

Aplicación web de romerías

PROGRAMA DE DOBLE TITULACIÓN: GRADO EN INGENIERÍA
INFORMÁTICA Y GRADO EN ADMINISTRACIÓN Y DIRECCIÓN
DE EMPRESAS

Laura Sarmiento Cárdenes

TUTORIZADO POR: FRANCISCO ALEXIS QUESADA ARENCIBIA Y JORGE MARIN
RODRIGUEZ DIAZ | DICIEMBRE 2024

SOLICITUD DE DEFENSA DE TRABAJO DE FIN DE TÍTULO

D/D^a Laura Sarmiento Cárdenes, autor/a del Trabajo de Fin de Título Aplicación web de romerías
correspondiente a la titulación Doble Grado Ingeniería Informática y Administración y Dirección de Empresas plan 41,
en colaboración con la empresa/proyecto (indicar en su caso) _____

S O L I C I T A

que se inicie el procedimiento de defensa del mismo, para lo que se adjunta la documentación requerida, haciendo constar que

☒ se autoriza / ☐ no se autoriza, la grabación en audio de la exposición y turno de preguntas.

Asimismo, con respecto al registro de la propiedad intelectual/industrial del TFT, declara que:

- ☐ Se ha iniciado o hay intención de iniciarlo (defensa no pública).
☒ No está previsto.

Y para que así conste firma la presente. (fecha en firma electrónica)

El/la estudiante

Fdo.: _____

A rellenar y firmar **obligatoriamente** por el/la/los/las tutor/a/es/as.
En relación a la presente solicitud, se informa: (firmar donde corresponda)

Positivamente (en caso de detección de copia, esta firma quedará invalidada)

Negativamente (la justificación en caso de informe negativo deberá incluirse en el TFT05)

Fdo.: _____

Fdo.: _____

DIRECTORA DE LA ESCUELA DE INGENIERÍA INFORMÁTICA

RESUMEN

El objetivo de este trabajo consiste en desarrollar una aplicación web para la gestión de romerías, destinada a unificar toda la información de romerías y facilitar el proceso de compra-venta de entradas en las carrozas que participan en estas. Así, se pretende convertir de esta aplicación la herramienta principal en las festividades tradicionales de las Islas Canarias, resolviendo las dificultades que se presentan actualmente en la celebración de estas fiestas a los distintos participantes. Se desarrolla un producto mínimo viable con un *frontend* amigable al usuario y un *backend* robusto y escalable. Así mismo, se elabora un modelo de negocio con la metodología *canvas* para seguir una estrategia que asegure una alta rentabilidad del proyecto.

PALABRAS CLAVE

Romerías, carrozas, modelo de negocio, aplicación web, Islas Canarias.

ABSTRACT

The aim of this work is to develop a web application for the management of pilgrimages, aimed at unifying all the information on pilgrimages and facilitating the process of buying and selling tickets on the floats that participate in them. The aim is to turn this application into the main tool for the traditional festivities of the Canary Islands, resolving the difficulties that currently arise in the celebration of these festivities for the different participants. A minimum viable product is developed with a user-friendly frontend and a robust and scalable backend. Likewise, a business model is developed with the canvas methodology to follow a strategy that ensures a high profitability of the project.

KEY WORDS

Pilgrimages, floats, business model, web application, Canary Islands.

ÍNDICE

1.	Introducción	8
1.1.	Guía de lectura	8
1.2.	Motivación	8
1.3.	Estado actual y objetivos iniciales	9
1.4.	Justificación de las competencias específicas cubiertas	9
1.5.	Aportaciones	12
1.6.	Normativa y legislación	12
2.	Modelo de negocio	13
2.1.	Análisis de la propuesta de valor	13
2.1.1.	Identificación del problema	13
2.1.2.	Propuesta de valor	14
2.1.3.	Diferenciación frente a otras soluciones	16
2.1.4.	Beneficios de la propuesta de valor	16
2.1.5.	Resultados esperados	17
2.1.6.	Conclusión	17
2.2.	Análisis de la competencia	17
2.2.1.	Identificación de competidores directos e indirectos	17
2.2.2.	Comparativa de funcionalidades	23
2.2.3.	Análisis DAFO	24
2.2.4.	Conclusión	26
2.3.	Análisis del segmento de clientes	27
2.3.1.	Identificación de los segmentos de clientes	27
2.3.2.	Análisis del segmento: participantes en las romerías	27
2.3.3.	Análisis del segmento: cocheros	27
2.3.4.	Análisis del segmento: organizadores de romerías	28
2.3.5.	Conclusión	29
2.4.	Análisis de los canales	29
2.4.1.	Identificación de los canales	29
2.4.2.	Canales de comunicación	29
2.4.3.	Canales de distribución	29
2.4.4.	Canales de soporte	30
2.4.5.	Conclusión	30

2.5.	Análisis de las relaciones con el cliente	30
2.5.1.	Identificación de las relaciones con el cliente	30
2.5.2.	Relaciones con los participantes de las romerías	31
2.5.3.	Relaciones con los cocheros	31
2.5.4.	Relaciones con los organizadores de las romerías	32
2.5.5.	Evaluación de la eficacia de las relaciones	32
2.5.6.	Conclusión	33
2.6.	Análisis del flujo de ingresos	33
2.6.1.	Introducción al análisis de los flujos de ingresos	33
2.6.2.	Ingresos por comisiones en las reservas de las carrozas	33
2.6.3.	Ingresos por suscripciones <i>premium</i> para usuarios	34
2.6.4.	Ingresos por suscripción <i>premium</i> para cocheros	34
2.6.5.	Ingresos por suscripción <i>premium</i> para organizadores de romerías	34
2.6.6.	Ingresos por publicidad	35
2.6.7.	Conclusión	35
2.7.	Análisis de los recursos claves	36
2.7.1.	Introducción	36
2.7.2.	Tecnología y plataforma digital	36
2.7.3.	Equipo de desarrollo y soporte técnico	36
2.7.4.	Infraestructura de seguridad y plataformas de pago	37
2.7.5.	Conclusión	37
2.8.	Análisis de las actividades claves	38
2.8.1.	Desarrollo y mantenimiento de la plataforma	38
2.8.2.	Gestión de seguridad y privacidad de datos	38
2.8.3.	Relaciones con cocheros y responsables de romerías	39
2.8.4.	<i>Marketing</i> y promoción	39
2.8.5.	Gestión de soporte y atención al cliente.	40
2.8.6.	Conclusión	40
2.9.	Análisis de los aliados	40
2.9.1.	Cocheros y organizadores de romerías	40
2.9.2.	Oficinas de turismo y entidades locales	41
2.9.3.	Empresas locales	41
2.9.4.	Conclusión	42
2.10.	Análisis de la estructura de costes	42

