



Voz del Este

**Obligatorio
Programación 3**

Franco Nieto, Rodrigo Ciosek

Profesores: Martín Álvarez, José Ramos

2025

Introducción

La emisora radial "Voz del Este", una radio FM con sede en Maldonado (Uruguay), busca desarrollar una aplicación web moderna en ASP.NET MVC para la gestión completa de su operación digital.

El objetivo es ofrecer a sus oyentes una experiencia interactiva: consultar la programación, comentar programas, leer noticias actualizadas, acceder a datos del clima local y cotizaciones de monedas extranjeras. Internamente, el sistema permitirá la gestión de usuarios, roles, clientes, patrocinadores, contenidos y programación radial.

Índice

Presentación del Cliente	4
Objetivos del proyecto:	4
Requerimientos Funcionales	5
1. Autenticación y autorización	5
2. Programa actual	5
3. Gestión de Clientes	5
4. Programación de la Radio	5
5. Sección de Patrocinadores	5
6. Gestión de Noticias	5
7. Comentarios en Programas	5
8. Clima en Maldonado	6
9. Cotizaciones de monedas	6
10. Diseño atractivo y funcional	6
Alcance del Sistema	7
Autenticación y Autorización	7
Página Principal	7
Panel de Administración	7
Programación	7
Noticias	7
Clima	7
Cotizaciones	7
Herramientas utilizadas	8
Arquitectura del Sistema	9
	2

1. Assets	9
2. Servicios	9
3. Controladores	9
4. Modelos	10
5. View Models	11
6. Vistas	12
Climas	12
CotizacionMoneda	12
Noticias	13
Patrocinadores	13
Permisos	13
Programacion	14
ProgramaRadios	14
Roles	14
Usuarios	14
7. SQL Server	15
8. API Externas	16
9. Manual de Usuario	16
Encabezado:	16
Pie de página:	17
Login:	18
Registro:	19
Principal:	20
Programación Semanal:	20
Noticias	21
Clima	21
Cotización de Monedas	22
Funcionamiento Estándar de los Módulos CRUD	22
10. Conclusión Grupal	23

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.

Objetivos del proyecto:

1. Autenticación y autorización

- Registro solo por parte de un Administrador
- Login por todos los tipos de usuario
- Permisos y roles definidos:
 - **Administrador**: acceso total al sistema
 - **Editor**: acceso a noticias y programación
 - **Cliente**: ver programación, dejar comentarios en programas

2. Programa actual

- Al entrar a la página se tiene que mostrar en alguna parte de la página principal, el programa que se está escuchando actualmente, según el día y horario de ingreso a la web.

3. Gestión de Clientes

- Alta, baja y modificación solo por administrador
- Vista de sus datos y comentarios

4. Programación de la Radio

- Gestión de programas, con asignación de múltiples conductores
- Visualización de la grilla semanal (ej. vista tipo calendario o tabla)

5. Sección de Patrocinadores

- CRUD de patrocinadores y visualización de planes
- Asociar cuántas veces aparece su anuncio al día

6. Gestión de Noticias

- CRUD completo de noticias por administradores y editores
- Visualización pública de las noticias ordenadas por fecha

7. Comentarios en Programas

- Los clientes logueados pueden dejar comentarios en los programas
- Estos comentarios se muestran en la página del programa

8. Clima en Maldonado

- Mostrar clima actual y pronóstico
- Usar API de OpenWeatherMap
- Mostrar temperatura, clima (nublado, lluvioso, etc.), mostrar toda la información posible e íconos representativos

9. Cotizaciones de monedas

- Obtener y guardar valores de USD, EUR, BRL y ARS diariamente
- Mostrar histórico con gráficos o tablas

- Usar la API de CurrencyLayer

10. Diseño funcional

- Debe haber un fuerte enfoque en la presentación visual
- La página debe tener una interfaz clara, amigable y visualmente profesional
- Responsiva para visualizarse correctamente en dispositivos móviles

Requerimientos Funcionales

1. Autenticación y autorización

- Registro solo por parte de un Administrador ✓
- Login por todos los tipos de usuario ✓
- Administrador: acceso total al sistema ✓
- Editor: acceso a noticias y programación ✓
- Cliente: ver programación, dejar comentarios en programas ✓

