



Voz del Este

**Obligatorio
Taller de
Tecnología**

Franco Nieto, Rodrigo Ciosek

Profesores: Martin Álvarez

2025

Introducción

Se solicita el desarrollo de una aplicación móvil multiplataforma (Android/IOS/Windows) utilizando .NET MAUI para una emisora de radio FM ubicada en Punta del Este. Esta aplicación estará orientada a ofrecer múltiples servicios a los oyentes, desde noticias, clima, cotizaciones, cine, hasta interacción con patrocinadores. La solución deberá estar orientada a la experiencia del usuario, con una estética amigable y preferencia por los colores azules.

Índice

Contenido

Presentación del Cliente	3
Requerimientos Funcionales.....	3
Alcance del Sistema.....	5
Herramientas utilizadas.....	7
Arquitectura del Sistema	7
Manual de Usuario.....	10
Login	10
Registro	11
Menu de Usuario	12
Página Principal.....	12
Menu de Usuario	13
Clima.....	14
Cotizaciones	14
Películas.....	15
Patrocinadores	16
Conclusión.....	17
Conclusión Grupal	Error! Bookmark not defined.

Presentación del Cliente

Voz del Este FM es una emisora local del departamento de Maldonado. En su proceso de modernización, ha apostado por la digitalización para fortalecer el vínculo con los oyentes y mejorar la eficiencia en su operación.

Requerimientos Funcionales

1. Login de Usuario

- Implementar autenticación por huella dactilar (biometría). ✗
- En caso de que la huella no esté disponible o falle, permitir el login mediante usuario y contraseña. ✓
- Guardar sesión de usuario autenticado. ✓

2. Registro de Usuario

- Los usuarios podrán registrarse desde la app ingresando:
 - Usuario, Contraseña, Nombre completo, Dirección, Teléfono, Email, Foto de perfil. ✓
 - Los datos deben almacenarse en una base de datos local SQLite. ✓
 - Validar datos obligatorios y formato (por ejemplo, email y teléfono). ✓

3. Servicio de Clima

- Consumir la API de OpenWeatherMap para:
 - Mostrar el clima actual en Punta del Este. ✓
 - Mostrar el pronóstico extendido de los próximos 5 días. ✓
 - Debe incluir iconos de clima, temperaturas y descripción textual (ej: "llovizna ligera"). ✓

4. Servicio de Cotizaciones

- Usar la API de CurrencyLayer (versión gratuita) para mostrar:
 - Cotización del euro, dólar y real brasileño frente al peso uruguayo. ✓
 - Actualización automática diaria. ✓

5. Servicio de Noticias

- Integrar la API de NewsData.io para mostrar:
 - Noticias actuales sobre Uruguay. ✓
 - Incluir Título, imagen, resumen, enlace al artículo original. ✓
 - Permitir filtrar por categorías o palabras clave. ✓

6. Servicio de Cine

- Usar la API gratuita de TheMovieDB para:
 - Mostrar próximos estrenos de cine. ✓
 - Incluir imagen del poster, título, sinopsis, fecha de estreno. ✓
 - Posibilidad de buscar por género o palabra clave.

7. Gestión de patrocinadores

- **Sección de gestión de patrocinadores con:**
 - Nombre del patrocinador. ✓
 - Logo (imagen) ✓
 - Dirección, seleccionada mediante un mapa interactivo (Geolocalización). ✓
- Sección para usuarios:
 - Ver patrocinadores en mapa mostrando su ubicación ✓
 - La ubicación actual del usuario. ✕
 - Mostrar ruta sugerida para llegar. ✕

8. Preferencias del Usuario

- Permitir a cada usuario configurar que secciones desea ver al inicio de la app. ✓

9. Requisitos de Diseño

- Estética limpia, moderna y profesional. ✓
- Preferencia por colores azules en la paleta de diseño ✓
- El diseño debe ser responsivo y adaptado tanto a smartphones como tablets. ✓
- Interacción intuitiva y amigable para el usuario promedio. ✓

Alcance del Sistema

El sistema fue concebido como una aplicación móvil multiplataforma orientada a brindar servicios informativos y personalizados a usuarios en Uruguay. La aplicación incluyó funcionalidades de autenticación, registro, visualización de datos externos, gestión de contenido local y configuración personalizada, con un diseño moderno y adaptativo.

Autenticación y Registro

La aplicación permitió el inicio de sesión mediante usuario y contraseña, con persistencia de sesión para usuarios autenticados. Los usuarios pudieron registrarse ingresando datos personales como nombre completo, dirección, teléfono, correo electrónico, foto de perfil, usuario y contraseña. Los datos se almacenaron en una base de datos local SQLite, y se validaron tanto los campos obligatorios como el formato de los datos (por ejemplo, correo electrónico y teléfono).

Servicio de Clima

Se integró la API de OpenWeatherMap para mostrar el clima actual en Punta del Este y el pronóstico extendido de los siguientes cinco días. La información incluyó iconos representativos del clima, temperaturas y descripciones textuales.

