



ULPGC
**Universidad de
Las Palmas de
Gran Canaria**



eii

**ESCUELA DE
INGENIERÍA INFORMÁTICA**

Trabajo de Fin de Título

Desarrollo de una Plataforma Web y Móvil para Gestión de Eventos y Comunidad de Rally

TITULACIÓN: Grado en Ingeniería Informática

AUTOR: Ismael Ramos Alonso

TUTORIZADO POR:

María Dolores Afonso Suárez

Marzo de 2025

Agradecimiento

En primer lugar, quiero expresar mi más sincero agradecimiento a mi tutora, María Dolores Afonso Suárez, por su apoyo, ánimos y la libertad que me ha brindado para desarrollar este Trabajo de Fin de Grado. Sin duda, su orientación ha sido fundamental para hacerlo posible.

A mis padres, quienes han sido mi mayor inspiración y soporte. A mi madre, por ser mi guía, mi pilar y la persona que más se alegra de mis logros. A mi padre, por ser un ejemplo de rectitud, seriedad, educación y enfoque, valores que me han acompañado en cada paso de este camino.

A mi hermano, Daniel Ramos, por darme todo lo que está en su mano y aún más, y por apoyarme siempre a seguir creciendo como persona y como profesional.

A mi pareja, María José Galván, por confiar en mí desde el principio y estar a mi lado en los momentos difíciles. Sin duda, has sido un apoyo imprescindible y fundamental en este proceso.

A mi familia y amigos, porque sin ustedes esto no habría sido posible. Cada uno de ustedes tiene un pedacito en este proyecto. Gracias por formar parte de mi vida.

Gracias a todos los que han formado parte de este camino.

Resumen

Este Trabajo de Fin de Título se centra en el desarrollo de una aplicación móvil y una plataforma web dedicadas al mundo del rally. La aplicación móvil, desarrollada con Android Studio, permite acceder a eventos, noticias actualizadas, una sección específica del Campeonato Mundial de Rally, con clasificaciones y resultados oficiales. Además, ofrece la posibilidad de guardar eventos favoritos, visualizarlos en un calendario personal y recibir notificaciones. Por su parte, la plataforma web, implementada con Angular, está diseñada para gestionar eventos, noticias y usuarios, así como para resolver incidencias y analizar estadísticas, optimizando la administración del contenido. Este proyecto tiene como objetivo fomentar el interés por el rally, mejorando la interacción entre los aficionados y facilitando la gestión eficiente del ecosistema digital asociado a este deporte.

Abstract

This Final Degree Project focuses on developing a mobile application and a web platform dedicated to the world of rallying. The mobile application, built with Android Studio, provides access to events, real-time news, and a dedicated section for the World Rally Championship, featuring official rankings and results. Additionally, it allows users to save their favorite events, view them in a personal calendar, and receive relevant notifications. Meanwhile, the web platform, developed with Angular, is designed to manage events, news, and users, as well as to address issues and analyze statistics, thereby optimizing content administration. This project aims to promote interest in rallying by enhancing interaction among enthusiasts and facilitating the efficient management of the digital ecosystem associated with this sport.

Índice general

Capítulo 1: Introducción	12
Capítulo 2: Estado actual y objetivos iniciales.....	13
2.1. Motivación.....	13
2.2. Estado Actual.....	13
2.2.1. Aplicaciones móviles.....	13
2.3. Objetivos.....	17
Capítulo 3: Aportaciones del trabajo y Competencias específicas.....	18
3.1. Aportaciones Socioeconómicas del Proyecto	18
3.2. Competencias específicas	19
3.3. Alineamiento con los objetivos de desarrollo sostenible.....	21
Capítulo 4: Desarrollo	24
4.1. Metodología.....	24
4.1.1. Estructura de los Incrementos.....	24
4.1.2. Fases de cada incremento	26
4.1.3. Gestión y Documentación de Tareas con Kanban y Trello	27
4.2. Lógica de la aplicación y la web	30
4.2.1. Flujo de la aplicación.....	30
4.2.2. Flujo de la web	31
4.3. Herramientas utilizadas en el desarrollo.....	33
4.3.1. Android Studio	33
4.3.2. Visual Studio	33
4.3.3. RapidAPI,	33
4.3.4. Git.....	34
4.3.5. Firebase.....	34
4.3.6. Trello	35
4.3.7. Figma.....	35
4.4. Arquitectura de la Base de Datos	36
4.4.1. Ventajas de Utilizar Firebase Firestore.....	36
4.4.2. Estructura y Organización de las Colecciones.....	36
4.5. Implementación del Patrón MVVM	38

4.6. Funcionamiento de la API	40
4.6.1. Características de la API.....	40
4.6.2. Métodos de solicitud y RapidAPI Code Snippet	40
4.7. Desarrollo de la aplicación móvil.....	42
4.7.1. Iniciación y configuración del proyecto en Android Studio.....	42
4.7.2. Diseño de la aplicación móvil	43
4.7.3. Arquitectura del Backend	44
4.7.4. Integración con la API.....	47
4.7.5. Estructura de ficheros	50
4.7.6. Funcionalidades	52
4.8. Desarrollo de la web	65
4.8.1. Iniciación y configuración del proyecto en Visual Studio.....	65
4.8.2. Diseño de panel de control web.....	66
4.8.3. Arquitectura del Backend	70
4.8.4. Estructura de ficheros	71
4.8.5 Funcionalidades	73
Capítulo 5: Conclusiones y trabajo futuro.....	82
5.1. Conclusiones.....	82
5.2. Trabajo futuro	83
Capítulo 6: Bibliografía.....	84
Capítulo 7: Anexo I- Ampliación de imágenes	86
7.1. Diagrama de Flujo de la Web	86
7.2. Diagrama de Flujo de la App.....	87

Índice de figuras

Ilustración 2.1: Pantalla de Inicio de Rally TV	14
Ilustración 2.2: Pantalla de competiciones en directo de Rally TV	14
Ilustración 2.3: Pantalla de Calendario de Rallys de Rally TV	14
Ilustración 2.4: Pantalla de Inicio de la Web de WRC.....	15
Ilustración 2.5: Pantalla de Calendario de Eventos de la Web de WRC	15
Ilustración 2.6: Pantalla principal de Dakar Rally	16
Ilustración 4.1: Diagrama de Incrementos	26
Ilustración 4.2: Progreso Trello 1	28
Ilustración 4.3: Progreso Trello 2	28
Ilustración 4.4: Progreso Trello 3	29
Ilustración 4.5: Progreso Trello 4	29
Ilustración 4.6: Diagrama de Flujo de la aplicación.....	31
Ilustración 4.7 Diagrama de Flujo de la Web.....	32
Ilustración 4.8: Logo Android Studio	33
Ilustración 4.9: Logo Visual Studio	33
Ilustración 4.10: Logo RapidAPI.....	34
Ilustración 4.11: Logo git.....	34
Ilustración 4.12: Logo Firebase	35
Ilustración 4.13: Logo Trello	35
Ilustración 4.14: Logo Figma.....	35
Ilustración 4.15: Diagrama de la base de datos.....	38
Ilustración 4.16: Diagrama MVVM.....	39
Ilustración 4.17: Método de solicitud	41
Ilustración 4.18: Código identificador WRC	41
Ilustración 4.19: Creación de Proyecto	42
Ilustración 4.20: Diseño Ejemplo 1	43
Ilustración 4.21: Diseño Ejemplo 2	44
Ilustración 4.22: Firebase Authentication	45
Ilustración 4.23: Cloud Firestore	45
Ilustración 4.24: Firebase Storage.....	46
Ilustración 4.25: Código de configuración de retrofit.....	48
Ilustración 4.26: Definición de Endpoints	49

Ilustración 4.27: Estructura de ficheros de la aplicación	51
Ilustración 4.28: Estructura fichero basados en MVVM.....	51
Ilustración 4.29: Acceso de Usuarios.....	53
Ilustración 4.30: Pantalla de Inicio	53
Ilustración 4.31: Eventos	54
Ilustración 4.32: Detalles de Noticia.....	55
Ilustración 4.33: Notificaciones de Foro.....	55
Ilustración 4.34: Sección del mundial de Rally	56
Ilustración 4.35: Temporadas y clasificación.....	57
Ilustración 4.36: Detalles de Temporada	57
Ilustración 4.37: Pilotos del Mundial de Rally.....	58
Ilustración 4.38: Coches y equipos de WRC	59
Ilustración 4.39: Foro.....	60
Ilustración 4.40: Otras secciones	60
Ilustración 4.41: Sección pilotos favoritos.....	61
Ilustración 4.42: Eventos guardados	61
Ilustración 4.43: Calendario de eventos.....	62
Ilustración 4.44: Galería de imágenes.....	62
Ilustración 4.45: Sección de vídeos	63
Ilustración 4.46: Sección Perfil de usuario	63
Ilustración 4.47: Sección de Ajustes.....	64
Ilustración 4.48: Sección de Soporte	64
Ilustración 4.49: Proyecto creado en Visual Studio	66
Ilustración 4.50: Diseño inicio de sesión	66
Ilustración 4.51: Inicio y Barra Lateral	67
Ilustración 4.52: Análisis de eventos	68
Ilustración 4.53: Calendario de eventos y noticias creadas.....	68
Ilustración 4.54: Listado de eventos	69
Ilustración 4.55: Gestión de administradores.....	69
Ilustración 4.56: Colección events.....	70
Ilustración 4.57: Colección news.....	71
Ilustración 4.58: Estructura de ficheros	72
Ilustración 4.59: Inicio de sesión	73
Ilustración 4.60: Registro de administrador	74

Ilustración 4.61: Creación de evento	75
Ilustración 4.62: Editar evento	75
Ilustración 4.63: Detalles noticias.....	77
Ilustración 4.64: Editar noticia.....	77
Ilustración 4.65: Generar código de administrador	78
Ilustración 4.66: Listado de administradores	78
Ilustración 4.67: Editar rol de un administrador	79
Ilustración 4.68: Subir imagen y crear colección.....	79
Ilustración 4.69; Editar Colección	80
Ilustración 4.70: Añadir Imagen a Colección.....	80
Ilustración 4.71: Listado de usuarios	81
Ilustración 4.72: Listado de informes y preguntas	81
Ilustración 7.1: Flujo web 1	86
Ilustración 7.2: Flujo web 2	86
Ilustración 7.3: Flujo web 3.....	86
Ilustración 7.4: Flujo App 1.....	87
Ilustración 7.5: Flujo App 2.....	87
Ilustración 7.6: Flujo App 3.....	88

Índice de cuadro

Tabla 3.1: Tabla de grado de relación de los ODS.....	21
---	----

Capítulo 1: Introducción

El rally es una disciplina automovilística caracterizada por su dinamismo, su diversidad de entornos y su complejidad logística; aspectos que la diferencian de otras modalidades del deporte del motor. Lejos de disputarse en circuitos cerrados, el rally se desarrolla en tramos cronometrados a lo largo de recorridos sobre asfalto, grava, nieve o incluso barro, integrándose en el medio natural y ofreciendo un espectáculo singular y atractivo.

En este contexto, se hace evidente la necesidad de contar con herramientas que contribuyan a difundir el rally y a facilitar el seguimiento de su actividad. Más allá de la mera obtención de datos, los aficionados requieren soluciones que les permitan mantenerse informados sobre las últimas noticias como consultar clasificaciones oficiales, descubrir nuevos eventos y conectar con otros seguidores. Además, para aquellos que desean asistir a competiciones en vivo o no perder detalle de los avances en el campeonato mundial, resulta indispensable disponer de recursos que ayuden a planificar la experiencia, recordando fechas clave y brindando un panorama completo de lo que sucede en tiempo real.

Es precisamente esta necesidad la que impulsa el desarrollo de la solución técnica que se describe a continuación. Mediante la integración de una aplicación móvil y una plataforma web administrativa, el proyecto busca satisfacer las demandas de los aficionados al rally, promoviendo un mayor conocimiento de este deporte, fortaleciendo su presencia y garantizando un acceso más sencillo, ordenado y eficaz a la información. De esta forma, se aspira a fomentar una comunidad más unida, informada y comprometida con el mundo del rally.

La solución propuesta consta de dos componentes principales, diseñados para cubrir tanto las necesidades del público general como las de los gestores del contenido. Por un lado, se presenta una aplicación nativa para dispositivos Android, desarrollada con el objetivo de brindar a los usuarios finales acceso a eventos y noticias actualizadas, un foro para compartir experiencias y un apartado dedicado al campeonato mundial de rally. En esta sección se integran datos oficiales sobre clasificaciones, resultados y detalles de cada etapa, lo que permite ofrecer una visión completa y contrastada de la competición. Además, se incorpora una galería de imágenes, cuyo fin es contextualizar y complementar la información suministrada, enriqueciendo así la experiencia del usuario.

Adicionalmente se ha implementado una plataforma web con un enfoque exclusivamente administrativo, ideada para facilitar la gestión interna de la aplicación. Este entorno permite la creación y actualización de eventos, la publicación de noticias, la moderación de usuarios y contenidos, la subida de imágenes, así como el acceso a estadísticas relacionadas con el número de eventos, publicaciones y aficionados registrados. Esta separación de funciones entre aplicación móvil y plataforma web busca optimizar el flujo de trabajo y garantizar la coherencia, actualidad y fiabilidad de la información ofrecida al público.

Capítulo 2: Estado actual y objetivos iniciales

2.1. Motivación

La idea principal de este proyecto surge de la necesidad de solucionar la dispersión de información que caracteriza al mundo del rally. Aunque esta disciplina cuenta con un amplio reconocimiento y seguimiento a nivel internacional, no existe una plataforma centralizada que permita a los aficionados acceder de manera rápida, sencilla y confiable a datos oficiales, noticias, resultados, detalles de etapas, y que además les brinde la posibilidad de interactuar con otros usuarios, guardar eventos y seguir a sus pilotos favoritos.

En la actualidad, el aficionado al rally enfrenta una situación en la que mantener un seguimiento constante y completo de la actividad deportiva implica visitar múltiples fuentes, contrastar datos y disponer de recordatorios o herramientas externas para no perder de vista las fechas clave de cada competición. Esta falta de centralización dificulta el acceso a información global y actualizada, y limita la interacción ordenada entre seguidores. Asimismo, la ausencia de un entorno integrado que provea, por un lado, un canal de comunicación y difusión de datos dirigido a los entusiastas y, por otro, un medio de gestión interno para organizadores y administradores de contenido impide la optimización de procesos y la mejora en la experiencia informativa.

2.2. Estado Actual

2.2.1. Aplicaciones móviles

En la actualidad, las aplicaciones móviles se han convertido en herramientas fundamentales para el seguimiento y la participación en competiciones deportivas de gran envergadura. Dentro del ámbito del rally, la adopción de plataformas digitales ha permitido a organizadores, participantes y aficionados acceder a información en tiempo real, contenido multimedia y funcionalidades interactivas que enriquecen la experiencia de cada evento.

A continuación, se describen dos aplicaciones destacadas en este sector, acompañadas de las principales características que ofrecen a su audiencia:

Rally.TV

Rally.TV [14] se presenta como un servicio de transmisión en vivo centrado en competiciones oficiales de rally y rallycross, principalmente del FIA World Rally Championship, European Rally Championship y World Rallycross Championship. Asimismo, brinda la oportunidad de acceder a repeticiones y resúmenes, favoreciendo la actualización permanente de quienes no pueden visualizar los eventos en directo.

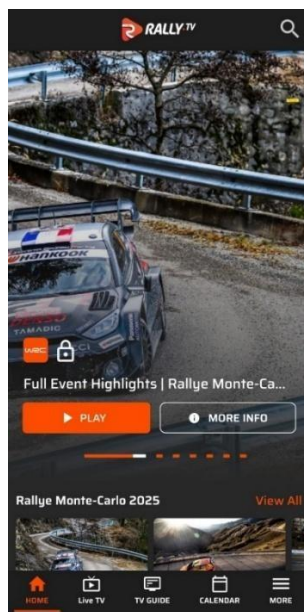


Ilustración 2.1: Pantalla de Inicio de Rally TV

La propuesta de Rally.TV se orienta a la especialización en el segmento del rally, lo que conlleva un enfoque informativo y de entretenimiento. Este factor se traduce en la posibilidad de cubrir distintos aspectos de cada prueba, incluyendo análisis de los equipos e información técnica.

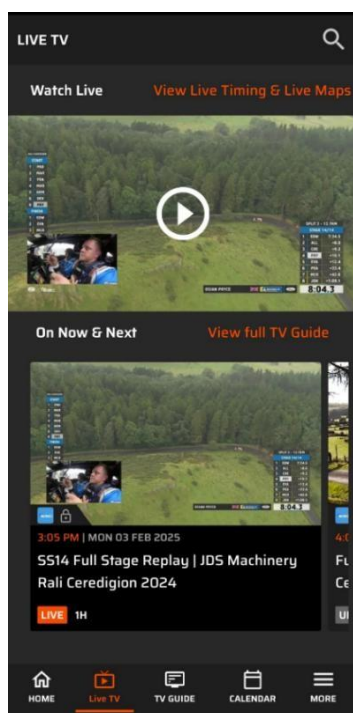


Ilustración 2.2: Pantalla de competiciones en directo de Rally TV

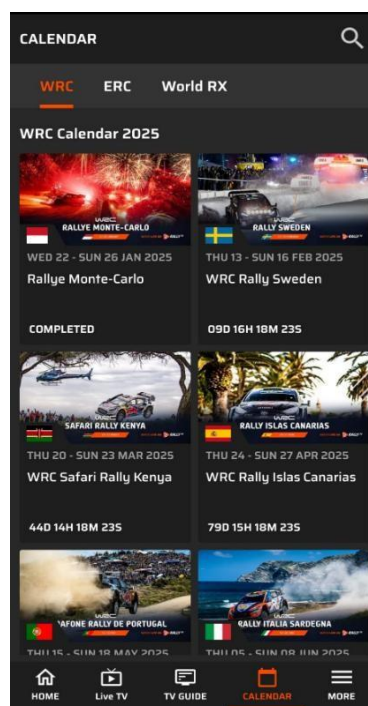


Ilustración 2.3: Pantalla de Calendario de Rallys de Rally TV

CAPÍTULO 2: ESTADO ACTUAL Y OBJETIVOS INICIALES

Rally.TV se encuentra disponible tanto para dispositivos móviles que operan con Android como para aquellos que utilizan iOS [7], así como en versión web accesible desde navegadores de escritorio o portátiles .

Web WRC (FIA World Rally Championship)

La web de WRC brinda noticias de las diferentes carreras que componen el campeonato, así como resultados y clasificaciones en tiempo real. Además, incorpora una sección de calendario que especifica las fechas y ubicaciones de los próximos eventos. El sitio ofrece igualmente la opción de seguir las competencias tanto en directo como bajo demanda mediante su servicio asociado Rally.TV, donde los aficionados pueden ver videos de cada jornada[8].

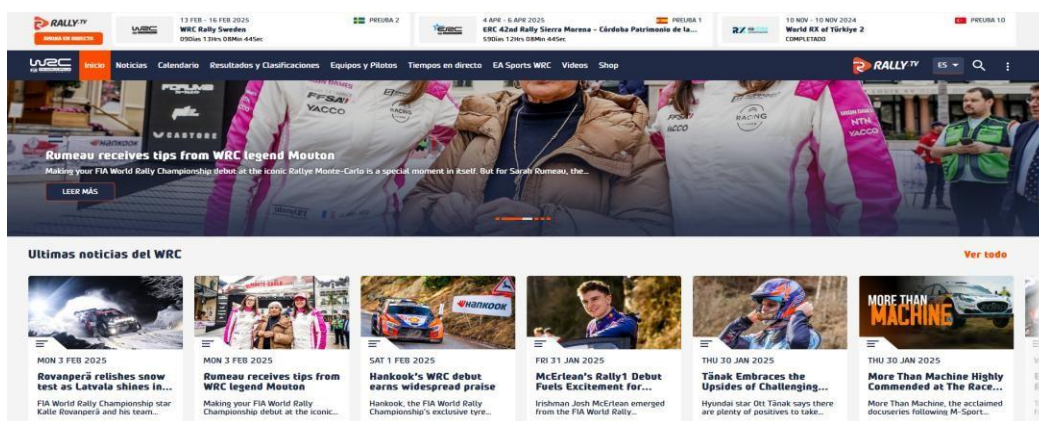


Ilustración 2.4: Pantalla de Inicio de la Web de WRC

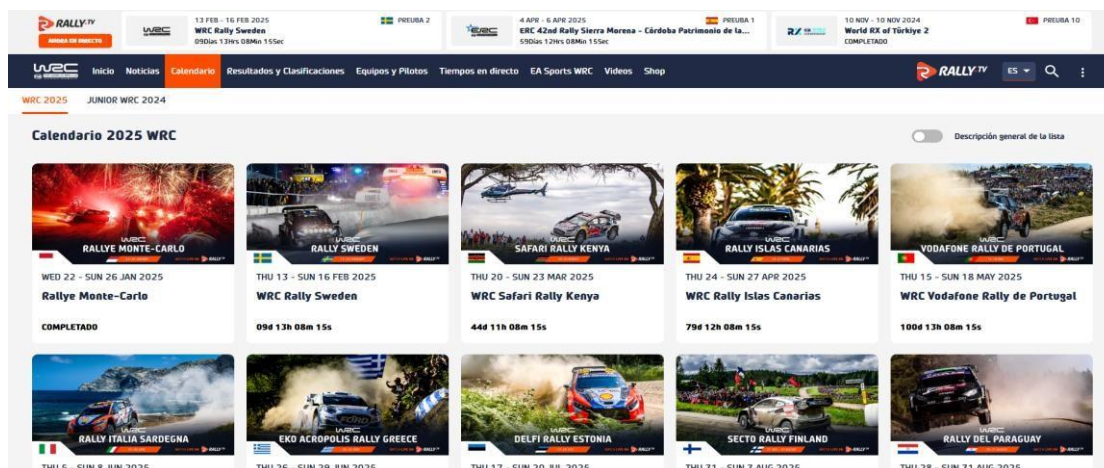


Ilustración 2.5: Pantalla de Calendario de Eventos de la Web de WRC

Dakar Rally

La aplicación oficial del Dakar Rally [1] presenta un diseño orientado a ofrecer a los aficionados una experiencia informativa y dinámica, respaldada por una serie de funcionalidades, entre las que destacan:

- **Transmisiones en Vivo:** A través de la aplicación, los usuarios tienen la oportunidad de seguir la competencia en tiempo real.

CAPÍTULO 2: ESTADO ACTUAL Y OBJETIVOS INICIALES

- **Detalles de Etapas y Competidores:** Proporciona información sobre cada etapa, desde las condiciones del terreno hasta el desempeño de los equipos y pilotos. Además, incluye perfiles y estadísticas de los competidores.



Ilustración 2.6: Pantalla principal de Dakar Rally

En la actualidad, aunque existen diversas aplicaciones que ofrecen diferentes funcionalidades relacionadas con el automovilismo, no se encuentra ninguna que abarque de manera conjunta las siguientes características:

- Consulta de noticias y eventos.
- Acceso a información detallada de la WRC, con pilotos y resultados.
- Foro de interacción para la comunidad.
- Posibilidad de guardar eventos para recibir notificaciones.
- Seguimiento de pilotos favoritos.
- Visualización de galerías de imágenes.
- Calendario personalizado con los eventos guardados.

Para responder a esta necesidad, se ha desarrollado una aplicación que reúne todas estas funcionalidades en un único entorno, con el propósito de fortalecer la experiencia del usuario y contribuir al crecimiento de la comunidad dedicada al rally.

2.3. Objetivos

En el presente proyecto se plantean objetivos generales que harán posible la implementación tanto de la aplicación como de la web:

- **Realizar un estudio de mercado y revisión de aplicaciones similares:** Evaluar plataformas actuales enfocadas al rally con el fin de detectar aspectos mejorables, funcionalidades ausentes y buenas prácticas. Esta información servirá como base para orientar la arquitectura del sistema y definir los puntos críticos de la aplicación.
- **Definir y diseñar las funcionalidades clave:** Establecer los requisitos esenciales del sistema, contemplando tanto la perspectiva de los usuarios que consumirán la información y participarán en la comunidad, como la de los administradores que gestionarán el contenido y los perfiles de los participantes.
- **Conectar la aplicación y el panel de administración a una base de datos en la nube:** Permitir sincronizar de manera simultánea y confiable los datos ingresados desde la aplicación y el panel de administración. El uso de una plataforma en la nube garantiza la disponibilidad de la información y simplifica la escalabilidad del sistema.
- **Desarrollar la aplicación móvil:** Desarrollar una aplicación que permita ser un punto de acceso principal para los aficionados, permitiéndoles consultar información actualizada, organizar sus eventos de interés y participar en foros de discusión. Su diseño y arquitectura deben ser intuitivos y escalables, para adaptarse a futuros requerimientos. Sus funcionalidades serían las siguientes:
 - Registro e inicio de sesión de usuarios
 - Sistema de noticias y calendario de eventos
 - Gestión de pilotos favoritos y guardado de eventos
 - Integración de notificaciones y recordatorios
 - Modo oscuro y apartado multimedia (galería de imágenes y vídeos):
 - Foro interno
- **Implementar el panel de administración web:** Desarrollar una plataforma para que gestores y moderadores mantengan el sistema actualizado y seguro. Contando con una interfaz clara, con controles de edición y supervisión que agilicen la gestión de usuarios y contenido, además de proveer reportes que faciliten la toma de decisiones. Para ello deberá contar con las siguientes funcionalidades:
 - Gestión de usuarios administradores (alta, modificación, baja)
 - CRUD (Crear, Leer, Actualizar, Eliminar) de noticias y eventos
 - Administración de contenidos multimedia
 - Análisis de la aplicación, sección de preguntas frecuentes (FAQ) y mejoras estéticas

Capítulo 3: Aportaciones del trabajo y Competencias específicas

3.1. Aportaciones Socioeconómicas del Proyecto

El proyecto contribuye significativamente a la modernización y dinamización del rally, consolidando su proyección a nivel local, nacional e internacional. La centralización de la información en una única plataforma facilita el acceso a datos para aficionados, organizadores, medios y patrocinadores. Esto favorece un seguimiento más estructurado del deporte, aumentando su visibilidad y atractivo para patrocinadores e inversores, lo que fortalece el financiamiento de eventos y abre nuevas oportunidades de negocio en el sector automovilístico. Este impacto se traduce en los siguientes beneficios clave:

- **Creación de Comunidad y Fortalecimiento del Ecosistema del Rally:** La plataforma fomenta la interacción y participación activa de los aficionados, creando un espacio de intercambio de experiencias y conocimientos. Facilita la promoción de equipos y pilotos, así como la organización de eventos, asegurando una mayor difusión y cohesión dentro del ecosistema del rally. Este enfoque fortalece la comunidad y refuerza la presencia del automovilismo en el entorno digital.
- **Fomento del Turismo Deportivo y Desarrollo Económico Regional:** El sistema actúa como impulsor del turismo deportivo, permitiendo a los aficionados planificar su asistencia a eventos con información detallada sobre fechas, ubicaciones y servicios. Esto beneficia sectores como la hostelería, el transporte y el comercio local, dinamizando la economía en las regiones donde se celebran las competiciones. Además, contribuye a la descentralización del turismo, generando oportunidades económicas para pequeñas y medianas empresas.
- **Innovación y Digitalización en la Gestión del Rally:** La implementación de esta plataforma introduce un modelo innovador de gestión digital, optimizando la administración de eventos, noticias y clasificaciones. Al integrar herramientas avanzadas de análisis y automatización, se mejora la eficiencia operativa y se reduce la carga administrativa para organizadores y gestores.

En conclusión, el proyecto no solo moderniza y optimiza la gestión del rally, sino que también impulsa su crecimiento a través de la digitalización, el fortalecimiento de la comunidad y el fomento del turismo deportivo. La plataforma centraliza la información y mejora la interacción entre aficionados, organizadores y patrocinadores, creando un ecosistema más dinámico y accesible. Además, su impacto en la economía local y regional refuerza la sostenibilidad del rally como un evento deportivo de gran alcance. Con estos avances, se garantiza la evolución del automovilismo en un entorno digital, promoviendo su continuidad y expansión en el futuro.

3.2. Competencias específicas

En relación con la Tabla 3.5 de Competencias Específicas de la rama de informática [17], este proyecto ha integrado diversas capacidades clave en el desarrollo de la aplicación y la plataforma web. La implementación de funcionalidades ha seguido principios fundamentales de fiabilidad, seguridad, calidad y accesibilidad, asegurando que el sistema sea eficiente y escalable. A continuación, se detallan las competencias aplicadas y su impacto en el proyecto.

Competencia CI1: Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente.

La gestión de accesos y la seguridad de la información han sido aspectos prioritarios en el diseño del sistema. Se ha implementado un módulo de autenticación que permite el registro e inicio de sesión, con la asignación de roles y el uso de notificaciones automáticas para optimizar el control de permisos.

