
                if (viewCell.View != null)
                {
                    // Abrir una nueva página, pasando la receta seleccionada como parámetro
                    await Navigation.PushAsync(new RecipeDetailsPage(receta,null));
                }
            };
        }
    }
    recipesTable.IsVisible = true;
}

0 referencias
private void Button_Clicked(object sender, EventArgs e)
{
    // Abrir una nueva página, misma que se coloca en la cima de la pila de páginas
    Navigation.PushAsync(new PruebaPostAPI());
}
}
}
```

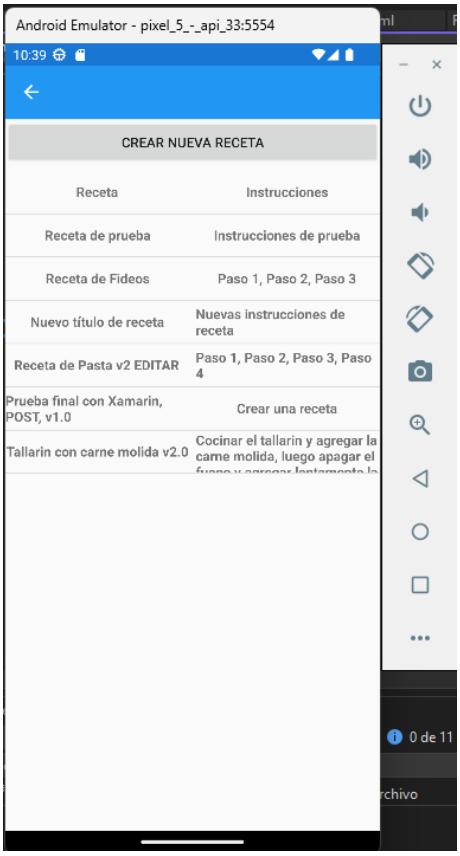
- RecipesPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:local="clr-namespace:AppRecetas.Paginas"
             x:Class="AppRecetas.Paginas.RecipesPage">
    <ContentPage.Content>
        <StackLayout>
            <Button Text="Crear Nueva Receta" Clicked="Button_Clicked"></Button>
            <TableView x:Name="recipesTable" IsVisible="False">
                <TableRoot>
                    <TableSection Title="Recetas">
                        <ViewCell>
                            <Grid>
                                <Grid.ColumnDefinitions>
                                    <ColumnDefinition Width="*" />
                                    <ColumnDefinition Width="*" />
                                </Grid.ColumnDefinitions>
                                <Label Grid.Column="0" Text="Titulo" FontAttributes="Bold" HorizontalOptions="Center" VerticalOptions="Center" />
                                <Label Grid.Column="1" Text="Instrucciones" FontAttributes="Bold" HorizontalOptions="Center" VerticalOptions="Center" />
                            </Grid>
                        </ViewCell>
                        <TextCell Text="No hay recetas cargadas." x:Name="emptyCell" />
                    </TableSection>
                </TableRoot>
            </TableView>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>
```

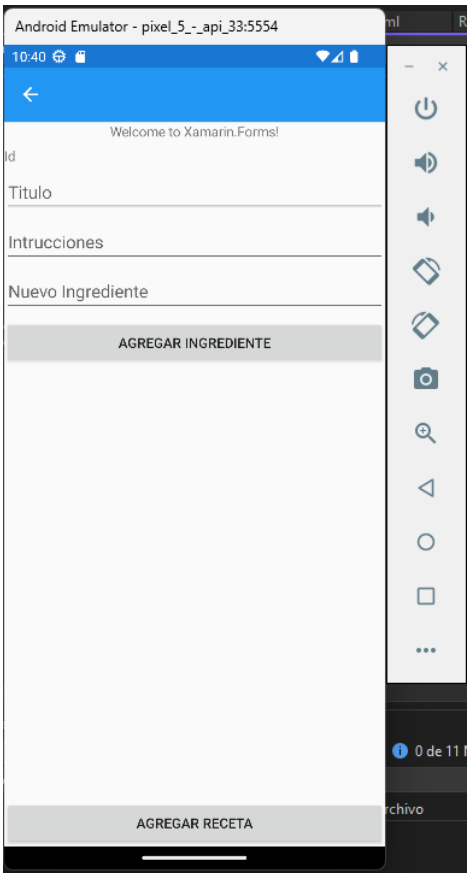
3. Pantallas creadas:
   - Pantalla Principal