Servicio de Cotizaciones

Se utilizó la API de CurrencyLayer (versión gratuita) para mostrar la cotización del euro, dólar y real brasileño frente al peso uruguayo. La información se actualizó automáticamente una vez al día.

Servicio de Noticias

La aplicación integró la API de NewsData.io para mostrar noticias actuales sobre Uruguay. Cada noticia incluyó título, imagen, resumen y enlace al artículo original. Se permitió además el filtrado por palabras clave.

Servicio de Cine

Se integró la API gratuita de TheMovieDB para mostrar próximos estrenos de cine. Cada película incluyó imagen del póster, título, sinopsis y fecha de estreno. Se habilitó la búsqueda por género o palabra clave.

Gestión de Patrocinadores

La aplicación contó con una sección de gestión de patrocinadores, donde se registraron nombre, logo e ingreso de dirección mediante un mapa interactivo. Los usuarios pudieron visualizar los patrocinadores en el mapa, incluyendo su ubicación.

Preferencias del Usuario

Cada usuario tuvo la posibilidad de configurar qué secciones deseaba visualizar al inicio de la aplicación, personalizando su experiencia.

Diseño y Experiencia de Usuario

El diseño se mantuvo limpio, moderno y profesional, con preferencia por colores azules en la paleta visual. La interfaz fue responsiva y se adaptó tanto a smartphones como a tablets, con una interacción intuitiva y amigable para el usuario promedio.

Funcionalidades Fuera de Alcance

Las siguientes funcionalidades no fueron incluidas en el alcance del sistema:

- Autenticación biométrica mediante huella dactilar.
- Visualización de la ubicación actual del usuario.
- Cálculo de ruta sugerida hacia patrocinadores.

Herramientas utilizadas

- **Plataforma de desarrollo:** .NET MAUI (Multiplataforma: Android, iOS, Windows)
- **Lenguaje de programación:** C#
- **Base de datos local:** SQLite (mediante SQLite-net-pcl)
- **Patrón de arquitectura:** MVVM con CommunityToolkit.MVVM
- **Inyección de dependencias**
- **Mapas y geolocalización:**
 - **Leaflet.js** para renderizado de mapas interactivos
 - **OpenStreetMap** como proveedor de cartografía base

Arquitectura del Sistema

La arquitectura se estructuró siguiendo el patrón MVVM (Model-View-ViewModel), con separación clara de responsabilidades, inyección de dependencias, persistencia local en SQLite y consumo de APIs externas. El proyecto se organizó en capas funcionales distribuidas en carpetas, cada una con una función específica.

2. Estructura Arquitectónica

2.1 Datos

Contiene la lógica de acceso a datos y persistencia local.

- **Interfaces/**
Define contratos como IClimaRepositorio.cs, que abstraen el acceso a datos climáticos. Esto permite desacoplar la lógica de persistencia del resto de la aplicación.
- **Repositorios/**
Implementa las interfaces, por ejemplo ClimaRepositorio.cs, que gestiona la lectura/escritura en SQLite usando SQLite-net-pcl. Aquí se encapsulan las operaciones CRUD y se aplican validaciones de formato.

2.2 Modelos

Define los DTOs que representan los datos consumidos desde APIs externas o persistidos localmente.

- ClimaForecastDTO.cs: estructura para el pronóstico extendido.
- ClimaMainDTO.cs: datos del clima actual.

Estos modelos se deserializan desde JSON y se adaptan a la UI mediante mapeo en los servicios.

2.3 Orquestadores

Actúan como capa intermedia entre los servicios y la base de datos. Su función principal es coordinar la actualización de datos desde APIs externas hacia el repositorio local, y exponer datos ya persistidos para su visualización.

- `ClimaOrquestador.cs`: gestiona la sincronización entre Clima servicio y Clima repositorio.
- `CotizacionOrquestador.cs`: realiza la misma función que `ClimaOrquestador` pero para cotizaciones.

Esta capa permite abstraerse de la lógica de recuperación de datos externos (servicios) y la comunicación directa con la base de datos (repositorios).

2.4 ViewModels

Contiene la lógica de presentación y comportamiento de las vistas. Cada ViewModel expone propiedades observables (`ObservableCollection`, `ObservableProperty`, `ObservableObject`, etc) y comandos (`ICommand`) que son enlazados desde la UI.

- Ejemplos:
- `ClimaViewModel.cs`: gestiona la visualización la vista Clima.

Los ViewModels son inyectados mediante constructor y asignados al `BindingContext` de cada vista. Se utiliza **CommunityToolkit.MVVM** para simplificar la implementación de bindings (`[ObservableProperty]`).

2.5 Servicios

Encapsulan la lógica de integración con APIs externas.

