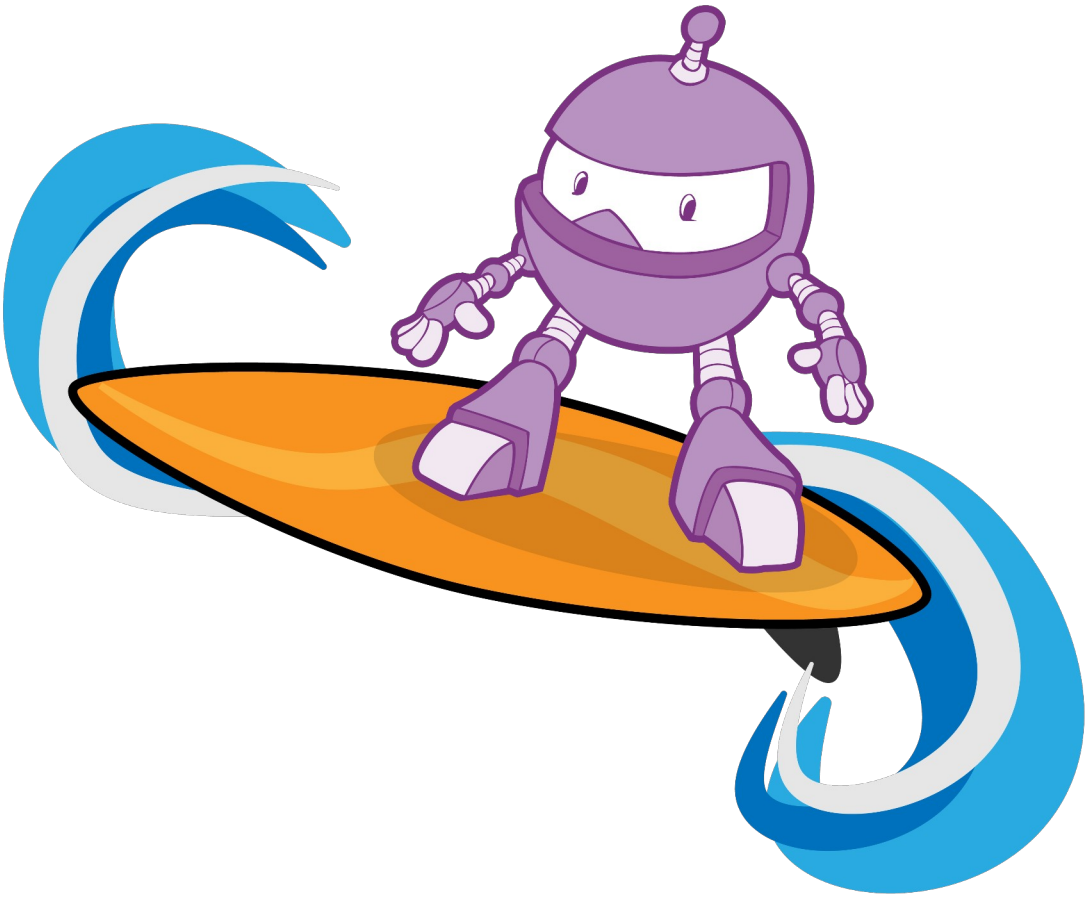
Gestión de Niveles de Idioma: Desarrollo con .NET MAUI



MARIO FERNÁNDEZ

DIN

2ºDAM

Índice

1. Introducción

En esta actividad, diseñé y construí una aplicación en .NET MAUI dedicada a gestionar niveles de idiomas como Valenciano, Inglés y Francés. El objetivo era crear algo práctico, visual y súper intuitivo para el usuario.

Este proyecto no solo buscaba cubrir una necesidad funcional, sino también proporcionar una experiencia de usuario fluida y agradable. La interfaz está diseñada para facilitar la interacción y adaptarse a diferentes niveles de habilidad.

2. Objetivos

- Crear una app funcional y amigable en .NET MAUI.
- Diseñar una interfaz atractiva y fácil de usar.
- Permitir que los usuarios gestionen sus niveles de idioma.
- Utilizar métodos asincrónicos para optimizar la experiencia.

3. Implementaciones

Antes de meterme de lleno en el desarrollo, repasé conceptos clave como:

- La estructura de .NET MAUI y cómo sacarle partido.
- Trabajar con XAML para crear interfaces atractivas.
- Implementar métodos asincrónicos en C#.

Además, investigué las mejores prácticas de diseño y usabilidad para garantizar que la aplicación fuera accesible y funcional para todos los usuarios.