2. Programa actual

- Mostrar el programa actual según día y hora en la página principal ✓

3. Gestión de Clientes

- Alta, baja y modificación solo por administrador ✓
- Vista de sus datos y comentarios ✓

4. Programación de la Radio

- Gestión de programas con múltiples conductores ✓
- Visualización de grilla semanal (calendario o tabla) ✓

5. Sección de Patrocinadores

- CRUD de patrocinadores y visualización de planes ✓
- Asociar frecuencia diaria de aparición de anuncios ✓

6. Gestión de Noticias

- CRUD completo de noticias por administradores y editores ✓
- Visualización pública de noticias ordenadas por fecha ✓

7. Comentarios en Programas

- Clientes logueados pueden dejar comentarios ✓
- Comentarios visibles en la página del programa ✓

8. Clima en Maldonado

- Mostrar clima actual y pronóstico ✓
- Usar API de OpenWeatherMap ✓
- Mostrar temperatura, estado del clima, íconos y más ✓

9. Cotizaciones de monedas

- Obtener y guardar USD, EUR, ARS y BRL diariamente ✓
- Mostrar histórico con gráficos o tablas ✓
- Usar API de CurrencyLayer ✓

10. Diseño atractivo y funcional

- Enfoque en presentación visual ✓
- Interfaz clara, amigable y profesional ✓
- Diseño responsive para móviles ✓

Alcance del Sistema

Autenticación y Autorización

- Inicio de sesión para tres tipos de usuarios: Administrador, Editor y Cliente.
- Registro de usuarios.

Página Principal

- Reproductor de radio en vivo con programa actual según día y hora.
- Programación del día destacada.
- Últimas noticias con imagen y fecha.
- Patrocinadores activos visibles.

Panel de Administración

Solo accesible para Administradores (y Editores en el módulo de noticias). Incluye:

- CRUD de Programas y su programación semanal.
- CRUD de Noticias (Administradores y Editores).
- CRUD de Usuarios y asignación de roles.
- CRUD de Permisos y Roles personalizados.
- CRUD de Patrocinadores y su frecuencia de aparición.
- CRUD de Cliente.

Programación

- Visualización pública de la grilla semanal.
- Navegación por días de la semana.

Noticias

- Listado con acceso a noticias completas.
- Imagen destacada en cada noticia.

Clima

- Clima actual en Maldonado con ícono representativo.
- Pronóstico extendido por 5 días.

Cotizaciones

- Cotización diaria de monedas: USD, EUR, BRL y ARS.
- Gráfico interactivo con historial de cotizaciones.

Herramientas utilizadas

- **ASP.NET MVC**: Arquitectura modelo-vista-controlador para separar capas.
- **Razor**: Motor de vistas dinámico en C# integrado con HTML.
- **jQuery**: Manipulación de DOM, eventos y AJAX.
- **RestSharp**: Librería para consumo de APIs REST.
- **Newtonsoft.Json**: Deserialización de respuestas JSON.
- **Entity Framework**: ORM para interacción con base de datos.
- **ApexCharts**: Visualización interactiva de datos con gráficos.

Arquitectura del Sistema

Este proyecto se desarrolló siguiendo la arquitectura **Modelo-Vista-Controlador (MVC)**, una estructura que permite separar las responsabilidades de presentación, lógica y acceso a datos. El enfoque adoptado para la gestión de persistencia fue **Database First**, en donde la base de datos es diseñada previamente y sirve como origen para la generación de modelos de entidades en el sistema.

1. Assets

Carpeta de recursos estáticos del frontend (estéticos y visuales).

Iconos: Archivos SVG utilizados para la sección Clima y Cotización.

- **Imágenes:** Material gráfico, fotos de programas de radio y noticias.

2. Servicios

Clases que encapsulan la lógica de negocio y comunicación externa.