- `ClimaServicio.cs` y `ServicioClima.cs`: gestionan las llamadas HTTP a OpenWeatherMap, deserializan la respuesta y la transforman en DTOs.
- Servicios similares se implementan para CurrencyLayer, NewsData.io y TheMovieDB.

Estos servicios se inyectan mediante Dependency Injection, desacoplamiento.

2.6 Vistas

Contiene las interfaces gráficas definidas en XAML y sus code-behind.

- `Clima.xaml`: vista que muestra el clima.

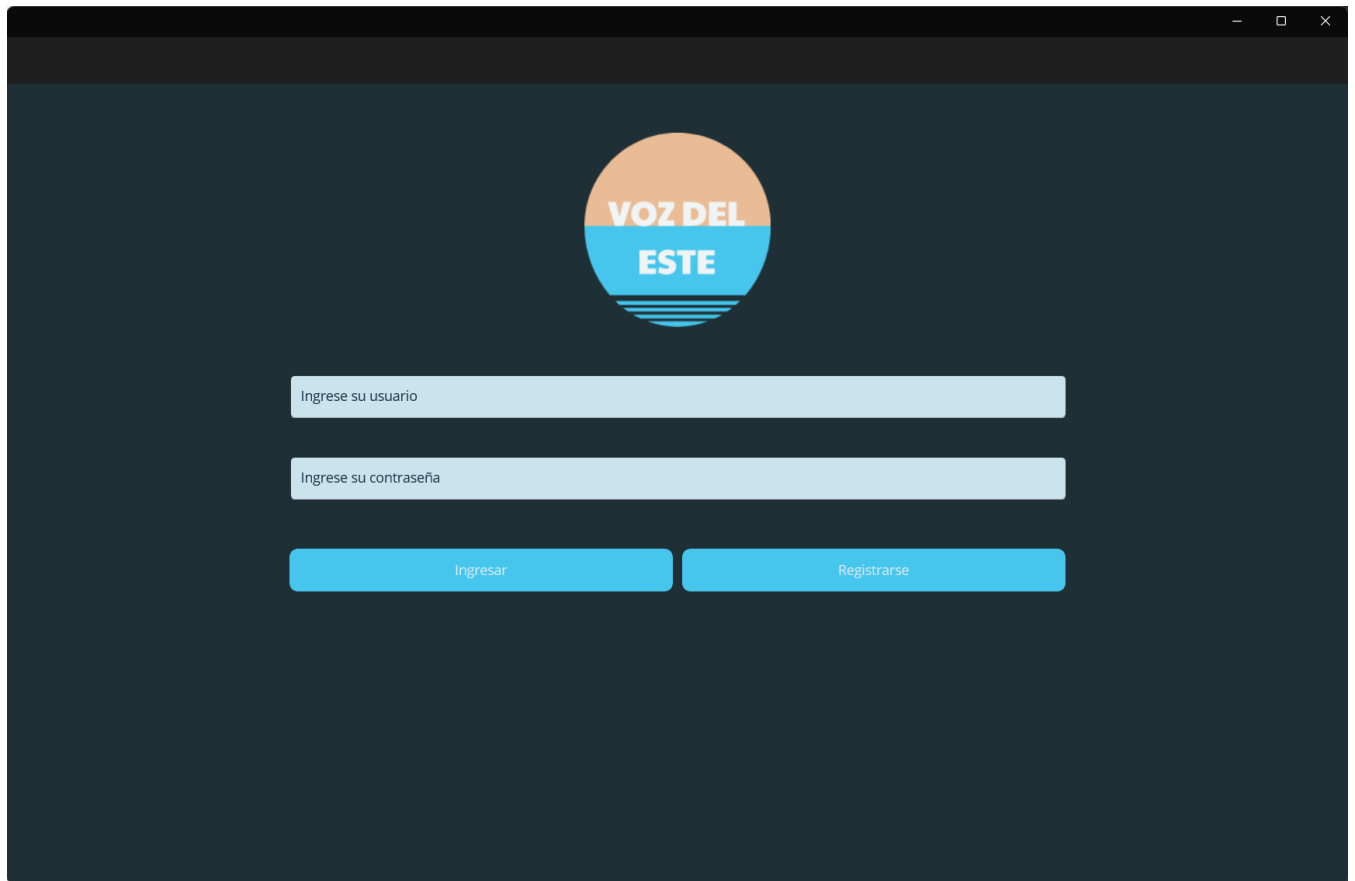
Las vistas están conectadas a sus ViewModels mediante bindings, y utilizan layouts responsivos (`Grid`, `FlexLayout`) y estilos definidos para mantener coherencia visual.

2.7 Mapas y Geolocalización

La visualización de patrocinadores se realiza mediante un componente WebView que integra **Leaflet.js**, utilizando **OpenStreetMap** como proveedor de cartografía base. Las ubicaciones se gestionan desde el ViewModel correspondiente, y se renderizan dinámicamente en el mapa con marcadores personalizados.