2.10.1.	Costes de desarrollo de la aplicación	42
2.10.2.	Infraestructura en la nube	42
2.10.3.	Soporte al cliente	43
2.10.4.	<i>Marketing</i> y promoción	43
2.10.5.	Transacciones bancarias	43
2.10.6.	Costes legales	43
2.10.7.	Conclusión	44
2.11.	Lienzo de modelo de negocio	44
2.12.	Análisis de la viabilidad económica y financiera	45
2.12.1.	Público objetivo	45
2.12.2.	Estimación de ingresos	47
2.12.3.	Estimación de gastos	48
2.12.4.	Análisis escenarios	49
3.	Herramientas y tecnologías de desarrollo	51
4.	Análisis, diseño y desarrollo tecnológico	54
4.1.	Análisis y diseño	54
4.1.1.	Casos de uso	54
4.1.1.1.	Actores	54
4.1.1.2.	Casos de uso: usuario común	54
4.1.1.3.	Casos de uso: usuario cochero	55
4.1.1.4.	Casos de uso: usuario administrador	55
4.1.2.	Estructura de la base de datos	55
4.1.3.	Arquitectura	57
4.1.4.	Interfaz gráfica	69
4.2.	Desarrollo	77
4.3.	Pruebas del sistema	86
5.	Resultados, conclusiones y trabajos futuros	90
5.1.	Resultados	90
5.2.	Conclusiones	90
5.3.	Trabajos futuros	91
6.	Fuentes de información	93
	ANEXO I: Manual de Usuario	98

ÍNDICE DE FIGURAS

Figura 1. Análisis DAFO. Fuente: elaboración propia.	24
Figura 2. Modelo Canvas. Fuente: elaboración propia.	45
Figura 3. Crecimiento de usuarios. Fuente: elaboración propia.	46
Figura 4. Gráfico de beneficios. Fuente: elaboración propia.	50
Figura 5. Casos de uso de usuario común. Fuente: elaboración propia.	54
Figura 6. Casos de uso de usuario cochero. Fuente: elaboración propia.	55
Figura 7. Casos de uso de usuario administrador. Fuente: elaboración propia.	55
Figura 8. Diagrama entidad-relación. Fuente: elaboración propia.	56
Figura 9. Arquitectura hexagonal. Fuente: Imágenes de Google.	58
Figura 10. Arquitectura backend. Fuente: elaboración propia.	59
Figura 11. Estructura de directorios. Fuente: elaboración propia.	61
Figura 12. Arquitectura del frontend. Fuente: elaboración propia.	62
Figura 13. Desarrollo react-router-dom. Fuente: elaboración propia.	63
Figura 14. Archivo index.html. Fuente: elaboración propia.	64
Figura 15. Archivo main.jsx. Fuente: elaboración propia.	64
Figura 16. Funciones de lógica. Fuente: elaboración propia.	65
Figura 17. Componente Home. Fuente: elaboración propia.	65
Figura 18. Componente Pilgrimage. Fuente: elaboración propia.	66
Figura 19. Componente Box de MUI. Fuente: elaboración propia.	66
Figura 20. Uso de AuthProvider. Fuente: elaboración propia.	67
Figura 21. Desarrollo de AuthContext y AuthProvider. Fuente: elaboración propia.	68
Figura 22. Uso de useAuth. Fuente: elaboración propia.	68
Figura 23. Página inicio. Fuente: elaboración propia.	69
Figura 24. Página romería. Fuente: elaboración propia.	69
Figura 25. Detalles romería y tarjetas opiniones. Fuente: elaboración propia.	70
Figura 26. Tarjeta de carroza. Fuente: elaboración propia.	70
Figura 27. Página inicio de sesión. Fuente: elaboración propia.	71
Figura 28. Página registro. Fuente: elaboración propia.	71
Figura 29. Caja usuario común. Fuente: elaboración propia.	72
Figura 30. Caja usuario cochero. Fuente: elaboración propia.	73
Figura 31. Configuración perfil. Fuente: elaboración propia.	73
Figura 32. Notificación perfil editado. Fuente: elaboración propia.	74
Figura 33. Modal añadir comentario. Fuente: elaboración propia.	74
Figura 34. Modal comprar entrada. Fuente: elaboración propia.	74
Figura 35. Página mis entradas. Fuente: elaboración propia.	75
Figura 36. Modal detalles entrada. Fuente: elaboración propia.	75
Figura 37. Botón añadir romería. Fuente: elaboración propia.	75
Figura 38. Modal editar romería. Fuente: elaboración propia.	76
Figura 39. Modal añadir carroza. Fuente: elaboración propia.	76
Figura 40. Modal seleccionar carroza. Fuente: elaboración propia.	77
Figura 41. Generador de token. Fuente: elaboración propia.	78
Figura 42. Método login. Fuente: elaboración propia.	78
Figura 43. Filtro de seguridad. Fuente: elaboración propia.	79

