

Universidad Claeh  
Centro CTC



**Analista Programador  
Taller de Tecnologías**

**Proyecto  
Trabajo Obligatorio  
La Voz Del Este**

Matias Delgado  
Danlee Garcia

**2025**

## Resumen:

Se realizó un proyecto basado en el desarrollo de una aplicación móvil multiplataforma utilizando .NET MAUI, cuyo propósito fue brindar a los usuarios una experiencia moderna e intuitiva para mantenerse informados y conectados con los servicios de una emisora de radio de Punta del Este. La aplicación fue diseñada con un estilo visual azul que transmite frescura y profesionalismo, y se implementaron distintas funcionalidades que buscan cubrir las necesidades de los oyentes. Para el inicio de sesión se incorporó autenticación biométrica mediante huella digital, complementada con la posibilidad de acceder con usuario y contraseña, garantizando así seguridad y practicidad. Los datos de los usuarios se almacenaron en una base local SQLite, permitiendo además un registro completo con información personal y fotografía de perfil.

Dentro de los módulos desarrollados se integró un servicio de clima utilizando la API de OpenWeatherMap, donde se presenta tanto el estado actual como el pronóstico de los próximos cinco días en Punta del Este. También se incluyó una sección de cotizaciones a través de la API de Currency Layer, con valores actualizados de las principales monedas extranjeras frente al peso uruguayo. Para mantener al usuario informado se trabajó con la API de New Data.io, mostrando noticias relevantes de Uruguay acompañadas de título, descripción, imagen y enlace a la fuente, además de un sistema de filtros para personalizar la búsqueda. En el área de entretenimiento se implementó un módulo de cine conectado a la API TheMovieDB, el cual muestra próximos estrenos con su cartelera, sinopsis y fecha de lanzamiento.

Uno de los apartados más importantes fue el de patrocinadores, que se gestionan desde una base de datos SQLite donde se almacenan sus datos junto con su logo e información de contacto. Cada patrocinador puede visualizarse en un mapa interactivo junto con la ubicación del usuario, lo que permite mostrar incluso la ruta sugerida para llegar al lugar. Además, se incorporó una sección de preferencias para que cada usuario pueda personalizar qué módulos desea ver al inicio, adaptando la aplicación a sus propios intereses.

El proyecto fue entregado con código fuente completo, un APK instalable en Android y un ejecutable para Windows, junto con la documentación técnica que describe las funcionalidades, el uso de APIs, las pruebas realizadas y las capturas de pantalla. En general se logró una aplicación sólida, funcional y moderna, que integra distintos servicios en un mismo entorno digital, demostrando la utilidad de .NET MAUI para crear soluciones multiplataforma con un diseño atractivo y orientado al usuario.

<b>Resumen:</b>	<b>1</b>
<b>1. Introducción:</b>	<b>3</b>
<b>2. Tecnologías:</b>	<b>4</b>
2.1 Visual Studio	4
2.2 .NET	5
2.3 MAUI (Multi-platform App UI)	5
2.4 SQLite	5
2.5 APIs	6
2.6 C#	6
2.7 XAML	6
2.8 MVVM (Model-View-ViewModel)	7
2.9 Control de versiones (Git/GitHub)	7
<b>3. Manual de Usuario – Aplicación Radio FM Punta del Este</b>	<b>7</b>
3.1 Introducción	7
3.2 Requisitos	8
3.3 Registro de Usuario	9
3.4 Inicio de Sesión	10
3.5 Módulos	11
3.5.1 Clima	11
3.5.2 Cotizaciones	13
3.5.3 Noticias	14
3.5.4 Cine	15
3.5.5 Patrocinadores	17
3.5.6 Clientes	19
Acciones sobre Clientes	20
Funcionalidad Adicional	20
3.6 Preferencias del Usuario	22
<b>4. Diagrama de clase</b>	<b>24</b>
<b>5. Conclusiones</b>	<b>25</b>
5.1 Conclusion General:	25
5.2 Conclusión Individual	26
5.2.1 Danlee Garcia:	26
5.2.2 Matias Delgado:	26
<b>6. Bibliografía</b>	<b>26</b>

## 1. Introducción:

El presente proyecto tuvo como finalidad el desarrollo de una aplicación móvil multiplataforma orientada a la emisora de radio “Voz del Este”, utilizando como base tecnológica .NET MAUI. La propuesta surgió de la necesidad de contar con una herramienta moderna, accesible y adaptable tanto a dispositivos Android como a equipos con sistema operativo Windows, que permitiera centralizar distintos servicios de información y entretenimiento en un mismo entorno digital. La aplicación fue concebida con un diseño visual moderno en tonos azules, transmitiendo una identidad clara, profesional y a la vez amigable para los usuarios.

En su construcción se integraron múltiples funcionalidades apoyadas en APIs externas y bases de datos locales, lo que permitió enriquecer la experiencia del usuario con datos actualizados en tiempo real, al mismo tiempo que se garantiza la persistencia de información de manera segura en el dispositivo. El proyecto no solo buscó satisfacer necesidades prácticas como la consulta del clima, las cotizaciones o las noticias, sino también ofrecer un espacio interactivo con secciones de entretenimiento, patrocinadores y preferencias personalizables.

A través de esta iniciativa se puso en práctica el uso de tecnologías actuales para el desarrollo de software multiplataforma, con un enfoque que combina diseño moderno, usabilidad, seguridad y escalabilidad, demostrando las capacidades de .NET MAUI para implementar soluciones completas que respondan a las demandas actuales de los usuarios.

## 2. Tecnologías

### 2.1 Visual Studio

Visual Studio es el IDE que centraliza todo el desarrollo de la aplicación. No solo permite escribir código, sino que ofrece herramientas avanzadas para depuración, pruebas unitarias, análisis de rendimiento y gestión de dependencias. Su integración con Git y GitHub facilita el control de versiones y la colaboración en equipo.

En el proyecto, Visual Studio permitió trabajar de manera modular: separar vistas, modelos y servicios, compilar para distintas plataformas con MAUI y ejecutar la aplicación en simuladores de Android, iOS y Windows sin cambiar la base de código. Uno de los retos fue

configurar los emuladores correctamente, ya que la aplicación debe funcionar igual en móvil y escritorio, pero Visual Studio ofrece soluciones integradas que simplifican este proceso.

## 2.2 .NET

.NET actúa como la columna vertebral de la aplicación. Este framework proporciona la base para la ejecución de código, manejo de memoria, seguridad y compatibilidad multiplataforma. Gracias a .NET, se pudo crear un proyecto con una arquitectura consistente, donde los modelos de datos, la lógica de negocio y los servicios externos se comunican sin conflictos.