Además, se han definido procedimientos de supervisión y mantenimiento de calidad mediante la gestión estructurada de eventos y noticias, asegurando la integridad de la información y la fiabilidad del sistema. Estas prácticas garantizan que los datos almacenados y procesados sean precisos y seguros, cumpliendo con los estándares de calidad exigidos en entornos digitales.

Competencia CI8: Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

Para garantizar un desarrollo seguro y eficiente, se llevó a cabo un análisis detallado de los requisitos funcionales y técnicos del sistema. La aplicación implementa la autenticación de usuarios, la gestión de noticias y eventos, así como la integración con APIs externas para la obtención de datos del Campeonato Mundial de Rally (WRC).

La arquitectura desarrollada asegura la escalabilidad del sistema y facilita el control de accesos y la ejecución de operaciones de creación, lectura, actualización y eliminación dentro del panel administrativo. La selección de tecnologías y paradigmas de programación se realizó priorizando el rendimiento, la modularidad y el mantenimiento del software a largo plazo.

Competencia CI16: Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería del software.

El desarrollo del proyecto ha seguido un enfoque estructurado basado en metodologías de ingeniería del software. Se inició con un análisis de requisitos detallado, definiendo criterios precisos para la implementación de funcionalidades. Posteriormente, se diseñó una arquitectura modular que facilita la separación de responsabilidades y la reutilización de componentes.

Durante la fase de implementación, se desarrollaron las distintas capas del sistema asegurando la trazabilidad entre los requisitos y las funcionalidades construidas. Finalmente, se realizaron pruebas exhaustivas para validar la corrección del software, permitiendo optimizar los componentes y asegurar un funcionamiento estable y confiable en la plataforma.

Competencia CI17: Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.

Se ha priorizado la experiencia del usuario mediante un diseño intuitivo y accesible. Se han organizado las principales secciones de la aplicación, tales como Home, Notificaciones, WRC, Foro y Otros, proporcionando una navegación clara y fluida.

Además, la inclusión de filtros para noticias, la opción de guardar eventos y la gestión de pilotos favoritos facilita la interacción con la aplicación. La barra de navegación estructurada mejora la accesibilidad y garantiza una experiencia óptima para todo tipo de usuarios, independientemente de su nivel de conocimiento tecnológico.

En conjunto, estas competencias han sido aplicadas a lo largo del desarrollo del proyecto, garantizando la calidad del software, su seguridad y su capacidad para ofrecer una experiencia de usuario eficiente e intuitiva.

Competencia TI3: Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, ergonomía y usabilidad de los sistemas

En relación con la Tabla 3.6 de Competencias Específicas en Tecnologías de la Información [17], se ha aplicado un enfoque centrado en el usuario para garantizar la accesibilidad, ergonomía y usabilidad del sistema.

El desarrollo de la aplicación ha estado enfocado en mejorar la experiencia del usuario, incorporando funcionalidades clave que optimizan la interacción y el acceso a la información. Se han implementado notificaciones personalizadas, las cuales permiten a los usuarios recibir avisos sobre eventos próximos, respuestas en el foro y otras actividades relevantes dentro de la plataforma.

Además, se ha desarrollado un foro de discusión, fomentando la participación y la interacción entre los aficionados al rally, proporcionando un espacio donde pueden compartir opiniones y conocimientos sobre el deporte.

Asimismo, se ha integrado un calendario con recordatorios automáticos, que permite a los usuarios visualizar y gestionar eventos de manera eficiente, resaltando las fechas más importantes.

En la parte administrativa, se ha diseñado un panel de control con roles y permisos diferenciados, asegurando que cada usuario con privilegios administrativos tenga acceso únicamente a las funciones necesarias según su perfil, lo que facilita una gestión estructurada y organizada del sistema.

Este enfoque ha permitido garantizar que la aplicación sea accesible, intuitiva y funcional, optimizando tanto la experiencia de los usuarios finales como la eficiencia en la administración de los contenidos y eventos relacionados con el rally.

3.3. Alineamiento con los objetivos de desarrollo sostenible

Los Objetivos de Desarrollo Sostenible (ODS), también llamados Metas Globales, fueron establecidos por las Naciones Unidas en 2015 como un plan de acción global para erradicar la pobreza, preservar el medioambiente y garantizar condiciones de vida dignas y equitativas para todas las personas antes del año 2030 [13].

En el contexto de este Trabajo de Fin de Título (TFT), se ha identificado una alineación significativa con tres de estos objetivos, los cuales se reflejan en la propuesta tecnológica y su impacto en el entorno:

- **ODS 8: Trabajo decente y crecimiento económico**
- **ODS 9: Industria, innovación e infraestructuras**
- **ODS 17: Alianzas para lograr los objetivos**

A continuación, se detalla de manera más extensa cómo la solución propuesta contribuye al logro de cada uno de estos objetivos y por qué el nivel de relación con ellos es alto o medio.

ODS	Grado de relación			
	0 No procede	1 Bajo	2 Medio	3 Alto
1 Fin de la Pobreza	X			
2 Hambre cero	X			
3 Salud y Bienestar	X			
4 Educación de calidad	X			
5 Igualdad de género	X			
6 Agua limpia y saneamiento	X			
7 Energía Asequible y no contaminante	X			
8 Trabajo decente y crecimiento económico				X
9 Industria, Innovación e Infraestructuras				X
10 Reducción de las desigualdades	X			
11 Ciudades y comunidades sostenibles	X			
12 Producción y consumo sostenibles	X			
13 Acción por el clima	X			
14 Vida submarina	X			
15 Vida de ecosistemas terrestres	X			
16 Paz, justicia e instituciones sólidas	X			
17 Alianzas para lograr objetivos			X	

Tabla 3.1: Tabla de grado de relación de los ODS

3.3.1. ODS 8: Trabajo decente y crecimiento económico (Grado de relación: Alto)

El ODS 8 busca promover un crecimiento económico sostenido, inclusivo y sostenible, así como el empleo pleno y productivo y un trabajo decente para todas las personas. La aplicación y la plataforma web desarrolladas en este proyecto inciden de manera directa en este objetivo de varias formas:

- **Impulso económico en el sector deportivo y comercial:**
Al facilitar la organización y difusión de eventos de automovilismo, se generan nuevas oportunidades de negocio para organizadores, patrocinadores y comercios asociados, contribuyendo así a la dinamización de la economía local y regional.
- **Profesionalización y especialización:**
El uso de herramientas digitales para la gestión de competiciones y la difusión de información potencia la profesionalización de la industria deportiva. Esto se traduce en la demanda de perfiles especializados (desarrolladores, administradores de sistemas, expertos en marketing digital, etc.), fomentando la creación de empleo cualificado.

De esta forma, el alto grado de relación con el ODS 8 se justifica por la clara contribución del proyecto al fortalecimiento económico y al impulso de oportunidades de trabajo decente en el ámbito deportivo y comercial.

ODS 9: Industria, innovación e infraestructuras (Grado de relación: Alto)

El ODS 9 promueve la construcción de infraestructuras resilientes, la promoción de la industrialización inclusiva y sostenible, y el fomento de la innovación. El proyecto cumple de manera destacada con este objetivo en los siguientes aspectos:

- **Digitalización y modernización de procesos:**
La adopción de tecnologías web y móviles para la gestión y el seguimiento de las competiciones introduce innovaciones en el sector del rally, impulsando la modernización de sus procesos organizativos y logísticos.
- **Automatización y eficiencia:**
La presencia de funcionalidades automatizadas (notificaciones, registro de participantes, clasificación, entre otras) reduce errores y agiliza trámites operativos, elevando la calidad e innovación en el servicio ofrecido al entorno deportivo.

En consecuencia, el grado de relación alto con el ODS 9 se explica por la contribución significativa a la innovación tecnológica dentro de la industria deportiva, ayudando a establecer infraestructuras digitales más eficientes y actualizadas.

ODS 17: Alianzas para lograr los objetivos (Grado de relación: Medio)

El ODS 17 está orientado a revitalizar la Alianza Mundial para el Desarrollo Sostenible, promoviendo la cooperación entre diversos actores (gobiernos, sector privado, sociedad civil, etc.). La plataforma propuesta se alinea de forma notable con este objetivo, aunque en un grado medio, a través de:

➤ **Fomento de la colaboración:**

Al reunir a organizadores, participantes, aficionados y otros interesados en un espacio digital común, se promueve el intercambio de información, la coordinación de eventos y la cooperación mutua en beneficio del deporte.

➤ **Creación de redes y colaboraciones:**

La posibilidad de compartir datos relevantes, noticias y resultados en tiempo real genera oportunidades para establecer alianzas estratégicas y patrocinios que fortalezcan el ecosistema deportivo.

Aunque la relación no es tan sólida como con los ODS 8 y 9, la aplicación sí promueve la interacción y el trabajo conjunto de múltiples agentes, contribuyendo así al progreso colectivo de la disciplina deportiva y a la consecución de objetivos compartidos.

Capítulo 4: Desarrollo

4.1. Metodología

En las etapas iniciales del proyecto, se adoptó un enfoque en cascada, siguiendo un desarrollo estructurado a través de fases secuenciales como definición de requisitos, diseño, implementación y verificación. Este enfoque proporcionó claridad en la planificación y ejecución del proyecto, sin embargo, presentó ciertas limitaciones a la hora de adaptarse a cambios inesperados en las funcionalidades y en la integración de datos relacionados con el rally.

Para mejorar la capacidad de adaptación ante nuevos requerimientos, se optó por una metodología iterativa incremental, la cual permite construir el sistema de manera progresiva a través de desarrollos sucesivos denominados incrementos. Cada incremento incorpora la implementación, prueba y validación de nuevas funcionalidades, garantizando que el producto sea funcional y utilizable en cada etapa del desarrollo. Este enfoque facilita la retroalimentación continua, permitiendo ajustes y mejoras constantes sin comprometer la estabilidad del sistema, lo que resulta en un desarrollo más flexible, eficiente y alineado con las necesidades del proyecto.

Este enfoque presenta múltiples ventajas que contribuyen a un desarrollo más eficiente y adaptable:

- **Flexibilidad y Adaptabilidad:** Facilita la incorporación de nuevos requerimientos o modificaciones en función de la retroalimentación obtenida, sin necesidad de rehacer el sistema completo, lo que permite una evolución continua del producto.
- **Validación Temprana del Producto:** Al contar con entregables funcionales desde las primeras etapas, se pueden detectar y corregir errores de manera oportuna, minimizando fallos en la versión final y asegurando una mayor calidad del software.
- **Reducción de Riesgos:** La división del desarrollo en incrementos permite una gestión progresiva de los posibles problemas, evitando que estos se acumulen hacia las fases finales del proyecto y garantizando una mayor estabilidad en cada iteración.
- **Mejor Gestión del Proyecto:** La planificación basada en incrementos permite una estimación más precisa de tiempos y recursos, facilitando un seguimiento eficiente del progreso y optimizando la asignación de esfuerzos en cada fase del desarrollo.

4.1.1. Estructura de los Incrementos

Para estructurar el desarrollo de manera eficiente, el proyecto se organizó en tres incrementos, cada uno con objetivos específicos que permitieron una evolución controlada del sistema. Este enfoque progresivo garantizó la validación y optimización de cada versión antes de avanzar a la siguiente, asegurando una mejora continua en la funcionalidad y estabilidad del producto.

Incremento 1: Implementación de la Base Funcional de la Aplicación

En esta primera fase, se estableció la base funcional de la aplicación móvil, garantizando que los aspectos esenciales estuvieran operativos en una versión inicial. Se implementaron funcionalidades fundamentales como el registro e inicio de sesión de usuarios, asegurando un sistema de autenticación seguro con validaciones de credenciales.

Se desarrollaron los módulos de visualización de eventos y noticias, permitiendo a los usuarios acceder a contenido actualizado en tiempo real mediante la integración con la API. Además, se implementó la sección del Campeonato Mundial de Rally (WRC), donde los usuarios pueden consultar clasificaciones, resultados y estadísticas obtenidas directamente desde la API.

Otro elemento clave de este incremento fue la incorporación del foro, una funcionalidad que brinda a los usuarios la posibilidad de interactuar y debatir sobre temas relacionados con el rally.

Para garantizar una base estructurada y escalable, se estableció una arquitectura organizada, asegurando que futuras iteraciones pudieran incorporar nuevas funcionalidades sin afectar la estabilidad del sistema. Este primer incremento sirvió como un punto de partida estable, que permitió realizar pruebas iniciales para evaluar el rendimiento y corregir posibles errores antes de avanzar al siguiente desarrollo.

Incremento 2: Desarrollo de la Plataforma Web para Gestión y Administración

En el segundo incremento, se amplió el alcance del sistema con el desarrollo de una plataforma complementaria para la gestión y administración del contenido. Esta plataforma, diseñada principalmente para administradores y moderadores, permitió la creación, edición y eliminación de noticias y eventos, asegurando que la información en la aplicación móvil estuviera siempre actualizada y correctamente gestionada.

Además, se incluyó un sistema de administración de usuarios, proporcionando herramientas para gestionar cuentas, asignar roles y realizar modificaciones en los permisos de acceso. Se diseñó una interfaz intuitiva y eficiente, asegurando que los administradores pudieran operar de manera rápida y sin complicaciones.

Durante este incremento, también se reforzaron los controles de seguridad y supervisión, asegurando la correcta gestión de la información dentro del sistema. Se implementaron mecanismos para la detección y corrección de errores, optimizando la plataforma antes de su integración definitiva con la aplicación móvil.

Incremento 3: Incorporación de Mejoras y Nuevas Funcionalidades

El tercer y último incremento se centró en la optimización de la experiencia del usuario y la incorporación de nuevas funcionalidades en la aplicación móvil y la plataforma web.

Se implementó un sistema de galería de fotos, permitiendo a los usuarios visualizar imágenes de eventos y competencias, enriqueciendo la experiencia dentro de la aplicación. Además, se añadieron opciones para que los usuarios pudieran guardar eventos y seleccionar pilotos favoritos en la sección del WRC, facilitando el acceso rápido a la información más relevante para cada usuario.

Otro aspecto clave en este incremento fue la incorporación de la gestión del perfil de usuario, permitiendo a los usuarios personalizar su experiencia dentro de la aplicación, administrar sus datos y preferencias.

En la plataforma web, se añadieron herramientas avanzadas para el análisis de datos y reportes, proporcionando a los administradores una supervisión más detallada del rendimiento de la aplicación y la gestión del contenido. Además, se optimizó la comunicación entre la aplicación móvil y la plataforma web, asegurando una sincronización rápida y eficiente de los datos.

Este último incremento permitió consolidar la aplicación y la plataforma web como un sistema completo y funcional, garantizando estabilidad, eficiencia y una experiencia de usuario mejorada. Gracias a este enfoque incremental, el desarrollo del sistema se realizó de manera estructurada, progresiva y adaptable, asegurando flexibilidad para futuras mejoras y actualizaciones.

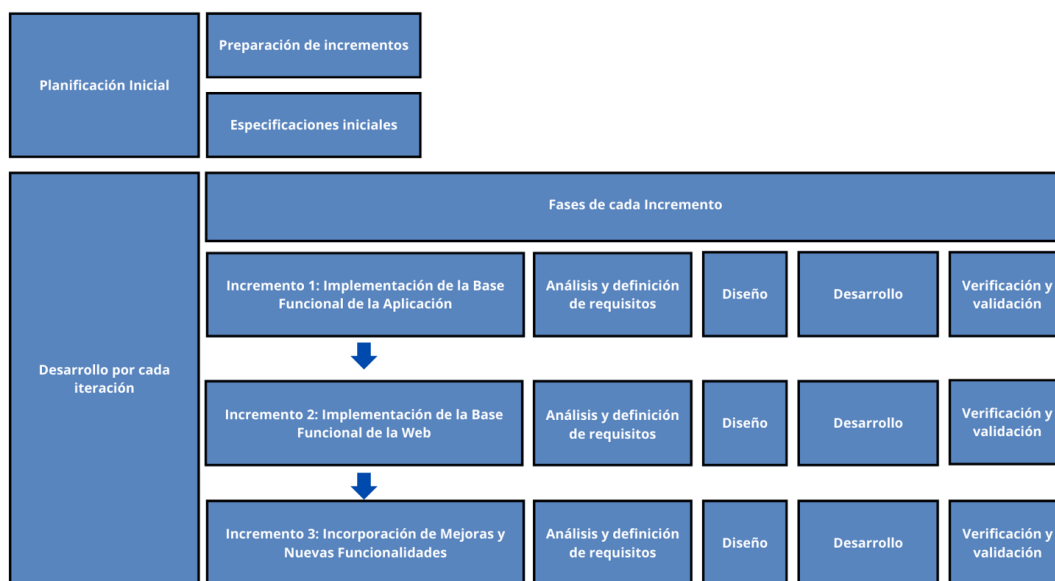


Ilustración 4.1: Diagrama de Incrementos

4.1.2. Fases de cada incremento

El desarrollo de la aplicación se estructuró en cuatro fases fundamentales: análisis y definición de requisitos, diseño, implementación y verificación, organizadas dentro de un enfoque iterativo e incremental. Esta metodología permitió abordar el proyecto de manera estructurada, asegurando que cada versión de la aplicación fuera funcional y se mejorara continuamente antes de avanzar a la siguiente etapa.

1. Análisis y Definición de Requisitos

La primera fase consistió en un estudio detallado de las necesidades del sistema, con el objetivo de identificar y documentar las funcionalidades esenciales, así como los requerimientos técnicos necesarios para garantizar su correcto funcionamiento. Durante esta etapa, se establecieron las bases del proyecto, asegurando que la planificación estuviera alineada con los objetivos definidos y las expectativas de los usuarios. La correcta identificación de estos requisitos permitió estructurar un plan de desarrollo sólido y minimizar posibles desviaciones en etapas posteriores.

2. Diseño

En la segunda fase, enfocada en el diseño, se definió la estructura y organización del sistema, garantizando una solución escalable, eficiente y alineada con las mejores prácticas en desarrollo de software. Durante esta etapa, se establecieron los lineamientos visuales y funcionales del sistema, asegurando una interfaz intuitiva y una integración coherente de todos los elementos de la aplicación. Se diseñaron tanto la arquitectura técnica como la disposición de los distintos módulos, permitiendo optimizar la experiencia del usuario y facilitar la interacción con el sistema.

3. Implementación

En la tercera fase, correspondiente a la implementación, se procedió a la programación y desarrollo de las funcionalidades clave del sistema. Se siguió un enfoque modular, lo que permitió garantizar una integración eficiente y facilitar el mantenimiento del software a largo plazo. Se implementaron los módulos fundamentales, asegurando su correcto funcionamiento individual y su integración con los demás componentes del sistema.

4. Verificación y Validación

Finalmente, en la cuarta fase, se llevó a cabo la verificación y validación, con el propósito de garantizar que el sistema cumpliera con los requisitos establecidos y funcionara correctamente en distintos escenarios.

4.1.3. Gestión y Documentación de Tareas con Kanban y Trello

Para la organización y documentación de las tareas a lo largo del proyecto, se ha utilizado un tablero Kanban en la plataforma Trello. Este sistema permitió estructurar el flujo de trabajo de manera visual, optimizar la gestión de actividades y facilitar la mejora continua del desarrollo. El tablero se estructuró en tres columnas principales, reflejando el estado de las tareas en cada momento:

1. **Tareas a realizar:** Actividades pendientes o por iniciar.
2. **En proceso:** Actividades en desarrollo.
3. **Finalizado:** Tareas completadas con éxito.

Evolución del Tablero Trello

A lo largo del proyecto, el tablero de Trello permitió gestionar y visualizar el progreso de las tareas, lo que ayudó a documentar y organizar el flujo de trabajo de forma clara y eficiente.

1. **Primera Etapa: Inicio del Desarrollo:** En la fase inicial, la mayor parte de las tareas se encontraban en la columna "Tareas a realizar", lo que reflejaba el inicio del desarrollo del proyecto. Solo unas pocas actividades estaban en "En proceso", mientras que la columna "Finalizado" permanecía prácticamente vacía, indicando que aún no se habían completado funcionalidades clave.

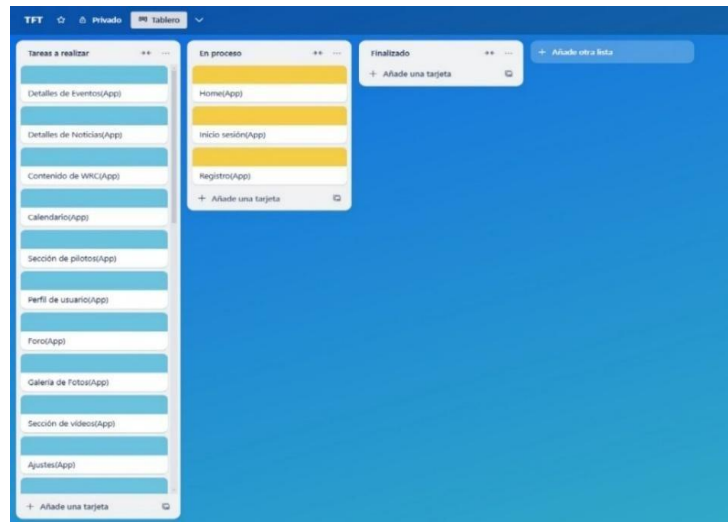


Ilustración 4.2: Progreso Trello 1

2. **Segunda Etapa:** A medida que el proyecto avanzaba, muchas tareas que inicialmente se encontraban en "Tareas a realizar" fueron movidas a "En proceso" y, progresivamente, a "Finalizado". Esto evidenciaba el desarrollo activo de nuevas funcionalidades y la finalización de módulos clave. La reducción de tareas pendientes indicaba un progreso significativo en la ejecución del proyecto.

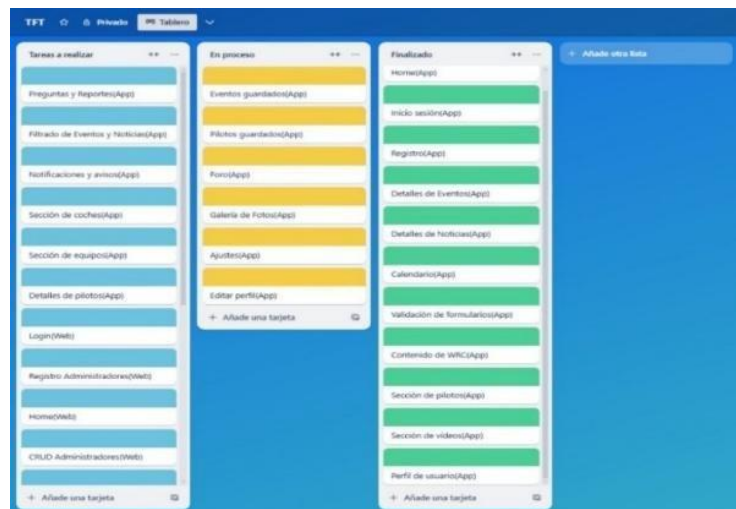


Ilustración 4.3: Progreso Trello 2

3. **Tercera Etapa:** En esta fase, el tablero Trello reflejaba un desarrollo más maduro y equilibrado. Se observó una distribución homogénea de tareas en las columnas "En proceso" y "Finalizado", lo que evidenciaba un avance constante en la implementación tanto de la aplicación móvil como del panel de administración web.

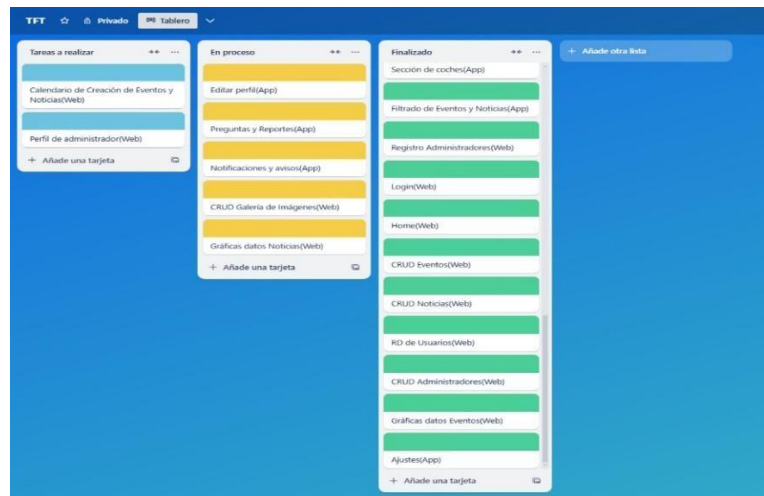


Ilustración 4.4: Progreso Trello 3

4. **Cuarta Etapa:** En la última etapa del desarrollo, la mayoría de las tareas se encontraban en "Finalizado", indicando la conclusión de las principales funcionalidades del proyecto. Solo unas pocas actividades permanecían en "Tareas a realizar" o "En proceso", lo que reflejaba que el desarrollo estaba en su fase final.

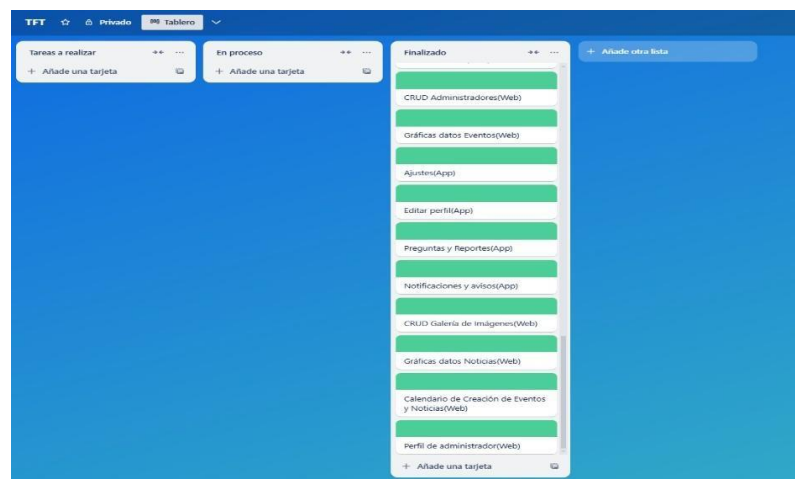


Ilustración 4.5: Progreso Trello 4

El uso de la metodología Kanban y Trello resultó fundamental para la gestión del proyecto, proporcionando orden, visibilidad y control sobre cada tarea. La evolución progresiva del tablero reflejó la transición estructurada entre las distintas fases del desarrollo, asegurando un flujo de trabajo eficiente y una correcta priorización de actividades.

Gracias a esta herramienta, se logró mantener un seguimiento detallado del avance del proyecto, permitiendo adaptaciones ágiles y asegurando la finalización exitosa de la aplicación y la plataforma web.

4.2. Lógica de la aplicación y la web

4.2.1. Flujo de la aplicación

El flujo de la aplicación inicia cuando el usuario accede a la pantalla inicial, donde se evalúa si dispone de una cuenta registrada. En caso de no estar registrado, se le ofrece la opción de crear una nueva cuenta, lo que implica proporcionar y validar sus datos personales. Si el usuario ya cuenta con credenciales, se procede con la autenticación correspondiente para garantizar un acceso seguro.

Una vez completado el registro o validada la identidad, el usuario es dirigido a la página de inicio, donde se encuentra un menú que organiza las secciones principales:

- **Home:** Acceso a noticias relevantes y eventos programados, con la opción de guardar eventos para consulta y seguimiento personal.
- **Notificaciones:** Visualización de avisos relacionados con la actividad del usuario, especialmente interacciones en el foro.
- **WRC:** Consulta de datos proporcionados por una API externa sobre temporadas, rallies, clasificaciones, pilotos y equipos, con la posibilidad de marcar a sus pilotos favoritos.
- **Foro:** Espacio de interacción social para responder, visualizar y crear preguntas o votaciones, con opciones de edición y eliminación de aportaciones.
- **Otros:** Funcionalidades adicionales como gestión de pilotos favoritos, eventos guardados, un calendario, galería de imágenes, vídeos integrados (YouTube), configuración de cuenta y un sistema de preguntas y reportes para soporte.

El diseño del menú asegura que todas las secciones principales estén al mismo nivel, permitiendo al usuario navegar de forma directa entre ellas según sus necesidades.