- **CotizacionService:** Lógica para consumir y gestionar datos de cotizaciones (API de CurrencyLayer).
- **ClimaService:** Se conecta con la API de OpenWeather para obtener el clima actual y pronóstico en Maldonado.
- **NoticiaService:** Obtener datos específicos de la tabla Noticias.
- **PatrocinadorService:** Obtener datos específicos de la tabla Patrocinadores.
- **ProgramacionService:** Obtener datos específicos de la tabla Programacion.

3. Controladores

Clases Controller de ASP.NET que gestionan las rutas, solicitudes y respuestas.

- **ClimasController.cs:** Controla acciones relacionadas con el clima.
- **CotizacionMonedaController.cs:** Controla consulta del valor de monedas.
- **HomeController.cs:** Controlador principal para la página de inicio.
- **NoticiasController.cs:** CRUD de noticias y vista pública ordenada por fecha.
- **PatrocinadoresController.cs:** CRUD de patrocinadores.
- **PermisosController.cs:** CRUD de permisos.
- **ProgramacionController.cs:** CRUD de programación.
- **ProgramaRadiosController.cs:** Gestión individual de programas radiales, comentarios y conductores.
- **RolesController.cs:** CRUD de roles de usuario.

- **UsuariosController.cs:** CRUD de usuarios.
-

4. Modelos

- **Clima:** Representa el clima actual y el pronóstico para Maldonado. Guarda la fecha, la temperatura, una descripción general del clima (por ejemplo: "nublado", "lluvioso") y un ícono representativo.
- **ComentarioPrograma:** Maneja los comentarios que los clientes dejan en los programas. Guarda el texto del comentario, la fecha, el cliente que lo hizo y el programa al que pertenece. Sirve para la interacción con el público.
- **Conductor:** Esta clase representa a los conductores de los programas. Guarda datos personales y una pequeña biografía. Se vincula con los programas de radio que conducen.
- **CotizacionMoneda:** Esta clase representa las cotizaciones de monedas extranjeras (USD, EUR, ARS y BRL) en relación al peso uruguayo. Guarda la fecha, el tipo de moneda y su valor. Se usa para mostrar la cotización del día y también para guardar el historial.
- **Noticia:** Clase que representa una noticia publicada por la radio. Guarda el título, contenido, imagen y fecha.
- **Patrocinador:** Esta clase representa a las empresas o marcas que auspician programas en la radio. Guarda datos como nombre, descripción, y cuántas veces al día se transmite su anuncio.
- **Permiso:** Define acciones específicas que puede tener un rol, como por ejemplo "puede crear noticias" o "puede ver cotizaciones". Se usa si se quiere tener control más detallado que solo el nombre del rol.
- **Persona:** Modelo base para representar a una persona con nombre, apellido, cédula, email, etc. Se usa como base común para los usuarios del sistema.
- **Programación:** Se usa para armar la programación semanal. Guarda el día, hora y a qué programa le corresponde ese espacio. Permite saber qué programa está al aire y generar la visualización semanal.
- **ProgramaRadio:** Modelo que representa cada programa de la emisora. Guarda el nombre, descripción, horario, imagen y la lista de productores. Es clave para armar la grilla de programación y saber qué programa está al aire.
- **Rol:** Representa el rol que tiene cada usuario dentro del sistema. Define si

es Administrador, Editor o Cliente. Se usa para dar permisos y limitar lo que puede ver o hacer un usuario.

- **Usuario:** Es la clase que maneja a los usuarios del sistema. Guarda nombre, email, contraseña y está relacionada con un rol. Sirve para iniciar sesión y controlar el acceso a las distintas partes del sistema.

5. View Models

Estructuras intermedias entre la vista y el modelo.

- **Validadores**

- **TienePermiso:** Clase que verifica si un usuario tiene permisos para acceder a ciertas rutas o funcionalidades.
- **FechaValidador:** Clase para validar el campo fecha para que se encuentre en un rango definido.