El proyecto aprovechó la biblioteca de clases de .NET para manejar excepciones, trabajar con colecciones, interactuar con SQLite y consumir APIs externas. Esto permitió que la aplicación tenga funcionalidades robustas como gestión de usuarios, preferencias dinámicas y actualizaciones de información en tiempo real. Además, la compatibilidad multiplataforma de .NET con MAUI fue crucial para mantener un único proyecto en lugar de desarrollar versiones separadas para cada sistema operativo.

## 2.3 MAUI (Multi-platform App UI)

MAUI es un framework que permite crear interfaces gráficas nativas en múltiples plataformas con un solo proyecto de .NET. La ventaja principal es que elimina la necesidad de desarrollar por separado para Android, iOS, Windows o macOS.

En el proyecto, MAUI permitió que los botones de la interfaz sean dinámicos según las preferencias del usuario. Por ejemplo, si un usuario decide ocultar el botón de clima, MAUI permite actualizar la interfaz de manera inmediata sin reiniciar la aplicación. Además, su compatibilidad con XAML facilita la creación de layouts adaptativos, animaciones y efectos visuales modernos. Un desafío fue asegurar que los elementos visuales se vean consistentes en todas las plataformas, ya que cada sistema operativo interpreta ligeramente los estilos.

## 2.4 SQLite

SQLite es una base de datos ligera y embebida. A diferencia de las bases de datos tradicionales, no requiere servidor y almacena los datos localmente en un archivo. Esto la hace ideal para aplicaciones móviles o de escritorio donde la conectividad a internet no siempre es confiable.

En el proyecto, SQLite almacena información crítica del usuario: perfiles, preferencias de visualización, historial de actividades y configuraciones de la aplicación. Esto asegura que, incluso sin conexión, el usuario pueda interactuar plenamente con la app. La integración con C# se realizó mediante librerías de acceso a datos, creando métodos CRUD (Crear, Leer, Actualizar, Eliminar) que manejan los datos de manera segura y eficiente.

## 2.5 APIs

Las APIs permiten a la aplicación conectarse con servicios externos para obtener datos en tiempo real. En este proyecto, se usaron APIs para servicios como:

OpenWeatherMap: Para mostrar el clima actual y el pronóstico extendido de Punta del Este.

CurrencyLayer: Para mostrar la cotización de dólar, euro y real brasileño frente al peso uruguayo.

NewsData.io: Para mostrar noticias de Uruguay con imágenes, resúmenes y enlaces.

TheMovieDB: Para mostrar próximos estrenos de cine, con pósters, sinopsis y fechas.

Estas APIs permiten ofrecer datos en tiempo real, enriqueciendo la experiencia del usuario.

Esto aporta valor al usuario, ya que recibe información actualizada sin que la aplicación tenga que manejar grandes volúmenes de datos internamente. Un desafío fue manejar la latencia y errores de conexión, lo que se soluciona implementando validaciones y mensajes claros al usuario.

## 2.6 C#

C# es el lenguaje principal del proyecto. Su orientación a objetos y fuerte tipado facilita la creación de clases, interfaces y métodos robustos.

En el proyecto, C# se utilizó para:

Controlar la lógica de negocio (gestión de usuarios, preferencias, botones dinámicos).

Acceder a SQLite mediante clases de servicio.

Consumir APIs externas y procesar la información obtenida.

Implementar MVVM, un patrón que separa la vista de la lógica y los datos, haciendo el proyecto más mantenible y escalable. Un reto frecuente con C# fue asegurarse de que la lógica de negocio funcione de manera uniforme en todas las plataformas soportadas por MAUI.

## 2.7 XAML

XAML se usó para diseñar la interfaz de usuario de forma declarativa. Esto permite separar la lógica de negocio del diseño visual.

En el proyecto, XAML se utilizó para crear:

Botones dinámicos visibles según preferencias del usuario.

Listas, formularios y tarjetas animadas.

Layouts responsive que se adaptan a diferentes tamaños de pantalla.

XAML junto con MAUI permitió aplicar animaciones, efectos de transparencia y estilos modernos (como glass morphism), garantizando una experiencia visual atractiva y profesional.

## 2.8 MVVM (Model-View-ViewModel)

MVVM es un patrón de arquitectura que facilita la separación entre interfaz, lógica y datos.

Model: Define los datos y estructura de la aplicación.

ViewModel: Contiene la lógica de negocio, actúa como intermediario entre el modelo y la vista.

View: La interfaz creada con XAML.

En el proyecto, MVVM permitió:

Actualizar automáticamente la interfaz cuando cambian los datos del usuario.

Mantener el código limpio y fácil de mantener.

Facilitar pruebas unitarias de la lógica de negocio sin depender de la interfaz.