Figura 44. Método addPilgrimage. Fuente: elaboración propia.	79
Figura 45. Método getPilgrimages. Fuente: elaboración propia.	79
Figura 46. Método getPilgrimagesPagination. Fuente: elaboración propia.	80
Figura 47. Clase Comment. Fuente: elaboración propia.	81
Figura 48. Clase CommentRequest. Fuente: elaboración propia.	82
Figura 49. Clase CommentResponse. Fuente: elaboración propia.	83
Figura 50. Métodos convertToResponse y convertFromRequest. Fuente: elaboración propia.	83
Figura 51. Interfaz CommentRepository. Fuente: elaboración propia.	84
Figura 52. Interfaz ICommentService. Fuente: elaboración propia.	84
Figura 53. Métodos getCommentsByPilgrimage y getCommentsByUser de CommentService. Fuente: elaboración propia.	85
Figura 54. Métodos getCommentsByUsername y getCommentsByPilgrimage de CommentController. Fuente: elaboración propia.	86
Figura 55. Configuración test base de datos. Fuente: elaboración propia.	87
Figura 56. Test shouldSavePilgrimageTest. Fuente: elaboración propia.	88
Figura 57. Test when_you_call_pilgrimage_with_put_id_then_update. Fuente: elaboración propia.	89

ÍNDICE DE TABLAS

Tabla 1. Comparativa de funcionalidades entre Romería Viva y sus competidores. Fuente: elaboración propia.	23
Tabla 2. Ingresos anuales. Fuente: elaboración propia.	48
Tabla 3. Gastos anuales. Fuente: elaboración propia.	49

1. INTRODUCCIÓN

1.1. GUÍA DE LECTURA

Apartados comunes:

Motivación (1.2.)

Estado actual y objetivos iniciales (1.3.)

Aportaciones (1.5.)

Normativa y legislación (1.6.)

Resultados, conclusiones y trabajos futuros (5.)

Apartados estrictamente relacionados con el grado de Administración y Dirección de Empresas:

Justificación de las competencias específicas cubiertas (Competencias del Grado en Administración y Dirección de Empresas) (1.4.)

Modelo de negocio (2.)

Apartados estrictamente relacionados con el grado de Ingeniería Informática

Justificación de las competencias específicas cubiertas (Competencias del Grado en Ingeniería Informática) (1.4.)

Análisis de la competencia (lectura recomendable, no obligatoria) (2.2.)

Herramientas y lenguajes de desarrollo (3.)

Análisis, diseño y desarrollo tecnológico (4.)

1.2. MOTIVACIÓN

En la actualidad, no existe demasiada información sobre las romerías en Gran Canaria, esta está limitada y genera una dificultad en el acceso a la información. A esta problemática se le añade que la participación activa es complicada ya que depende de los contactos personales dado que funciona con una gestión “boca a boca”, limitando el acceso a nuevos asistentes y generando una menor participación. Así, se crea una

barrera de entrada para gente turista o locales de municipios distintos al de la romería correspondiente, motivo por el que en muchos casos no pueden disfrutar de estas festividades. Este trabajo busca proporcionar una solución a estos procesos manuales y poco eficientes en la gestión de romerías, desarrollando una aplicación web que digitalice el proceso de compra-venta de entradas y centralice toda la información de las romerías, consiguiendo que sean más accesibles a turistas e isleños. A su vez, se pretende analizar la viabilidad económica del modelo de negocio correspondiente.

1.3. ESTADO ACTUAL Y OBJETIVOS INICIALES

En el contexto actual se presenta una gestión manual y poco eficiente que dificulta la logística y la participación en las romerías. A pesar de que exista información de las romerías entre distintas páginas de turismo o portales oficiales de los municipios, la información se encuentra incompleta. Esto significa que el acceso a la información y a la compra de entradas de carrozas es desigual y pobre, afectando negativamente a la experiencia de los usuarios interesados.

El objetivo principal del proyecto consiste en desarrollar un modelo de negocio para una aplicación web de romerías, elaborándolo de tal manera que resulte un proyecto rentable. Asimismo, se elaborará un producto viable mínimo (MVP) que ofrecerá un sistema que mejore la experiencia de todas las partes interesadas, integrando información de romerías, gestión de carrozas y experiencias de usuarios.

1.4. JUSTIFICACIÓN DE LAS COMPETENCIAS ESPECÍFICAS CUBIERTAS

Competencias del Grado en Ingeniería Informática:

- CII01: 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.
- CII02: Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y mejora continua y valorando su impacto económico y social.
- CII08: Capacidad para analizar, diseñar, construir y mantener aplicaciones en un contexto tecnológico avanzado, con especial énfasis en plataformas web.

- IS01: Conocimiento de metodologías y estrategias de planificación de proyectos para el desarrollo de software.
- IS02: Capacidad de aplicar buenas prácticas en la integración de soluciones informáticas en el ámbito empresarial.
- IS04: Competencia en la administración y monitorización de recursos computacionales y en la gestión del ciclo de vida de sistemas complejos

El desarrollo de un producto mínimo viable de la aplicación web para digitalizar el proceso de reserva de entradas de carrozas en las romerías de la isla de Gran Canaria cumple con todas las competencias específicas listadas, incorporándolas en distintas fases del proyecto. Este trabajo mezcla el diseño tecnológico, la implementación práctica y el análisis de calidad, mostrando un enfoque completo para solucionar la problemática actual usando herramientas avanzadas tecnológicas.

La planificación y diseño de la aplicación empezaron con el estudio de los requerimientos y la creación de una arquitectura sólida, enfocada en la modularidad y la expansión. La inclusión de *Spring Boot* en el *backend* y *React JS* en el *frontend* ha permitido distinguir las distintas responsabilidades y garantizar un desarrollo fácil de mantener. Las pruebas unitarias y de integración son vitales para asegurar la confiabilidad y la calidad de la aplicación, usando *mocks* para simular escenarios realistas y verificar el comportamiento esperado.

Además, la seguridad es un aspecto clave en el diseño del sistema, haciendo uso de *Spring Security* en la autenticación de usuarios mediante *tokens* y controlando el acceso a las distintas funciones según los roles asignados. Así, se ha garantizado que los usuarios accedan únicamente a las funciones en las que están autorizados, asegurando también la protección de datos y una experiencia adaptada. En el *frontend*, el uso de *AuthContext* en *React* permite una gestión del estado de autenticación y los roles, posibilitando controlar de forma dinámica el acceso a las vistas y mejorando el rendimiento al recargar únicamente los componentes necesarios.