- **DTO (Data Transfer Objects)**

- **ClimaForecastRespuestaDTO:** Esta clase representa la respuesta de la API de OpenWeather para el pronóstico del clima. Contiene una lista con los datos horarios del clima y la información de la ciudad consultada. Se usa para deserializar el JSON que devuelve la API y obtener los datos necesarios para mostrar el pronóstico en el sistema.
- **ClimaWeatherRespuestaDTO:** Esta clase representa la respuesta de la API de OpenWeather para el clima actual. Contiene la lista de condiciones climáticas, datos principales como temperatura y humedad, información del viento y la fecha/hora en formato timestamp. Se usa para deserializar el JSON de la API y mostrar el clima actual en el sistema.
- **CurrencyLayerRespuestaDTO:** Es una clase que se utiliza para representar la respuesta que devuelve la API de CurrencyLayer. Se usa cuando el sistema consulta las cotizaciones de monedas extranjeras. Esta clase permite deserializar la respuesta de la API y trabajar con los datos desde el código.

- **ProgramaDTO:** Esta clase es un objeto de transferencia de datos que representa la información principal de un programa de radio para mostrarse en la interfaz. Contiene el Id de la programación, la fecha y horario, duración, nombre, descripción, URL de la imagen y una lista con los nombres de los conductores asociados. Se usa para enviar datos del programa de manera simplificada y clara desde el backend hacia la vista.
- **UsuarioSesion:** Esta clase representa la información básica de un usuario que ha iniciado sesión en el sistema. Contiene el Id del usuario, su nombre de usuario, el nombre completo y una lista de permisos asignados. Se utiliza para manejar la sesión activa y controlar qué acciones puede realizar el usuario durante su uso del sistema.

6. Vistas

Archivos .cshtml que definen la interfaz gráfica mostrada al usuario.

Climas

- **Home.cshtml:** Muestra el clima actual y el pronóstico, obtenido mediante la API de OpenWeatherMap. Vista pública informativa.

CotizacionMoneda

- **Home.cshtml:** Presenta cotizaciones actualizadas de monedas (USD, EUR, UYU, BRL) y su histórico. Vista informativa con posible gráfico.
- **About.cshtml:** Información sobre la radio “Voz del Este”, su misión, visión, etc.
- **Contact.cshtml:** Formulario de contacto o información para comunicarse con la radio.
- **Index.cshtml:** Página principal. Muestra resumen del sitio (programa en curso, clima, noticias recientes, etc.).

Noticias

- **Create.cshtml:** Formulario para que administradores o editores creen una nueva noticia.
- **Delete.cshtml:** Confirmación para eliminar una noticia existente.
- **Details.cshtml:** Vista de los detalles completos de una noticia.
- **Edit.cshtml:** Formulario para editar el contenido de una noticia existente.
- **Home.cshtml:** Listado de noticias públicas ordenadas por fecha.
- **Index.cshtml:** Panel de administración/listado completo de noticias para CRUD.

Patrocinadores

- **Create.cshtml:** Formulario para crear un nuevo patrocinador y asignarle detalles como imagen, cantidad de transmisiones diarias, etc.
- **Delete.cshtml:** Confirmación para eliminar un patrocinador.
- **Details.cshtml:** Vista detallada de un patrocinador (nombre, imagen, descripción, etc.).
- **Edit.cshtml:** Modificación de datos del patrocinador.
- **Index.cshtml:** Listado general de patrocinadores con acciones disponibles (editar, borrar, ver).

Permisos

- **Create.cshtml:** Crear un nuevo permiso en el sistema.
- **Delete.cshtml:** Confirmación para eliminar un permiso.
- **Details.cshtml:** Vista informativa de un permiso específico.
- **Edit.cshtml:** Modificar un permiso existente.
- **Index.cshtml:** Lista de todos los permisos del sistema.

Programacion

- **Create.cshtml:** Agregar nueva programación (día, hora, asignar programa).
- **Delete.cshtml:** Confirmación para eliminar un bloque de programación.
- **Details.cshtml:** Ver la información de una programación puntual.
- **Edit.cshtml:** Editar fecha, hora o programa asignado a una programación.
- **Index.cshtml:** Grilla completa de programación, posiblemente con vista semanal.

ProgramaRadios

- **Create.cshtml:** Formulario para agregar un nuevo programa.
- **Delete.cshtml:** Confirmación para eliminar un programa existente.
- **Details.cshtml:** Muestra información detallada de un programa.
- **Edit.cshtml:** Permite modificar los datos de un programa.
- **Index.cshtml:** Lista general de todos los programas de radio.