## 2.9 Control de versiones (Git/GitHub)

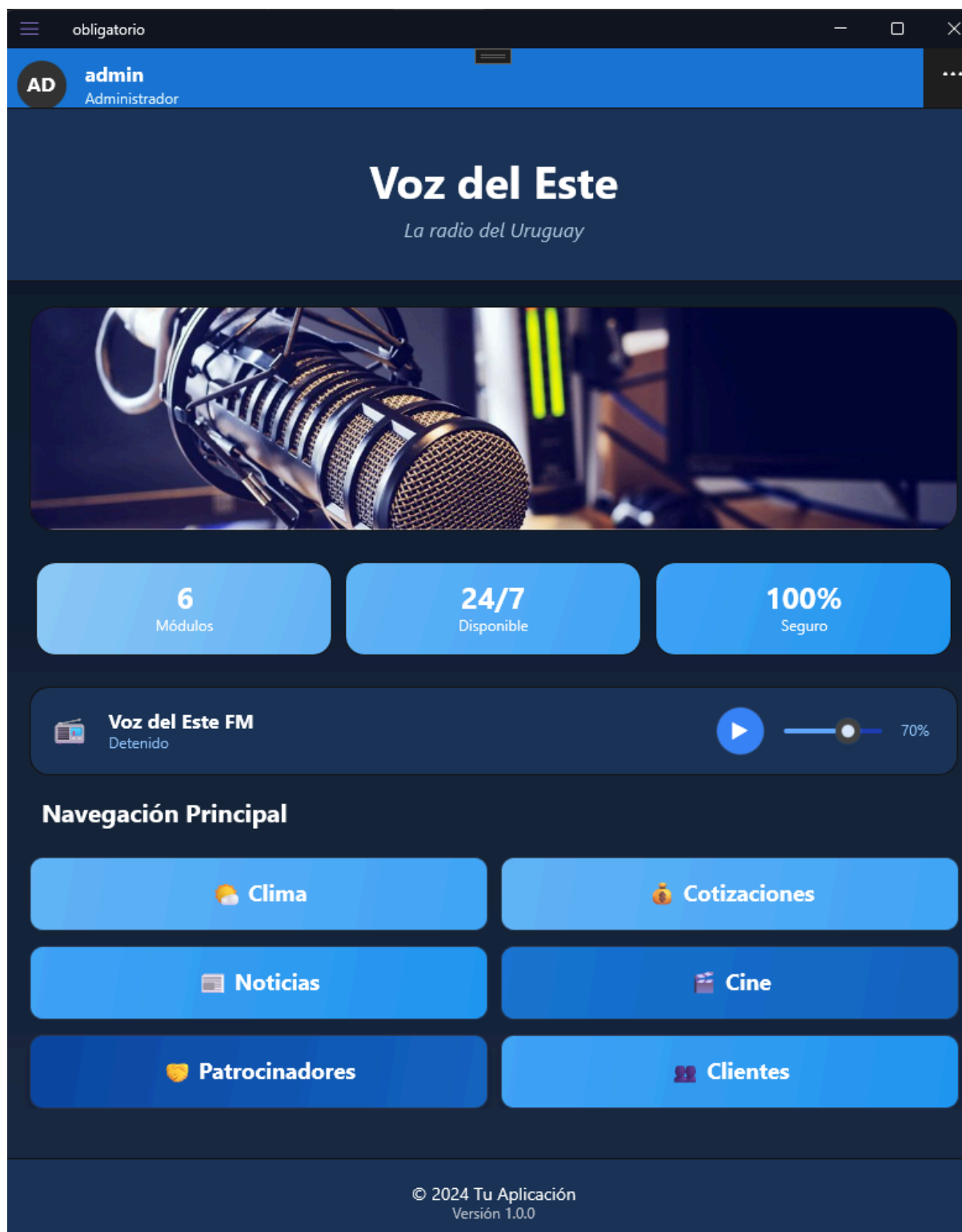
Git permite llevar un historial completo del proyecto, creando versiones y ramas para experimentar sin afectar la versión estable. GitHub sirvió como repositorio remoto y plataforma de colaboración, asegurando que los cambios de cada desarrollador se integren correctamente. Esto fue esencial para manejar el desarrollo de nuevas funcionalidades, corrección de errores y despliegues seguros.

# 3. Manual de Usuario – Aplicación Radio FM Punta del Este

## 3.1 Introducción

Esta aplicación permite acceder a los servicios de la emisora de radio FM de Punta del Este desde tu dispositivo móvil o computadora. Incluye módulos de clima, noticias, cotizaciones, cine, patrocinadores y personalización de la experiencia mediante preferencias de usuario. La interfaz está diseñada para ser limpia, intuitiva y responsiva, predominando los colores azules de la identidad de la emisora.





## 3.2 Requisitos

- Dispositivo Android, iOS o Windows.
- Conexión a Internet para servicios en línea.

- Espacio suficiente para la instalación de la app.

### 3.3 Registro de Usuario

1. Abre la aplicación y selecciona “**Registrarse**”.
2. Completa los campos obligatorios:
  - Usuario
  - Contraseña
  - Nombre completo
  - Dirección
  - Teléfono
  - Email
  - Foto de perfil (tomada con cámara o seleccionada de galería)
3. Toca “**Registrarse**” para guardar los datos en la base de datos local SQLite.

The screenshot shows a mobile application interface for user registration. At the top, a dark navigation bar contains a back arrow, a hamburger menu icon, the text 'obligatorio', and standard mobile window controls (minimize, maximize, close). The main content area has a light purple background. The title 'Registro de Usuario' is centered in a bold blue font, with the subtitle 'Completa los campos para crear tu cuenta' below it in a smaller, gray font. Five white input fields with rounded corners are stacked vertically, labeled 'Usuario', 'Contraseña', 'Dirección', 'Teléfono', and 'Email'. Below these fields is a large gray square placeholder for a profile picture. Under the placeholder are two blue buttons: 'Tomar foto' and 'Seleccionar foto'. At the bottom is a large blue button labeled 'Registrarse'.