La integración y el despliegue continuo, gestionado a través de *GitHub*, asegura un seguimiento constante del progreso. Además, el uso de una arquitectura hexagonal y

la implementación de pruebas automatizadas, son prácticas modernas que aseguran la evolución de acuerdo a las necesidades futuras.

En cuanto a los impactos económico y social, el proyecto pretende facilitar el acceso a las romerías, fomentando la inclusión de turistas y residentes. Esto refleja el compromiso con el sistema de mejora continua y su impacto positivo socio-económico.

La integración de las distintas competencias, muestran la capacidad para resolver problemas reales con un enfoque profesional.

Competencias del Grado en Administración y Dirección de Empresas:

- CE1: Capacidad de aplicar los conocimientos en la práctica.
- CE2: Habilidad para el diseño y gestión de proyectos.
- CE7: Comprensión de la relación entre la empresa y su entorno.
- CE9: Identificación y resolución de problemas económicos en empresas, utilizando herramientas específicas.
- CE11: Capacidad para tomar decisiones estratégicas basadas en modelos empresariales.
- CE17: Habilidad para redactar proyectos de gestión global o de áreas funcionales de una empresa.
- CE19: Derivación de información relevante de datos complejos que no sean reconocibles para no profesionales

La elaboración del modelo de negocio sobre la aplicación web de romerías cumple con todas las competencias específicas listadas previamente. A través del análisis en profundidad de cada punto del modelo, se abordan cuestiones relacionadas con la estrategia, el análisis del mercado y la viabilidad económica y financiera.

Se identifican las problemáticas actuales en el sector de las festividades de la isla, permitiendo definir una propuesta de valor clara. Asimismo, el análisis de la competencia y la identificación de los segmentos de clientes y canales ha permitido identificar un contexto y lograr una solución adaptada a las necesidades del mercado.

El análisis económico y financiero se ha desarrollado considerando múltiples variables, por lo que se logra que surja un modelo sostenible y escalable. Con esto, se determina los flujos de ingresos, la estructura de costes y las actividades principales para el éxito del negocio. Además, a través de las relaciones con los clientes y las alianzas se refuerza el valor generado a largo plazo para todos los actores.

Adicionalmente, se destaca la capacidad de transformar datos que pueden ser complejos en información relevante para la toma de decisiones estratégicas. Así, cada decisión está respaldada con fundamentos gracias a los análisis previos que se verán reflejados en el modelo de negocio.

1.5. APORTACIONES

Este trabajo pretende contribuir en distintos entornos:

- Cultural y social: pretende hacer una mejora en la experiencia de romerías para locales y turistas al digitalizar un proceso actualmente muy manual y centralizar la información en un solo sitio.
- Tecnológico y de negocio: la introducción de una herramienta tecnológica que facilitará la organización y gestión de romerías.
- Administración pública: se proveerá de un sistema que facilitará a las autoridades la promoción y la gestión de estas festividades tradicionales.
- Desarrollo sostenible: se refuerza un impacto económico en las romerías respetando sus valores tradicionales.

1.6. NORMATIVA Y LEGISLACIÓN

Reglamento General de Protección de Datos (RGPD) – UE 2016/679

Este reglamento regula la protección de datos, obligando a solicitar el consentimiento explícito para tratar los datos, implementando medidas técnicas que garanticen la seguridad de los datos e incluyendo funcionalidades de derecho al olvido y portabilidad de datos.

Actualmente, el proyecto desarrollado garantiza el encriptado de la contraseña de los usuarios y se incluye un mecanismo de tokens en el *backend* implementados con *Spring Security*.

JetBrains Educational Licences (IntelliJ IDEA)

Esta licencia educativa se ha utilizado para poder hacer uso del editor de código *IntelliJ IDEA*, pudiendo disfrutar de todos sus beneficios al hacer uso de una versión completa.

Microsoft 365 Educational License (Word)

Para el desarrollo de la memoria presente, se ha utilizado la licencia educativa ofrecida por la universidad, permitiendo un acceso completo a la herramienta.

Licencias Open Source (Spring Boot y React)

Los frameworks mencionados se han utilizado bajo licencias abiertas.

Licencia GPL v2 (GNU General Public License)

Git se ha utilizado de forma gratuita bajo por la licencia mencionada.

2. MODELO DE NEGOCIO

En este capítulo se presenta el Modelo *Canvas*, una metodología creada por Alexander Osterwalder e Yves Pigneur, que facilita el desarrollo, la visualización, la evaluación y la modificación de modelos de negocios nuevos o existentes.

Este modelo está compuesto por nueve módulos: propuesta de valor, segmento de clientes, relaciones con clientes, canales, fuentes de ingresos, actividades clave, recursos clave, alianzas y costes de estructura. Además, también se presentará un análisis de la competencia que es clave para diferenciarse en el mercado y conseguir el éxito.

2.1. ANÁLISIS DE LA PROPUESTA DE VALOR

2.1.1. IDENTIFICACIÓN DEL PROBLEMA

Actualmente, una dificultad que encuentran los participantes de las romerías en la isla de Gran Canaria, es la falta de una plataforma centralizada que ofrezca información

completa y actualizada sobre este tipo de eventos, así como la posibilidad de reservar una plaza en las carrozas participantes de forma eficiente y con anticipación. A día de hoy, la información disponible está dispersa entre medios analógicos, páginas web o redes sociales, y en muchas ocasiones se encuentra desactualizada, por lo que se dificulta la participación tanto para turistas como para residentes. Además, el proceso de reserva de carrozas es manual, lo cual limita la oferta y provoca un uso ineficiente de los recursos disponibles por parte de los cocheros.

Así mismo, los cocheros carecen de una herramienta digital eficaz que les permita gestionar las entradas a sus carrozas, optimizando así las reservas y aumentando la visibilidad entre residentes y turistas de la isla. Por otra parte, los organizadores de las romerías también encuentran dificultades en cuanto a la coordinación y la logística del evento.