Algunas acciones específicas, como eliminar un evento o activar notificaciones, requieren confirmación del usuario antes de proceder. Finalmente, la acción de cerrar sesión devuelve al usuario a la pantalla inicial, concluyendo el ciclo de uso del sistema. Este flujo garantiza una experiencia intuitiva, organizada y adaptada a las necesidades tanto funcionales como de interacción del usuario. [\[Ver Anexo I para más claridad\]](#)

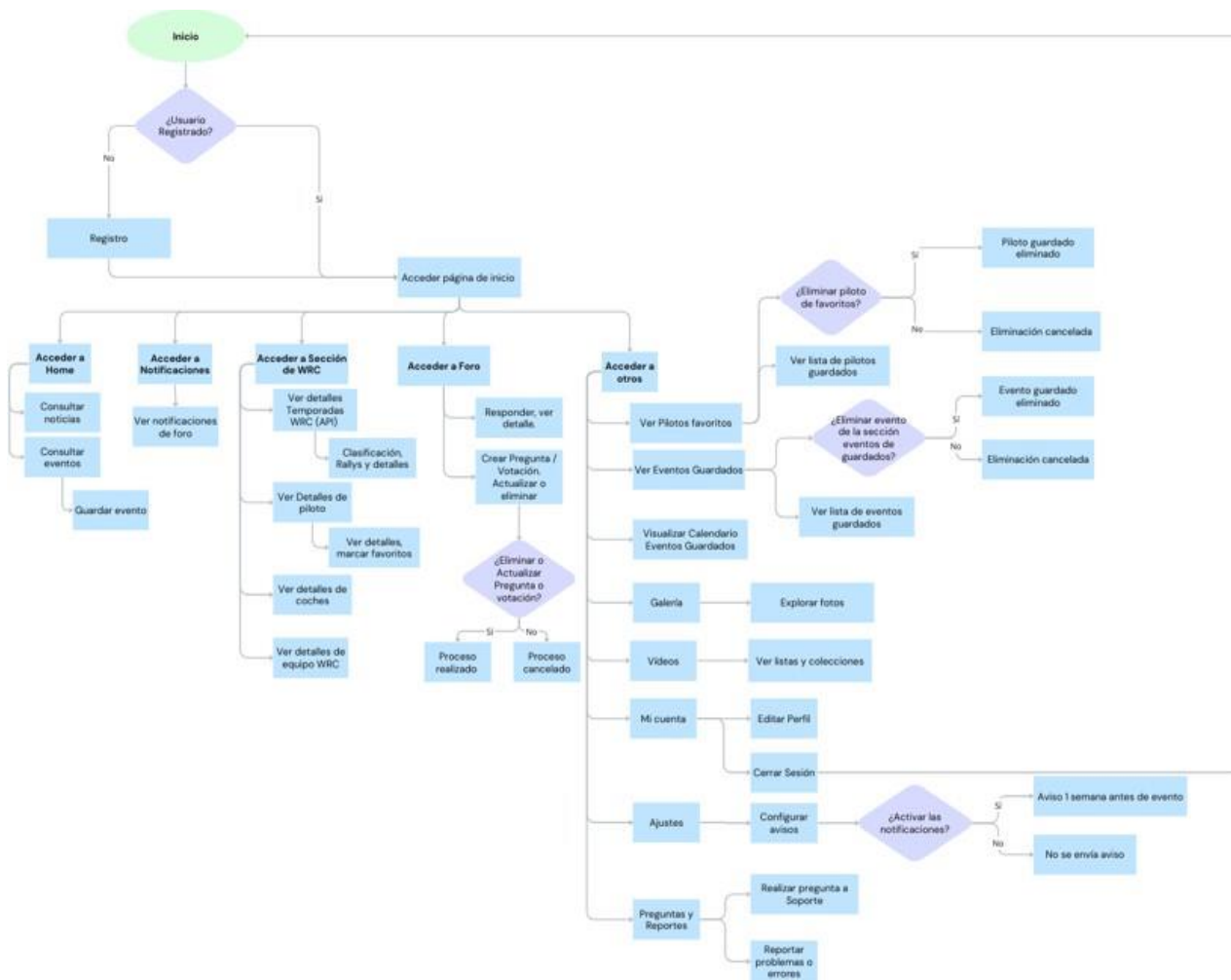


Ilustración 4.6: Diagrama de Flujo de la aplicación

4.2.2. Flujo de la web

El proceso se inicia cuando un administrador intenta acceder al panel de gestión de la plataforma. Inicialmente, el sistema verifica si el usuario posee credenciales válidas. En caso de no tenerlas, el administrador debe solicitar un código temporal al administrador principal. Dicho código, combinado con los datos de registro, habilita la creación de una cuenta administrativa. Si el usuario ya cuenta con un registro previo, puede autenticarse directamente e ingresar al panel principal.

Tras la autenticación, se presenta una interfaz inicial con un menú que organiza las secciones principales del sistema, todas ubicadas al mismo nivel jerárquico, permitiendo la navegación libre entre ellas según las necesidades del administrador. Estas secciones son:

- **Panel de Control:** Muestra un resumen visual con estadísticas clave, como el número de eventos creados, noticias publicadas y usuarios registrados, así como un calendario para consultar las actividades programadas.

- **Gestión de Administradores:** Permite visualizar y gestionar los administradores existentes, incluyendo la asignación o modificación de roles, así como la creación de nuevos administradores mediante un código temporal. También incluye opciones para dar de baja administradores.
- **Gestión de Eventos:** Facilita la creación, edición y eliminación de eventos que serán visibles en la aplicación principal. Los administradores pueden modificar detalles como fechas, ubicaciones y descripciones.
- **Gestión de Noticias:** Ofrece un entorno para publicar, editar o eliminar noticias destinadas a los usuarios, con opciones para gestionar contenido multimedia y categorías.
- **Reportes y Preguntas:** Centraliza las incidencias y consultas generadas por los usuarios en la aplicación principal, permitiendo a los administradores revisar, responder y solucionar problemas reportados.
- **Galería de Imágenes:** Proporciona herramientas para subir, editar o eliminar imágenes que se utilizarán en la aplicación, organizándolas en categorías específicas.
- **Sección "Más":** Incluye la gestión del perfil del administrador, como la actualización de datos personales y la acción de cerrar sesión, que finaliza la interacción y devuelve al usuario al estado previo a la autenticación.

El sistema está diseñado para limitar el acceso a ciertas secciones según el rol asignado al administrador, asegurando que solo quienes tienen los permisos necesarios puedan realizar acciones específicas. Con este enfoque, se garantiza un control eficiente y seguro de las funcionalidades del panel administrativo. [\[Ver Anexo I para más claridad\]](#)

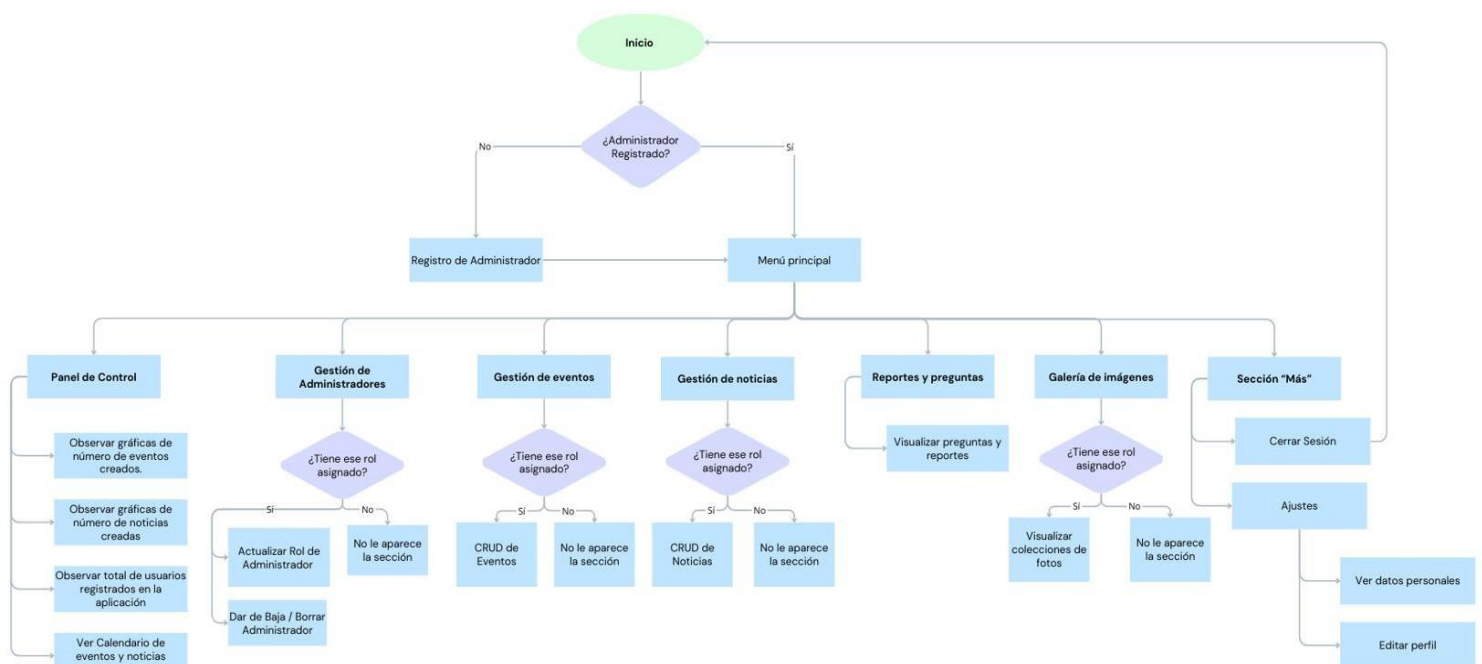


Ilustración 4.7 Diagrama de Flujo de la Web

4.3. Herramientas utilizadas en el desarrollo

4.3.1. Android Studio

Android Studio [9] es el entorno de desarrollo integrado (IDE) oficial diseñado para la creación de aplicaciones en el ecosistema Android. Construido sobre la potente base del editor de código y las avanzadas herramientas de IntelliJ IDEA, este entorno ofrece características adicionales que optimizan el proceso de desarrollo y mejoran la productividad durante la construcción y ejecución de proyectos.



Ilustración 4.8: Logo Android Studio

4.3.2. Visual Studio

Visual Studio Code [18] es un editor de código fuente altamente eficiente y versátil, caracterizado por su diseño ligero y su óptimo rendimiento. Está disponible para sistemas operativos de escritorio como Windows, macOS y Linux, proporcionando una experiencia de desarrollo fluida e intuitiva. Diseñado para simplificar el proceso de programación, ofrece compatibilidad nativa con lenguajes como JavaScript, TypeScript y Node.js. Además, su amplio ecosistema de extensiones permite la integración con múltiples entornos y lenguajes de programación, incluyendo C++, C#, Java, Python, PHP, Go y .NET, ampliando así sus capacidades y adaptabilidad a distintos proyectos.



Ilustración 4.9: Logo Visual Studio

4.3.3. RapidAPI,

RapidAPI [15] es una plataforma líder a nivel mundial para la gestión de APIs, utilizada por muchos desarrolladores y empresas para explorar, probar, integrar y administrar diversas interfaces de programación. Su infraestructura centralizada permite acceder a múltiples APIs a través de una única clave de autenticación y un panel de control intuitivo, facilitando su

monitoreo y administración. Esta herramienta mejora la eficiencia operativa, acelera el desarrollo de aplicaciones y optimiza la integración de servicios externos.



Ilustración 4.10: Logo RapidAPI

4.3.4. Git

Git [10] es un sistema de gestión de versiones distribuido que permite a cada clon local de un proyecto funcionar como un repositorio autónomo y plenamente operativo. Gracias a esta arquitectura, los desarrolladores pueden trabajar sin conexión y mantener un historial completo de modificaciones, lo que facilita la trazabilidad de cambios y la recuperación de versiones previas. Las modificaciones se guardan en el repositorio local a través de confirmaciones, y posteriormente pueden sincronizarse con un repositorio remoto, garantizando una coordinación eficiente en proyectos colaborativos y una organización estructurada del código fuente.



Ilustración 4.11: Logo git

4.3.5. Firebase

Firebase [12] es una plataforma en la nube desarrollada por Google, está orientada a la creación y gestión de aplicaciones móviles y web, brindando compatibilidad con múltiples entornos, como iOS, Android y la web. Su propósito es agilizar el proceso de desarrollo y escalabilidad de las aplicaciones, permitiendo a los desarrolladores centrarse en la funcionalidad sin preocuparse por la infraestructura. Entre sus características más destacadas se encuentran:

- **Gestión de autenticación (Firebase Authentication):** Ofrece un sistema unificado para la administración de usuarios, soportando diversas opciones de acceso, como correo electrónico, redes sociales y proveedores externos. Además, incorpora mecanismos avanzados de seguridad, como la verificación de identidad y la autenticación en dos pasos, garantizando la protección de los datos y el acceso seguro a la aplicación.
- **Almacenamiento en la nube (Cloud Storage):** Proporciona una solución robusta para almacenar archivos de distintos formatos y tamaños de manera segura. Su integración con Firebase facilita la subida, descarga y gestión del contenido, con controles de acceso basados en reglas de seguridad personalizadas para restringir el uso no autorizado.
- **Base de datos en tiempo real (Realtime Database):** Implementa un sistema NoSQL que permite la sincronización instantánea de la información entre dispositivos

conectados. Cualquier actualización en los datos se refleja de inmediato en todos los clientes, lo que lo convierte en una opción ideal para aplicaciones interactivas, como chats o plataformas colaborativas. Su modelo basado en nodos JSON permite organizar y recuperar información de manera eficiente, reduciendo la latencia y mejorando el rendimiento.



Ilustración 4.12: Logo Firebase

4.3.6. Trello

Trello [6] es una plataforma de gestión de proyectos basada en un sistema de tableros, listas y tarjetas, diseñada para facilitar la organización y el seguimiento de tareas en equipo. Su interfaz visual permite estructurar flujos de trabajo de manera intuitiva, adaptándose a distintos tipos de proyectos, desde la planificación de tareas individuales hasta la coordinación de equipos de gran escala.



Ilustración 4.13: Logo Trello

4.3.7. Figma

Figma [2] es una plataforma de diseño colaborativo basada en la nube que permite a equipos y profesionales crear, prototipar y compartir interfaces de usuario de manera eficiente. Su enfoque principal es democratizar el diseño, proporcionando herramientas accesibles para que cualquier persona pueda plasmar sus ideas de forma visual, independientemente de su nivel de experiencia. Gracias a su sistema de trabajo en tiempo real, múltiples usuarios pueden colaborar simultáneamente en un mismo proyecto, optimizando la productividad y fomentando la creatividad colectiva.



Ilustración 4.14: Logo Figma

4.4. Arquitectura de la Base de Datos

Para la gestión de datos en la aplicación, se ha utilizado Firebase Firestore, un sistema de base de datos en la nube que emplea un modelo no relacional basado en colecciones y documentos. Este enfoque proporciona mayor flexibilidad y escalabilidad en comparación con las bases de datos relacionales tradicionales, permitiendo estructuras anidadas y referencias mediante identificadores únicos. La elección de Firebase Firestore se debe a su capacidad de ofrecer sincronización en tiempo real, almacenamiento seguro en la nube y fácil integración con la aplicación, garantizando un acceso rápido y eficiente a los datos desde múltiples dispositivos.

4.4.1. Ventajas de Utilizar Firebase Firestore

El uso de Firebase Firestore aporta múltiples beneficios a la aplicación:

1. **Escalabilidad y Flexibilidad:** La estructura no relacional permite modificar y expandir la base de datos sin restricciones rígidas.
2. **Sincronización en Tiempo Real:** Los cambios en los datos se reflejan instantáneamente en la aplicación sin necesidad de realizar consultas adicionales.
3. **Integración Directa con Otras Herramientas de Firebase:** Se puede combinar con Firebase Authentication para gestionar el acceso de usuarios o con Cloud Storage para almacenar imágenes y archivos.

4.4.2. Estructura y Organización de las Colecciones

La base de datos se ha diseñado con una estructura modular, dividiendo la información en distintas colecciones, cada una con documentos que representan registros individuales. A continuación, se presentan las principales colecciones organizadas por su funcionalidad:

1. Gestión de Contenido

- **events:** Almacena los eventos disponibles en la aplicación, permitiendo la creación, edición y eliminación de eventos. Cada evento contiene información clave como título, fecha, hora, ubicación, tipo (mundial o amateur), imágenes y descripción. Además, permite realizar filtros por fecha y ubicación, facilitando la búsqueda de eventos relevantes.
- **news:** Contiene las noticias publicadas en la aplicación, permitiendo una organización clara para mostrar contenido actualizado a los usuarios. Cada noticia incluye título, imagen, descripción y fecha de publicación.
- **gallery:** Administra la galería de imágenes, organizando el contenido por categorías y fechas. Las imágenes pueden asociarse a eventos específicos, mejorando la presentación visual de la aplicación.

2. Interacción del Usuario

- **votations:** Gestiona el sistema de encuestas y votaciones, permitiendo a los usuarios participar en consultas interactivas dentro de la comunidad. Cada votación incluye opciones de respuesta y un recuento de votos en tiempo real.
- **questions:** Contiene todas las preguntas del foro, permitiendo la interacción entre usuarios. Cada pregunta está asociada a un usuario y contiene respuestas, fecha de publicación y metadatos como la imagen y el nombre del autor.
- **notifications:** Almacena las notificaciones enviadas a los usuarios, como recordatorios de eventos, respuestas en el foro o cualquier actividad relevante dentro de la aplicación.

3. Soporte e Incidencias

- **bugReports:** Recopila reportes de errores enviados por los usuarios, ayudando a priorizar y clasificar los problemas según su gravedad. Cada reporte contiene información detallada como descripción del error, fecha, usuario que lo reportó y nivel de criticidad.
- **questionsAdmin:** Sección dedicada a las preguntas administrativas y de soporte, utilizadas por los administradores para responder dudas o gestionar incidencias dentro de la plataforma.

4. Gestión Administrativa

- **administrator:** Contiene la información de los administradores, incluyendo roles y privilegios para gestionar el acceso a funciones avanzadas. Cada administrador tiene asociado un correo, un identificador único y su respectivo rol dentro del sistema (por ejemplo, "creador de noticias" o "gestor de eventos").
- **adminCodes:** Almacena los códigos de acceso temporales generados para habilitar nuevos administradores. Cada código posee un correo asociado y una fecha de expiración, garantizando mayor seguridad en la asignación de privilegios.

5. Información Específica del Rally

- **wrc_teams:** Contiene los datos de los equipos que participan en el Campeonato Mundial de Rally (WRC), incluyendo nombre del equipo, país de origen, coche utilizado y pilotos asociados.
- **cars:** Almacena información detallada sobre los vehículos utilizados en la competición, incluyendo modelo, potencia, peso y precio.
- **categories:** Clasifica los coches dentro de diferentes categorías según su relevancia y características técnicas, permitiendo una mejor segmentación en la visualización.

6. Gestión de Usuarios

- **users:** Registra la información de los usuarios finales, almacenando nombre, correo, imagen de perfil y preferencias personales, como eventos guardados o pilotos favoritos. Esto permite personalizar la experiencia del usuario dentro de la aplicación.

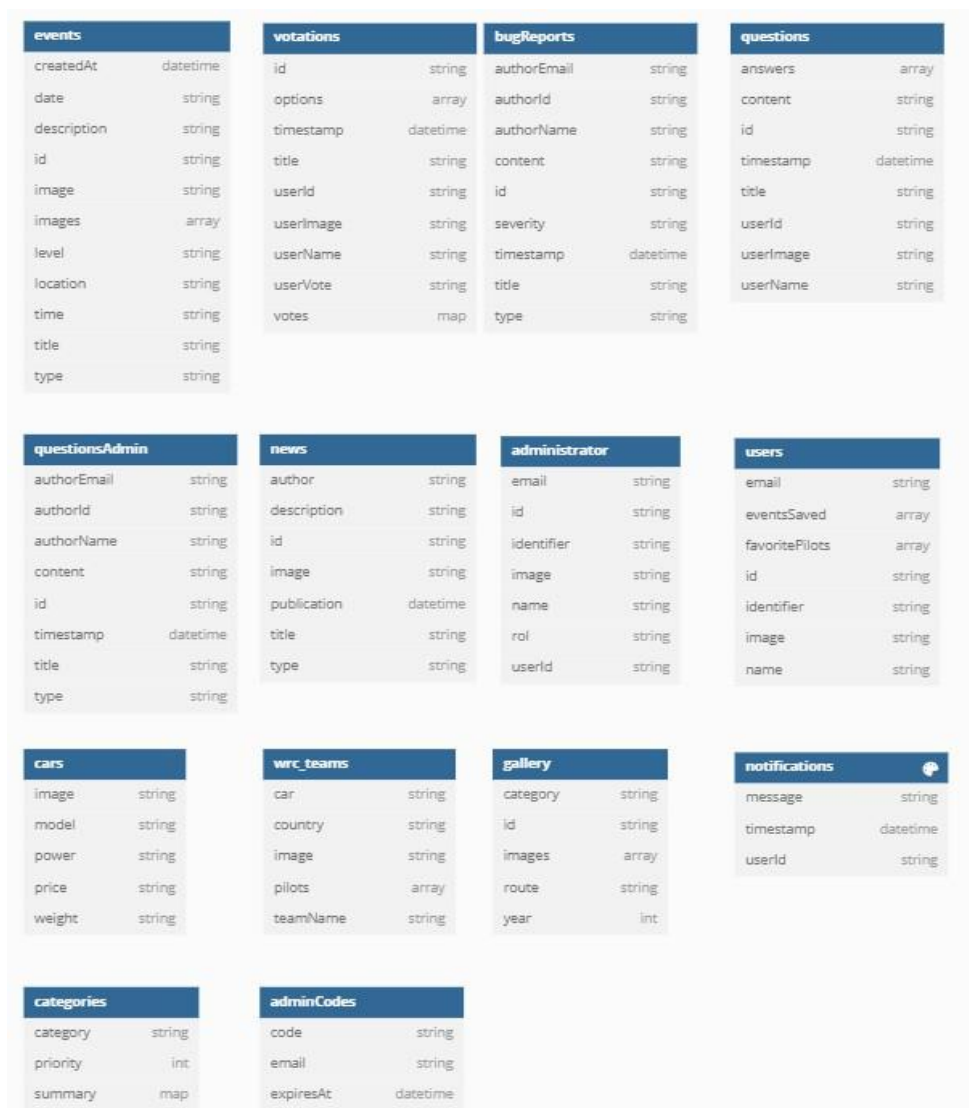


Ilustración 4.15: Diagrama de la base de datos

4.5 Implementación del Patrón MVVM

Para el desarrollo tanto de la aplicación móvil como de la plataforma web, se ha adoptado el patrón arquitectónico MVVM (Model-View-ViewModel) [11], el cual supone una evolución respecto a modelos tradicionales como MVC (Model-View-Controller) y MVP (Model-View-Presenter). Esta metodología permite estructurar el código de manera modular y organizada, estableciendo una separación clara entre los diferentes componentes del sistema.

Gracias a esta segmentación, se mejora la mantenibilidad, se incrementa la escalabilidad del software y se optimiza el proceso de pruebas sin comprometer la estabilidad de la aplicación.

La implementación del patrón MVVM en ambas plataformas ofrece múltiples beneficios que favorecen un desarrollo más eficiente y estructurado:

- **Mayor mantenibilidad:** La diferenciación entre la lógica de negocio y la interfaz de usuario facilita la actualización o modificación de cada componente de forma independiente, sin generar impactos en el resto del sistema. Esto permite realizar mejoras o correcciones sin necesidad de reestructurar toda la arquitectura.
- **Facilidad en la ejecución de pruebas unitarias:** Al mantener la vista separada de la lógica de negocio, el ViewModel puede ser evaluado de manera aislada, sin requerir la interacción directa con la interfaz gráfica. Esto facilita la detección y resolución de errores en la lógica de procesamiento de datos.

MVVM [11] se compone de tres capas principales, cada una con una función específica dentro de la aplicación:

- **Model:** Representa la capa de gestión de datos y lógica de negocio. Es responsable de obtener, procesar y almacenar la información, interactuando con bases de datos o servicios externos como APIs para recuperar o modificar los datos requeridos.
- **View:** Se encarga de la presentación de la información en la interfaz de usuario, mostrando los datos obtenidos del ViewModel y registrando las acciones del usuario. Su independencia de la lógica de negocio permite cambios en la interfaz sin afectar el funcionamiento interno de la aplicación.
- **ViewModel:** Es el intermediario entre el Model y la View. Su función principal es transformar los datos recibidos de la capa de negocio en un formato adecuado para ser mostrado en la interfaz gráfica. Asimismo, gestiona las interacciones del usuario y los eventos generados dentro de la aplicación.

Gracias a la adopción de este patrón, se ha logrado implementar una arquitectura flexible, modular y eficiente tanto en la aplicación móvil como en la plataforma web. Esta organización ha permitido mejorar la estructuración del código, optimizar la gestión del sistema, garantizar una mayor escalabilidad y facilitar la incorporación de nuevas funcionalidades sin comprometer la estabilidad del software.

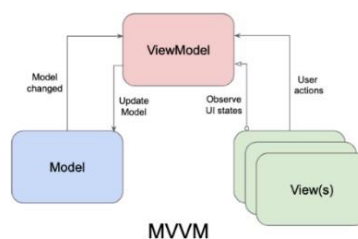


Ilustración 4.16: Diagrama MVVM

4.6 Funcionamiento de la API

Para la integración y obtención de datos en la aplicación, se ha utilizado MotorsportApi [16], una API especializada en proporcionar información en tiempo real sobre diversas competiciones de automovilismo. MotorsportApi está alojada en RapidAPI, lo que facilita su acceso mediante una infraestructura segura y optimizada. Esta API ha sido utilizada para obtener información detallada sobre el Campeonato Mundial de Rally (WRC), incluyendo clasificaciones de pilotos, tiempos por etapa, detalles individuales de cada piloto y su historial de posiciones en competiciones anteriores.

4.6.1 Características de la API

MotorsportApi proporciona una amplia cobertura de eventos de automovilismo y motociclismo, incluyendo MotoGP, Nascar, el Campeonato Mundial de Rally (WRC), Moto2 y Moto3. A lo largo de la temporada, esta API ofrece datos en tiempo real sobre las competiciones, asegurando que los usuarios tengan acceso a información actualizada y precisa.

Entre sus principales funcionalidades se encuentran:

- **Calendario de carreras:** Proporciona un registro detallado de todas las competiciones programadas dentro de la temporada, incluyendo fechas, ubicaciones y horarios de inicio.
- **Clasificaciones en tiempo real:** Permite visualizar la posición actual de cada piloto en el campeonato, mostrando su evolución en cada etapa y los puntos acumulados.
- **Tiempos de cada piloto:** Se registran los tiempos individuales de cada piloto en cada jornada del rally, permitiendo conocer el rendimiento en cada tramo de la competición.
- **Detalles individuales de los pilotos:** Se proporciona información detallada de cada piloto, incluyendo su nacionalidad, equipo, coche y estadísticas de desempeño.
- **Historial de posiciones:** Los datos de MotorsportApi permiten consultar el rendimiento histórico de un piloto, analizando sus posiciones finales en temporadas anteriores.
- **Resultados de cada jornada:** Al finalizar cada carrera, la API actualiza automáticamente los resultados finales y los tiempos obtenidos por cada participante.

4.6.2 Métodos de solicitud y RapidAPI Code Snippet

Para realizar solicitudes a MotorsportApi, RapidAPI proporciona automáticamente un código objetivo (target code) que permite generar la solicitud con las credenciales adecuadas. Esto simplifica la integración con la API en distintos entornos de desarrollo sin la necesidad de configurar manualmente los encabezados de autenticación. De esta manera, los desarrolladores pueden obtener respuestas inmediatas sin preocuparse por la gestión de permisos o seguridad en las peticiones.

Además, RapidAPI permite seleccionar el método de cliente (Client Method) más adecuado según el stack tecnológico utilizado en el proyecto. Entre las opciones disponibles para la integración, se encuentran:

- **Java** (mediante OkHttp o Unirest).
- **Kotlin** (con Retrofit para una gestión optimizada de las solicitudes HTTP).
- **Python** (empleando la librería requests para facilitar la comunicación con la API).

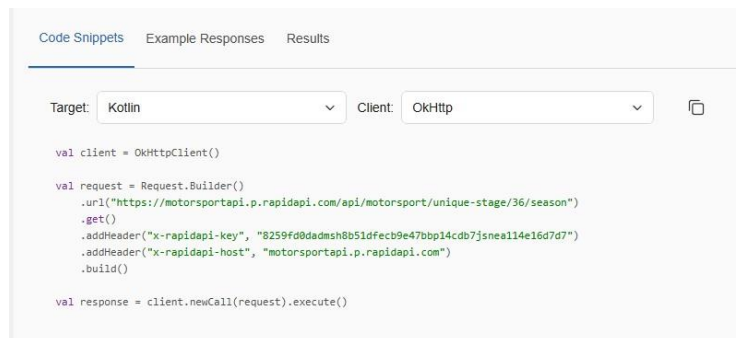


Ilustración 4.17: Método de solicitud

En el caso particular de MotorsportApi, cada tipo de competición dentro de la API posee un código identificador único. Para el Campeonato Mundial de Rally (WRC), el código correspondiente es el 36. Por lo tanto, cualquier solicitud relacionada con esta competición debe incluir dicho código para obtener datos precisos sobre clasificaciones, tiempos de etapa, detalles de pilotos y resultados históricos.