obligatorio

## Registro de Usuario

Completa los campos para crear tu cuenta

Usuario

Contraseña

Dirección

Teléfono

Email

Tomar foto

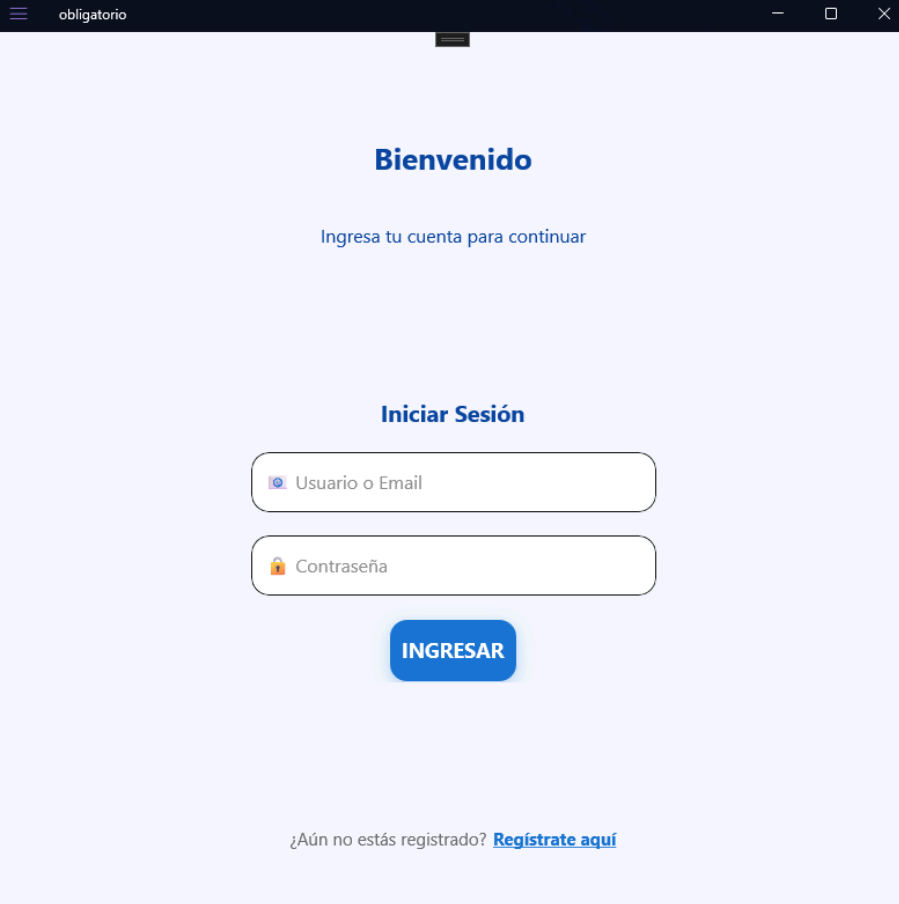
Seleccionar foto

Registrarse

### 3.4 Inicio de Sesión

1. Selecciona “**Iniciar sesión**” en la pantalla principal.
2. Ingresa tu **usuario y contraseña**.

3. Si tu dispositivo lo permite, puedes usar **huella dactilar**.
4. Pulsa **“Ingresar”** para acceder a la aplicación.



The screenshot shows a web browser window with the title 'obligatorio'. The page has a light purple background. At the top, it says 'Bienvenido' in bold blue text. Below that, it says 'Ingresa tu cuenta para continuar' in a smaller blue font. In the center, there's a section titled 'Iniciar Sesión' in bold blue text. Below this title are two input fields: the first is labeled 'Usuario o Email' with a user icon, and the second is labeled 'Contraseña' with a lock icon. Below these fields is a blue button with the text 'INGRESAR' in white. At the bottom, there's a link that says '¿Aún no estás registrado? [Regístrate aquí](#)'.

## 3.5 Módulos

### 3.5.1 Clima

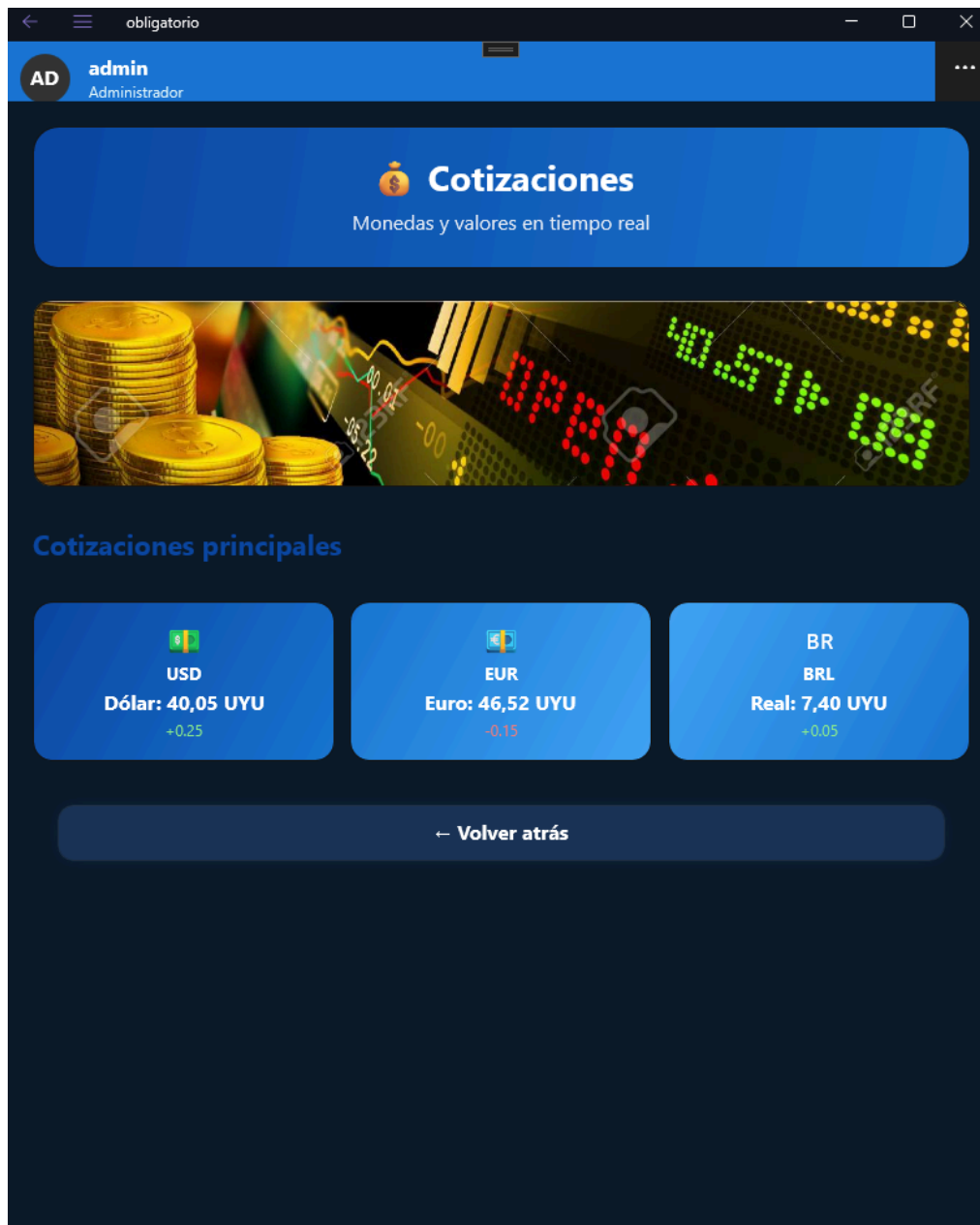
1. En el menú principal, selecciona **“Clima”**.
2. Se mostrará:
  - Clima actual de Punta del Este

- Pronóstico extendido 5 días
  - Iconos representativos de cada clima
  - Temperatura y descripción textual
3. La información se obtiene mediante la API de OpenWeatherMap.



### 3.5.2 Cotizaciones

1. Selecciona “**Cotizaciones**” desde el menú.
2. Visualiza las cotizaciones actuales de:
  - Dólar (USD)
  - Euro (EUR)
  - Real brasileño (BRL)
3. Las cotizaciones se actualizan automáticamente cada 24 horas mediante la API de CurrencyLayer.