Roles

- **Create.cshtml:** Crear un nuevo rol de usuario.
- **Delete.cshtml:** Eliminar un rol.
- **Details.cshtml:** Ver los detalles de un rol.
- **Edit.cshtml:** Editar información de un rol.
- **Index.cshtml:** Mostrar todos los roles disponibles.

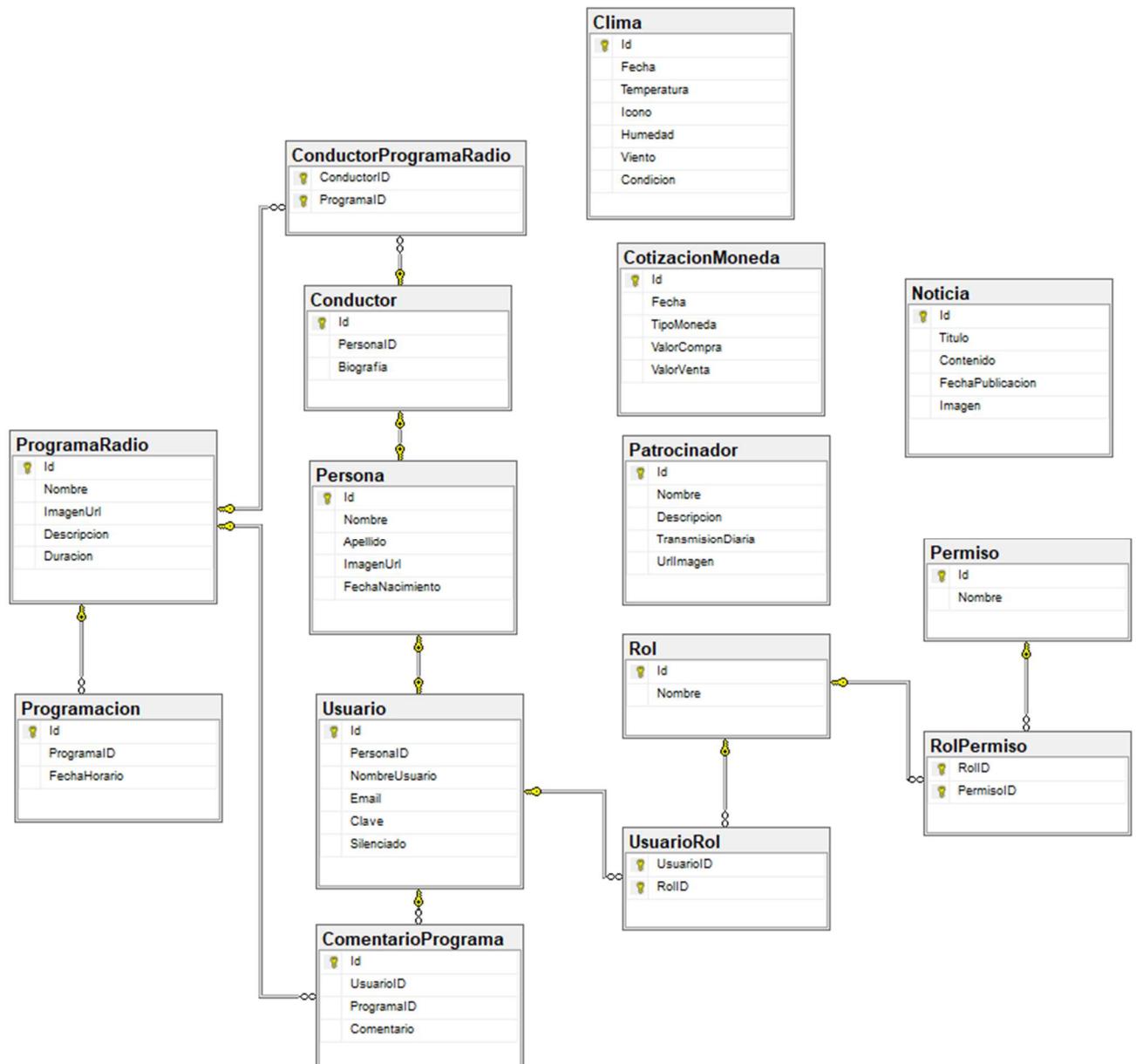
Usuarios

- **Delete.cshtml:** Eliminar usuario.
- **Details.cshtml:** Ver perfil detallado de un usuario.
- **Edit.cshtml:** Editar datos de usuario.