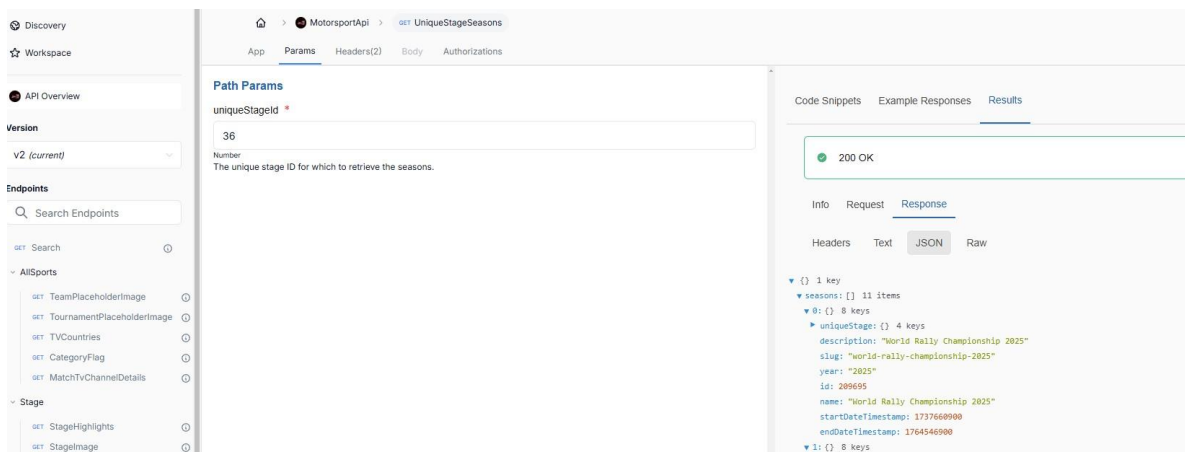


Ilustración 4.18: Código identificador WRC

La integración de MotorsportApi ha permitido dotar a la aplicación de una cobertura en tiempo real y datos detallados sobre el Campeonato Mundial de Rally (WRC). Gracias a esta API, los usuarios pueden acceder a clasificaciones actualizadas, tiempos individuales de cada piloto, información detallada de los competidores y el historial de posiciones en temporadas anteriores.

4.7 Desarrollo de la aplicación móvil

4.7.1. Iniciación y configuración del proyecto en Android Studio

En esta sección se abordará el proceso inicial para el desarrollo de una aplicación móvil utilizando Android Studio, el entorno de desarrollo integrado (IDE) oficial para crear aplicaciones en el ecosistema Android.

Se explicarán los primeros pasos necesarios para configurar un nuevo proyecto, incluyendo la instalación de Android Studio, la creación de un proyecto base y la descripción de las principales herramientas y componentes que proporciona el IDE. Además, se detallarán las configuraciones iniciales esenciales, como la selección del lenguaje de programación, la definición de la versión mínima de Android soportada y la estructura básica del proyecto.

Para crear un proyecto en Android Studio [4], se comienza abriendo el IDE y seleccionando la opción "Create New Project". A continuación, se elegirá una plantilla inicial según el tipo de aplicación que deseas desarrollar, como una actividad en blanco o una basada en Jetpack Compose. Se configurará los detalles básicos, incluyendo el nombre del proyecto, el paquete, el lenguaje de programación (Kotlin) y la versión mínima del SDK soportada.

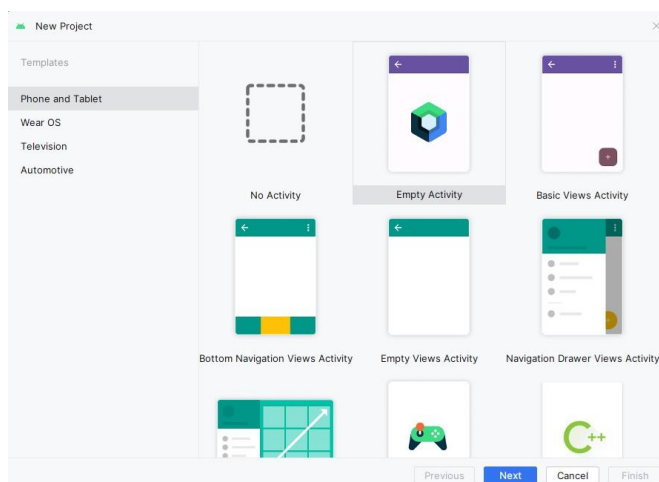


Ilustración 4.19: Creación de Proyecto

Una vez configurados estos parámetros, el proceso finaliza haciendo clic en "Finish", tras lo cual Android Studio generará automáticamente la estructura del proyecto. Este incluye carpetas clave como java para el código fuente, res para los recursos (diseños, imágenes, cadenas de texto) y el archivo AndroidManifest.xml para definir la configuración principal de la aplicación. También se crean archivos Gradle para gestionar dependencias y configuraciones de compilación.

Con el proyecto creado, se puede configurar un emulador desde el AVD Manager o conectar un dispositivo físico para probar la app. Se hará clic en el botón "Run" para compilar y ejecutar la aplicación, que abrirá la pantalla inicial definida en la actividad principal. Este proceso establece la base para desarrollar y ampliar las funcionalidades de la aplicación.

4.7.2. Diseño de la aplicación móvil

A continuación, se presentan las capturas de pantalla de las principales secciones de la aplicación móvil, permitiendo visualizar su diseño y la disposición de los elementos interactivos. Cada pantalla refleja la organización de la plataforma y la navegación entre sus distintas secciones, garantizando una experiencia de usuario estructurada e intuitiva.

Asimismo, se incluyen las capturas correspondientes a las pantallas de inicio de sesión y registro de usuario, donde se puede observar la distribución de los campos de entrada, botones de acceso y opciones adicionales, facilitando la autenticación y la creación de cuentas dentro del sistema.

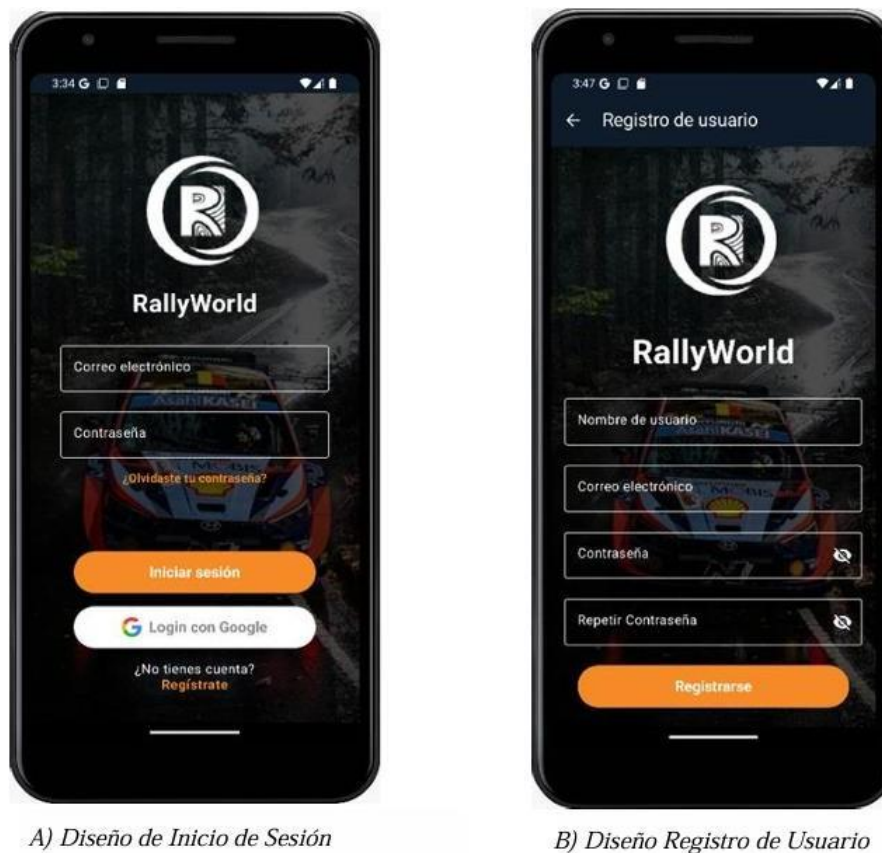


Ilustración 4.20: Diseño Ejemplo 1

Dentro de la aplicación se presentan las principales secciones, destacando un diseño organizado y accesible para el usuario.

La pantalla de inicio (Home) cuenta con una interfaz limpia, con un banner promocional en la parte superior, seguido de un título y una sección de noticias recientes en tarjetas con imágenes y descripciones. En la parte inferior, se muestran los próximos eventos, mientras que la barra de navegación permite un acceso rápido a las secciones principales, resaltando la pestaña activa en color amarillo.

La pantalla del mundial de Rally (WRC) presenta un diseño estructurado con tarjetas visuales que facilitan la navegación. En la parte superior, se encuentra un banner representativo

del campeonato, seguido de secciones diferenciadas para consultar clasificaciones y pilotos, cada una con una imagen ilustrativa y un icono representativo.

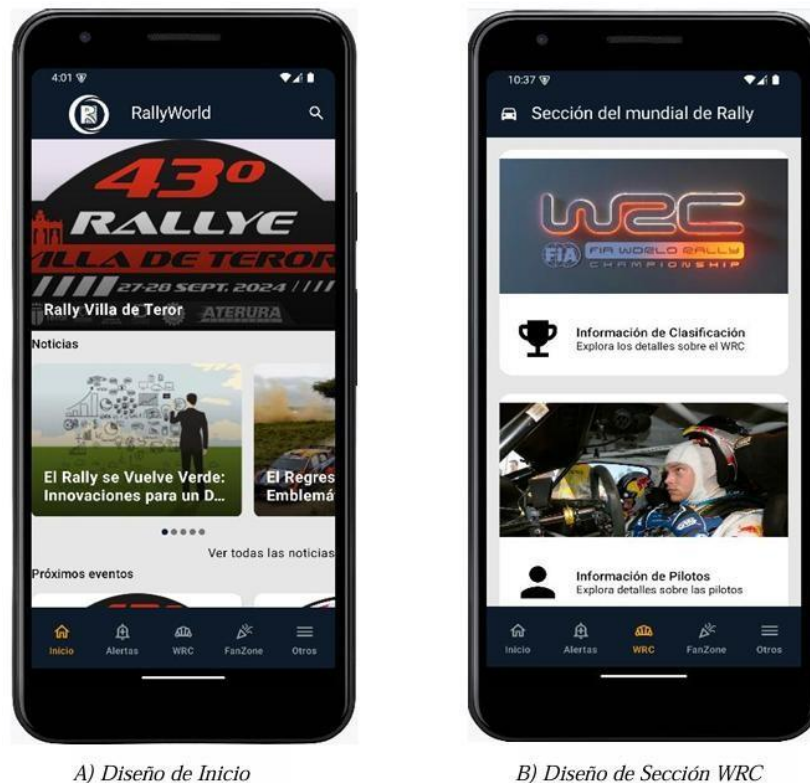


Ilustración 4.21: Diseño Ejemplo 2

Las capturas de pantalla presentadas permiten visualizar la estructura y el diseño de la aplicación, destacando la organización de los elementos interactivos, la paleta de colores y la coherencia estética en las distintas secciones. No obstante, para un análisis más detallado del diseño, distribución y funcionalidades de la interfaz, es posible acceder al prototipo completo en Figma, donde se encuentran todas las vistas y elementos interactivos en su totalidad, seleccionando [aquí](#).

4.7.3. Arquitectura del Backend

La arquitectura del backend de la aplicación móvil se basa en Firebase[3], una plataforma en la nube que proporciona un Backend como Servicio (BaaS), eliminando la necesidad de administrar servidores propios y permitiendo el desarrollo ágil y escalable de la aplicación. Gracias a esta infraestructura, se pueden gestionar la autenticación de usuarios, el almacenamiento de datos en tiempo real y el envío de notificaciones, garantizando un rendimiento eficiente y una sincronización inmediata entre dispositivos.

El backend sigue un modelo basado en eventos, donde Firebase actúa como el núcleo de la infraestructura para gestionar todas las interacciones de la aplicación. Esta arquitectura

CAPÍTULO 4: DESAROLLO

permite optimizar la escalabilidad del sistema, asegurando que cualquier modificación en la base de datos se refleje de inmediato en todos los dispositivos conectados[2].

Uno de los elementos fundamentales de la arquitectura es Firebase Authentication, encargado de la gestión del inicio de sesión y registro de usuarios. Esta funcionalidad permite la autenticación mediante correo electrónico y contraseña, así como la integración con proveedores externos como Google.

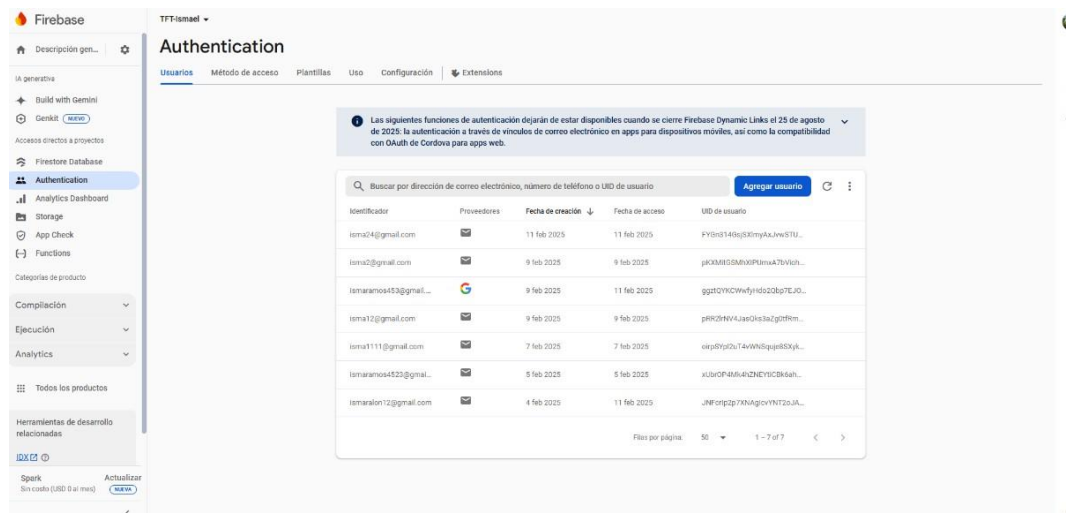


Ilustración 4.22: Firebase Authentication

El almacenamiento y gestión de datos se basa en Cloud Firestore, una base de datos NoSQL que organiza la información en colecciones y documentos estructurados. Este modelo permite realizar consultas escalables y en tiempo real, mejorando significativamente la experiencia del usuario al proporcionar actualizaciones inmediatas en secciones clave como eventos, noticias y foros. La estructura de la base de datos ha sido diseñada de manera segmentada, asegurando una organización eficiente de la información y optimizando las consultas para un mejor rendimiento.

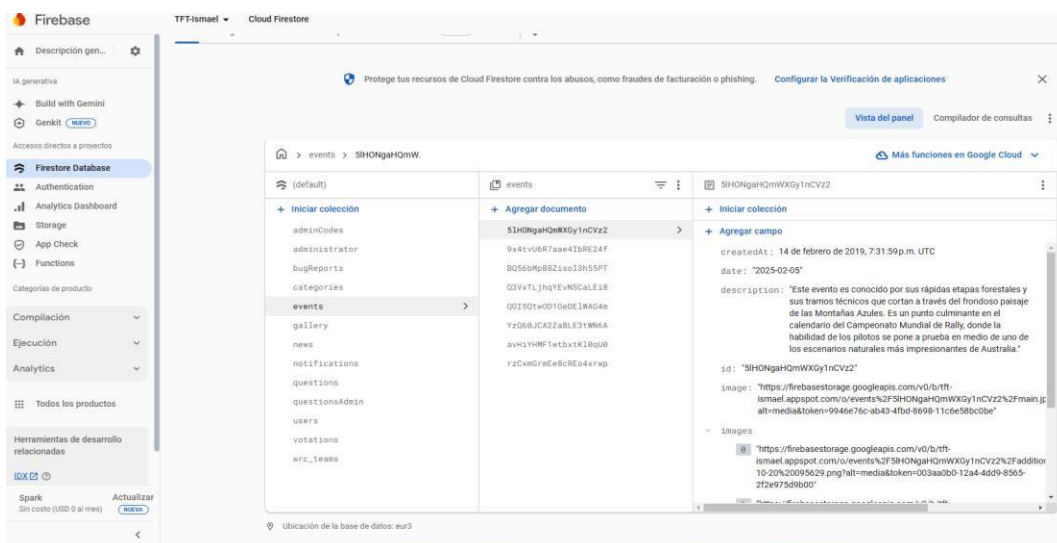


Ilustración 4.23: Cloud Firestore

CAPÍTULO 4: DESAROLLO

Para el manejo de contenido multimedia, la aplicación utiliza Firebase Storage, un sistema que permite almacenar imágenes a los usuarios. Cada archivo se encuentra vinculado a metadatos dentro de Firestore, lo que facilita su organización y acceso eficiente. Este sistema garantiza que los archivos sean gestionados de manera segura y optimizada, asegurando que la carga y recuperación de contenido multimedia se realice sin comprometer el rendimiento general del backend.

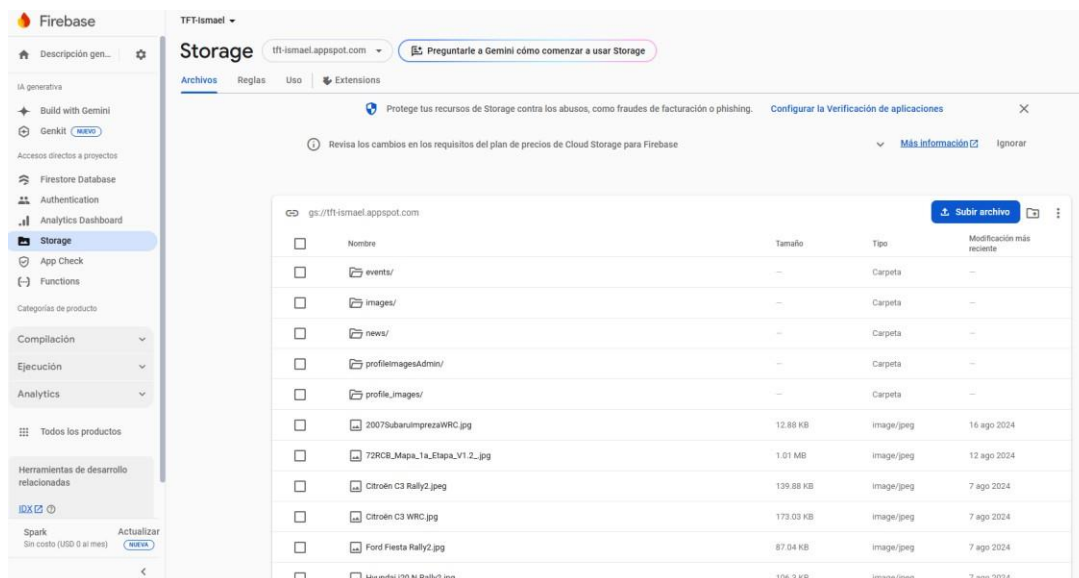


Ilustración 4.24: Firebase Storage

Flujo de Datos en la Arquitectura Backend

El flujo de datos en el backend sigue una estructura definida para garantizar la correcta gestión de la información y su sincronización en tiempo real:

- **Autenticación y acceso:** El usuario inicia sesión a través de Firebase Authentication, validando su identidad mediante tokens de acceso y reglas de seguridad en Firestore.
- **Recuperación de datos:** Se obtiene la información del usuario desde Cloud Firestore para personalizar su experiencia dentro de la aplicación.
- **Consulta y visualización:** La aplicación solicita datos de eventos, noticias y foros desde Firestore, mostrando la información en tiempo real.
- **Interacción del usuario:** Si el usuario comenta en el foro o guarda un evento, la aplicación envía una solicitud de actualización a Firestore, reflejando los cambios de inmediato en todos los dispositivos conectados.
- **Manejo de contenido multimedia:** Las imágenes y videos se almacenan en Firebase Storage, vinculándose con registros en Firestore para una organización y recuperación eficiente.

4.7.4 Integración con la API

La integración de MotorsportApi [16] en la aplicación se ha realizado a través de Retrofit, un cliente HTTP eficiente y optimizado para Android, permitiendo la obtención de datos en tiempo real desde la API alojada en RapidAPI. Esta API proporciona información esencial sobre el Campeonato Mundial de Rally (WRC), incluyendo clasificaciones de pilotos, tiempos por etapa, historial de posiciones y detalles individuales de cada competidor.

Para garantizar una integración óptima, se ha seguido una arquitectura modular que facilita la gestión de las peticiones y la manipulación de datos dentro de la aplicación. A continuación, se detallan los aspectos clave de la implementación.

1. Configuración de Retrofit para la Conexión con MotorsportApi

Para establecer la conexión con MotorsportApi, se ha implementado Retrofit, una biblioteca ampliamente utilizada en Android que permite realizar peticiones HTTP de manera sencilla y eficiente. Su integración en la aplicación permite la obtención y actualización de datos en tiempo real, asegurando una comunicación estable y confiable con la API.

En primer lugar, se define la URL base de la API, que servirá como punto de entrada para todas las solicitudes realizadas desde la aplicación. Esta URL define la dirección del servidor donde se encuentran alojados los datos del WRC y otros eventos automovilísticos. A partir de esta URL base, se pueden realizar diferentes consultas a los endpoints específicos, como clasificaciones, detalles de pilotos o resultados de cada jornada.

Además, es fundamental incluir las cabeceras de autenticación proporcionados por RapidAPI, ya que MotorsportApi requiere una clave de acceso para validar la identidad del cliente. Estas cabeceras garantizan que solo los usuarios autorizados puedan acceder a la información, protegiendo los datos y evitando solicitudes no autorizadas. Las cabeceras de autenticación esenciales son:

- **La clave de acceso única:** permite autenticar la aplicación y validar los permisos para consumir la API.
- **El host:** asegura que las solicitudes sean dirigidas correctamente al servidor adecuado.

Para garantizar un manejo adecuado de las solicitudes HTTP y mejorar la seguridad de la comunicación con la API, se utiliza OkHttpClient, un cliente HTTP que permite agregar interceptores para el control de errores y la autenticación.

Finalmente, se construye la instancia de Retrofit utilizando el Retrofit Builder, lo que permite definir la configuración de las solicitudes y establecer un convertidor encargado de transformar las respuestas en formato JSON en objetos de Kotlin de manera automática. Esto simplifica la manipulación de los datos en la aplicación, ya que se pueden trabajar directamente con estructuras de datos en lugar de procesar respuestas en texto plano.

Gracias a esta configuración, la aplicación puede realizar solicitudes a MotorsportApi de manera eficiente, asegurando que la información obtenida sea precisa y se mantenga actualizada.

```
object RetrofitInstance {
    private const val BASE_URL = "https://motorsportapi.p.rapidapi.com/api/motorsport/"

    private val loggingInterceptor = HttpLoggingInterceptor().apply { this: HttpLoggingInterceptor
        level = HttpLoggingInterceptor.Level.BODY
    }

    val client = OkHttpClient.Builder()
        .addInterceptor(loggingInterceptor)
        .addInterceptor { chain ->
            val request = chain.request().newBuilder()
                .addHeader( name: "x-rapidapi-key", value: "key")
                .addHeader( name: "x-rapidapi-host", value: "motorsportapi.p.rapidapi.com")
                .build()
            chain.proceed(request) ^addInterceptor
        }
        .build()

    val api: WrcApiService by lazy {
        Retrofit.Builder()
            .baseUrl(BASE_URL)
            .client(client)
            .addConverterFactory(GsonConverterFactory.create())
            .build()
            .create(WrcApiService::class.java)
    }
}
```

Ilustración 4.25: Código de configuración de retrofit

2. Definición de Endpoints para Obtener Datos del WRC

En la aplicación, se ha definido la interfaz `WrcApiService` para gestionar la comunicación con MotorsportApi utilizando Retrofit, un cliente HTTP eficiente para Android. Esta interfaz establece los métodos que permiten realizar solicitudes a la API y obtener información detallada sobre el Campeonato Mundial de Rally (WRC). Cada método corresponde a un endpoint específico, lo que facilita una integración estructurada y optimizada con la API.

Uno de los principales métodos implementados permite obtener todas las etapas de la temporada, utilizando un código de identificación único para el WRC, que en este caso es el 36. A través de este endpoint, la aplicación recupera información sobre cada fase del campeonato, asegurando que los usuarios tengan acceso a un calendario completo y actualizado. De manera complementaria, existe un método que permite obtener la imagen representativa de cada temporada, proporcionando una mayor personalización visual en la interfaz de usuario.

Además, la API ofrece un método para recuperar las sub-etapas de una etapa específica, permitiendo desglosar el rally en secciones más detalladas. También se incluye la funcionalidad para consultar la clasificación de una etapa en particular, mostrando la posición actual de los pilotos dentro de la competencia. Ambos endpoints requieren el identificador de la etapa correspondiente para realizar la solicitud.

Otro aspecto fundamental es la obtención de información detallada sobre los pilotos. Para ello, se ha implementado un método que permite consultar los datos individuales de cada piloto, incluyendo su equipo, estadísticas y desempeño en la temporada. Asimismo, se dispone de un endpoint que proporciona el historial de carreras de un piloto en una temporada específica, permitiendo a los usuarios analizar su rendimiento a lo largo del campeonato.

```
interface WrcApiService {
    // Obtener todas las etapas
    @GET("unique-stage/36/season")
    suspend fun getSeasons(): SeasonsResponse

    @GET("unique-stage/{seasonId}/image")
    suspend fun getSeasonImage(@Path("seasonId") seasonId: String): ImageResponse

    // Obtener substages de una etapa específica
    @GET("stage/{stageId}/substages")
    suspend fun getSubstages(@Path("stageId") stageId: String): SubstagesResponse

    // Obtener los standings de una etapa específica
    @GET("stage/{stageId}/standings/competitor")
    suspend fun getStandings(@Path("stageId") stageId: String): RallyStandingsResponse2

    @GET("team/{pilotId}")
    suspend fun getPilotDetail(@Path("pilotId") pilotId: Int): PilotDetailResponse

    @GET("team/{id}/stage/season/{seasonId}/races")
    suspend fun getPilotRaces(
        @Path("id") id: Int,
        @Path("seasonId") seasonId: Int
    ): RaceResponse
}
```

Ilustración 4.26: Definición de Endpoints

Todos estos métodos están diseñados para interactuar con Retrofit, que facilita la conversión de respuestas JSON en objetos Kotlin, optimizando el manejo de datos en la aplicación. La arquitectura de la API permite obtener información en tiempo real sobre las clasificaciones, tiempos y detalles de cada piloto, garantizando una experiencia de usuario fluida y actualizada. Esta integración asegura que la aplicación disponga de datos precisos y actualizados sin necesidad de almacenamiento local, mejorando así la eficiencia y escalabilidad del sistema.

4.7.5. Estructura de ficheros

La estructura de ficheros del proyecto se ha organizado de manera modular para favorecer la separación de responsabilidades, lo que facilita tanto el desarrollo como el mantenimiento del sistema [5]. En el directorio principal de código, ubicado en *TFT\app\src\main\java\com\example\tft*, se encuentran varias carpetas que agrupan funcionalidades específicas de la aplicación.

En primer lugar, el directorio *data* contiene, entre otros, el subdirectorio *api*, en el que se alojan todos los archivos relacionados con la realización de llamadas a la API y el procesamiento de los datos obtenidos, centralizando así la lógica de integración y el manejo de datos externos.

Adicionalmente, se dispone del directorio *services*, que reúne todos los servicios encargados de la conexión y la gestión de la comunicación con la base de datos. Junto a este, el directorio *worker* se dedica a gestionar el sistema de notificaciones automáticas. Al mismo nivel que el directorio *data*, se encuentra el directorio *model*. En este se ubican las clases de datos (dataclasses) correspondientes a diversas entidades de la aplicación, tales como etapas, foros, pilotos, preguntas, reportes, usuarios, vídeos, y otros elementos relacionados con la API. Este nivel de abstracción permite encapsular de forma ordenada la lógica de negocio y la estructura de datos.

El directorio *navigation*, también situado al mismo nivel que *data*, se encarga de definir el flujo y la lógica de navegación dentro de la aplicación. Por otro lado, el directorio *templates_App* agrupa componentes reutilizables que se utilizan de manera frecuente en diferentes partes de la aplicación, asegurando la coherencia visual y funcional en toda la interfaz.