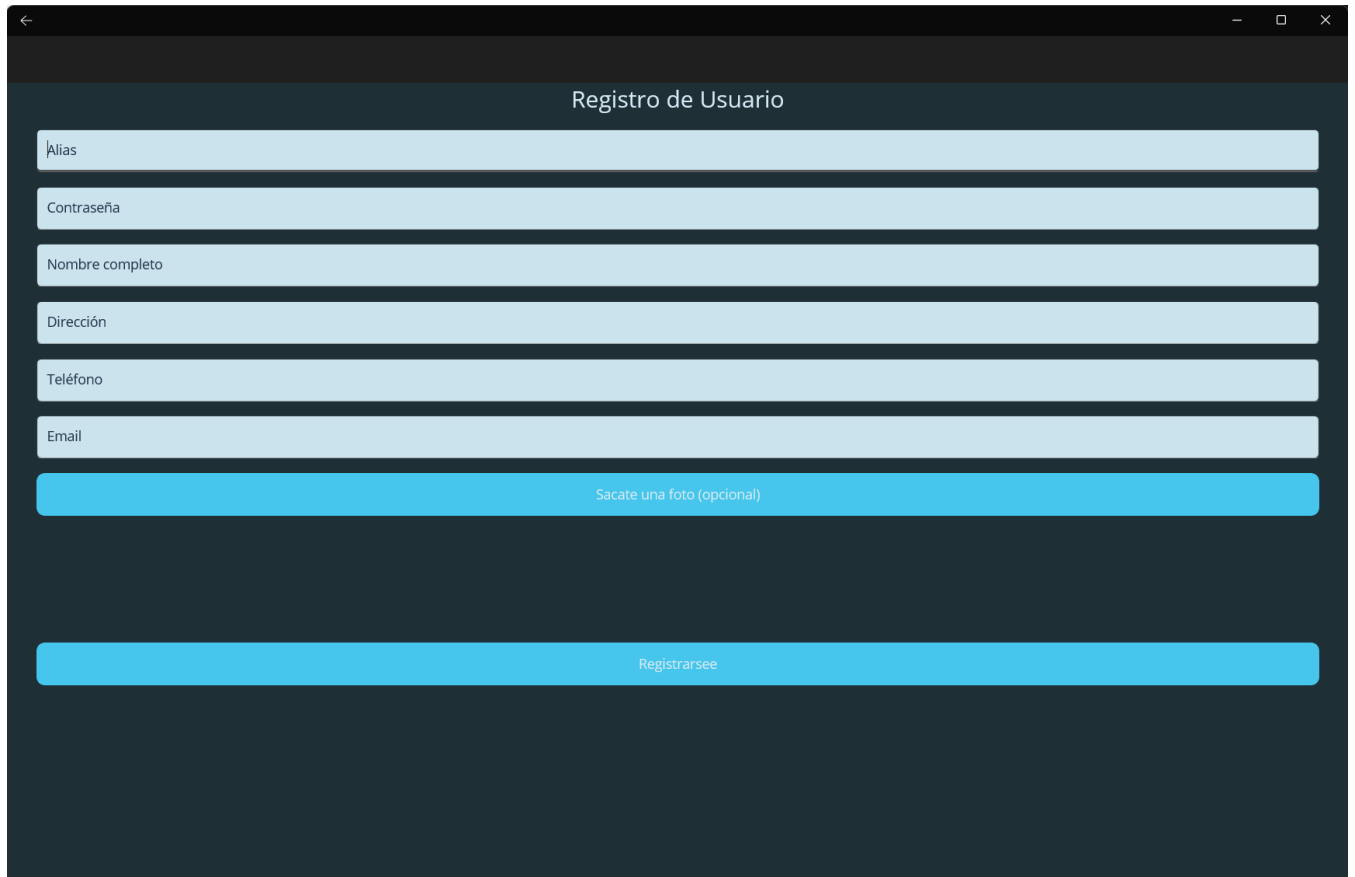
Manual de Usuario

Login

The image shows a login screen for an application named 'VOZ DEL ESTE'. The background is a dark blue-grey. At the top center is a circular logo with an orange top half containing the text 'VOZ DEL' and a blue bottom half containing the text 'ESTE'. Below the logo are two light blue input fields. The first field is labeled 'Ingrese su usuario' and the second is labeled 'Ingrese su contraseña'. Below these fields are two light blue buttons. The left button is labeled 'Ingresar' and the right button is labeled 'Registrarse'. The entire interface is framed by a dark window border with standard OS window controls (minimize, maximize, close) in the top right corner.

- **Ingresar:** valida usuario y contraseña para acceder a la app.
- **Registrarse:** abre el formulario para crear una nueva cuenta.