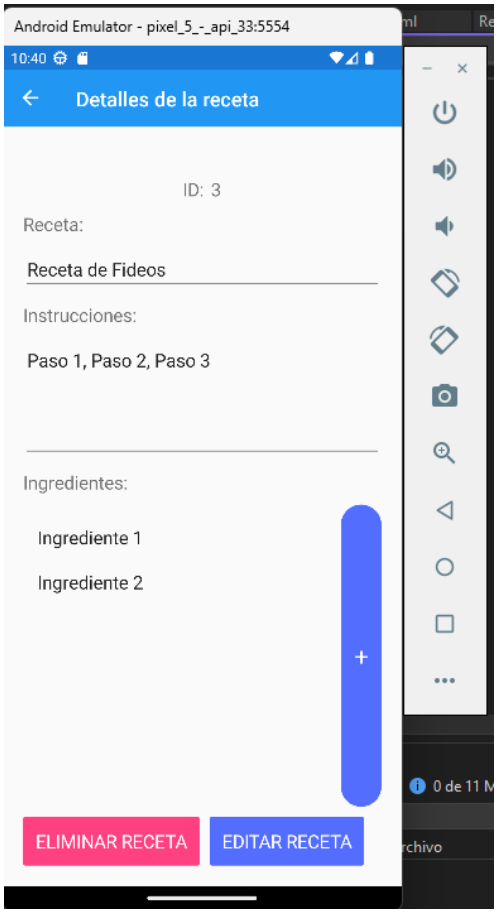
   - Pantalla de las Recetas



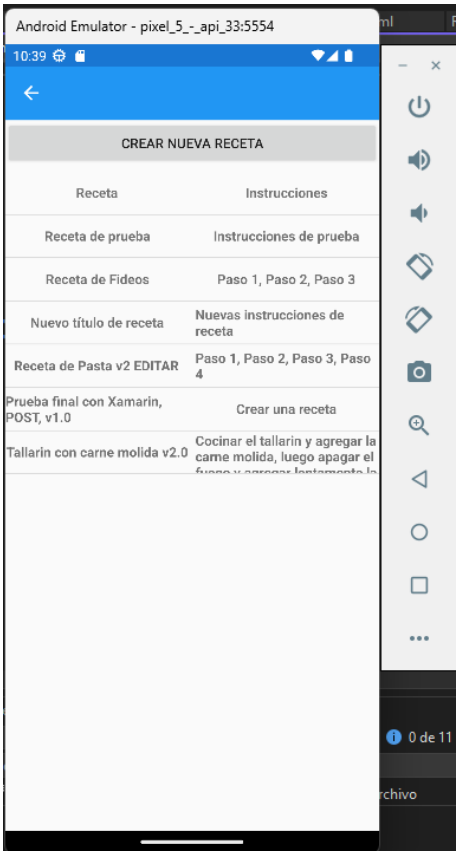
4. Probando funcionalidades
   - Crear Nueva Receta

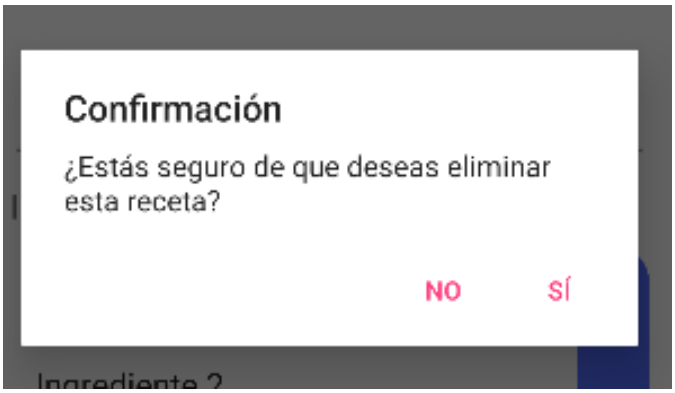


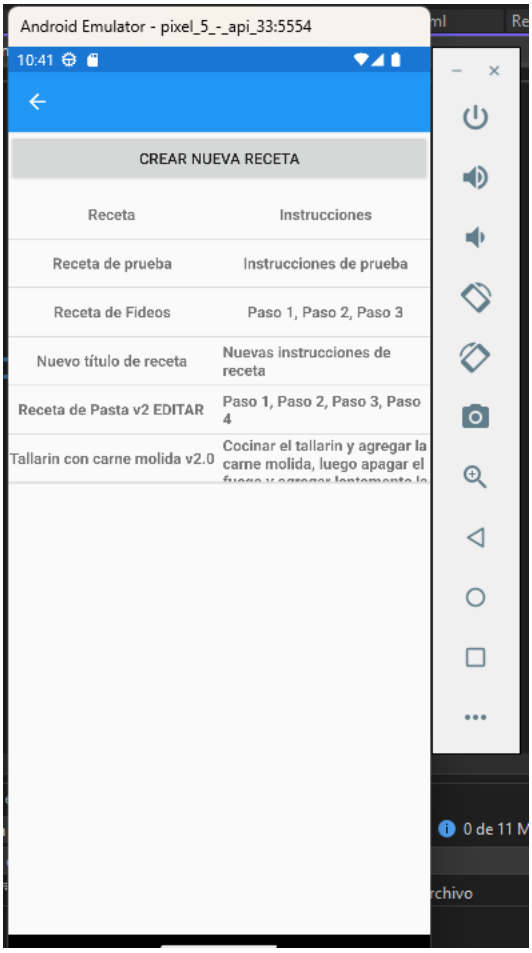
   - Editar una receta existente



- Eliminar una receta







CONCLUSIONES:

Xamarin.Forms es una opción increíblemente útil para desarrollar aplicaciones que funcionen en múltiples plataformas. Con esta herramienta, puedes escribir el código una vez y utilizarlo en iOS, Android y Windows, lo cual ahorra tiempo y recursos. No es necesario crear interfaces de usuario separadas para cada plataforma, ya que Xamarin.Forms permite crear interfaces compartidas que se adaptan automáticamente a cada sistema operativo.

A pesar de ser una solución multiplataforma, Xamarin.Forms ofrece una experiencia de usuario nativa en cada plataforma. Esto se logra al utilizar controles y componentes nativos de iOS, Android y Windows. La ventaja es que tus aplicaciones se verán y funcionarán de manera similar a las aplicaciones desarrolladas específicamente para cada plataforma, brindando a los usuarios una experiencia fluida y familiar.

Xamarin.Forms se basa en el sólido framework de desarrollo .NET, lo que te brinda acceso a una amplia gama de bibliotecas y herramientas para facilitar el proceso de desarrollo. Puedes aprovechar el poder y la flexibilidad de .NET para crear aplicaciones con características avanzadas. Además, Xamarin.Forms se integra sin problemas con otras tecnologías de Microsoft, como Azure y Xamarin Test Cloud, lo que te permite implementar y probar tus aplicaciones de forma más eficiente.