El directorio *ui* es el repositorio de los elementos visuales, albergando tanto los archivos correspondientes a las distintas pantallas (Screen.kt) de la aplicación como los respectivos ficheros *ViewModel*, que se encargan de gestionar la lógica de la interfaz. Esta separación facilita la implementación de patrones arquitectónicos como MVVM (Model-View-ViewModel), garantizando una clara distinción entre la presentación y la lógica de negocio.

El archivo *MainActivity*, situado al mismo nivel que la carpeta *data*, actúa como punto de entrada de la aplicación, coordinando la interacción entre los distintos componentes.

Fuera del directorio de código, en *TFT\Proyecto\TFT\app\src\main\res*, se ubican los recursos de la aplicación. El directorio *drawable* contiene las imágenes e iconos utilizados, mientras que los directorios *mipmap*, *values* y los archivos XML almacenan otros recursos gráficos y de configuración que definen la apariencia y el comportamiento de la interfaz. Finalmente, en el directorio *Gradle Scripts* se localizan todos los ficheros de configuración del proyecto, responsables de gestionar las dependencias y los ajustes de compilación, fundamentales para el correcto funcionamiento y despliegue del sistema.

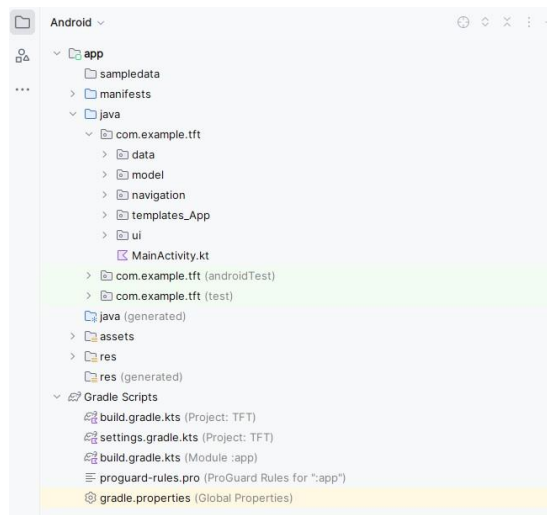


Ilustración 4.27: Estructura de ficheros de la aplicación

Cabe destacar que esta organización del proyecto se fundamenta en el patrón arquitectónico MVVM, que estructura los archivos en paquetes según su funcionalidad y ámbito de aplicación. Esta disposición modular no solo facilita la gestión eficiente del código, sino que también garantiza la separación de responsabilidades, contribuyendo al mantenimiento y la escalabilidad del sistema.

Cada directorio representa un módulo funcional dentro de la aplicación, agrupando tanto la vista (Screen.kt) como su correspondiente ViewModel, encargado de gestionar la lógica de la interfaz. Este enfoque evita la mezcla entre la capa de presentación y la lógica de negocio, asegurando un desarrollo más estructurado y organizado.

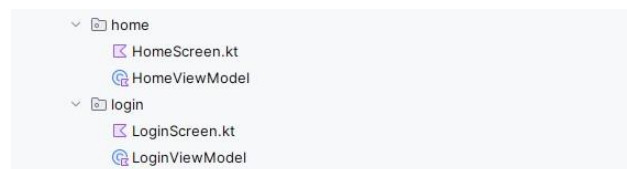


Ilustración 4.28: Estructura fichero basados en MVVM

El diseño modular del proyecto distribuye sus funcionalidades en directorios independientes, lo que permite mantener un código limpio, facilita la reutilización de componentes y posibilita la expansión del sistema sin comprometer su estabilidad general. Un principio clave en esta organización es la clara separación entre la interfaz de usuario y la lógica de negocio. Cada vista se encarga exclusivamente de la representación gráfica e interacción con el usuario, mientras que el ViewModel gestiona el procesamiento de datos y sincroniza la información con la interfaz, evitando la sobrecarga de responsabilidades en una única capa y mejorando la eficiencia del desarrollo.

Desde una perspectiva de mantenimiento y pruebas, esta estructura permite realizar modificaciones o correcciones de errores de manera localizada, sin afectar otras partes del código. Además, la independencia entre la vista y el ViewModel facilita la realización de pruebas unitarias, asegurando la calidad del software sin depender de la interfaz gráfica

Por último, esta organización modular favorece la escalabilidad del sistema, permitiendo la integración de nuevas funcionalidades sin alterar la arquitectura existente. La implementación de nuevas pantallas sigue un esquema estándar, manteniendo coherencia y asegurando buenas prácticas de desarrollo. En definitiva, este enfoque garantiza un código más eficiente, organizado y fácil de mantener a largo plazo.

4.7.6 Funcionalidades

A continuación, se presentan y describen las principales funcionalidades de la aplicación. Cada una de estas funciones ha sido diseñada para optimizar la experiencia del usuario, garantizar un acceso seguro a la información y mejorar la interacción dentro de la plataforma.

Inicio de Sesión

El sistema de autenticación de la aplicación ofrece múltiples métodos de acceso, garantizando una experiencia de usuario flexible y segura. Cuenta con las siguientes funcionalidades:

- **Métodos de autenticación:** El usuario puede iniciar sesión mediante correo electrónico y contraseña o utilizar la opción de inicio de sesión con Google, agilizando el acceso sin necesidad de recordar credenciales adicionales.
- **Sesión persistente:** Se implementa un sistema que permite mantener la sesión iniciada en el mismo dispositivo, evitando que el usuario deba autenticarse repetidamente.
- **Seguridad optimizada:** La autenticación está protegida con medidas de seguridad que garantizan la privacidad de los datos del usuario y previenen accesos no autorizados.

Este sistema de inicio de sesión facilita un acceso rápido, seguro y eficiente, optimizando la experiencia de uso dentro de la aplicación.

Registro de usuario:

El proceso de registro de usuario ha sido diseñado para garantizar una experiencia sencilla y segura, asegurando la correcta identificación dentro del sistema. Dispone de las siguientes funcionalidades:

- **Campos obligatorios:** Se requiere que el usuario proporcione nombre, correo electrónico y contraseña, permitiendo una identificación única en la plataforma.
- **Requisitos de seguridad:** Para fortalecer la protección de la cuenta, la contraseña debe contener al menos seis caracteres, incluyendo una letra mayúscula y un número.
- **Interfaz optimizada:** El formulario de registro ha sido diseñado para ofrecer una experiencia intuitiva y rápida, facilitando la creación de cuentas en pocos pasos.



A) Inicio de Sesión



B) Registro de Usuario

Ilustración 4.29: Acceso de Usuarios

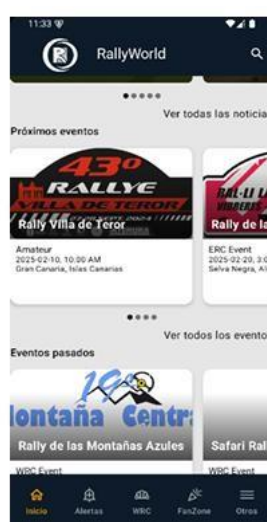
Sección de pantalla de inicio

La pantalla de inicio ha sido diseñada para proporcionar una experiencia organizada e intuitiva, permitiendo a los usuarios acceder de manera rápida a la información más relevante. Incluye las siguientes funcionalidades:

- **Sección de noticias:** Se presentan las últimas publicaciones sobre el mundo del rally, facilitando la consulta de información actualizada y asegurando que el usuario se mantenga informado.
- **Sección de eventos:** Se ofrece una lista de eventos recientes y pasados, organizada de manera clara, permitiendo a los usuarios explorar competiciones anteriores y futuras de manera sencilla.



A) Pantalla de Inicio 1



B) Pantalla de Inicio 2

Ilustración 4.30: Pantalla de Inicio

Sección de Eventos

La sección de eventos proporciona a los usuarios una visión completa y detallada de cada competición programada, facilitando el acceso a información clave y opciones de gestión personalizadas. Dispone de las siguientes funcionalidades:

- **Detalles del evento:** Se muestra información esencial como la hora, fecha de celebración, ubicación y una descripción detallada, permitiendo a los usuarios conocer todos los aspectos relevantes.
- **Opciones de seguimiento:** Se ofrece la posibilidad de guardar eventos de interés, permitiendo un acceso rápido para su consulta posterior y facilitando la planificación personal.

Estas funcionalidades permiten a los usuarios gestionar y seguir de cerca los eventos más relevantes dentro del mundo del rally, asegurando una experiencia organizada y accesible.



A) Detalles de evento



B) Alerta de Evento Guardado

Ilustración 4.31: Eventos

Sección de Noticias

Esta sección permite a los usuarios acceder a información actualizada sobre eventos, competiciones y novedades del mundo del rally, garantizando un acceso rápido y estructurado a los artículos publicados. Cuenta con las siguientes funcionalidades:

- **Acceso a publicaciones:** Los usuarios pueden consultar artículos informativos sobre el desarrollo de competiciones, análisis de pilotos y noticias relevantes del sector.
- **Diseño optimizado:** La interfaz está diseñada para ofrecer una lectura fluida y agradable, con un formato claro que facilita la navegación entre diferentes publicaciones.
- **Contenido estructurado:** La organización visual permite acceder de manera sencilla a las noticias más recientes, asegurando que el usuario se mantenga informado sobre las novedades del rally.



Ilustración 4.32: Detalles de Noticia

Sección de Notificaciones

El sistema de notificaciones del foro está diseñado para mejorar la interacción y participación de los usuarios dentro de la comunidad, asegurando que se mantengan informados sobre nuevas respuestas e interacciones en sus publicaciones. Incluye las siguientes funcionalidades:

- **Avisos automáticos:** Los usuarios reciben notificaciones en tiempo real cuando alguien responde a sus preguntas, permitiendo un seguimiento inmediato de las conversaciones.
- **Actualización continua:** Las notificaciones se generan de manera automática y sin retrasos, evitando que los usuarios deban revisar manualmente el foro para verificar nuevas respuestas.

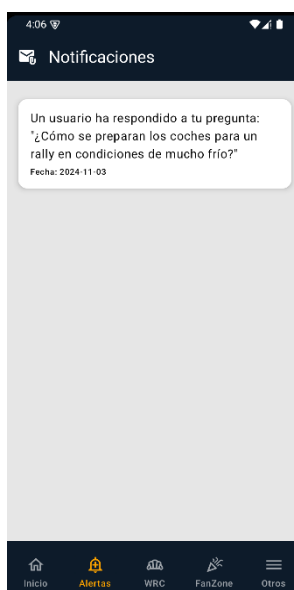


Ilustración 4.33: Notificaciones de Foro

Sección WRC

Esta sección agrupa toda la información relevante sobre el Campeonato Mundial de Rally, permitiendo a los usuarios acceder a datos detallados y actualizados. Dentro de esta sección, se incluyen diversas subsecciones que ofrecen una visión completa de la competición.

En los apartados siguientes, se describen las funcionalidades clave de cada subsección, como la clasificación en tiempo real, donde se pueden consultar las posiciones actualizadas de los pilotos, la información detallada de los competidores, incluyendo sus estadísticas y rendimiento, la sección de coches, que presenta las especificaciones técnicas de los vehículos utilizados en el campeonato, y finalmente, la sección de equipos, donde se muestran los datos de cada escudería y sus pilotos asociados.

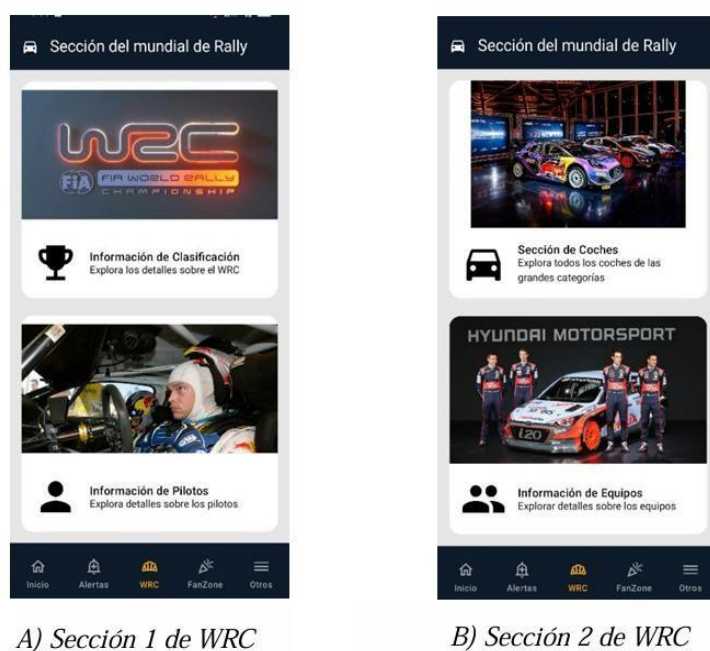


Ilustración 4.34: Sección del mundial de Rally

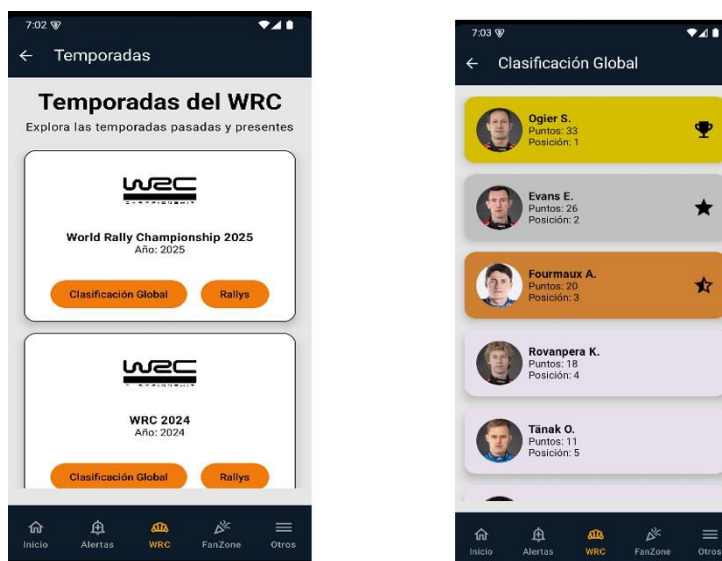
Información de Clasificación

El sistema de clasificación permite a los usuarios consultar la posición actual de cada piloto en el campeonato, basado en los puntos acumulados a lo largo de la temporada. Incluye las siguientes funcionalidades:

- **Clasificación general:** Se muestra la posición actual de cada piloto en la tabla del campeonato, con detalles sobre sus puntos acumulados.
- **Resultados por jornada:** Se pueden consultar los resultados diarios de cada etapa, permitiendo un seguimiento detallado del desempeño de los competidores.
- **Actualización en tiempo real:** Al estar integrada con una API externa, la información se actualiza automáticamente, garantizando datos precisos y continuos a lo largo de la competición.

CAPÍTULO 4: DESAROLLO

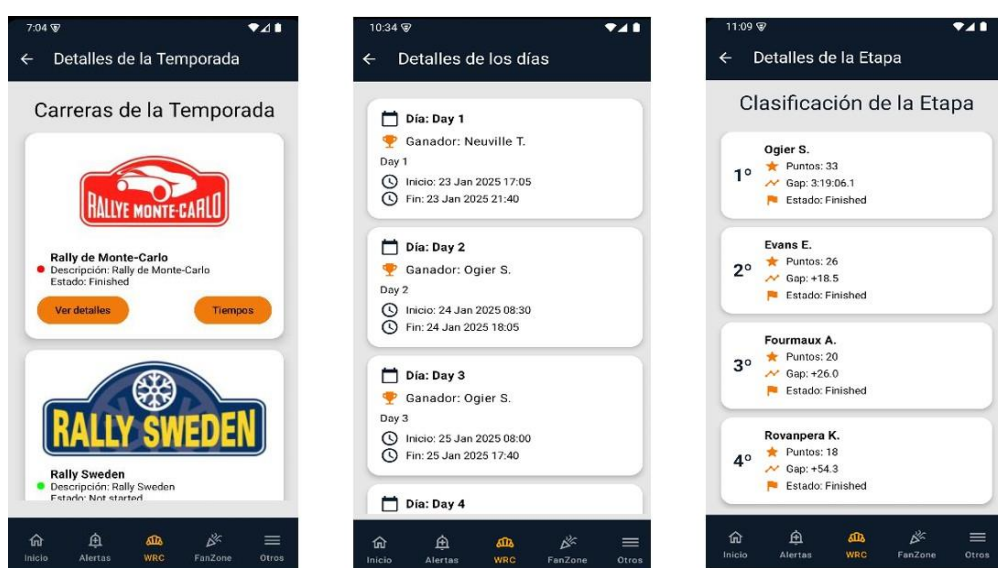
Estas funcionalidades proporcionan una visión completa y actualizada del campeonato, asegurando que los usuarios puedan seguir la evolución de la clasificación en tiempo real.



A) Temporadas

B) Clasificación global

Ilustración 4.35: Temporadas y clasificación



A) Carreras

B) Detalles por día

C) Clasificación de la etapa

Ilustración 4.36: Detalles de Temporada

Sección de Pilotos

La sección de pilotos permite a los usuarios acceder a un listado completo de todos los competidores del mundial de rally, proporcionando información detallada sobre cada uno de ellos. Dispone de las siguientes funcionalidades:

- **Datos del piloto:** Se muestran detalles como imagen, nombre, nacionalidad y lugar de nacimiento, brindando un perfil completo de cada competidor.

CAPÍTULO 4: DESAROLLO

- **Rendimiento por temporada:** Se pueden consultar estadísticas como posiciones finales, puntos acumulados y rallies ganados, ofreciendo una visión detallada de la trayectoria deportiva de cada piloto.
- **Gestión de favoritos:** Los usuarios pueden añadir pilotos a una lista personalizada, facilitando un seguimiento rápido de sus competidores preferidos.

Estas funcionalidades permiten conocer en profundidad a los pilotos del WRC y realizar un seguimiento detallado de su desempeño a lo largo de la temporada.

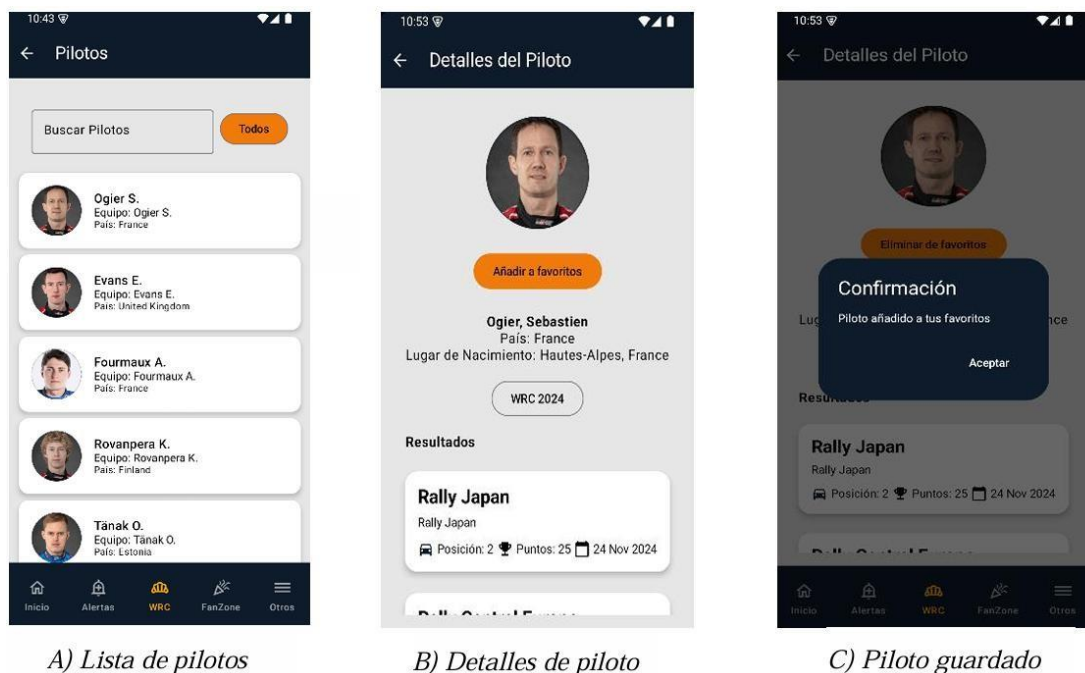


Ilustración 4.37: Pilotos del Mundial de Rally

Sección de Coches

Esta sección permite a los usuarios explorar los vehículos utilizados en el mundial de rally, organizados según su categoría, facilitando la comparación entre distintos modelos. Cuenta con las siguientes funcionalidades:

- **Clasificación por categoría:** Se presentan los coches agrupados por segmentos, permitiendo una mejor organización y comparación entre diferentes modelos.
- **Especificaciones técnicas:** Los usuarios pueden consultar datos clave como potencia, peso y precio, proporcionando información esencial sobre el rendimiento y características de cada vehículo.

Sección de Equipos WRC

La sección de equipos proporciona información detallada sobre cada escudería que compete en el mundial de rally, permitiendo a los usuarios conocer su estructura y participantes. Dispone de las siguientes funcionalidades:

- **Datos del equipo:** Se muestra información clave como nombre, país de origen y una imagen del coche oficial utilizado en la competición.
- **Listado de pilotos:** Se presenta la lista de pilotos asociados a cada equipo, con enlaces directos a sus perfiles individuales para consultar sus estadísticas y rendimiento.



A) Categorías de coches



B) Detalles de equipos de WRC

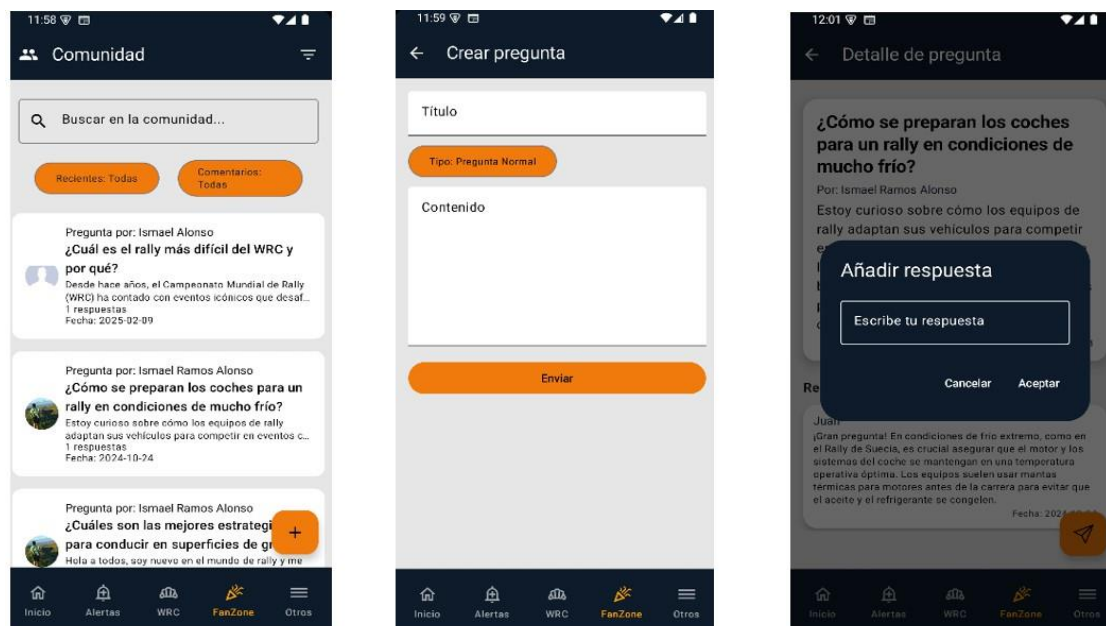
Ilustración 4.38: Coches y equipos de WRC

Sección de Foro

Este sistema permite a los usuarios participar en un espacio de discusión e intercambio de conocimientos dentro de la comunidad, fomentando la interacción y la colaboración entre los aficionados al rally. Dispone de las siguientes funcionalidades:

- **Exploración de contenido:** Los usuarios pueden buscar y visualizar preguntas y encuestas creadas por otros miembros, organizadas por relevancia o fecha de publicación.
- **Creación de publicaciones:** Se ofrece la opción de crear nuevas preguntas o votaciones, permitiendo a los usuarios aportar contenido y generar debates dentro del foro.
- **Interacción con otros usuarios:** Se habilita la posibilidad de responder preguntas, participar en debates y votar en encuestas, fomentando la participación y la construcción de una comunidad dinámica.

Esta funcionalidad proporciona un espacio donde los usuarios pueden compartir conocimientos, resolver dudas y debatir sobre el mundo del rally, enriqueciendo la experiencia de la aplicación.



A) Preguntas de usuarios

B) Creación de pregunta

C) Responder pregunta

Ilustración 4.39: Foro

Otras Secciones

La aplicación incluye una serie de funcionalidades complementarias que permiten a los usuarios personalizar y gestionar su experiencia de manera más eficiente. Estas secciones adicionales ofrecen herramientas para organizar contenido de interés, administrar preferencias y acceder a información relevante dentro de la plataforma. Son las siguientes:

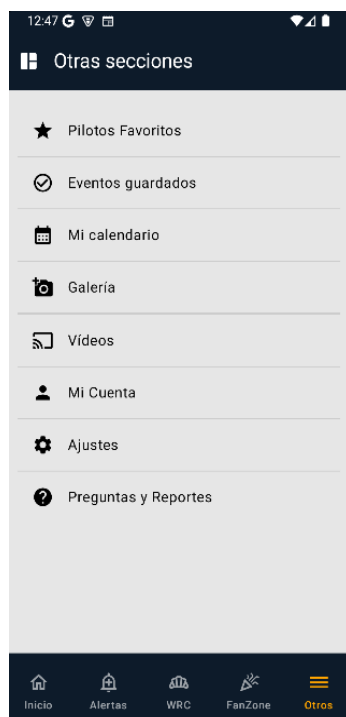
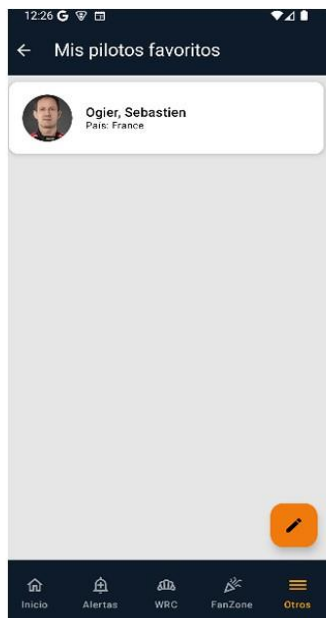


Ilustración 4.40: Otras secciones

1. **Pilotos favoritos:** Esta funcionalidad permite al usuario guardar y consultar rápidamente los pilotos que ha marcado como favoritos. De este modo, puede acceder fácilmente a su información, incluyendo estadísticas y resultados de la temporada y desempeño en cada rally sin necesidad de realizar búsquedas repetitivas.



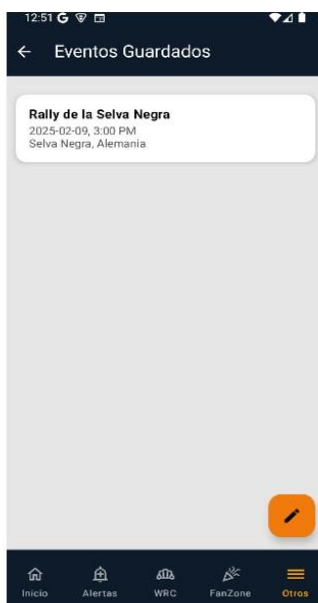
A) Lista de pilotos favoritos



B) Eliminación de piloto de favoritos

Ilustración 4.41: Sección pilotos favoritos

2. **Eventos guardados:** Sección en la que el usuario puede almacenar los eventos de su interés para hacer un seguimiento más eficiente. Esto facilita su consulta posterior, evitando que el usuario tenga que buscarlos manualmente cada vez que desee revisarlos.



A) Lista de eventos guardados



B) Eliminar evento guardado

Ilustración 4.42: Eventos guardados

3. **Calendario de eventos:** Ofrece una vista estructurada en formato de calendario, donde se resaltan las fechas clave de los eventos guardados. Además, permite al usuario recibir recordatorios automáticos sobre competiciones próximas, ayudando a la planificación y el seguimiento de los eventos más relevantes dentro del campeonato.



Ilustración 4.43: Calendario de eventos

4. **Galería de imágenes:** Espacio donde el usuario puede explorar una colección de imágenes organizadas por categorías, relacionadas con el mundo del rally. Esta funcionalidad facilita la visualización de fotografías de diferentes eventos.