Registro

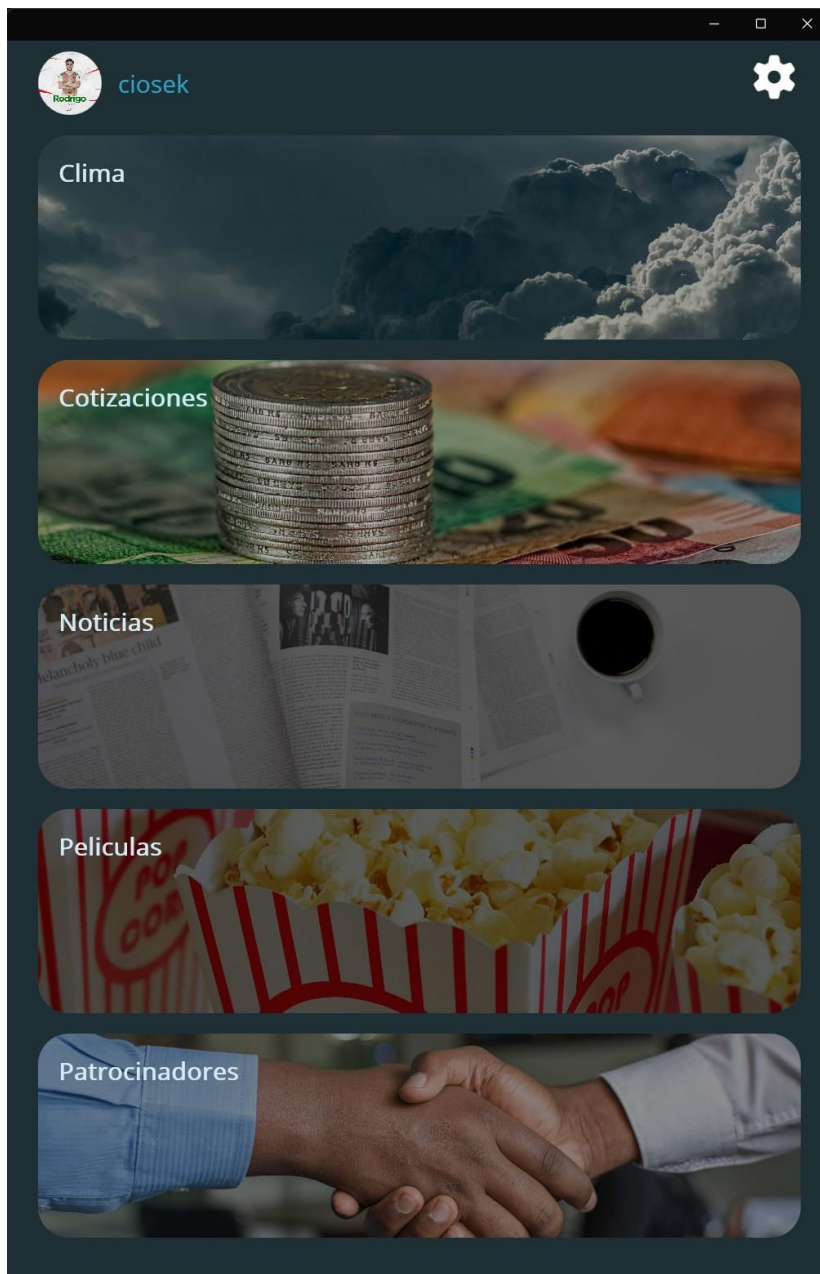


A mobile application registration screen with a dark blue background. At the top, there is a title bar with a back arrow on the left and window controls on the right. Below the title bar, the title 'Registro de Usuario' is centered. The form consists of several light blue input fields stacked vertically, each with a label on the left: 'Alias', 'Contraseña', 'Nombre completo', 'Dirección', 'Teléfono', and 'Email'. Below these fields is a light blue button with the text 'Sacate una foto (opcional)'. At the bottom of the form is a wide, light blue button with the text 'Registrarse'.

- **Suba una foto (opcional):** permite seleccionar y cargar una imagen de perfil desde el dispositivo.
- **Registrarse:** guarda los datos ingresados y crea la cuenta del usuario.