- **Index.cshtml**: Mostrar listado de usuarios.
- **Login.cshtml**: Página de inicio de sesión.
- **Registro.cshtml**: Registro de nuevos usuarios.

7. SQL Server

Diagrama de entidades:



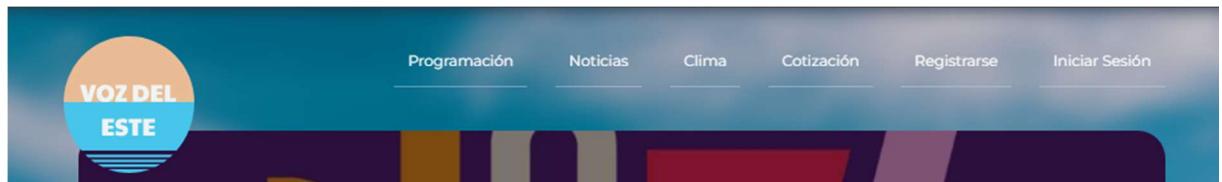
8. API Externas

CurrencyLayer provee datos de cotización de monedas (USD, EUR, BRL, ARS). Se utiliza para mostrar el historial y los datos actualizados diariamente.

OpenWeatherMap provee datos del clima actual y del pronóstico en Maldonado. Se utiliza para mostrar el historial y los datos actualizados diariamente.

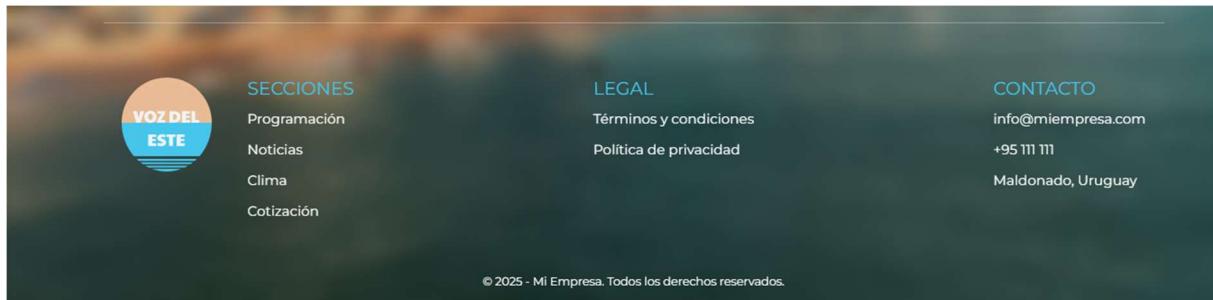
9. Manual de Usuario

Encabezado:



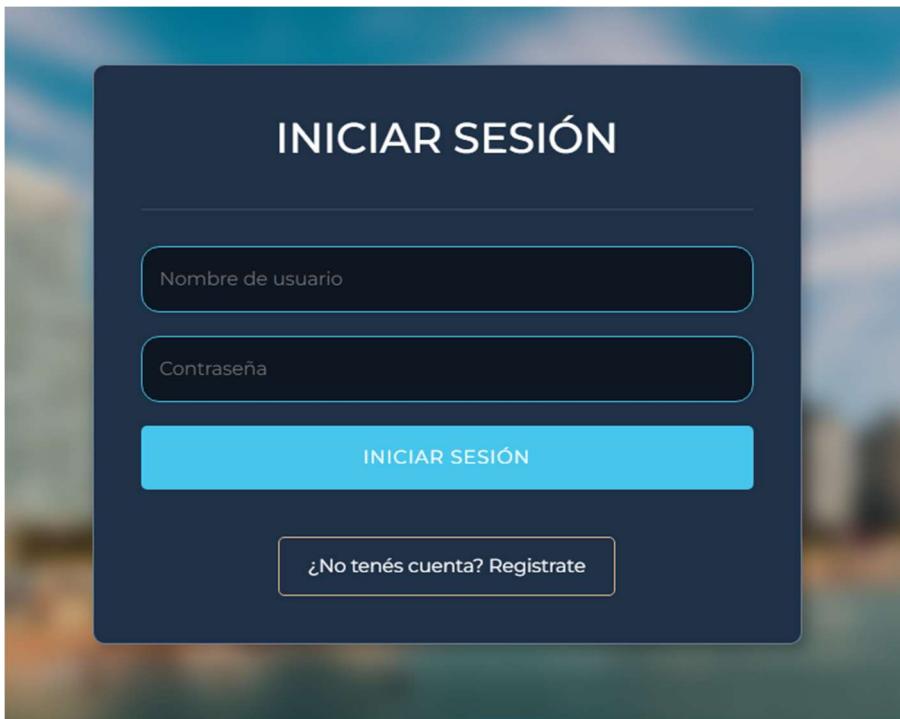
- **Logo:** Redirige a la página principal.
- **Programación:** Muestra la grilla de programas por día y horario.
- **Noticias:** Acceso al listado de noticias recientes.
- **Clima:** Información actual y pronóstico en Maldonado.
- **Cotización:** Cotizaciones y gráficos de monedas extranjeras.
- **Registrarse:** Alta de usuarios (solo administradores).
- **Iniciar Sesión:** Autenticación para usuarios registrados.

Pie de página:



- **Secciones del sitio:** Accesos rápidos a Programación, Noticias, Clima y Cotización.
- **Información legal:** Enlaces a Términos y condiciones y Política de privacidad.
- **Contacto:** Datos institucionales como correo electrónico, teléfono y ubicación en Maldonado.

Login:



- **Campo de nombre de usuario:** Ingresar el identificador personal del usuario.
- **Campo de contraseña:** Ingresar la contraseña asignada.
- **Botón “Iniciar sesión”:** Confirma los datos ingresados e intenta acceder al sistema.
- **Enlace de registro:** Visible solo para administradores que crean nuevos usuarios.

Registro:

The screenshot shows a mobile application interface for user registration. At the top, a dark blue header bar displays the text "INGRESE SUS DATOS" in white capital letters. Below the header, there are six input fields, each with a light blue placeholder text: "Nombre", "Apellido", "Email", "Nombre de Usuario", "Clave", and "dd/mm/aaaa". After the date field, there is a small calendar icon. Below these fields is a file selection area with a blue button labeled "Seleccionar archivo" and the text "Sin archivos seleccionados". At the bottom of the form is a large blue button with the white text "CREAR USUARIO". In the bottom left corner of the main form area, there is a small white "Volver" button.

- **Campo Nombre:** Para ingresar el nombre del usuario.
- **Campo Apellido:** Para ingresar el apellido del usuario.
- **Campo Email:** Dirección de correo electrónico válida.
- **Campo Nombre de Usuario:** Identificador único para iniciar sesión.
- **Campo Clave:** Contraseña asignada al usuario.
- **Campo Fecha de Nacimiento:** Selección en formato día/mes/año.
- **Campo Imagen de Perfil:** Opción para subir una imagen desde el dispositivo (opcional).

Principal:

- **Programa actual en vivo:** Aparece en la parte superior indicando el título del programa que está al aire en ese momento.
- **Próximo programa:** Muestra el nombre del siguiente programa que se emitirá y la hora de inicio estimada.
- **Programación de hoy:** Sección con el listado de todos los programas del día.
Incluye:
 - Nombre del programa
 - Horario de inicio y fin
 - Conductores asociados
 - Breve descripción del contenido del programa
- **Ultimas Noticias:** Sección con las ultimas noticias publicadas.
Incluye:
 - Título de la noticia
 - Descripción breve del contenido principal
 - Fecha y hora de publicación
 - Botón “Ver más” para acceder a la noticia completa

Programación Semanal:

La sección de programación presenta al usuario una grilla organizada por día de la semana. En cada bloque se incluyen los siguientes elementos que se encuentran en el día seleccionado:

- **Nombre del programa:** Título del contenido que se emite en ese horario.
- **Día y horario:** Indica cuándo se transmite cada programa, con hora de inicio y fin.
- **Conductores:** Lista de personas encargadas del programa.
- **Descripción:** Breve resumen informativo sobre el contenido del programa.