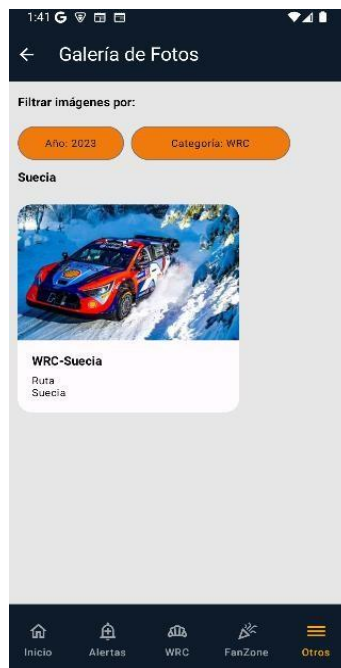


Ilustración 4.44: Galería de imágenes

5. **Publicaciones de videos:** Integración con plataformas como YouTube, permitiendo al usuario acceder a resúmenes de carreras, entrevistas, documentales y otro contenido multimedia relacionado con el rally.

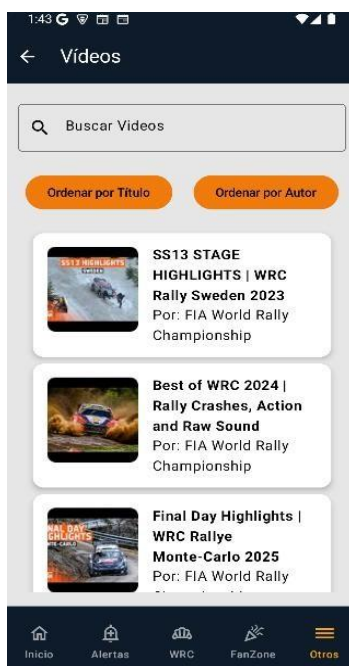
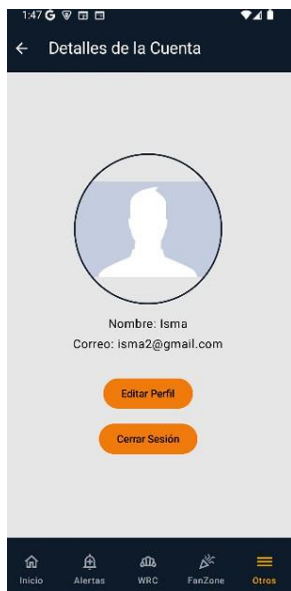


Ilustración 4.45: Sección de vídeos

6. **Mi cuenta:** Sección destinada a la gestión del perfil del usuario, donde puede actualizar su información personal, cambiar su imagen de perfil y cerrar sesión de manera segura. Además, permite modificar datos relacionados con su cuenta y actualizar su contraseña.



A) Perfil de usuario



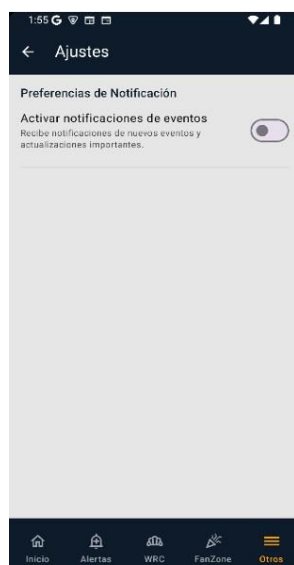
B) Editar perfil



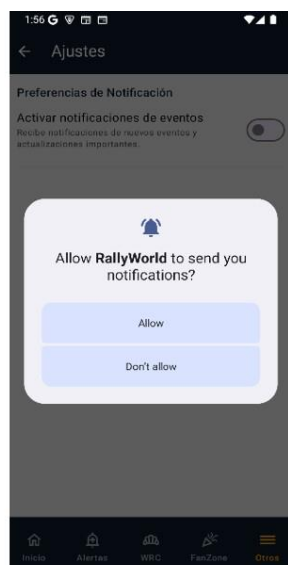
C) Cambiar contraseña

Ilustración 4.46: Sección Perfil de usuario

7. **Ajustes de notificaciones:** Permite al usuario activar o desactivar notificaciones de acuerdo con sus preferencias. Entre las opciones destacadas se encuentra la posibilidad de recibir recordatorios automáticos cuando falte menos de una semana para la celebración de un evento guardado, asegurando que el usuario no pierda ninguna competición de interés.



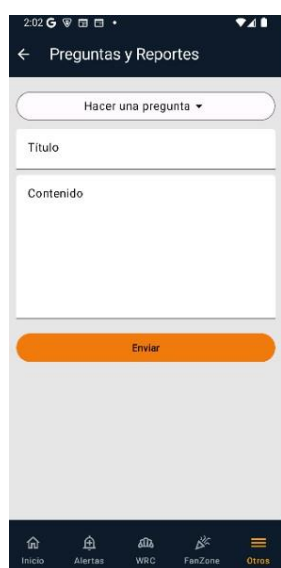
A) Ajustes



B) Permisos de notificación

Ilustración 4.47: Sección de Ajustes

8. **Preguntas y reportes:** Espacio dedicado a la comunicación con el soporte técnico de la aplicación, donde el usuario puede enviar consultas, sugerencias o reportar incidencias. Esta funcionalidad facilita la resolución de problemas y mejora la interacción con la plataforma, asegurando que el usuario pueda recibir asistencia en caso de inconvenientes o dudas sobre el funcionamiento de la aplicación.



A) Realizar consulta



B) Informar de un error

Ilustración 4.48: Sección de Soporte

4.8 Desarrollo de la web

4.8.1. Iniciación y configuración del proyecto en Visual Studio

En este apartado se describen los pasos necesarios para iniciar un nuevo proyecto en Angular utilizando Visual Studio Code, un entorno de desarrollo integrado (IDE) ligero y potente, ampliamente utilizado en el desarrollo web moderno.

Instalación de Node.js y Angular CLI

El primer paso es instalar **Node.js**, que incluye el gestor de paquetes npm necesario para trabajar con Angular. Una vez instalado, se procede a instalar la **Angular CLI** (Command Line Interface), la herramienta oficial de Angular, mediante el siguiente comando ejecutado en la terminal:

```
npm install -g @angular/cli
```

La Angular CLI proporciona comandos que permiten crear y gestionar proyectos Angular de manera estructurada y eficiente.

Creación de un Nuevo Proyecto

Para crear un nuevo proyecto Angular se ha ejecutado el siguiente comando:

```
ng new nombre-proyecto
```

Una vez completado, se generará una estructura base del proyecto en una carpeta con el nombre especificado.

Navegación al Proyecto y Ejecución Inicia

Accede al directorio del proyecto recién creado utilizando el comando:

```
cd nombre-proyecto
```

Luego, se debe iniciar el servidor de desarrollo con:

```
ng serve
```

Esto ejecutará la aplicación en un servidor local, accesible desde <http://localhost:4200>

Con esta configuración inicial, ya se dispone de un entorno completo y funcional para comenzar el desarrollo de una aplicación en Angular utilizando Visual Studio Code. A partir de aquí, se pueden añadir componentes, servicios, módulos y funcionalidades específicas, siguiendo buenas prácticas, para construir una aplicación web escalable y fácil de mantener.

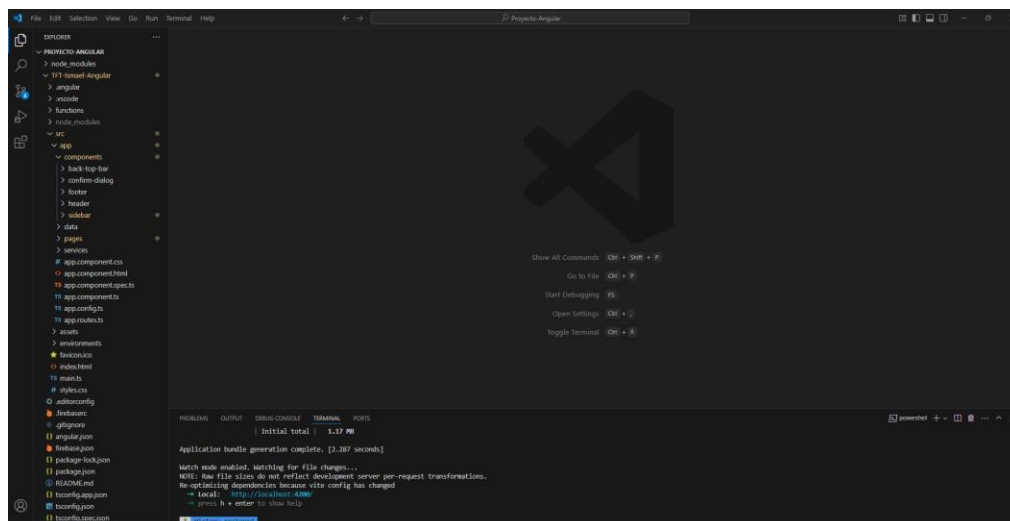


Ilustración 4.49: Proyecto creado en Visual Studio

4.8.2. Diseño de panel de control web

En las secciones siguientes, se presentan de manera integral las capturas de pantalla y el diseño de la plataforma web, poniendo especial atención en la coherencia visual. Las vistas expuestas ilustran la disposición de los elementos y la navegabilidad propuesta, con el objetivo de garantizar una experiencia intuitiva a los administradores, proporcionándoles las herramientas necesarias para una gestión eficiente del contenido. Este enfoque asegura la uniformidad en toda la interfaz y facilita una interacción fluida en cada sección de la plataforma. Para un análisis más detallado del diseño completo, se puede acceder a Figma [\[aquí\]](#).

1. Pantalla de Inicio de Sesión

En esta sección inicial, se prioriza la simplicidad y la claridad en la distribución de los elementos. Con esta disposición, se busca garantizar un acceso intuitivo y eficiente, a través de etiquetas descriptivas y de un botón de acción destacado, de modo que el proceso de inicio de sesión resulte claro y accesible para cualquier persona que interactúe con la plataforma.



Ilustración 4.50: Diseño inicio de sesión

2. Sección de Inicio (Home)

La pantalla de inicio presenta un diseño limpio y ordenado, en el cual se agrupan los elementos más relevantes de la plataforma en tarjetas o bloques diferenciados. En la vista principal, se presentan cuatro tarjetas clave; una dedicada a los eventos, otra a las últimas noticias, una más para los usuarios activos y, por último, una que muestra las estadísticas del mes. Esta disposición no solo permite una navegación ágil, sino que también otorga un acceso rápido a la información esencial.

3. Barra Lateral

Se incorpora una barra lateral como elemento principal de navegación, ofreciendo acceso directo a secciones clave de la plataforma, tales como “Soporte”, “Eventos”, “Noticias” y “Galería”. Este enfoque contribuye a una experiencia de usuario organizada y coherente, al concentrar en un solo lugar las funcionalidades más relevantes y permitir una interacción ágil con los distintos módulos de la aplicación.

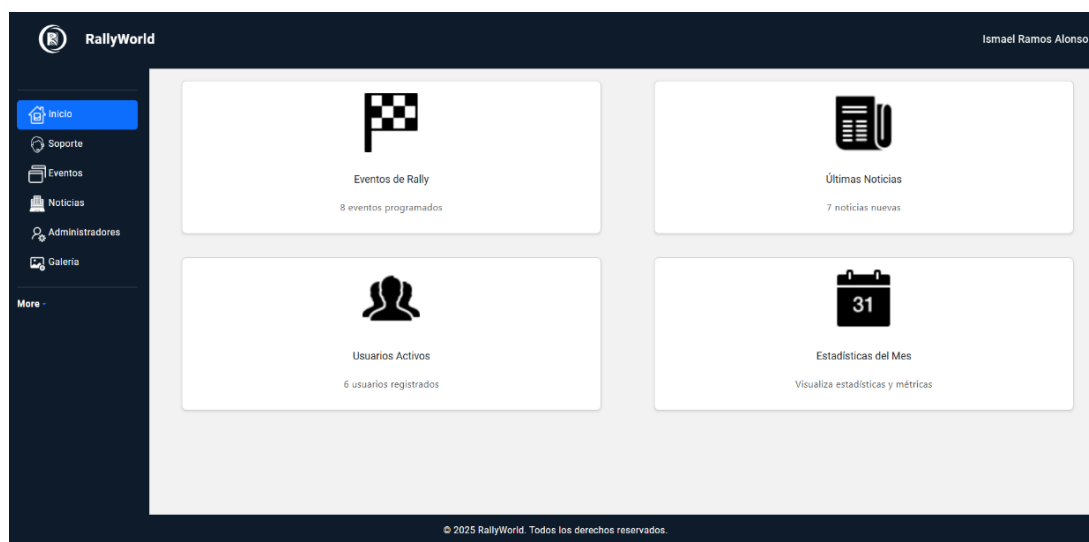


Ilustración 4.51: Inicio y Barra Lateral

4. Análisis de Eventos

Para representar la información relativa a los eventos, se utilizan diagramas de barras y gráficas circulares, complementados con listados que incluyen fechas y ubicaciones. Esta configuración permite apreciar de manera inmediata la cantidad de eventos por lugar, su nivel (ya sea mundial, nacional, provincial o local) y su evolución anual. El empleo de colores distintivos en los gráficos facilita la interpretación de los datos, mientras que cada evento cuenta con una breve descripción y la posibilidad de acceder a información más detallada.



Ilustración 4.52: Análisis de eventos

5. Calendario de Creación de Eventos y Noticias

Esta sección incorpora un calendario con un formato mensual en el que los días con noticias o eventos creados se distinguen mediante colores específicos. En la parte superior de la vista, se han incluido controles de navegación (anterior y siguiente) para desplazarse entre los meses y el año en curso. Al seleccionar una fecha resaltada, se despliega un recuadro informativo que detalla los contenidos registrados en ese día, contribuyendo así a un mayor control y claridad en la planificación de próximas publicaciones.

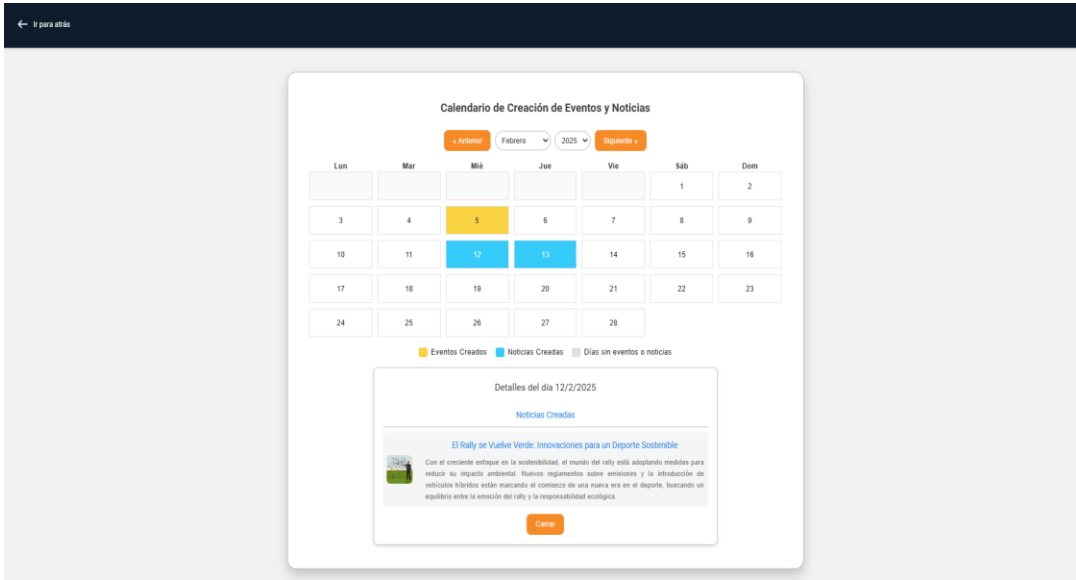


Ilustración 4.53: Calendario de eventos y noticias creadas

6. Listado de Eventos

En esta sección, se despliega un listado de los eventos registrados, cada uno acompañado de información relevante como la fecha, la ubicación y una breve descripción. Además, se incluye un botón de acción para añadir nuevos eventos, el cual, al ser seleccionado, dirige a un formulario donde el administrador puede especificar todos los detalles necesarios. El diseño de la sección mantiene una estructura limpia, con tarjetas que permiten diferenciar fácilmente cada evento y resaltar los datos más significativos, facilitando la consulta y la gestión de la información.

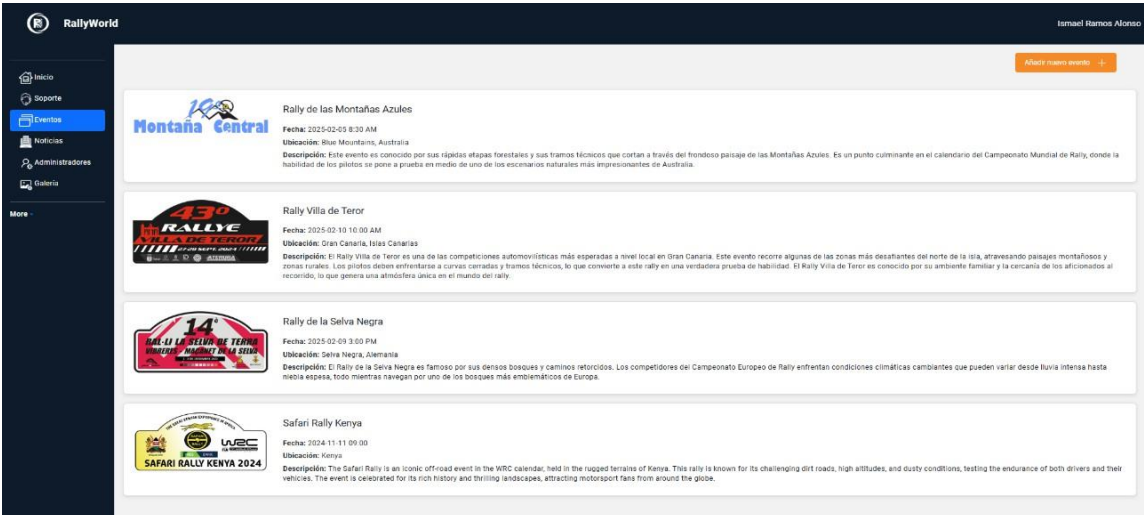


Ilustración 4.54: Listado de eventos

7. Panel de Administración

En cuanto al panel de administradores, se dispone de un listado que muestra los datos fundamentales de cada uno de ellos, entre los que se incluyen el nombre, el correo electrónico y la función que desempeñan dentro de la plataforma. Asimismo, se han habilitado botones de acción para modificar los roles, dar de alta a nuevos miembros o dar de baja a los existentes, lo que agiliza la supervisión global del sistema.

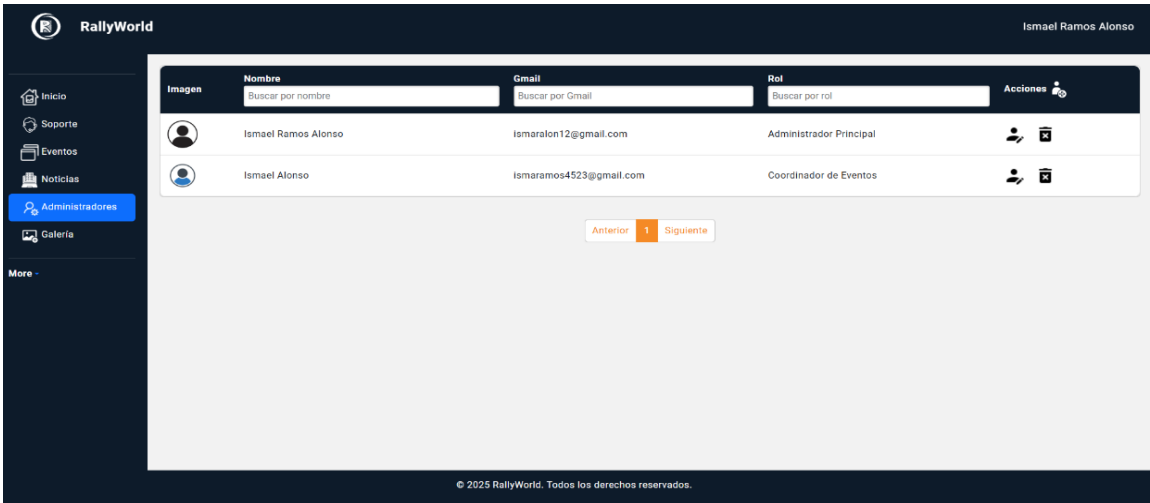


Ilustración 4.55: Gestión de administradores

8. Coherencia y Navegación

En resumen, el panel para administradores mantiene la misma línea gráfica y criterios de diseño responsivo que el resto de la plataforma, asegurando claridad y coherencia en cada sección. Los menús de navegación, botones de acción y enlaces presentan estilos y tipografías uniformes, propiciando una experiencia de uso fluida. Las capturas de pantalla evidencian un entorno enfocado en la simplicidad, facilitando la identificación de opciones y la interacción de los usuarios. Para una observación más detallada de la interfaz, se puede consultar el diseño general en Figma, donde se observa cada pantalla en su totalidad y la navegación completa entre secciones.

4.8.3. Arquitectura del Backend

En el desarrollo del backend, se ha optado por la integración de varios servicios de Firebase, lo que permite aprovechar su infraestructura escalable y sus funcionalidades de seguridad y autenticación. Esto incluye principalmente Firebase Authentication, Cloud Firestore y Firebase Storage, cada uno enfocado a solventar necesidades específicas en la gestión de usuarios y datos.

En primer lugar, Firebase Authentication gestiona el proceso de registro e inicio de sesión tanto para usuarios regulares como para administradores. Gracias a los mecanismos de roles que se implementan, cada administrador accede únicamente a las secciones y permisos asignados, garantizando un control de acceso seguro y centralizado.

Por otro lado, Cloud Firestore se emplea como base de datos principal para almacenar colecciones como *events*, *news*, *administrators* y *users*, entre otras. Esto permite la realización de operaciones CRUD (crear, leer, actualizar y eliminar) por parte de los administradores a través de funciones específicas en la aplicación. De esta manera, los cambios se reflejan de inmediato y en tiempo real, ya sea para agregar nuevos eventos, modificar el contenido existente o eliminar registros cuando sea necesario.

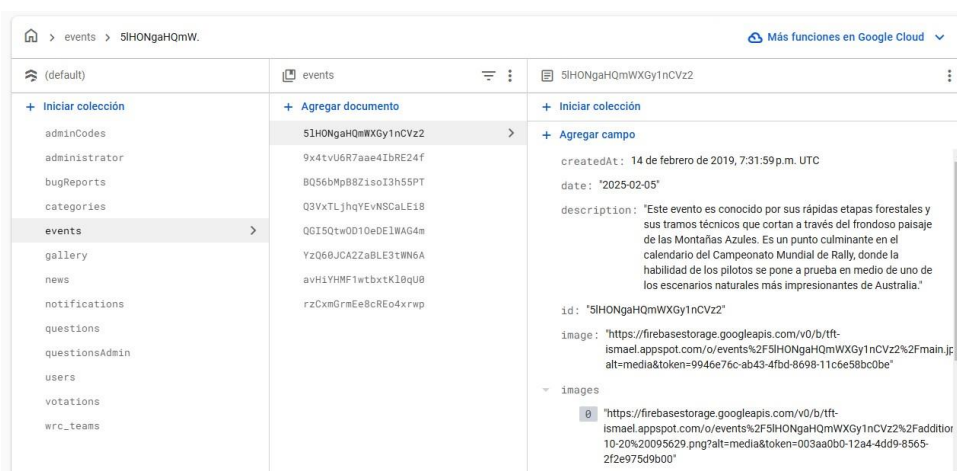


Ilustración 4.56: Colección events

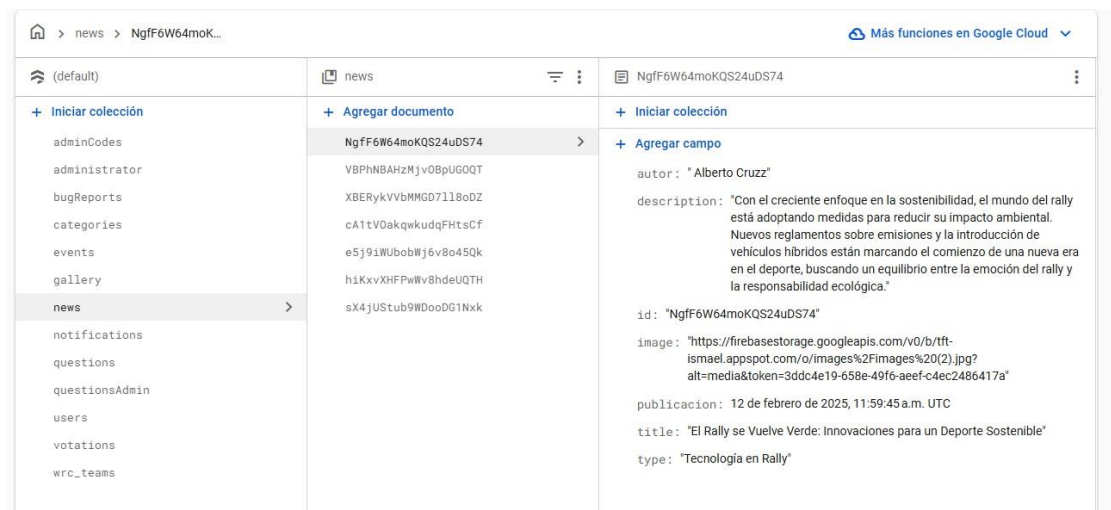


Ilustración 4.57: Colección news

En cuanto a la gestión de imágenes, Firebase Storage aloja los archivos multimedia asociados a noticias, eventos o galerías. Para cada imagen, se guarda una referencia en Firestore que permite su manejo a través de operaciones de subida, reemplazo y eliminación, todo ello desde el panel de administración. Este enfoque evita la necesidad de contar con servidores adicionales y facilita la manipulación de los archivos de gran tamaño.

En definitiva, la adopción de la arquitectura de Firebase en el backend agiliza la creación y el mantenimiento de contenido (eventos, noticias, usuarios, galerías, etc.), fortaleciendo la fiabilidad de la plataforma al permitir que la misma base de datos sea compartida con la aplicación móvil. Este enfoque sienta las bases de un sistema sólido y eficiente para la gestión de datos, capaz de evolucionar y adaptarse a medida que la plataforma crece.

4.8.4. Estructura de ficheros

A continuación, se presenta de forma detallada la estructura de ficheros y directorios que conforman el proyecto, con el propósito de mantener un orden lógico y una separación clara de responsabilidades. Esta organización contribuye tanto a una mejor comprensión del código como a la optimización de las tareas de mantenimiento y evolución de la aplicación.

Dentro de la carpeta **app/** se concentra la mayor parte de la funcionalidad del sistema. Aquí se incluyen los componentes reutilizables, las páginas específicas con las que interactúa el usuario, los servicios de comunicación con la base de datos y los archivos de configuración indispensables para el correcto funcionamiento de la plataforma.

En la carpeta **components/** se ubican los componentes reutilizables, como la barra de navegación para retroceder a la vista anterior, los cuadros de diálogo de confirmación, el footer, el header y la sidebar. Esta disposición facilita la reutilización y el mantenimiento de cada elemento, al quedar claramente definido y separado de los demás.

Por otro lado, en la carpeta **data/** se alojan los archivos que exportan las interfaces o modelos de datos. Este enfoque fomenta la coherencia a la hora de definir y utilizar la

información, ya que componentes, servicios y páginas pueden acceder de manera organizada a estructuras de datos previamente establecidas.

El directorio **pages/** agrupa las diferentes pantallas o vistas que el usuario final observa durante la navegación por la aplicación. Cada una de estas vistas cuenta con sus propios archivos de estructura, estilo y lógica, lo que simplifica tanto su implementación inicial como posibles ajustes o ampliaciones en el futuro.

En la carpeta **services/** se concentran los servicios que se encargan de gestionar la comunicación con las colecciones de la base de datos y de proveer datos a los componentes y páginas. Entre ellos, destacan los servicios para la administración de usuarios, noticias, eventos, autenticación y la gestión de preguntas, reportes y validaciones. Este enfoque centralizado aporta escalabilidad y facilita el mantenimiento del proyecto en el largo plazo.

En conjunto, esta distribución de archivos y directorios no solo promueve la claridad al facilitar la ubicación y el entendimiento de cada elemento del proyecto, sino que también refuerza la escalabilidad al permitir la incorporación ordenada de nuevas funcionalidades y módulos. Cada carpeta y subcarpeta responde a un propósito concreto, lo que evita confusiones y duplicar esfuerzos en el desarrollo. Además, esta arquitectura incentiva la mantenibilidad, pues delimita de manera precisa las distintas capas y responsabilidades, agilizando la identificación y corrección de errores. De esta forma, el proyecto se consolida como una base sólida y fiable para cualquier ampliación o modificación futura.