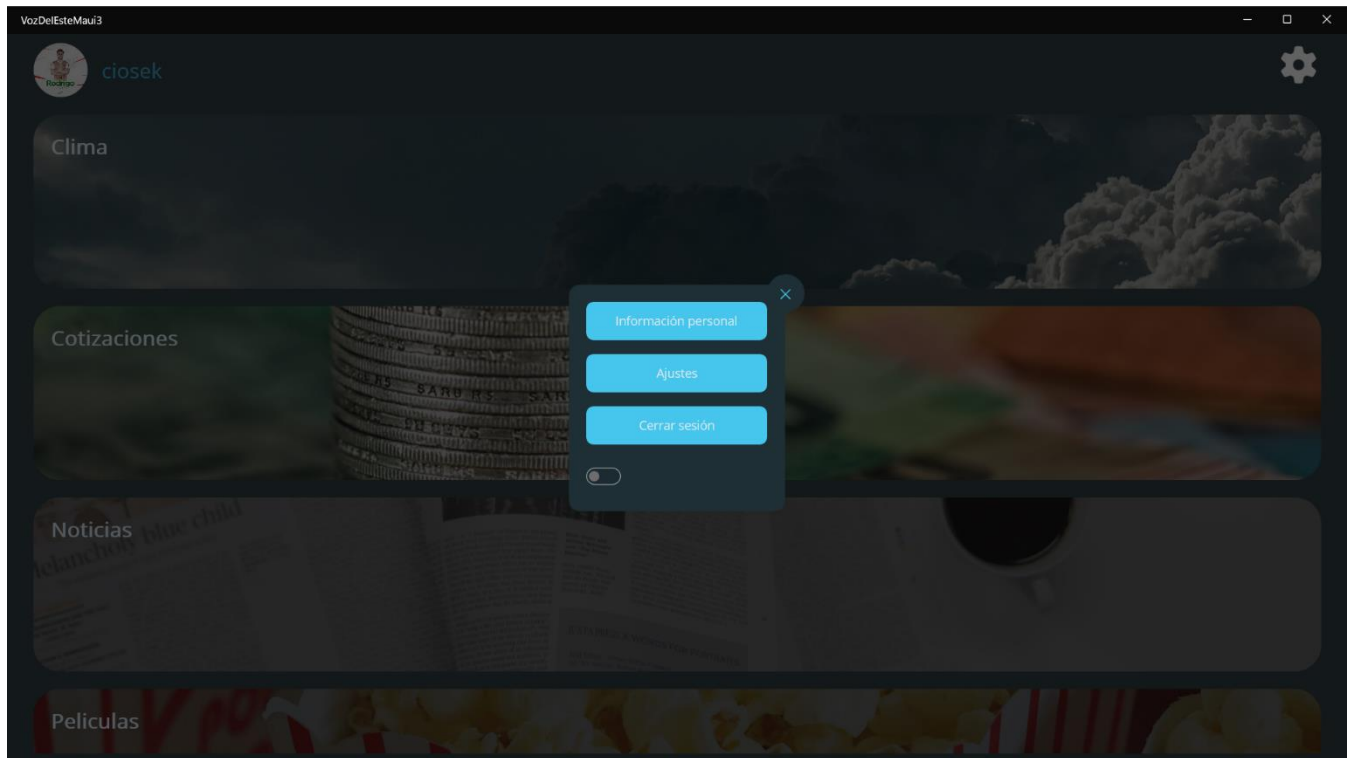
Menu de Usuario

Página Principal



- **Clima:** muestra el estado meteorológico actual.
- **Cotizaciones:** accede a valores de monedas.
- **Noticias:** abre el panel con búsqueda de artículos.
- **Películas:** abre el panel con búsqueda de películas.
- **Ajustes (ícono de engranaje):** permite modificar datos del perfil, cambiar contraseña y configurar preferencias del usuario