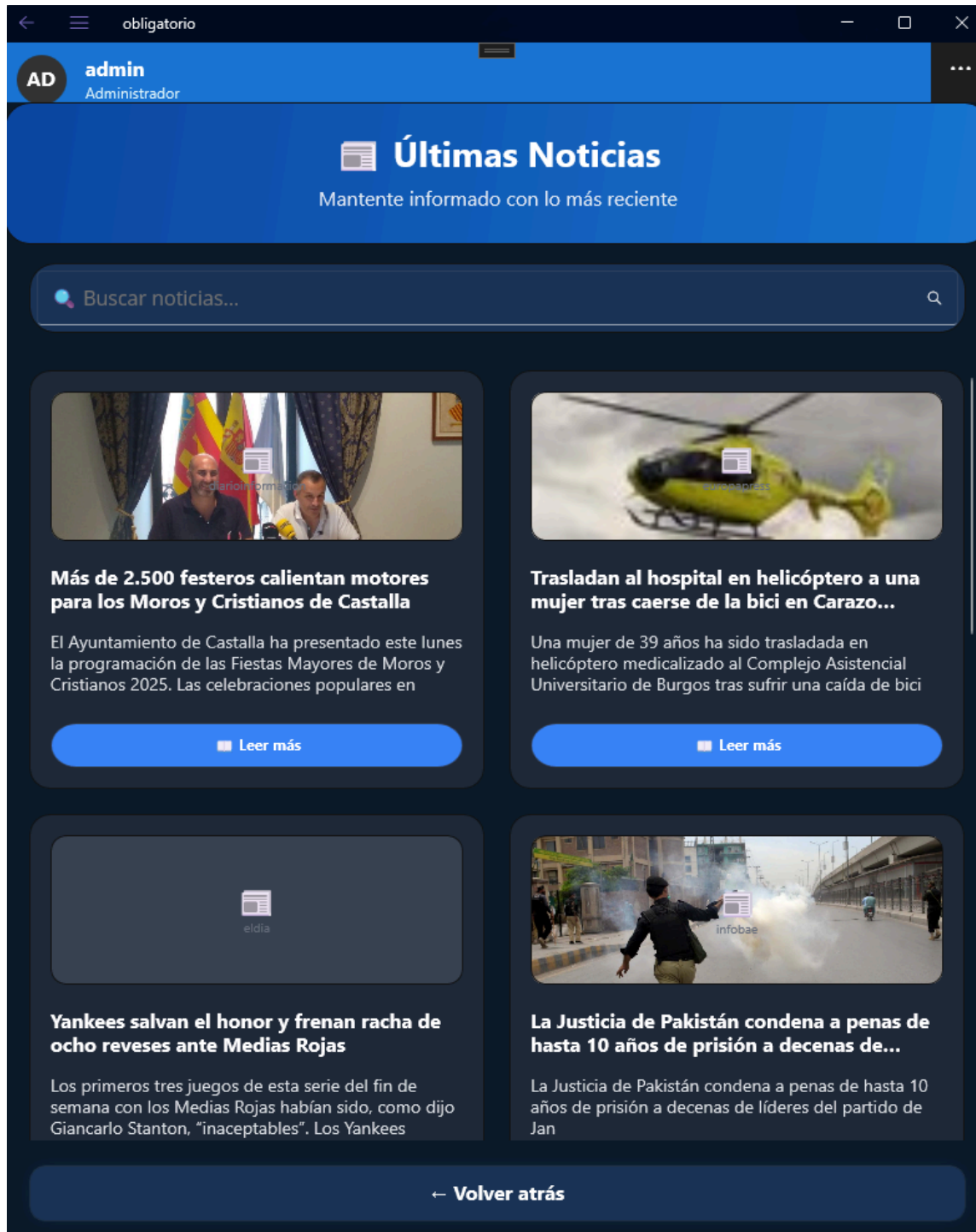


### 3.5.3 Noticias

1. Selecciona “**Noticias**” en el menú principal.
2. Visualiza noticias de Uruguay con:
  - Título
  - Imagen
  - Resumen

- Enlace al artículo completo

3. Puedes filtrar por categorías o palabras clave.



### 3.5.4 Cine



1. Accede a “**Cine**” desde el menú.
2. Se mostrarán próximos estrenos con:
  - Póster de la película
  - Título
  - Sinopsis
  - Fecha de estreno
3. Puedes buscar películas por género o palabra clave usando el buscador integrado (TheMovieDB API).

obligatorio

AD
admin
Administrador

Películas en Cartelera
Estrenos y funciones disponibles

Buscar películas por título...

**La guerra de los mundos**

Will Radford, un destacado analista de ciberseguridad, pasa sus días rastreando posibles amenazas a la seguridad...

29/07/2025 [Ver más](#)

**F1 la película**

El mítico piloto Sonny Hayes vuelve de su retiro, persuadido para liderar un equipo de Fórmula 1 en apuros y guiar...

25/06/2025 [Ver más](#)

**La conductora**

La conductora, un irreverente thriller de ritmo frenético, cuenta la historia de una mujer que, en su adolescencia, fue...

21/08/2025 [Ver más](#)