2.1.2. PROPUESTA DE VALOR

‘Romería Viva’ propone una solución integral que resuelve todas las problemáticas presentadas mediante la centralización de la información y la digitalización del proceso de compra-venta de entradas de carrozas. Esta aplicación web ofrece a usuarios, cocheros y responsables una plataforma de gestión de romerías para tratar todos los aspectos de estas.

Para los turistas y residentes (usuarios):

- Acceso a toda la información actualizada: la aplicación proporciona información actualizada sobre cada romería de la isla, aportando una descripción, el lugar donde se celebrará, el horario, la ruta, e incluso otros datos de interés como fotos de otros años y enlace a la página web oficial (ya sea del ayuntamiento del lugar, o una de la propia romería).
- Compra/venta de entradas: los usuarios pueden ver las carrozas participantes de cada romería y reservar una plaza mediante la compra de entradas. De esta forma, se elimina todo el proceso manual para poder acceder a una, aumentando también la accesibilidad a turistas o residentes de otros municipios que quieran experimentar esta vivencia tan característica de una romería.

- Facilitar la planificación y participación: al encontrar la información de todas las romerías de la isla en un solo lugar, los usuarios podrán conocer más de ellas y así planificar a cuáles asistirán de forma más cómoda y sencilla.

Para los cocheros:

- Herramienta de gestión de entradas: la aplicación proporciona un sistema sencillo de gestión de entradas para que los cocheros puedan añadir sus carrozas a las romerías que deseen, aportando información como el tipo de comidas/bebidas que ofrecerán, precios, y el máximo de personas/entradas que se venderán.
- Mayor visibilidad: al utilizar esta plataforma, los cocheros tienen un público objetivo más grande ya que conseguirán llegar a más usuarios, maximizando su rentabilidad.
- Optimización de recursos: con el uso de la página web, los cocheros podrán mejorar sus recursos y minimizar el riesgo de impago/cancelaciones, además de mejorar los posibles problemas de capacidad.

Para los organizadores:

- Gestión centralizada del evento: los organizadores de romerías, podrán asegurar una mejor coordinación entre los participantes y las carrozas del evento.
- Promoción mejorada: al tener toda la información de romerías de la isla en un solo lugar, se puede mejorar la difusión de los próximos eventos, además de lograr que la información sea accesible y actual.
- Gestión de romerías: con el uso de la plataforma, los organizadores de romerías podrán saber el total de carrozas que participarán, y el volumen aproximado de personas que asistirán al evento, por lo que se podrá realizar una mejor gestión y preparación, adecuándola al número de asistentes.

2.1.3. DIFERENCIACIÓN FRENTE A OTRAS SOLUCIONES

A día de hoy, no existe ninguna plataforma que ofrezca una solución que integre la gestión completa de romerías. Si bien es cierto que existen webs o redes sociales en las que se ofrece esta información sobre eventos locales, no incluye todas las romerías, y ninguno ofrece la posibilidad de comprar entradas para una carroza. La diferenciación de esta propuesta de valor se basa en:

- Funcionalidad única: además de la información centralizada de romerías, la aplicación permite la compra-venta de entradas para carrozas, que es algo que no está disponible en ninguna de las soluciones actuales.
- Enfoque en la gestión de carrozas: la aplicación permite a los cocheros gestionar sus carrozas y maximizar la venta de entradas, optimizando así su negocio, algo que actualmente no logra ninguno de los servicios actuales.
- Interacción entre usuarios y cocheros: facilita la conexión entre ambas partes participantes de la romería, lo cual mejora la experiencia y reduce cualquier problema en el proceso de reserva.

2.1.4. BENEFICIOS DE LA PROPUESTA DE VALOR

- Para los participantes: mayor accesibilidad a la información completa de todas las romerías, incluyendo horarios, lugares y carrozas disponibles. Un proceso manual, caótico y poco accesible como era la reserva de una plaza en una carroza, se convierte en una experiencia digital sencilla.
- Para los cocheros: mayor eficiencia en la gestión de sus carrozas, mayor atracción de clientes y optimización de la oferta, además de la gestión de la capacidad de la carroza y la reducción del riesgo de impago.
- Para los organizadores: mayor control del evento, centralización de la información y facilidad de mantenimiento para tener la información actualizada.

2.1.5. RESULTADOS ESPERADOS

La implementación de esta página web pretende generar una mejora significativa en la organización y disfrute de las romerías. Se espera:

- Mayor satisfacción de usuarios: al ofrecer acceso a toda la información centralizada y con la reserva de entradas en las carrozas, mejorará la experiencia de los participantes en las romerías, haciéndolas más atractivas para residentes de otros municipios y turistas.
- Optimización de los recursos para los cocheros: gracias a la visibilidad ofrecida por la plataforma, los cocheros podrán aumentar el volumen de ventas para las entradas de sus carrozas, maximizando su rentabilidad.
- Facilitar el trabajo de los organizadores: se dispondrá de una herramienta única para poder mantener la información actualizadas y así poder coordinar y gestionar de forma más sencilla las romerías.

2.1.6. CONCLUSIÓN

La propuesta de valor de ‘Romería Viva’ se basa en resolver la falta de información que actualmente hay sobre las romerías, la gestión de entradas de las carrozas (que en muchas ocasiones resulta inaccesible), y los posibles problemas de organización, proporcionando una solución integral para los distintos integrantes del evento. Con la implementación de esta plataforma, se mejora la experiencia de usuario y se maximiza la eficiencia del evento. Así, se genera un valor añadido inexistente hasta ahora en el mercado, consiguiendo que el uso de esta herramienta sea indispensable para la gestión de romerías.

2.2. ANÁLISIS DE LA COMPETENCIA

2.2.1. IDENTIFICACIÓN DE COMPETIDORES DIRECTOS E INDIRECTOS

A pesar de que no existe ninguna aplicación específica para la gestión de romerías, sí que existen competidores tanto directos como indirectos de aplicaciones similares.