Esta vista permite al usuario conocer qué se transmite cada día, acceder a la información de interés y anticiparse a los contenidos disponibles durante la semana.

Noticias

Esta sección muestra una selección de noticias actuales, presentadas en formato de tarjetas. Cada tarjeta incluye:

- Título de la noticia
- Descripción introductoria del contenido
- Fecha y hora de publicación
- Botón “Expandir” para acceder al desarrollo completo de la noticia
- Imagen ilustrativa relacionada con el tema

Las tarjetas se actualizan con contenido informativo relevante sobre temas como legislación, cultura, salud y tecnología. Están organizadas cronológicamente y permiten una lectura rápida o una exploración más detallada según el interés del visitante.

Clima

La sección de clima muestra el estado meteorológico actual y el pronóstico horario para Maldonado. Los elementos principales que se visualizan son:

- **Condición actual:** Presenta el tipo de cielo (por ejemplo, “CIELO CLARO”) y la temperatura medida.
- **Icono representativo:** Imagen del clima correspondiente al estado actual (sol, nubes, lluvia, etc.).
- **Pronóstico horario:** Listado de intervalos con fecha, hora, condición esperada y temperatura estimada.

Esta información se actualiza dinámicamente a partir de una API externa y ayuda al usuario a anticipar el clima en distintos momentos del día y la madrugada siguiente.

Cotización de Monedas

La sección muestra el valor actualizado de las monedas extranjeras con respecto al peso uruguayo, además de un gráfico con su evolución histórica. Los elementos incluidos son:

- **Cotización actual:** Se presentan los valores del USD, EUR, BRL y ARS, junto con la fecha y hora de la última actualización.
- **Visualización gráfica:** Gráfico comparativo que muestra la variación de cada moneda a lo largo del último mes.
- **Leyenda de colores:** Cada moneda está representada por una línea de color diferente para facilitar la lectura:
 - USD: azul
 - EUR: verde
 - BRL: amarillo
 - ARS: rojo

Esta sección permite al usuario consultar las cotizaciones del día y observar tendencias económicas mediante la evolución visual de los valores.

Funcionamiento Estándar de los Módulos CRUD

Los módulos del sistema que permiten **gestión de entidades** (como usuarios, programas, noticias, patrocinadores, etc.) siguen una estructura estándar de tipo CRUD (Crear, Leer, Actualizar, Eliminar). Esta estructura se presenta mediante una tabla con las siguientes características:

- **Listado de registros existentes** con columnas como nombre, descripción, fecha u otros datos relevantes según el tipo de entidad.
- En cada fila se incluyen botones para las siguientes acciones:
 - **Editar:** Permite modificar los datos del registro.
 - **Detalles:** Muestra la información completa del registro en vista extendida.
 - **Eliminar:** Elimina el registro del sistema, previa confirmación.
- En la parte superior de la tabla se encuentra el botón “**Crear [nombre de la entidad]**”, que abre un formulario para agregar un nuevo elemento.

Este diseño se replica en los distintos módulos administrativos del sistema, lo que garantiza una experiencia de uso coherente y simplificada.

10. Conclusión Grupal

Aprendimos bastante en el proceso, sobre todo cómo conectar todo: backend, base de datos y lo que ve el usuario. Quedamos conformes con lo que hicimos y con cómo lo resolvimos en equipo.

El trabajo nos sirvió para poner en práctica todo lo que aprendimos durante el curso. Pudimos organizarnos bien, repartir tareas y llevar el proyecto adelante sin dar vueltas. Terminamos con un sistema funcional, prolijo y completo, que cumple con lo que se pedía. Quedamos conformes con el resultado y con lo que fuimos mejorando a medida que lo hacíamos. Fue una buena experiencia para entender cómo es desarrollar algo real, en equipo y con todas las piezas conectadas.