**Jurassic World: El renacer**

**Los tipos malos 2**

**Guardianes de la noche:**

← Volver atrás

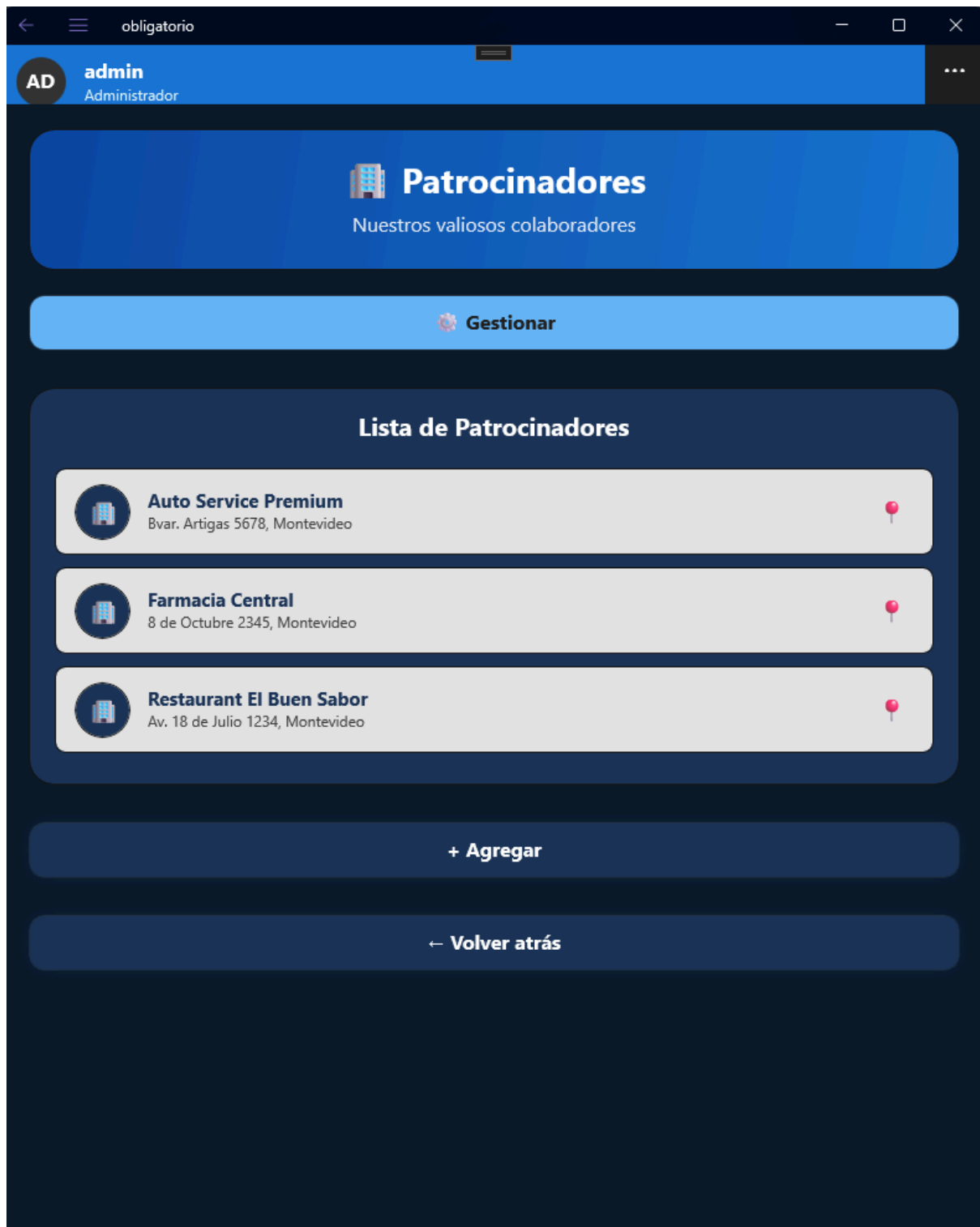
### 3.5.5 Patrocinadores

1. Selecciona “**Patrocinadores**”.

2. Visualiza un mapa interactivo con:

- Ubicación de patrocinadores
- Tu ubicación actual
- Ruta sugerida hacia un patrocinador (si está disponible)

3. Los administradores pueden agregar patrocinadores con nombre, logo y ubicación.



### 3.5.6 Clientes

1. En el menú principal, selecciona la sección **"Clientes"**.

2. Aquí puedes visualizar la lista de todos los clientes registrados en la aplicación. Cada cliente incluye información como:

- Nombre completo
- Usuario
- Teléfono
- Correo electrónico
- Dirección
- Foto de perfil (si fue cargada)

### Acciones sobre Clientes

- **Agregar Cliente:** Los administradores pueden registrar un nuevo cliente ingresando todos los datos requeridos.
- **Editar Cliente:** Modifica los datos de un cliente existente.
- **Eliminar Cliente:** Borra un cliente de la base de datos, con confirmación previa para evitar errores.
- **Buscar Cliente:** Filtra clientes por nombre o usuario mediante la barra de búsqueda.

### Funcionalidad Adicional

- La información de los clientes se almacena de manera segura en **SQLite**, garantizando que los datos estén disponibles incluso sin conexión a internet.
- Los cambios realizados se reflejan inmediatamente en la interfaz de usuario, gracias al patrón **MVVM**.