Menu de Usuario

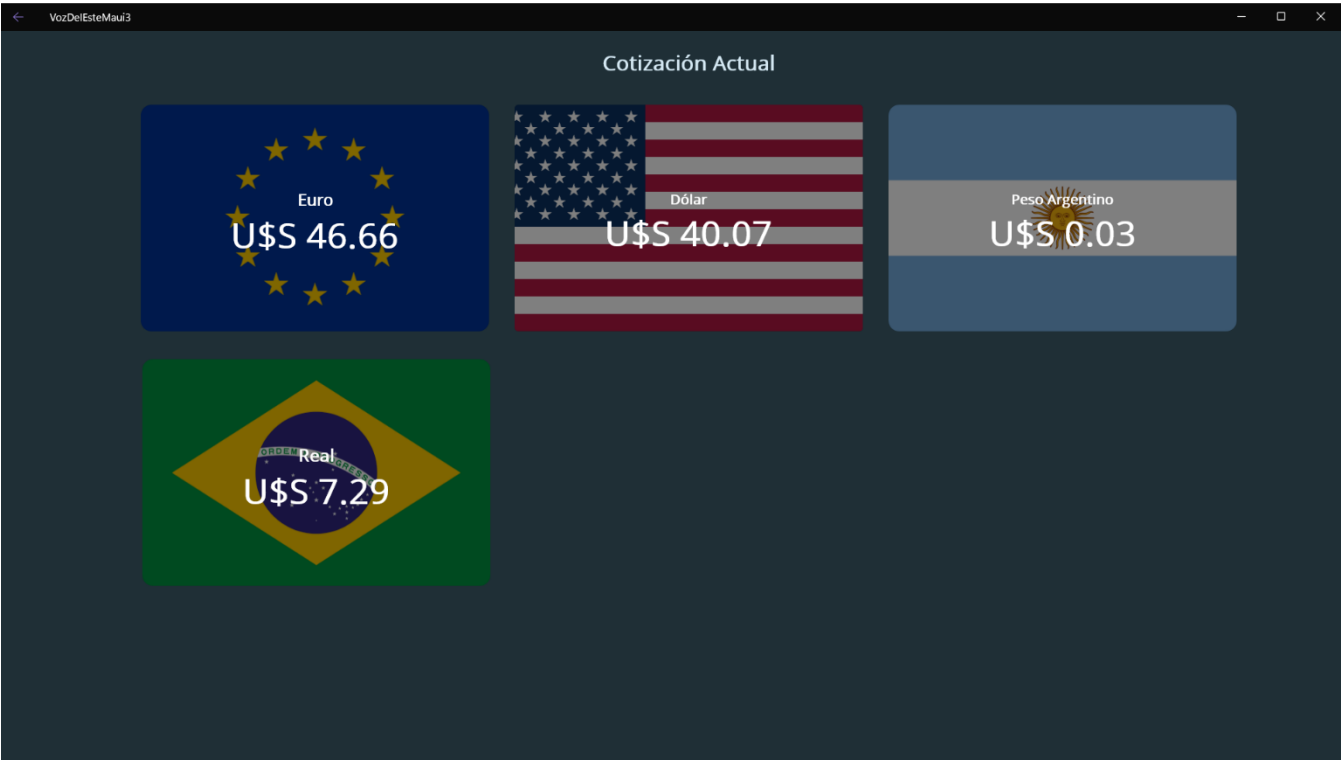


- **Información personal:** permite ver los datos del perfil.
- **Ajustes:** accede a configuraciones de visualización del usuario.
- **Cerrar sesión:** finaliza la sesión actual y vuelve a la pantalla de login.
- **Interruptor de tema:** cambia entre modo claro y modo oscuro de la interfaz.