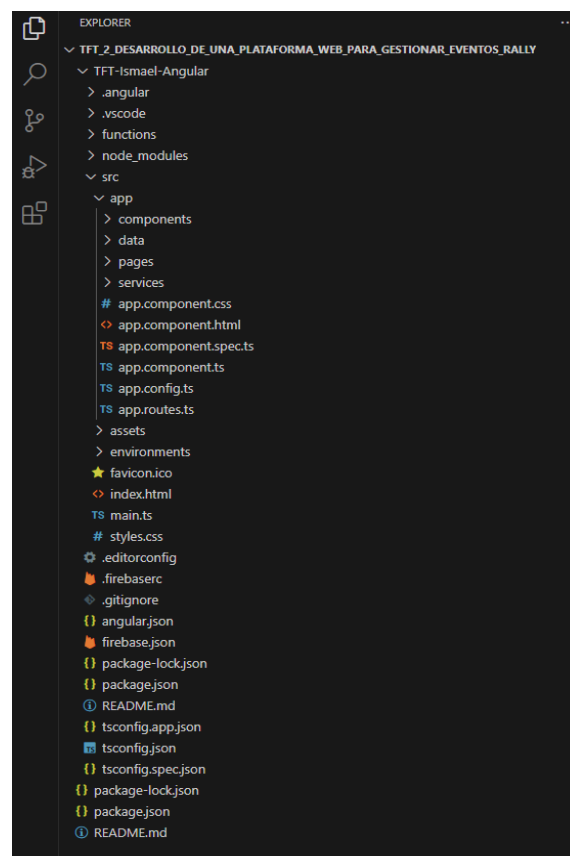


Ilustración 4.58: Estructura de ficheros

4.8.5 Funcionalidades

A continuación, se presentan las principales funcionalidades de la plataforma web, resaltando los aspectos clave que garantizan una experiencia de usuario fluida, eficiente y segura. Estas funciones han sido diseñadas para optimizar la gestión de contenido, mejorar la interacción entre los administradores y facilitar el acceso a la información de manera estructurada.

Inicio de Sesión

El proceso de inicio de sesión está diseñado para ofrecer una autenticación rápida y segura, asegurando que cada usuario pueda acceder a su cuenta sin complicaciones. Sus principales funcionalidades incluyen:

- **Formulario intuitivo:** El usuario introduce su correo electrónico y contraseña en un formulario de fácil uso, garantizando un acceso ágil y seguro.
- **Validación de credenciales:** El sistema verifica la información ingresada y, en caso de ser correcta, otorga acceso inmediato a la cuenta personal.
- **Gestión de sesiones:** Se permite mantener la sesión activa en el dispositivo, evitando que el usuario deba autenticarse repetidamente.
- **Inicio de sesión con Google:** Se ofrece la opción de autenticación con Google Sign-In, permitiendo un acceso más rápido sin necesidad de recordar credenciales adicionales.



Ilustración 4.59: Inicio de sesión

Registro de Usuario

Si el usuario no dispone de una cuenta, el sistema permite el registro mediante un proceso seguro y estructurado. Este proceso incluye:

- **Creación de cuenta:** Se solicita al usuario ingresar un correo electrónico, nombre y una contraseña segura que cumpla con los requisitos mínimos (seis caracteres, al menos una mayúscula y un número).

- **Almacenamiento seguro:** La información proporcionada es validada y almacenada en Firebase Authentication, garantizando la seguridad de los datos.
- **Código de verificación para administradores:** En el caso del registro de administradores, el administrador principal genera un código de verificación único, almacenado en Firebase y asociado al correo electrónico del nuevo administrador. Este código tiene una vigencia de dos minutos, asegurando que solo personas autorizadas puedan registrarse con permisos administrativos.
- **Confirmación de cuenta:** Una vez registrado, el usuario puede acceder a la plataforma con sus credenciales y utilizar las funcionalidades correspondientes según su rol.

Este sistema de autenticación proporciona una experiencia segura y estructurada, permitiendo a los usuarios acceder y gestionar su cuenta de manera eficiente dentro de la plataforma.

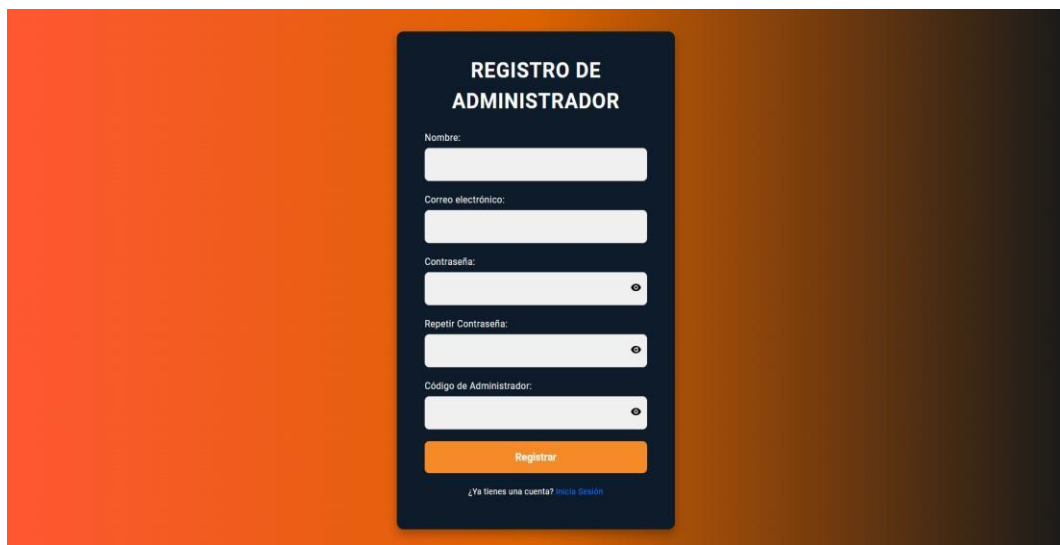
La imagen muestra una interfaz de usuario para el registro de administradores. El fondo es un gradiente de naranja a negro. En el centro hay un formulario con un fondo oscuro. El título del formulario es "REGISTRO DE ADMINISTRADOR". Los campos de entrada son: "Nombre:", "Correo electrónico:", "Contraseña:", "Repetir Contraseña:" y "Código de Administrador:". Cada campo tiene un icono de ojo para alternar la visibilidad. Debajo de los campos hay un botón naranja con el texto "Registrar". En la parte inferior del formulario, hay un enlace que dice "¿Ya tienes una cuenta? Inicia Sesión".

Ilustración 4.60: Registro de administrador

Gestión de eventos

La plataforma permite a los administradores gestionar eventos de manera eficiente mediante un panel de control especializado. Este sistema posibilita la creación, consulta, actualización y eliminación de eventos, asegurando que la información publicada sea precisa y accesible tanto en la versión web como en la aplicación móvil. Las siguientes funcionalidades permiten un manejo estructurado de los eventos:

- **Creación de eventos:** Los administradores pueden añadir nuevos eventos proporcionando información clave como título, ubicación, fecha, descripción e imágenes asociadas, asegurando una presentación clara y detallada.

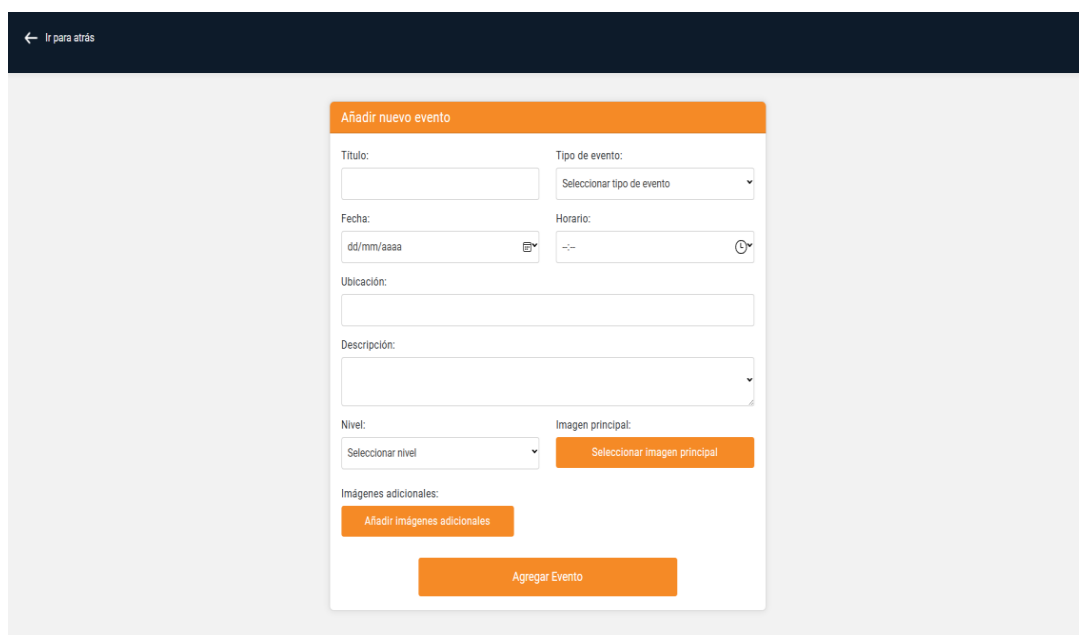


Ilustración 4.61: Creación de evento

- **Modificación de eventos:** Se permite consultar y editar los eventos existentes, asegurando que la información se mantenga actualizada ante posibles cambios en fechas, ubicaciones u otros detalles relevantes.

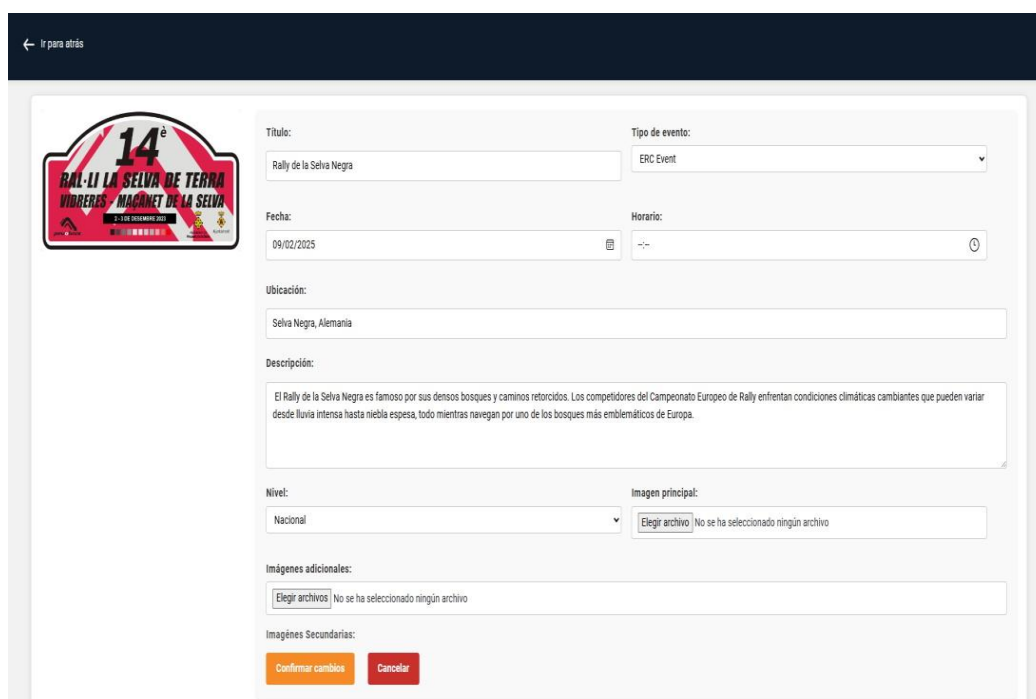
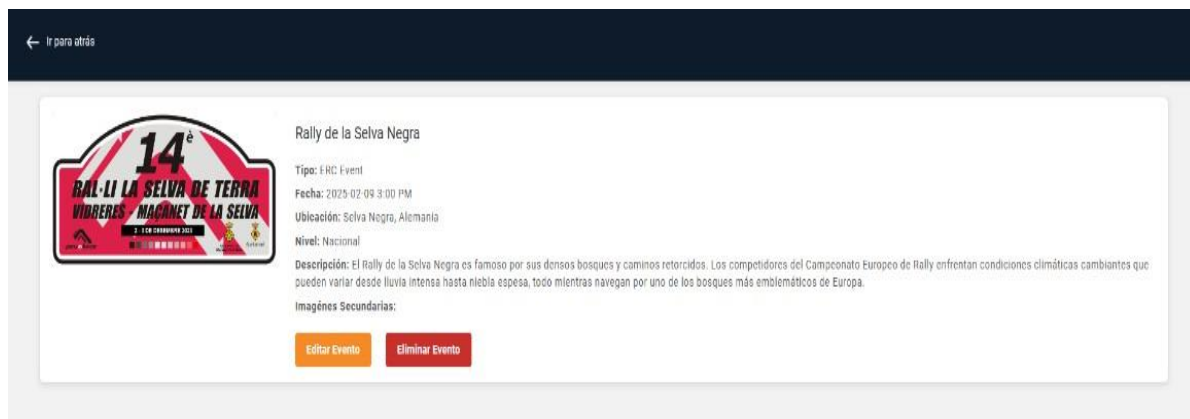


Ilustración 4.62: Editar evento

CAPÍTULO 4: DESAROLLO

- **Disponibilidad en la plataforma:** Todos los eventos creados se almacenan en la base de datos y quedan disponibles tanto en la web como en la app móvil, facilitando el acceso de los usuarios a la información.
- **Eliminación de eventos:** Si un evento deja de ser pertinente, los administradores tienen la opción de eliminarlo permanentemente, manteniendo la sección de eventos clara y libre de información obsoleta.



Gestión de noticias

Esta sección permite a los administradores crear, modificar y organizar publicaciones informativas para los usuarios. Este sistema garantiza que el contenido difundido en la aplicación sea preciso, actualizado y fácilmente accesible en todo momento. Las siguientes funcionalidades aseguran un manejo eficiente de las noticias:

- **Creación de noticias:** Los administradores pueden definir títulos, agregar imágenes, establecer categorías o etiquetas y redactar el texto de cada publicación, asegurando que la información se estructure de manera clara y organizada.

A screenshot of a mobile application interface showing a form to add a new news item. The form is titled 'AGREGAR NUEVA NOTICIA'. It contains several input fields: 'TÍTULO' (a single-line text field), 'DESCRIPCIÓN' (a multi-line text area), 'IMAGEN' (a button labeled 'Elegir archivo' with the text 'No se ha seleccionado ningún archivo' next to it), 'FECHA DE PUBLICACIÓN' (a date picker showing 'dd/mm/aaaa'), and 'TIPO' (a single-line text field). At the bottom of the form is an orange button labeled 'AGREGAR NOTICIA'. The form is set against a light gray background with a dark blue header at the top containing a back arrow and the text 'Ir para atrás'.

- **Disponibilidad en la plataforma:** Una vez publicadas, las noticias se visualizan tanto en la versión web como en la aplicación móvil, permitiendo a los usuarios acceder a la información en cualquier momento y desde cualquier dispositivo.
- **Eliminación de noticias:** Si una publicación deja de ser relevante o necesita ser retirada, el sistema ofrece la opción de eliminar la noticia, permitiendo mantener la sección clara y organizada, evitando la saturación de información innecesaria.

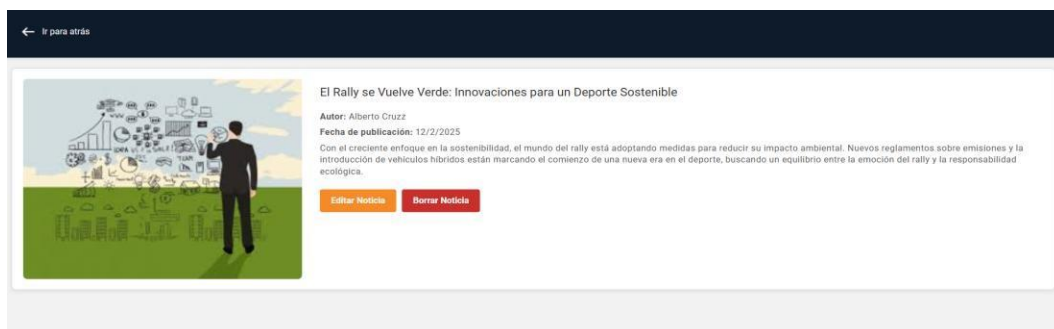


Ilustración 4.63: Detalles noticias

- **Modificación de contenido:** En caso de que una noticia necesite actualizaciones, el administrador puede editar el texto, cambiar imágenes y modificar categorías o etiquetas, asegurando que el contenido refleje información precisa y actualizada.

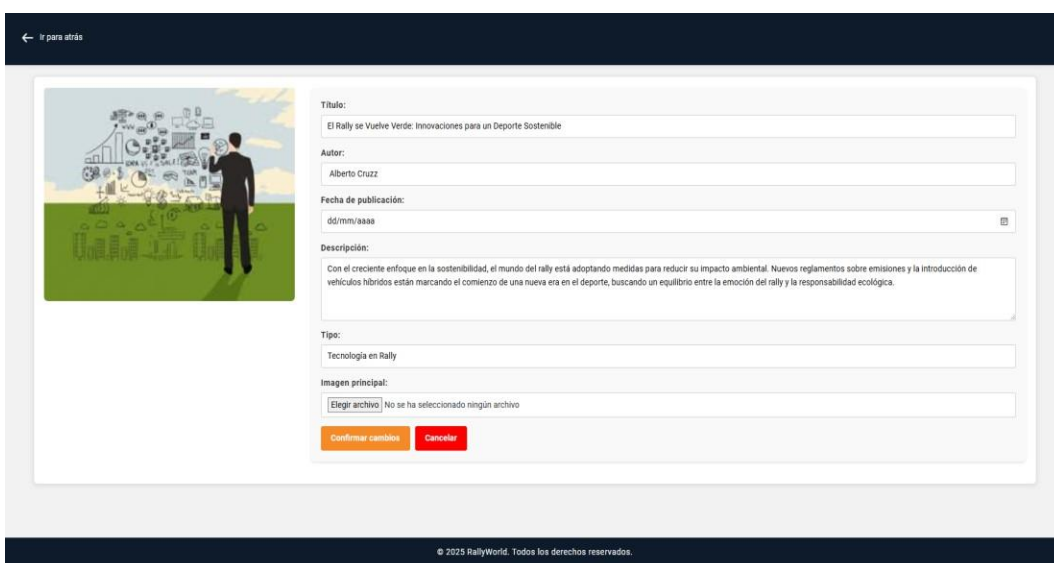
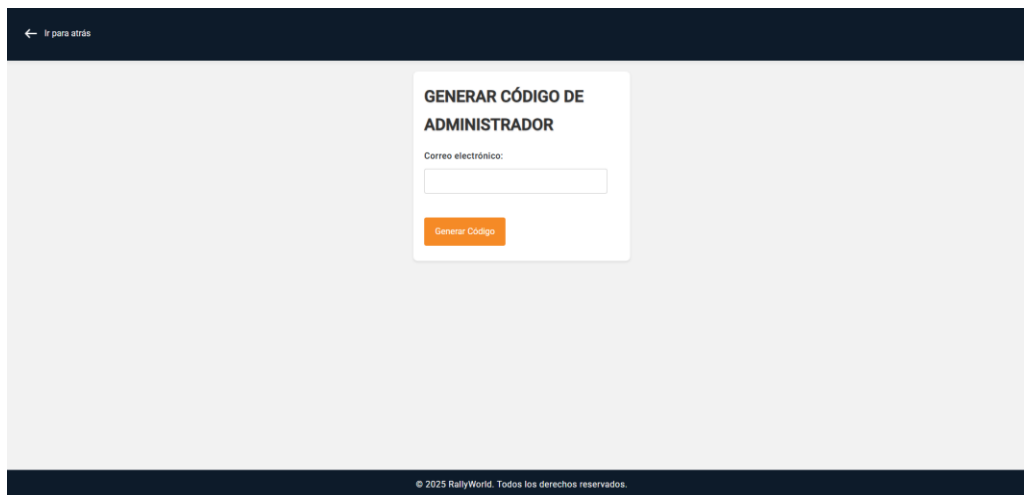


Ilustración 4.64: Editar noticia

Gestión de Administradores

La gestión de administradores dentro de la plataforma permite controlar y asignar privilegios a los usuarios con acceso a funciones de administración. Este sistema garantiza un manejo estructurado y seguro del entorno administrativo, permitiendo una adecuada distribución de responsabilidades. Las siguientes funcionalidades aseguran una administración eficiente y controlada:

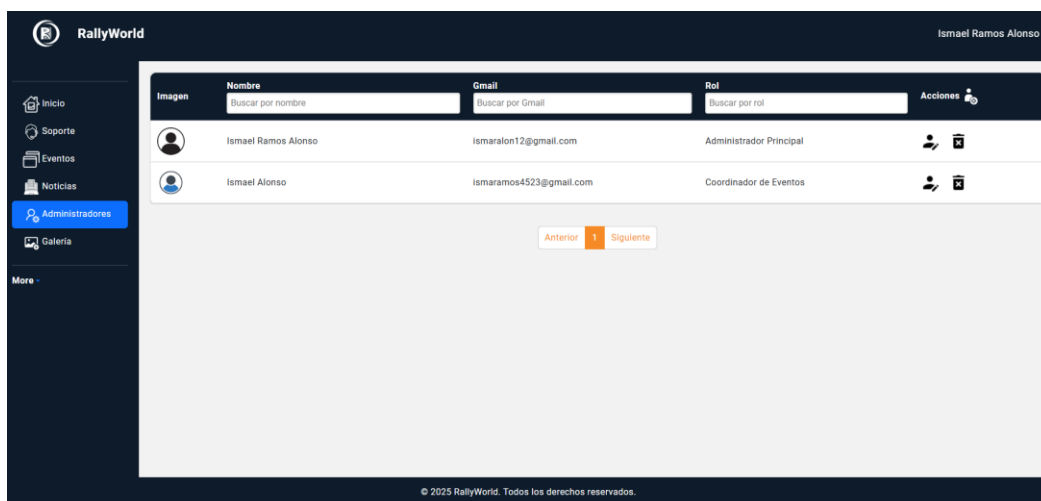
- **Generación de códigos de verificación:** Se generan códigos de acceso únicos, vinculados a un correo electrónico específico, con un periodo de validez limitado. Estos códigos permiten el registro de nuevos administradores, asegurando que solo usuarios autorizados obtengan privilegios de gestión.



The screenshot shows a web interface for generating an administrator code. At the top left, there is a dark blue header with a back arrow and the text 'Ir para atrás'. The main content area is light gray and contains a white card titled 'GENERAR CÓDIGO DE ADMINISTRADOR'. Inside the card, there is a label 'Correo electrónico:' followed by a text input field. Below the input field is an orange button labeled 'Generar Código'. At the bottom of the page, there is a dark blue footer with the text '© 2025 RallyWorld. Todos los derechos reservados.'

Ilustración 4.65: Generar código de administrador

- **Eliminación de administradores:** Se habilita la opción de revocar el acceso de administradores que ya no requieran permisos dentro del sistema. Esta funcionalidad permite mantener un control sobre los usuarios con privilegios especiales, garantizando la seguridad de la plataforma.
- **Visualización de datos esenciales:** Se proporciona un panel donde se muestran información clave de cada administrador, como su nombre y correo electrónico, facilitando la gestión y supervisión de los administradores activos dentro del sistema.



The screenshot shows a web interface for listing administrators. At the top left, there is a dark blue header with the 'RallyWorld' logo and the name 'Ismael Ramos Alonso'. The main content area is light gray and contains a table with the following columns: 'Imagen', 'Nombre', 'Email', 'Rol', and 'Acciones'. The table has two rows of data. Below the table, there are two orange buttons labeled 'Anterior' and 'Siguiente'. At the bottom of the page, there is a dark blue footer with the text '© 2025 RallyWorld. Todos los derechos reservados.'

Imagen	Nombre	Email	Rol	Acciones
	Ismael Ramos Alonso	ismaralon12@gmail.com	Administrador Principal	 
	Ismael Alonso	ismaramos4523@gmail.com	Coordinador de Eventos	 

Ilustración 4.66: Listado de administradores

- **Actualización de roles:** Se permite modificar los roles asignados a cada administrador, ajustando sus responsabilidades según las necesidades de la plataforma. Esto garantiza

que cada usuario tenga acceso únicamente a las funciones necesarias para su rol, manteniendo una distribución eficiente y segura de los privilegios administrativos.

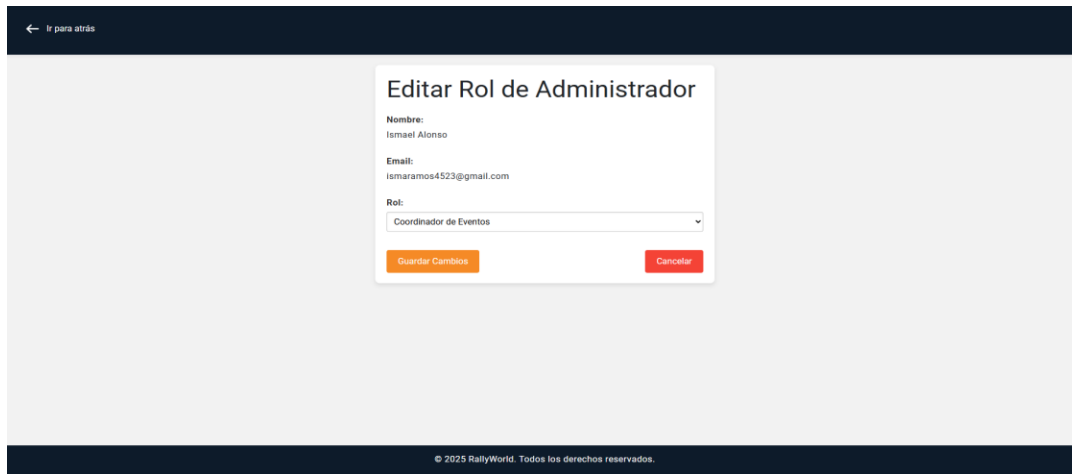
The screenshot shows a web interface with a dark blue header containing a back arrow and the text 'Ir para atrás'. The main content area is light gray and features a white form titled 'Editar Rol de Administrador'. The form contains the following fields: 'Nombre:' with the value 'Ismael Alonso', 'Email:' with the value 'ismaramos4523@gmail.com', and 'Rol:' with a dropdown menu showing 'Coordinador de Eventos'. At the bottom of the form are two buttons: 'Guardar Cambios' (orange) and 'Cancelar' (red). A dark blue footer at the bottom of the page contains the text '© 2025 RallyWorld. Todos los derechos reservados.'

Ilustración 4.67: Editar rol de un administrador

Esta estructura de gestión permite una administración eficiente del equipo de trabajo, asegurando un control adecuado de los permisos y optimizando la seguridad dentro de la plataforma.

Galería de Imágenes

La galería de imágenes en la plataforma web permite a los administradores gestionar el contenido visual disponible en la aplicación, asegurando una organización eficiente y una experiencia visual atractiva para los usuarios. Dispone de las siguientes funcionalidades:

- **Subida de imágenes:** Los administradores pueden añadir nuevas imágenes a las colecciones existentes, organizándolas dentro de categorías predefinidas.

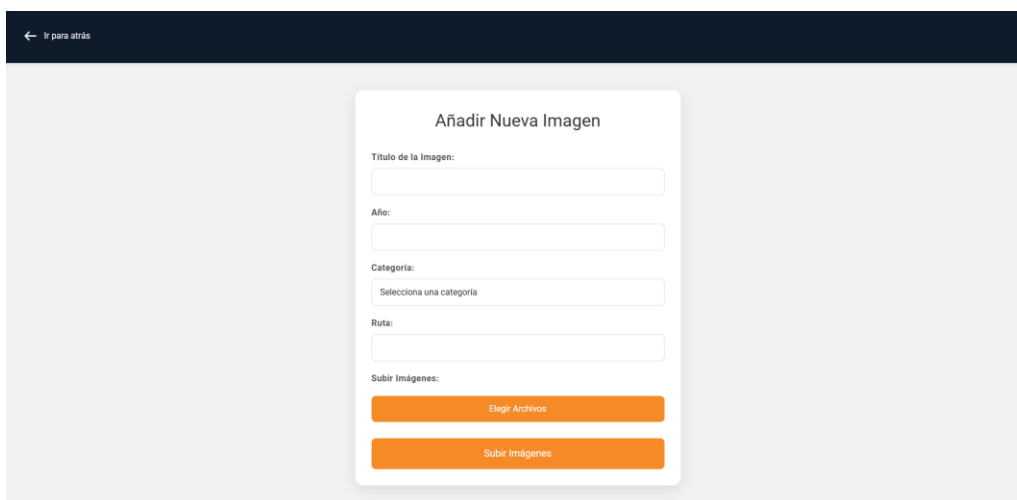
The screenshot shows a web interface with a dark blue header containing a back arrow and the text 'Ir para atrás'. The main content area is light gray and features a white form titled 'Añadir Nueva Imagen'. The form contains the following fields: 'Título de la Imagen:' (text input), 'Año:' (text input), 'Categoría:' (dropdown menu with 'Selecciona una categoría'), and 'Ruta:' (text input). At the bottom of the form, under the heading 'Subir Imágenes:', are two orange buttons: 'Elegir Archivos' and 'Subir Imágenes'.