Competencia directa

Dentro de la competencia directa, se pueden considerar las aplicaciones o sitios web de gestión de eventos y compra-venta de entradas.

Los competidores más relevantes son:

Eventbrite

Con más de 10 millones de descargas en *playstore*, *Eventbrite* es un sitio web de gestión de eventos que permite a los organizadores de eventos crear, promocionar y vender entradas para todo tipo de eventos.

Sus características principales se centran en la creación de eventos, la venta de entradas, la publicidad y el marketing, la gestión de invitados y posterior informe y análisis.

- Crear un evento: los organizadores pueden crear un evento con detalles como fecha, ubicación, descripción y categoría.
- Venta de entradas: promueve la venta de entradas y gestiona el registro de asistentes.
- Publicidad y *marketing*: se brindan herramientas para promocionar el evento a través de redes sociales y el correo electrónico.
- Gestión de invitados: se puede administrar una lista de invitados y controlar así el acceso a un evento.
- Informes y análisis: proporciona informes y análisis detallados sobre la asistencia y el rendimiento de los eventos.

La interfaz de usuario es intuitiva y fácil de usar tanto para los anfitriones como para los participantes. Además, proporciona una aplicación móvil que permite a los usuarios encontrar eventos cercanos y comprar entradas desde el dispositivo móvil.

En cuanto a las fuentes de ingresos, *Eventbrite* cobra a los organizadores de eventos una tarifa de transacción basada en el precio de entrada, además de las tarifas aplicadas en los procesamientos de pagos.

Eventbrite se utiliza ampliamente en todo el mundo y ha demostrado ofrecer una amplia gama de eventos, desde pequeñas reuniones locales hasta grandes festivales.

Las opiniones de los usuarios destacan la facilidad de uso y la eficacia de la plataforma para planificar y promover eventos.

Meetup

Meetup es una red social que facilita la creación y asistencia a eventos locales conectando a personas con intereses similares.

Sus características principales son la creación de grupos y de eventos, el descubrimiento de eventos, la interacción social, y las notificaciones y recordatorios.

- Crear grupos y eventos: permite a los usuarios crear grupos y organizar eventos locales.
- Descubrimiento de eventos: facilita a los usuarios descubrir eventos según sus intereses o ubicación.
- Interacción social: promueve la interacción social entre miembros de la comunidad local.
- Notificaciones y recordatorios: permite el envío de notificaciones y recordatorios de eventos.

Meetup se centra en construir una comunidad local y promueve la participación activa. La plataforma tiene una interfaz de usuario sencilla que facilita la navegación y la participación en eventos.

En cuanto a sus fuentes de ingreso, se ofrecen distintos planes de suscripción para organizadores de grupos. Además, aplica tarifas por evento a los organizadores.

Meetup ha tenido éxito en la conexión de personas locales con intereses comunes por todo el mundo.

Los usuarios aprecian la oportunidad de conocer gente con intereses similares y la variedad de eventos que se ofrecen.

Ticketmaster

Liderando la venta de entradas para ocio y espectáculos, *Ticketmaster* es una plataforma global de venta de entradas que conecta a los planificadores de eventos con personas interesadas en asistir a conciertos, eventos deportivos y otras experiencias en vivo.

Entre sus características principales se encuentran la venta de entradas, un mapa de asientos, y las notificaciones de los eventos.

- Venta de entradas: ofrece una amplia gama de eventos para los que los usuarios pueden adquirir entradas.
- Mapa de asientos: proporciona un mapa de asientos detallado para ayudar a los usuarios a seleccionar una ubicación específica.
- Notificaciones de eventos: los usuarios pueden recibir notificaciones sobre eventos de sus artistas o equipos favoritos.

La plataforma tiene una interfaz fácil de usar que facilita la exploración y obtención de entradas. Además, cuenta con una aplicación móvil que permite a los usuarios comprar y administrar entradas a través de sus dispositivos.

En cuanto a sus fuentes de ingreso, *Ticketmaster* cobra tarifas de transacción y tarifas de servicio según el precio de la entrada. Además, también crea colaboraciones con los planificadores de eventos para la venta exclusiva de entradas.

Ticketmaster es una de las plataformas de venta de entradas más grandes y populares en todo el mundo.

Si bien algunos usuarios cuestionan las tarifas adicionales, muchos aprecian la conveniencia y la variedad de eventos disponibles.

Stubhub

Stubhub es una plataforma de reventa de entradas que permite a los usuarios comprar y vender entradas para eventos en directo.

Entre sus características principales se encuentran la reventa de entradas, la garantías de entrada y el plano de asientos.

- Reventa de entradas: los usuarios pueden comprar y vender entradas para eventos que ya están agotados o no están disponibles para la venta general.
- Garantías de entrada: proporciona garantías para proteger a compradores y vendedores en las transacciones.
- Plano de asientos: proporciona un plano de asientos detallado.

La plataforma es fácil de usar tanto para la compra como para la venta de entradas. Además, cuenta con una aplicación móvil que permite a los usuarios acceder a *StubHub* desde cualquier lugar.

En cuanto a las fuentes de ingreso, se aplican tarifas de transacción tanto a los compradores como a los vendedores. Se cobra una tarifa por cada venta exitosa de entradas.

StubHub es una de las plataformas líderes en reventa de entradas y cada vez se está haciendo más popular, especialmente en eventos deportivos y conciertos.

Los usuarios aprecian poder asistir a eventos con entradas agotadas, pero algunos critican las tarifas adicionales.

Competencia indirecta

Dentro de la competencia indirecta, se pueden incluir los medios que proporcionan información sobre el evento, pero nada más allá, sin permitir una gestión avanzada del evento como la venta de entradas.

Facebook

Facebook es una red social global que permite a los usuarios conectarse y compartir contenido multimedia con amigos, familiares y la comunidad.

Dentro de sus rasgos destacados en el marco de gestión de eventos, encontramos la creación de eventos, los grupos y páginas y las notificaciones de eventos.

- Eventos en *Facebook*: los usuarios tienen la posibilidad de crear, buscar y participar en eventos locales.