obligatorio

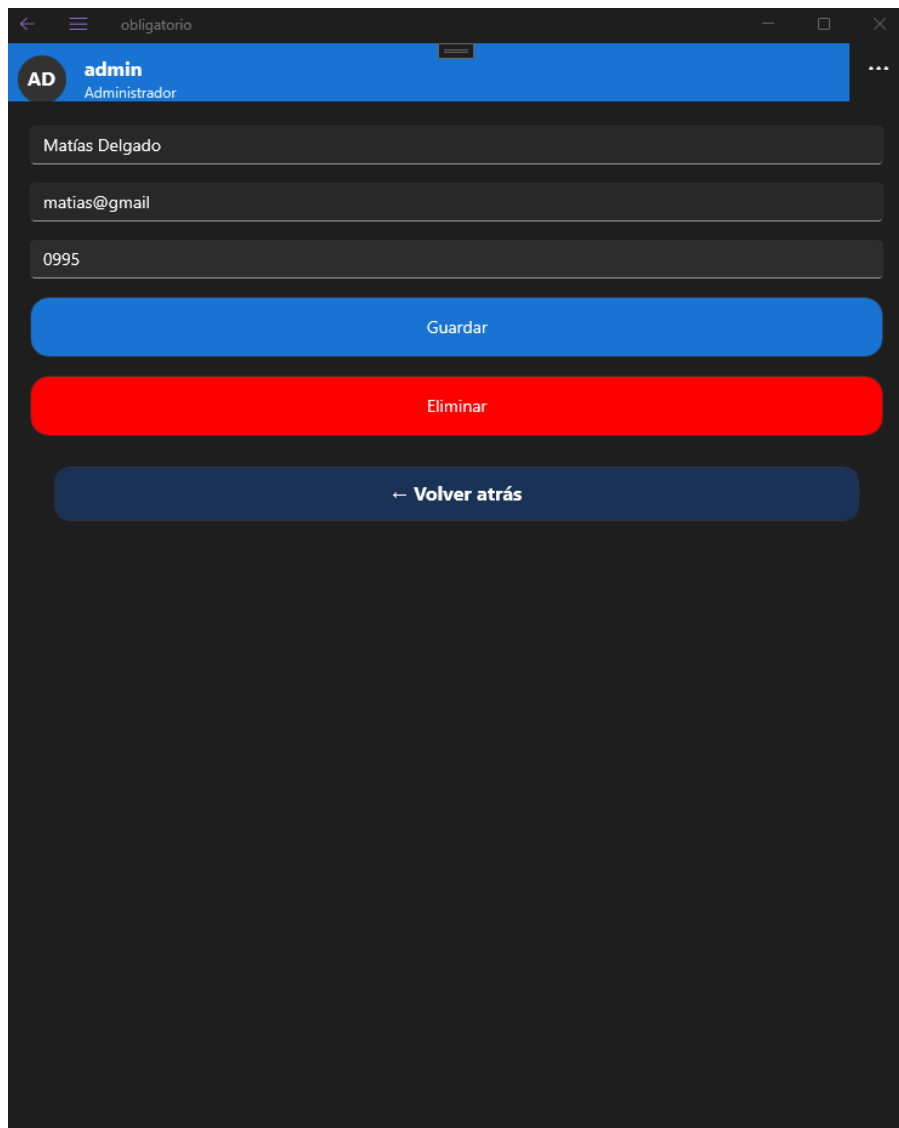
AD

**admin**  
Administrador

Agregar Cliente

matias@gmail  
0995

← Volver atrás

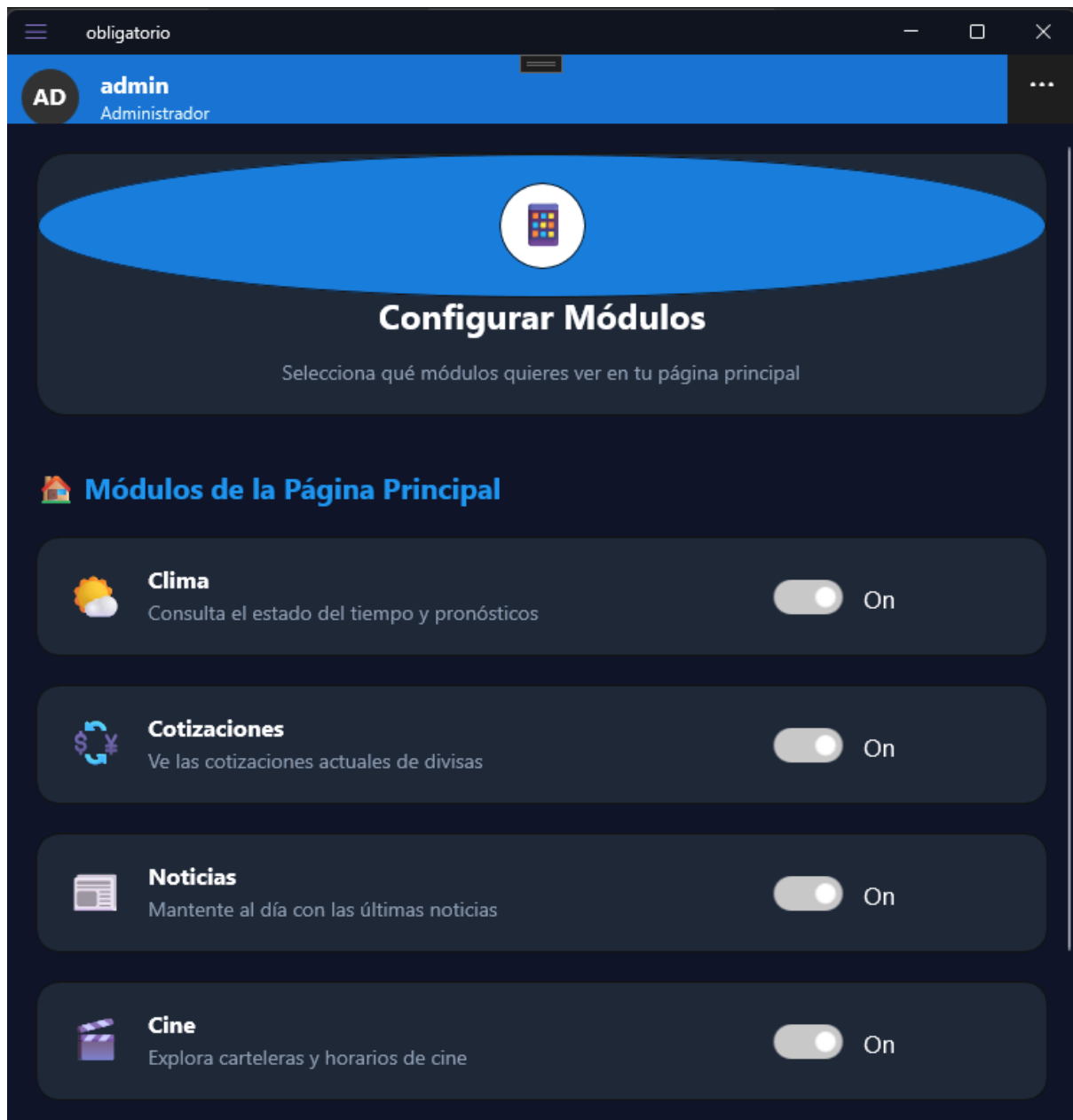


### 3.6 Preferencias del Usuario

1. Ve a “**Preferencias**” desde el menú.
2. Activa o desactiva los módulos que deseas ver:
  - Noticias
  - Clima
  - Cotizaciones
  - Cine

- Patrocinadores

3. Pulsa “**Guardar**” para aplicar los cambios.



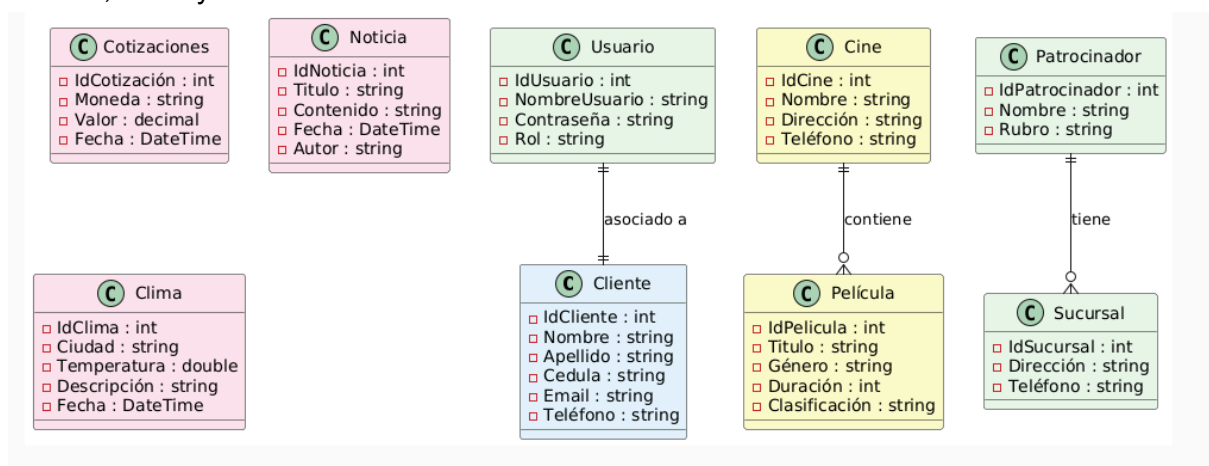




## 4. Diagrama de clase

### 4.1 Diagrama UML:

Este diagrama de clases representa la arquitectura de un sistema de gestión para una radio que integra múltiples funcionalidades: administración de usuarios y oyentes, cartelera cinematográfica, información de patrocinadores y visualización de datos externos como noticias, clima y cotizaciones.



## 5. Conclusiones

### 5.1 Conclusion General:

El desarrollo de este proyecto marcó un punto de inflexión en nuestro proceso de formación como futuros analistas programadores, ya que representó la primera experiencia práctica en el diseño e implementación de una aplicación móvil multiplataforma utilizando **.NET MAUI**. La construcción de este sistema de radio no solo implicó la programación en sí misma, sino también la integración de diversos componentes técnicos que nos obligaron a comprender la importancia de la planificación, el diseño de interfaces intuitivas y la correcta gestión de datos.

