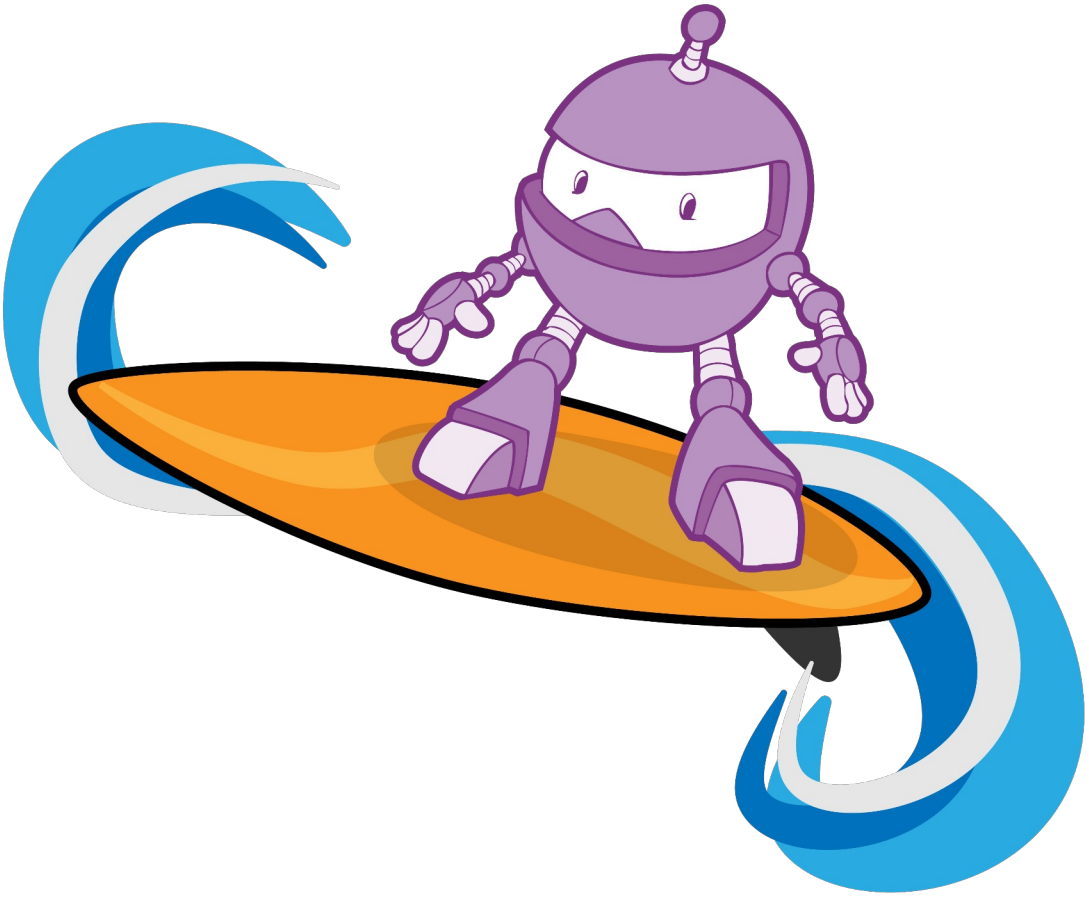
Gestión de Niveles de Idioma: Desarrollo con .NET MAUI



MARIO FERNÁNDEZ

DIN

2ºDAM

Indice

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.

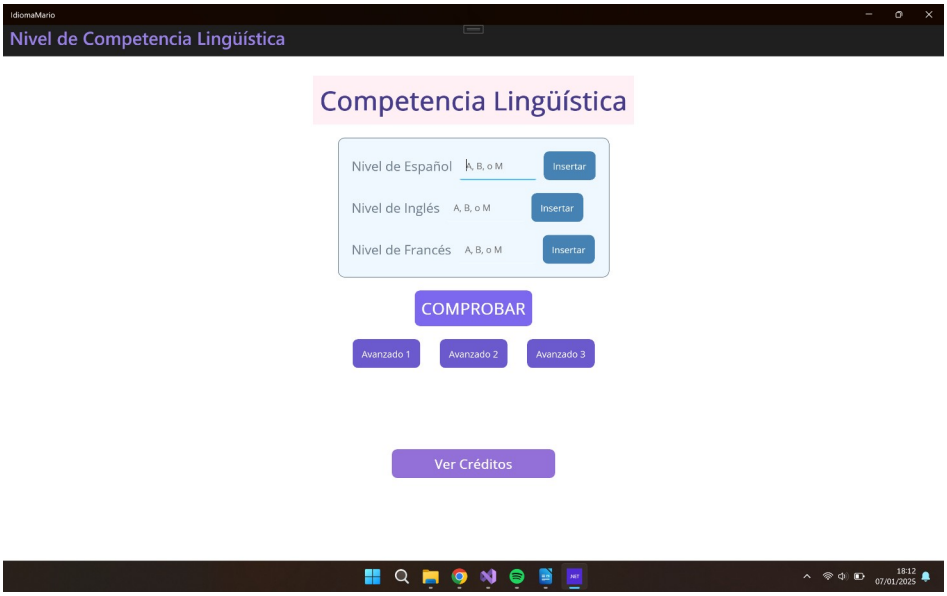
4. Planificación

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

1. **Interfaz de usuario:**
    - Organiza los idiomas (valenciano, inglés y francés) en un Grid con tres columnas dentro de un VerticalStackLayout. Los botones avanzados están dispuestos horizontalmente en un HorizontalStackLayout.

2. **Cambios visuales con Triggers:**
    - Las casillas (Entry) cambian de color según el nivel ingresado: verde para A, rojo para B, y blanco con fuente naranja para M.

3. **Función de botones de idiomas:**
    - **Valenciano:** Muestra un cuadro para escribir A, B, o M y lo inserta en el Entry.
    - **Inglés:** Permite seleccionar el nivel (A, B, M) de una lista y lo inserta en el Entry.
    - **Francés:** Igual que inglés, pero incluye un botón adicional para borrar el valor del Entry.

4. **Botón "Comprobar":**
    - Calcula cuántos idiomas tienen nivel avanzado (A). Activa los botones de "Avanzado 1", "Avanzado 2" o "Avanzado 3" según el número de idiomas con nivel alto.
    - Muestra un mensaje indicando el número de idiomas avanzados o limpia el mensaje si se cancela.

5. **Botón "Ver Créditos":**
    - Muestra un cuadro de diálogo con tus iniciales.

6. **Colores:**

- **Error en los cambios de color de las casillas (Entry):**
  Los colores de fondo y fuente de los Entry (verde para A, rojo para B, y blanco con fuente naranja para M) dependen de Property Triggers. Si se introduce un valor distinto de A, B, o M, las casillas no cambiarán de color y podrían quedar con el estilo predeterminado. Esto ocurre porque los Property Triggers no se activan si el valor no coincide con las condiciones definidas.

Ejemplo del XAML

Este `Label` muestra el texto **"Competencia Lingüística"** con un tamaño de fuente de 40, en negrita. Está centrado horizontalmente, con un color de texto **DarkSlateBlue**, un fondo **LavenderBlush** y un relleno de 10 unidades para separarlo del borde.

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

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/

- IA ChatGPT para informarme de algunos aspectos que no sabía que hacían.