- Grupos y páginas: los grupos y páginas dedicados a eventos locales brindan un espacio para promocionar y participar en el evento.
- Notificaciones de eventos: los usuarios reciben notificaciones sobre eventos de interés.

La interfaz de *Facebook* resulta familiar para la mayoría de los usuarios, lo cual facilita unirse a los eventos. Además, los usuarios pueden compartir fotos y videos de sus eventos para enriquecer su experiencia.

En cuanto a las fuentes de ingresos, *Facebook* gana dinero a través de la publicidad y permite a los organizadores de eventos promocionar sus eventos a través de anuncios pagados.

Facebook es la red social más grande del mundo y se utiliza para promocionar y descubrir diversos eventos.

Si bien la plataforma es elogiada por su conveniencia para encontrar y asistir a eventos locales, algunos usuarios pueden encontrar la experiencia abrumadora debido a la cantidad de contenido.

Páginas web de turismo

Las páginas web de turismo son sitios que proporcionan información de interés turístico, ya sea información sobre destinos, eventos locales o actividades de ocio para realizar en el destino. Están diseñadas para ayudar a los turistas a descubrir lugares de interés y eventos locales.

Dentro de sus rasgos destacados en el marco de gestión de eventos, se puede destacar la información de eventos, la promoción de festividades y los mapas y rutas.

- Listado de eventos: las páginas de turismo suelen contar con secciones específicas para conocer los eventos locales, ofreciendo información detallada de estos.
- Promoción de festividades: suelen ofrecer una sección para promocionar los eventos locales, aportando enlaces adicionales para mayor información.

- Mapas y rutas: algunas páginas de turismo ofrecen mapas interactivos para localizar más fácilmente los eventos.

La interfaz de una página de turismo suele ser intuitiva y amigable, aunque la calidad de experiencia de usuario varía según el diseño. Los usuarios pueden conocer información sobre los eventos, pero no se suele permitir una reserva directa de entradas.

En cuanto a los ingresos, las páginas de turismo suelen obtener sus ingresos a través de la publicidad de alojamientos, restaurantes y actividades, así como mediante colaboraciones.

A pesar de que estas páginas resultan confiables para encontrar información, carecen de funcionalidades interesantes como la reserva de entrada o gestión de eventos.

2.2.2. COMPARATIVA DE FUNCIONALIDADES

En la siguiente tabla 1, se muestra una comparación entre nuestra aplicación y las alternativas disponibles, enfocándose en las funcionalidades clave.

COMPARATIVA DE FUNCIONALIDADES

	ROMERÍA VIVA	EVENTBRITE	MEETUP	TICKETMASTER	STUBHUB	FACEBOOK	PÁGINAS TURISMO
Información sobre eventos	✓	✓	✓	✓	✓	✓	✓
Reserva plazas carrozas	✓	✗	✗	✗	✗	✗	✗
Compra/venta entradas	✓	✓	✗	✓	✓	✗	✗
Creación y promoción de eventos	✓	✓	✓	✗	✗	✓	✓
Notificaciones personalizadas	✓	✓	✓	✓	✗	✓	✗
Gestión de carrozas	✓	✗	✗	✗	✗	✗	✗
Mapas interactivos	✓	✗	✗	✓	✓	✗	✓
Comentarios y reseñas	✓	✓	✗	✗	✗	✓	✗
Monetización	Comisiones, planes, publicidad, colaboraciones	Tarifas de transacción	Suscripciones para organizadores	Tarifas de transacción	Tarifas de transacción	Publicidad pagada	Publicidad, colaboraciones
Promoción de cocheros	✓	✗	✗	✗	✗	✗	✗

Tabla 1. Comparativa de funcionalidades entre Romería Viva y sus competidores.

Fuente: elaboración propia.

2.2.3. ANÁLISIS DAFO

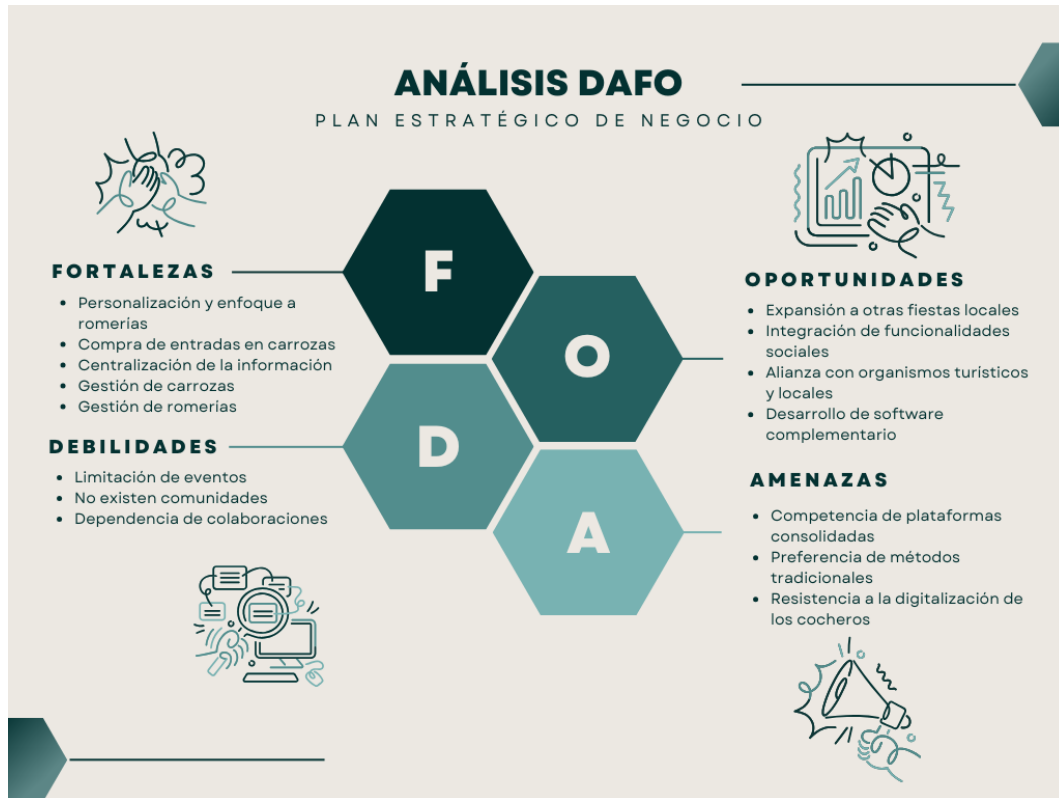


Figura 1. Análisis DAFO.