En términos técnicos, se logró implementar con éxito una **base de datos SQLite** para almacenar información de usuarios, clientes, órdenes de trabajo y contenidos comerciales, permitiendo una administración eficaz de la información. Asimismo, el uso de **APIs externas** y el consumo de datos dinámicos fueron un reto fundamental, ya que nos brindaron la oportunidad de trabajar con conceptos de conectividad, asincronía y validación de información proveniente de fuentes externas.

Otro aspecto clave fue la aplicación de **principios de diseño responsive**, garantizando que la aplicación mantuviera una experiencia de usuario consistente en distintas plataformas como **Android, iOS y Windows**. Esto nos permitió comprender la importancia de la adaptabilidad y la usabilidad como ejes centrales en el desarrollo de aplicaciones modernas.

Durante el proceso, enfrentamos múltiples dificultades técnicas —desde la organización de la arquitectura del proyecto, la estructuración de las capas de negocio y datos, hasta la correcta vinculación entre interfaces y lógica funcional—. Cada desafío requirió investigación, prueba y error, así como el uso de distintas herramientas de apoyo como **Visual Studio, PlantUML, Draw.io, Stack Overflow, tutoriales de YouTube y documentación oficial**. Esta combinación de fuentes demostró que el aprendizaje autónomo es una de las competencias más relevantes dentro del ámbito del desarrollo de software.

En conclusión, el proyecto no solo cumplió con el objetivo de desarrollar una aplicación móvil funcional, sino que también nos permitió **adquirir competencias prácticas en programación multiplataforma, diseño de bases de datos, consumo de servicios externos y desarrollo de interfaces modernas**. Además, consolidó nuestra capacidad de trabajar en equipo, investigar de manera independiente y resolver problemas técnicos de forma creativa, sentando así una base sólida para futuros proyectos más complejos en nuestra carrera profesional.

## 5.2 Conclusión Individual

### 5.2.1 Danlee Garcia:

Este proyecto representó para mí un gran desafío y, al mismo tiempo, una experiencia de aprendizaje muy enriquecedora. Durante el desarrollo tuve que enfrentarme a distintas dificultades técnicas, desde la organización de la base de datos hasta la implementación de las interfaces y la conexión entre las distintas partes del sistema. Sin embargo, cada obstáculo me obligó a investigar, probar alternativas y mejorar mis conocimientos, lo cual me ayudó a crecer como futuro analista programador. Aprendí que programar no es solamente escribir código, sino también planificar, diseñar y tomar decisiones que hagan que una aplicación sea funcional y fácil de usar. El uso de herramientas como Visual Studio, .NET MAUI y SQLite me permitió entender mejor cómo trabajar con proyectos más completos y portables. Además, el apoyo de documentación, foros y tutoriales me demostró la importancia de saber buscar información y no limitarse solo a lo aprendido en clase.

### 5.2.2 Matias Delgado:

Este proyecto representa mi primera experiencia desarrollando una aplicación móvil con .NET MAUI. El desarrollo de esta aplicación de radio me permitió integrar múltiples funcionalidades (gestión de usuarios, información de entretenimiento, datos externos y módulo comercial), implementando consumo de APIs, y aplicando principios de diseño responsive para garantizar una experiencia de usuario óptima en diferentes plataformas como Android, iOS y Windows. Como estudiante, este proyecto me proporcionó experiencia práctica en el desarrollo multiplataforma de MAUI, sentando las bases para futuros desarrollos más complejos.

## 6. Bibliografía

**Microsoft Visual Studio** – <https://visualstudio.microsoft.com/>

Documentación oficial del IDE utilizado para desarrollo, depuración y compilación de la aplicación multiplataforma.

**.NET MAUI Documentation** – <https://learn.microsoft.com/dotnet/maui/>

Guía oficial de desarrollo con MAUI, incluyendo componentes visuales, XAML, MVVM y compatibilidad multiplataforma.

**SQLite Official Documentation** – <https://www.sqlite.org/docs.html>

Referencia para implementación de base de datos local y manejo de almacenamiento seguro de datos.

**OpenWeatherMap API** – <https://openweathermap.org/api>

Documentación de la API para obtener información climática en tiempo real y pronósticos

extendidos.

**CurrencyLayer API** – <https://currencylayer.com/documentation>

Guía oficial para obtener cotizaciones de monedas en tiempo real.

**NewsData.io API** – <https://newsdata.io/docs>

Documentación de la API utilizada para integrar noticias de Uruguay y filtrado por categorías o palabras clave.

**TheMovieDB API** – <https://developers.themoviedb.org/3>

Referencia de la API para mostrar próximos estrenos de cine, con información de pósters, sinopsis y fechas de estreno.

**OpenAI (ChatGPT)** – <https://openai.com>

Herramienta de inteligencia artificial utilizada para generar documentación, explicaciones técnicas y ejemplos de código.

**Claude AI** – <https://www.anthropic.com/>

Asistente de inteligencia artificial usado para organizar y mejorar la redacción de la documentación.

**PlantUML** – <https://plantuml.com/>

Herramienta para crear diagramas UML a partir de texto, utilizada para documentar la arquitectura y relaciones entre clases.

**Draw.io (diagrams.net)** – <https://app.diagrams.net/>

Plataforma de diagramación gráfica utilizada para ilustrar la arquitectura y flujo de la aplicación.

**Stack Overflow** – <https://stackoverflow.com/>

Repositorio de conocimientos y comunidad de programadores usado para resolver dudas de C#, MAUI, consumo de APIs y buenas prácticas.

**Documentos de Google y Microsoft Word/Docs**

Herramientas de documentación utilizadas para redactar manuales de usuario, informes técnicos y consolidar información colaborativamente.

**YouTube – Tutoriales de desarrollo multiplataforma**

Canales y videos educativos utilizados para aprender sobre:

- .NET MAUI y XAML
- Consumo de APIs REST en C#
- Uso de SQLite en aplicaciones móviles

- MVVM y patrones de arquitectura  
Ejemplos: "dotNET", "IAmTimCorey", "Programming with Mosh", "Code with Andrea".

### **Blogs y Tutoriales Online**

- Medium, Dev.to, y blogs especializados en desarrollo móvil.
- Sitios de ejemplos y buenas prácticas de programación en C# y MAUI.

### **Repositorios de GitHub**

Proyectos de ejemplo y referencias de código para implementar módulos como noticias, clima, cine y mapas interactivos.