Ilustración 4.68: Subir imagen y crear colección

- **Edición de colecciones:** Los administradores pueden modificar las colecciones de imágenes, añadiendo o eliminando fotos dentro de cada categoría para mantener un contenido estructurado y actualizado.

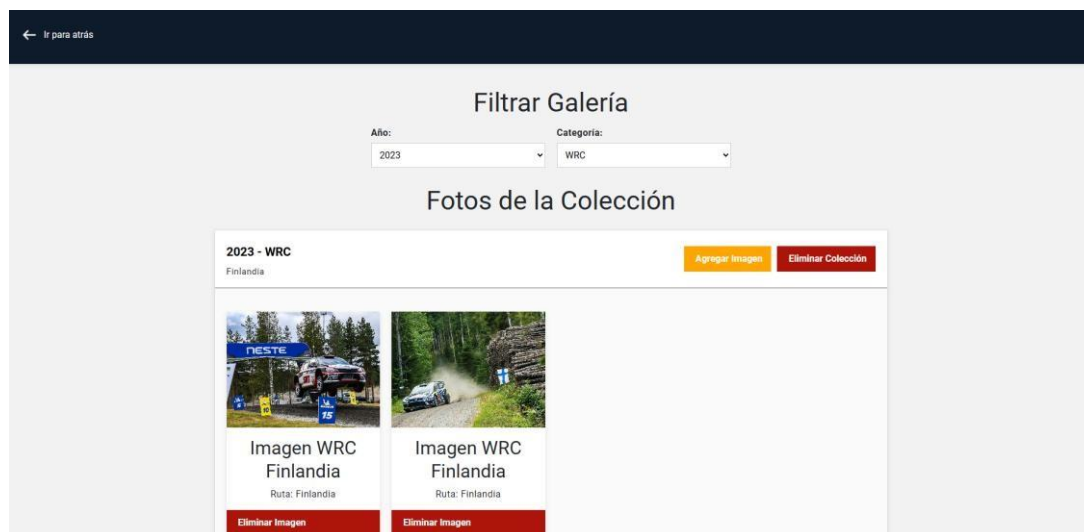


Ilustración 4.69; Editar Colección

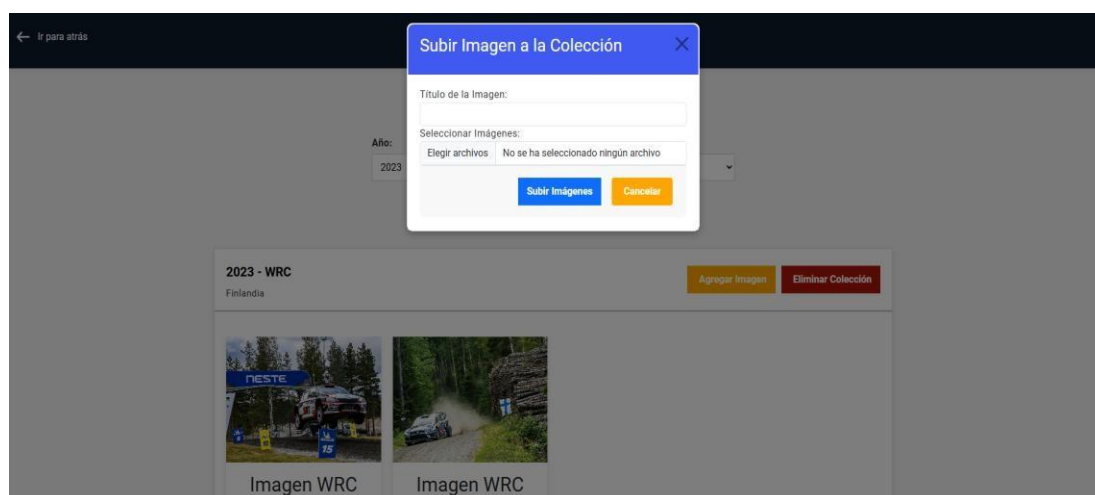


Ilustración 4.70: Añadir Imagen a Colección

Gestión de Usuarios

La Gestión de Usuarios en la plataforma web permite a los administradores supervisar y administrar a los usuarios registrados en la aplicación, asegurando un control adecuado sobre la comunidad. Dispone de las siguientes funcionalidades:

- **Visualización de usuarios:** Los administradores pueden consultar información esencial de los usuarios.
- **Eliminación de usuarios:** Se permite borrar usuarios de la plataforma, asegurando el mantenimiento de la comunidad y la gestión de accesos cuando sea necesario.

Esta funcionalidad proporciona un control eficiente sobre los usuarios registrados, permitiendo a los administradores gestionar la plataforma de manera organizada y optimizada.

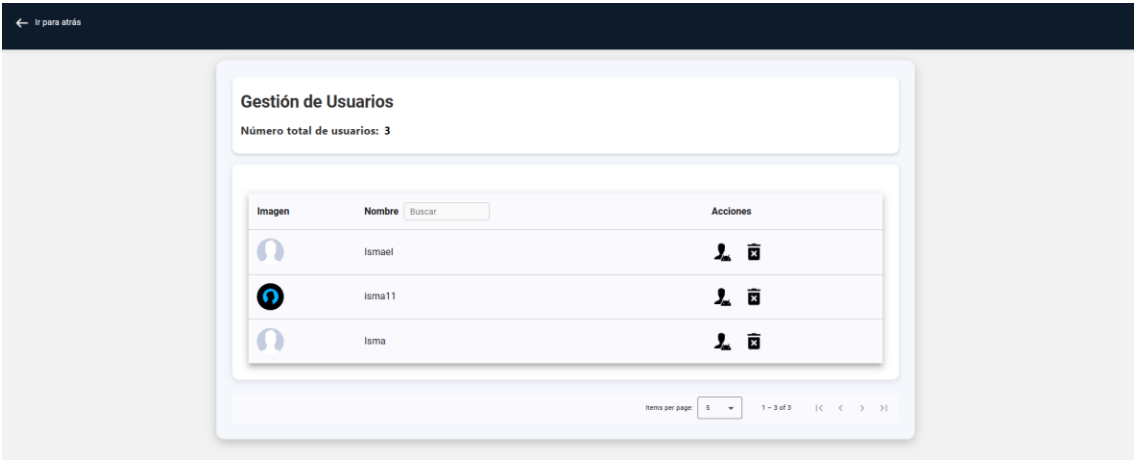


Ilustración 4.71: Listado de usuarios

Gestión de Soporte

La Gestión de Soporte permite a los administradores supervisar y atender consultas o incidencias reportadas por los usuarios, asegurando un adecuado mantenimiento de la plataforma. Tiene las siguientes funcionalidades:

- **Visualización de preguntas y reportes:** Los administradores pueden acceder a una lista de preguntas y errores reportados por los usuarios, organizados para su fácil gestión.
- **Acceso a detalles:** A través de una alerta o notificación, el administrador puede seleccionar un reporte para ver más detalles, facilitando su análisis y resolución.

Esta funcionalidad garantiza un mecanismo eficiente para la gestión de consultas y errores, permitiendo mantener la plataforma operativa y mejorar la experiencia del usuario.

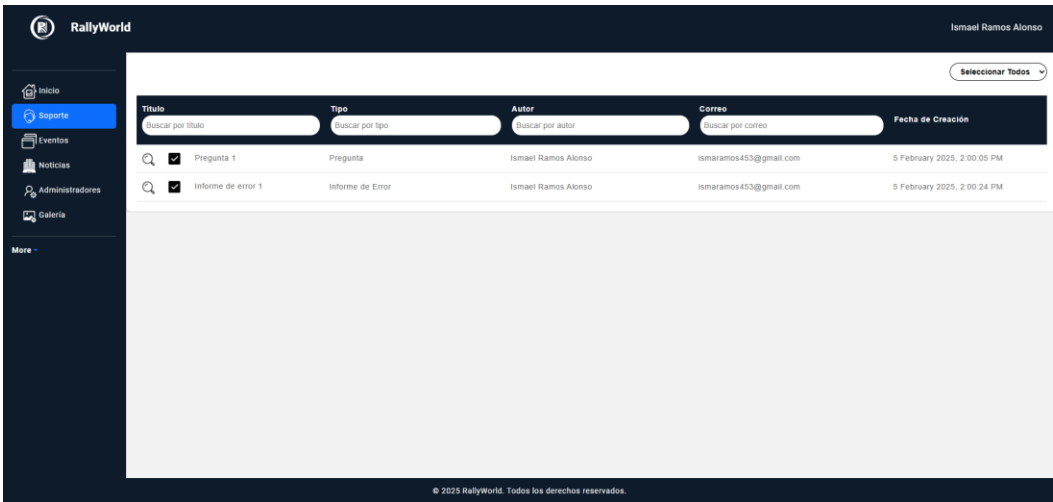


Ilustración 4.72: Listado de informes y preguntas

Capítulo 5: Conclusiones y trabajo futuro

5.1. Conclusiones

El desarrollo de este Trabajo de Fin de Titulación (TFT) ha permitido cumplir con los objetivos planteados en el TFT01, proporcionando una solución tecnológica eficiente para la gestión de información y la interacción de los usuarios en el ámbito del rally. Tanto la aplicación móvil como la plataforma web han sido diseñadas y desarrolladas siguiendo principios de escalabilidad, seguridad y usabilidad, lo que ha garantizado una experiencia óptima para los usuarios y administradores del sistema.

A lo largo del proyecto, se ha empleado el patrón arquitectónico MVVM, lo que ha facilitado una clara separación de responsabilidades entre la interfaz de usuario, la lógica de negocio y la gestión de datos. Esta estructura ha permitido una mejor organización del código, optimizando su mantenimiento y asegurando la posibilidad de futuras ampliaciones sin comprometer la estabilidad del sistema.

La integración con la API ha sido un elemento clave en el desarrollo, ya que ha permitido obtener datos actualizados sobre clasificaciones, tiempos y detalles de pilotos y equipos del Campeonato Mundial de Rally (WRC). Esta funcionalidad ha enriquecido la experiencia del usuario al proporcionar información en tiempo real dentro de la aplicación.

En cuanto a la gestión administrativa, la plataforma web ha dotado a los administradores de herramientas eficaces para la publicación y control de noticias, la gestión de eventos, la moderación del foro y la administración de usuarios. Asimismo, la implementación de Firebase como infraestructura backend ha garantizado un almacenamiento seguro y una sincronización eficiente de los datos en tiempo real.

Desde el punto de vista de la experiencia de usuario, la interfaz de la aplicación ha sido diseñada bajo principios de accesibilidad y ergonomía, asegurando una navegación intuitiva y estructurada. La integración de funcionalidades como notificaciones, gestión de eventos y foros interactivos ha permitido ofrecer una plataforma completa y dinámica para los aficionados al rally.

En conclusión, el presente TFT ha alcanzado los objetivos establecidos, consolidando una solución digital que combina innovación, tecnología y una experiencia de usuario optimizada. La implementación de una arquitectura bien definida, la integración con fuentes de datos externas y el desarrollo de un entorno administrativo eficiente han sido claves para el éxito del proyecto, garantizando su utilidad y viabilidad en el tiempo.

5.2. Trabajo futuro

A pesar de los logros alcanzados en este Trabajo de Fin de Titulación, existen diversas áreas de mejora y expansión que podrían abordarse en futuras versiones de la aplicación y la plataforma web. A continuación, se detallan algunas líneas de trabajo que permitirían ampliar y optimizar el sistema desarrollado:

- **Implementación de un Modo Offline:** Actualmente, la aplicación requiere una conexión a internet para acceder a la información. Una posible mejora sería la incorporación de un sistema de almacenamiento en caché que permita a los usuarios consultar datos previos sin conexión.
- **Expansión de Funcionalidades en el Foro:** Se podría mejorar la sección del foro con nuevas herramientas, como la incorporación de moderación de contenido, un sistema de mensajes privados entre usuarios o la posibilidad de reacciones en publicaciones.
- **Integración con Otras APIs:** Si bien la API proporciona datos en tiempo real del WRC, una futura actualización podría incluir la integración con otras APIs que ofrezcan información complementaria, como estadísticas históricas más detalladas, entrevistas o contenido multimedia en vivo.
- **Mayor Personalización para los Usuarios:** Actualmente, los usuarios pueden guardar eventos y pilotos favoritos, pero futuras mejoras podrían incluir un sistema de recomendaciones personalizadas, basado en la interacción previa de cada usuario con la aplicación.
- **Desarrollo de una Versión para iOS:** Actualmente, la aplicación está enfocada en Android. Un desarrollo futuro podría incluir la adaptación de la aplicación para dispositivos iOS, ampliando así la base de usuarios.

Capítulo 6: Bibliografía

- [1] (A.S.O), A. S. (9 de Enero de 2025). Recuperado el 28 de Enero de 2025, de Dakar Rally: <https://play.google.com/store/apps/details?id=com.aso.dakar&hl=es>
- [2] ¿Qué es Figma? (s.f.). Recuperado el 10 de Febrero de 2025, de Figma Learn: <https://help.figma.com/hc/es-es/articles/14563969806359--Qu%C3%A9-es-Figma>
- [3] Conecta tu app a Firebase. (10 de Febrero de 2025). Recuperado el 3 de Septiembre de 2024, de Firebase: <https://firebase.google.com/docs/database/android/start?hl=es-419>
- [4] Crear un proyecto. (13 de Junio de 2024). Recuperado el 5 de Febrero de 2025, de Android Developers: <https://developer.android.com/studio/projects/create-project>
- [5] Descripción general de proyectos. (21 de Diciembre de 2024). Recuperado el 8 de Febrero de 2025, de Android Developer: <https://developer.android.com/studio/projects?hl=es-419>
- [6] Descubre las funciones que ayudarán a tu equipo a lograr sus objetivos. (s.f.). Recuperado el 12 de Febrero de 2025, de Trello: <https://trello.com/es/tour>
- [7] GmbH, W. P. (8 de Octubre de 2024). Rally TV. Recuperado el 28 de Enero de 2025, de <https://play.google.com/store/apps/details?id=laola1.wrc&hl=es>
- [8] GmbH, W. P. (s.f.). WRC. Recuperado el 20 de Enero de 2025, de <https://www.wrc.com/>
- [9] Introducción a Android Studio. (Diciembre de 2024). Recuperado el 27 de Enero de 2025, de Android Developers: <https://developer.android.com/studio/intro?hl=es-419#:~:text=Android%20Studio%20es%20el%20entorno,desarrollo%20de%20apps%20para%20Android.>
- [10] Jacobs, M. (2023 de Noviembre de 5). ¿Qué es Git? Recuperado el 10 de Enero de 2025, de Microsoft: <https://learn.microsoft.com/es-es/devops/develop/git/what-is-git>
- [11] Michael Stonis. (s.f.). Modelo-Vista-Modelo de vista (MVVM). Recuperado el 8 de Febrero de 2025, de Microsoft Learn: <https://learn.microsoft.com/es-es/dotnet/architecture/maui/mvvm>
- [12] Mora, S. L. (17 de Mayo de 2020). ¿Qué es Firebase? Funcionalidades, ventajas, conclusiones. Recuperado el 20 de Enero de 2025, de Digital55: <https://digital55.com/blog/que-es-firebase-funcionalidades-ventajas-conclusiones/>
- [13] PNUD. (s.f.). Objetivos de Desarrollo Sostenible. Recuperado el 12 de Febrero de 2025, de Programa de las Naciones Unidas para el Desarrollo (PNUD): <https://www.undp.org/es/sustainable-development-goals>
- [14] Rally TV ,¿Qué es Rally TV? (s.f.). Recuperado el 4 de Febrero de 2025, de DAZN: <https://help.dazn.com/hc/es-es/articles/19002884413469-Rally-TV>
- [15] RapidAPI, ¿Qué es RapidAPI? (2022). Recuperado el 27 de Enero de 2025, de <https://docs.rapidapi.com/v1.0/docs/what-is-rapidapi>

- [16] REcodeX (Ed.). (s.f.). MotorsportApi. Obtenido de Rapid API:
<https://rapidapi.com/fluis.lacasse/api/motorsportapi>
- [17] ULPGC., E. d. (s.f.). Recuperado el 20 de Enero de 2025, de
<https://www.eii.ulpgc.es/sites/default/files/2022-05/Grado%20en%20Ingenier%C3%ADa%20Inform%C3%A1tica%20-%20memoria%20del%20plan%20de%20estudios%20-%202019.pdf>
- [18] Visual Studio Code. (s.f.). Recuperado el 27 de Enero de 2025, de Microsoft:
<https://visualstudio.microsoft.com/es/>

Capítulo 7: Anexo I- Ampliación de imágenes

7.1. Diagrama de Flujo de la Web

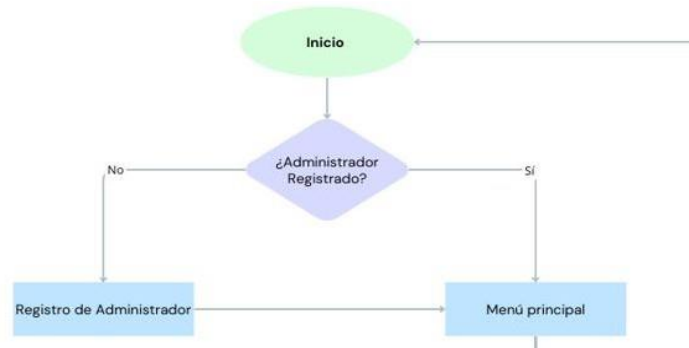


Ilustración 7.1: Flujo web 1

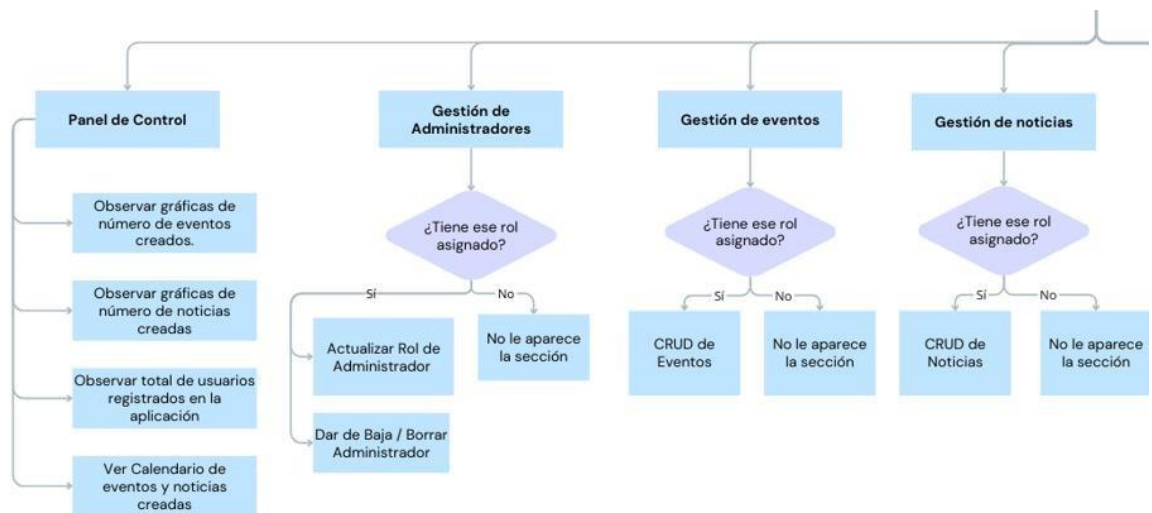


Ilustración 7.2: Flujo web 2

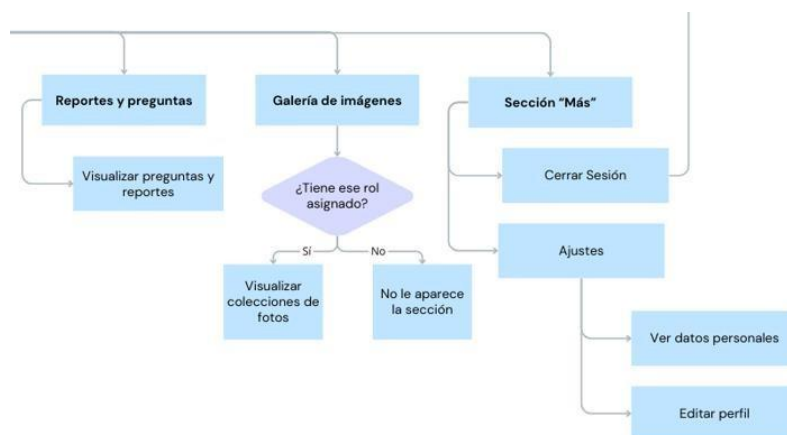


Ilustración 7.3: Flujo web 3

7.2. Diagrama de Flujo de la App

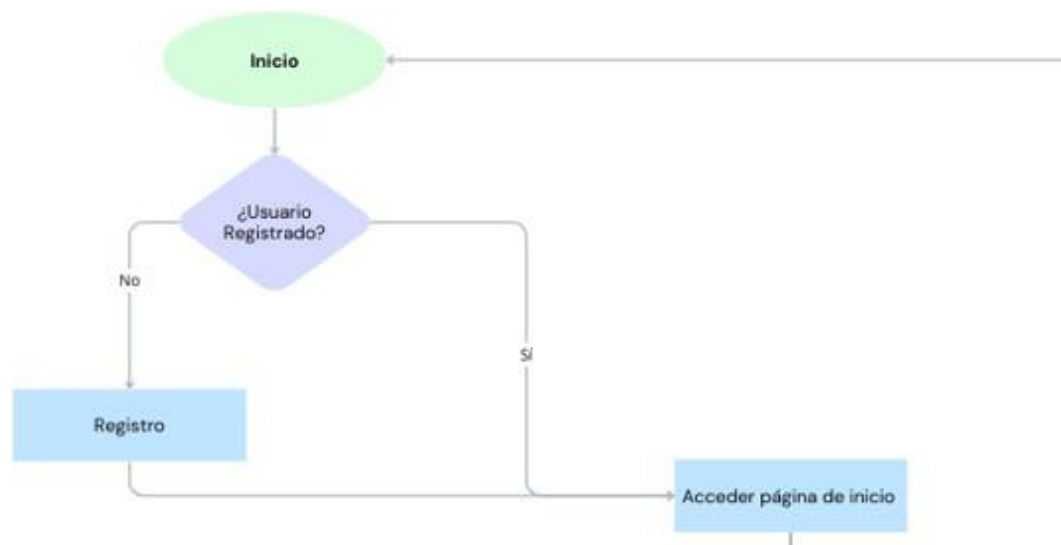


Ilustración 7.4: Flujo App 1

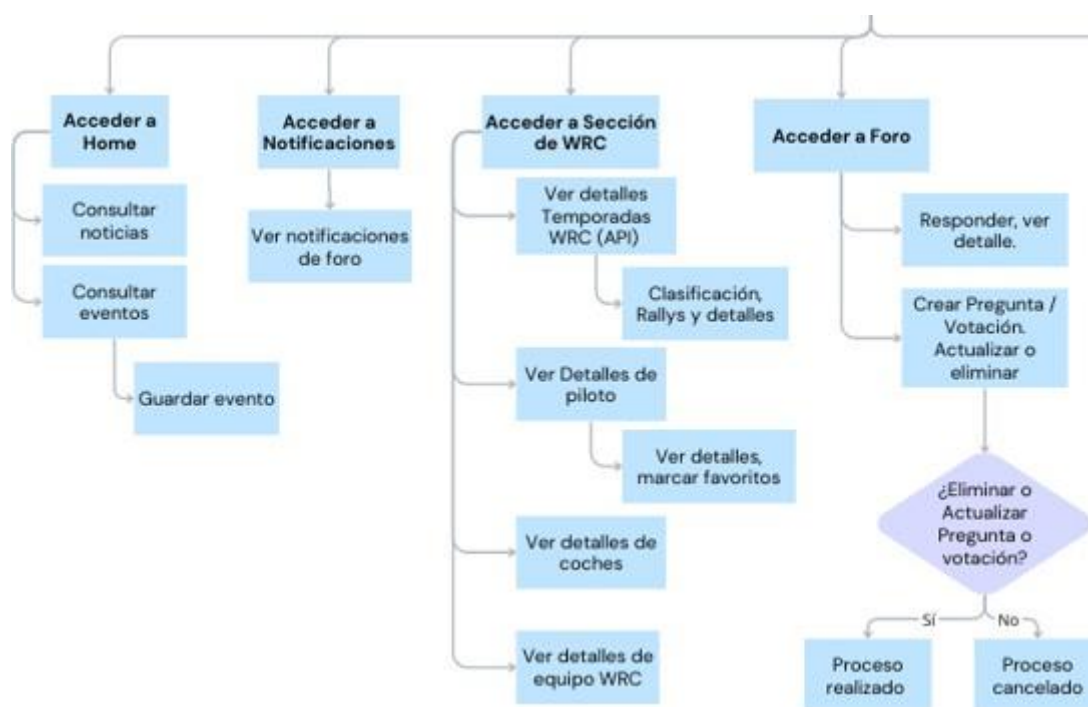


Ilustración 7.5: Flujo App 2

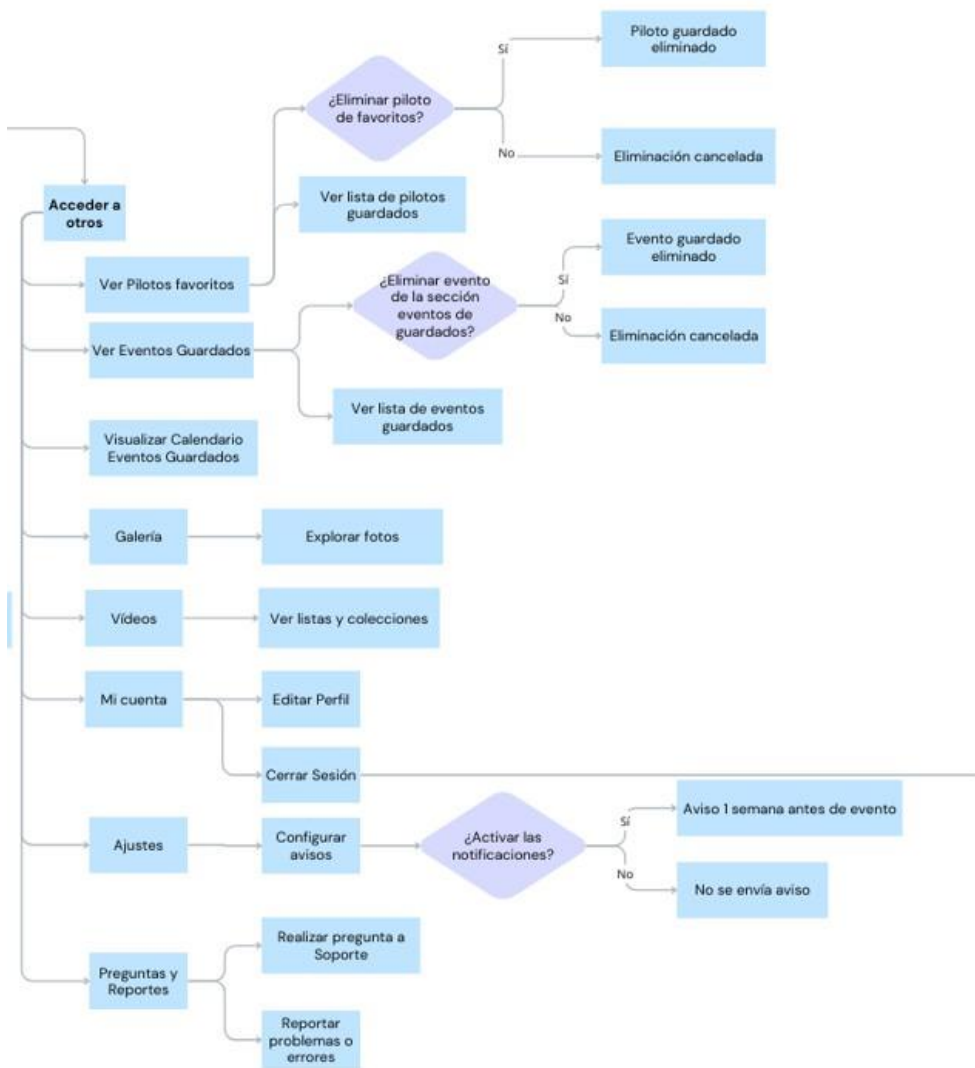


Ilustración 7.6: Flujo App 3