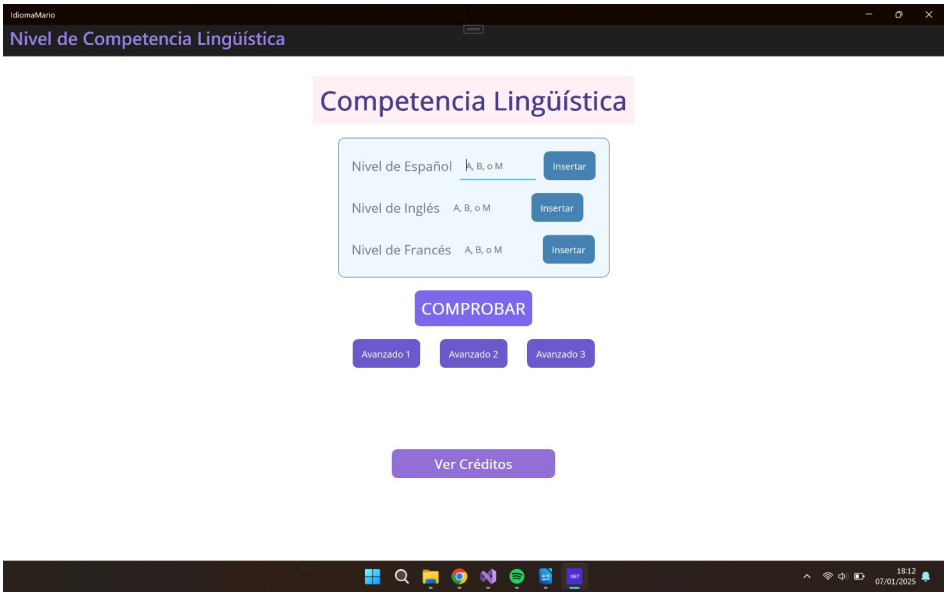
4. Planificación

5. Resultados Detallados

Diseño de la Interfaz

La interfaz de la app combina simplicidad y funcionalidad. Estos son algunos de los elementos clave:

- Colores y diseño: Fondo blanco, texto gris y azul para mantener un contraste atractivo.
- Organización vertical: Uso de `VerticalStackLayout` para una estructura limpia.
- Triggers interactivos: Los campos cambian de color según el nivel seleccionado (A, B o M).



Funcionalidades Clave

- Botones dinámicos: Los botones "Insertar" y "Comprobar" permiten gestionar los niveles fácilmente.
- Validaciones en tiempo real: Los triggers aseguran que los valores ingresados sean correctos.
- Opciones avanzadas: Botones deshabilitados que se activan solo con niveles avanzados.

6. Código Destacado

Ejemplo de Código XAML

```xaml
<!-- Título de la aplicación -->
<Label Text="Competencia Lingüística"
       FontSize="40"
       FontAttributes="Bold"
       HorizontalOptions="Center"
       TextColor="DarkSlateBlue"
       BackgroundColor="LavenderBlush"
       Padding="10"/>
```

Ejemplo de Código C#

```csharp
public async void OnInsertarEspanolClicked(object sender, EventArgs e)
{
    if (string.IsNullOrWhiteSpace(EntryEspanol.Text))
    {
        string nivel = await DisplayPromptAsync("Nivel de Español", "¿Cuál es tu nivel?", "Aceptar", "Cancelar", "A, B o M");

        if (!string.IsNullOrWhiteSpace(nivel))
        {
            EntryEspanol.Text = nivel.ToUpper();
        }
    }
}
```

7. Retos y Soluciones

Tuve problemas con la visibilidad de algunos métodos en C#, pero los resolví haciéndolos públicos.
Esto garantizó que la app funcionara sin errores. También aprendí a optimizar el uso de triggers en XAML para mejorar la experiencia de usuario, aparte error en lo de los colores ya que no realiza del todo bien su funcion.

Ninguna sugerencia.

8. Conclusión

Este proyecto fue un gran aprendizaje. Pude aplicar conceptos avanzados de .NET MAUI y crear algo funcional y atractivo. Estoy satisfecho con el resultado y emocionado por mejorar más en el futuro.

9. Fuentes

- Documentación de .NET MAUI: https://docs.microsoft.com/en-us/dotnet/maui/
- Guía de XAML: https://docs.microsoft.com/en-us/xamarin/xamarin-forms/xaml/
- Métodos asincrónicos en C#:

https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/concepts/async/