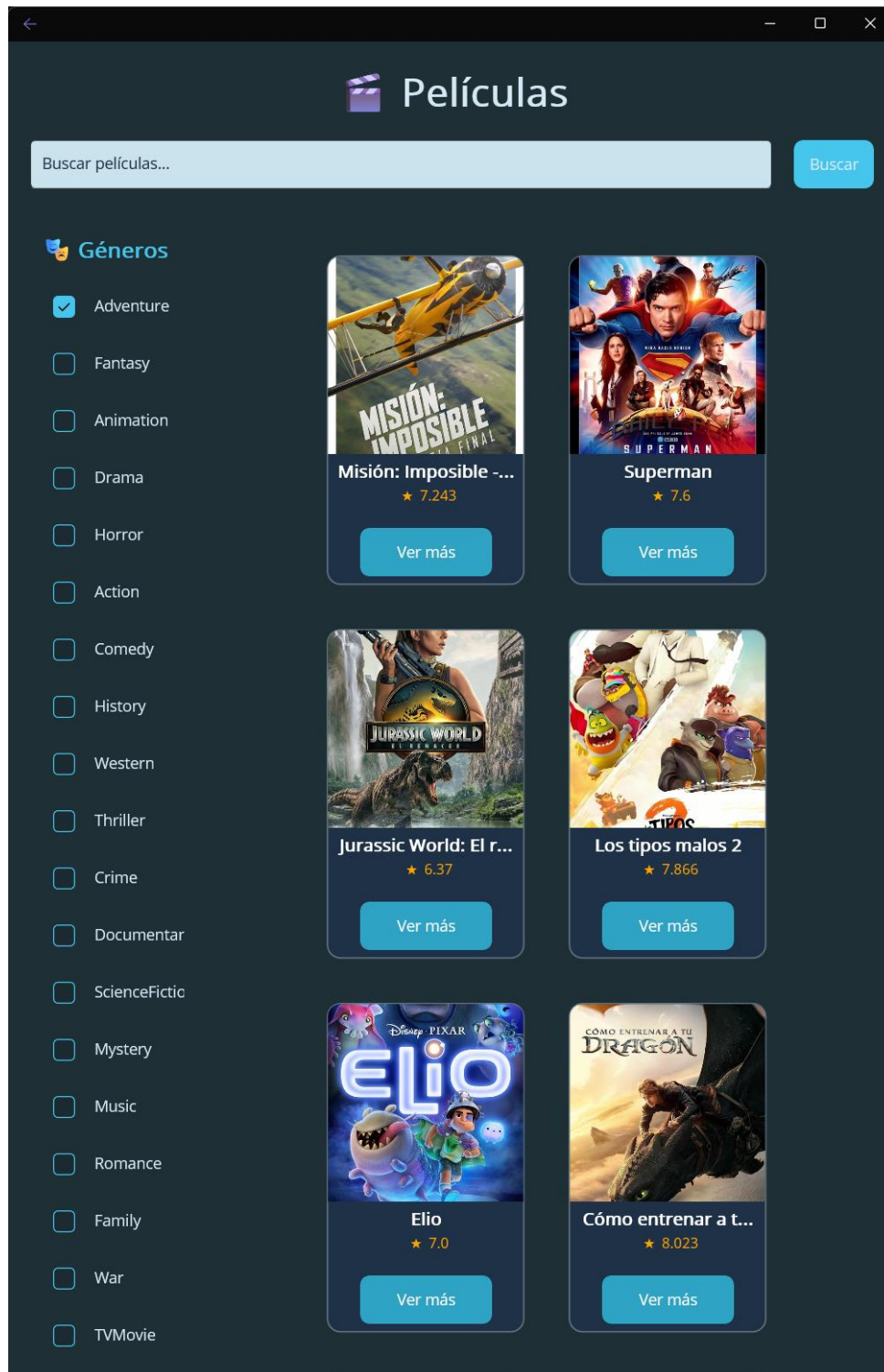
Clima



Cotizaciones

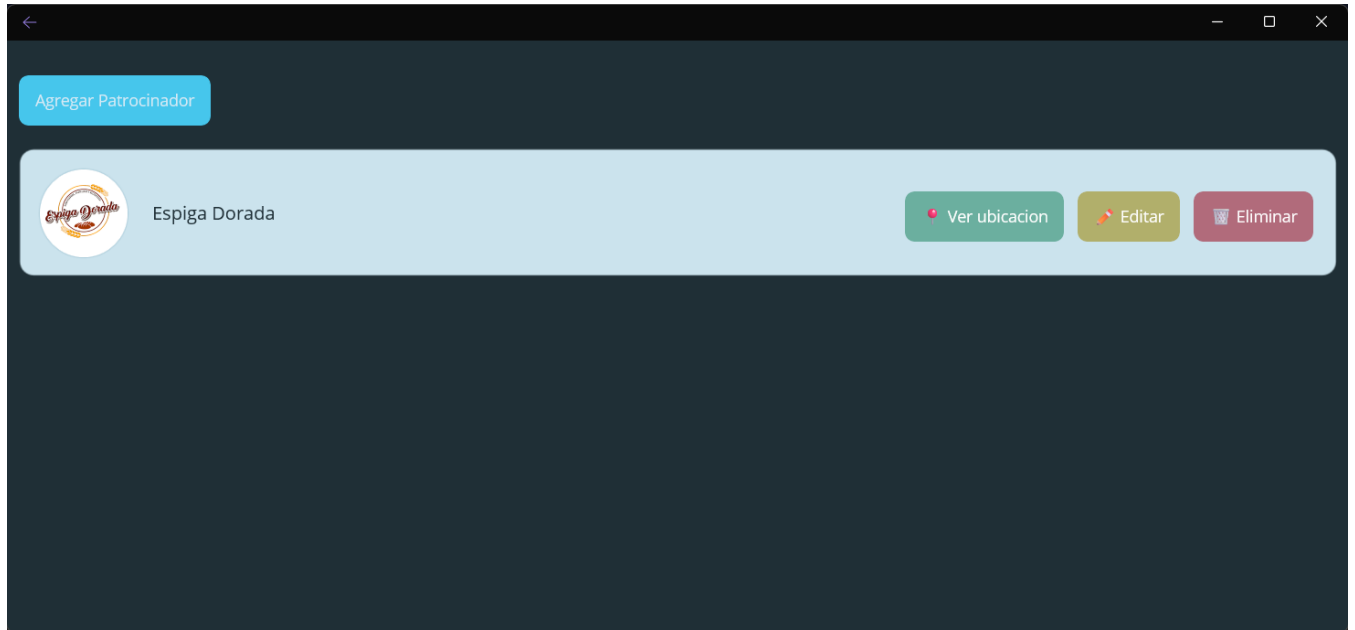


Películas



- Barra de búsqueda: permite buscar películas por título o palabra clave.
- Menú de géneros: lista vertical con categorías como Aventura, Drama, Comedia, etc.
- Grid de películas: muestra tarjetas con póster, título y puntuación.
- Botones "Ver más": permiten acceder a los detalles ampliados de cada película.

Patrocinadores



- Botón “Agregar Patrocinador”: permite añadir nuevos patrocinadores al sistema.
- Tarjeta de patrocinador: muestra logo y nombre del patrocinador (ej. Espiga Dorada).
- Botón “Ver ubicación”: muestra la localización del patrocinador.
- Botón “Editar”: permite modificar los datos del patrocinador.
- Botón “Eliminar”: elimina el patrocinador del listado.

Conclusión

Este trabajo nos permitió aprender sobre .NET MAUI y aplicarlo en el desarrollo de una aplicación multiplataforma con funciones útiles. Nos organizamos para repartir tareas, resolver problemas y mantener el código ordenado.

Cada parte de la app fue hecha con integración de APIs externas, cuidando también la documentación y el diseño para que el resultado sea funcional y presentable.

El proyecto nos ayudó a mejorar en planificación, trabajo en equipo y buenas prácticas de desarrollo. Quedamos conformes con lo que logramos.