Fuente: elaboración propia.

Una herramienta fundamental para valorar la situación actual y de crecimiento en el mercado con respecto a la competencia, es el análisis DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades), desarrollado por Albert Humphrey en la década de 1960.

Debilidades:

- Limitación de eventos: la aplicación se centra exclusivamente en romerías, lo cual puede limitar su atractivo.
- Interacción social: actualmente, la aplicación no permite crear grupos o fomentar una comunidad como Facebook o Meetup.
- Dependencia de colaboración con cocheros: la aplicación depende de la adaptabilidad de los cocheros para pasarse a un formato tecnológico, algo que es un desafío dado que hasta ahora ha funcionado con boca a boca.

Amenazas:

- Competencia de plataformas consolidadas: las plataformas actuales como Eventbrite, Meetup o Ticketmaster ofrecen una gestión de eventos más ampliada con funciones más desarrolladas en la compra-venta de entradas.
- Preferencia de métodos tradicionales: los usuarios residentes tienden a escoger un método más tradicional a la hora de comprar entradas para una carroza por su familiaridad.
- Resistencia a la digitalización de los cocheros: los cocheros tradicionales podrían ser reacios a un cambio de modelo digital si considerasen que la tecnología es demasiado complicada.

Fortalezas:

- Mayor personalización y enfoque de romerías: no existe ninguna aplicación que se centre específicamente en romerías, pues actualmente el resto tratan eventos de forma global.
- Compra de entradas en carrozas: la aplicación permite la compra-venta de entradas en carrozas en formato digital, lo cual puede facilitar la participación de un público que hasta ahora no ha podido debido a las limitaciones de la compra de entrada en formato boca a boca, que depende de conocer a alguno de los cocheros.
- Centralización de la información: a pesar de que algunas webs ofrezcan información sobre romerías, ninguna centraliza la información, por lo que en la aplicación se ofrece información más destallada y relevante, así como un listado de todas ellas.
- Gestión de carrozas: con la aplicación, se facilita la organización para los cocheros, pudiendo maximizar la venta de entradas al atraer a más público del usual, además de tener un mayor control sobre las entradas vendidas para poder organizar mejor la carroza en el evento.

- Gestión de romerías: gracias a la aplicación, se podrá tener una mayor gestión de las romerías, ya que se podrán ver la totalidad de carrozas participantes y la aproximación de asistentes.

Oportunidades:

- Expansión a otras fiestas locales: la aplicación fácilmente podría crecer y escalar para cubrir otras festividades tradicionales, ampliando así su mercado objetivo y fortaleciendo el ya creado.
- Integración de funcionalidades sociales: se pueden desarrollar herramientas que permitan a los usuarios interactuar entre ellos y crear grupos/comunidades en los que compartir experiencias.
- Alianzas con organismos turísticos y locales: colaborar con oficinas turísticas y empresas locales para promocionar la aplicación y captar un público mayor. La web podría convertirse en la herramienta principal que utilicen los turistas para conocer la cultura de la isla.
- Desarrollo de software complementario para los organizadores: se podría desarrollar una herramienta complementaria para los organizadores de romerías en el que pudiesen tener el control sobre otros aspectos de la romería, elaborando estadísticas y mejorando la organización

2.2.4. CONCLUSIÓN

Tras el análisis de la competencia junto al análisis DAFO, se revela que a pesar de que existen alternativas para obtener información sobre romerías o la gestión de eventos en general, ‘Romería Viva’ cuenta con un enfoque más específico. Si bien es cierto que se encuentra con desafíos importantes relacionados con la preferencia de los usuarios por canales más informales, la aplicación presenta una ventaja competitiva sobre las herramientas existentes al facilitar la gestión de carrozas para los cocheros.

Romería Viva ofrece una solución única para un amplio público desatendido de usuarios, cocheros y organizadores de romerías, ofreciéndoles una experiencia más integrada.

2.3. ANÁLISIS DEL SEGMENTO DE CLIENTES

2.3.1. IDENTIFICACIÓN DE LOS SEGMENTOS DE CLIENTES

El segmento de clientes se divide en tres grupos principales que utilizarán la aplicación:

- Participantes de las romerías (locales y turistas).
- Cocheros (personas que ofrecen carrozas).
- Organizadores de romerías (entidades públicas o privadas encargadas de gestionar y coordinar los eventos).

2.3.2. ANÁLISIS DEL SEGMENTO: PARTICIPANTES EN LAS ROMERÍAS

Características clave:

- Demografía: adultos jóvenes y familias residentes, así como los turistas que deseen conocer la cultura a través de los eventos tradicionales. Abarcarían un amplio rango de edad, aunque principalmente el uso de la aplicación será de mayor interés para personas entre 18 y 60 años.
- Comportamientos: búsqueda de información sobre romerías, como lugares y horarios; desean una experiencia cómoda para participar en primera línea a través de una carroza.
- Necesidades: información centralizada y actualizada de las romerías, facilidad de compra-venta de entradas a carrozas.

La aplicación cubre sus necesidades al tener toda la información de romerías en un mismo lugar, además de la posibilidad de comprar entradas para carrozas de forma ágil y segura.

2.3.3. ANÁLISIS DEL SEGMENTO: COCHEROS

Características clave:

- Demografía: principalmente propietarios de carrozas con experiencia en la gestión de estos vehículos durante una romería.

- Comportamientos: gestionan la disponibilidad de sus carrozas manualmente mediante mensajes o llamadas, recibiendo dinero en efectivo; desean atraer clientela para aumentar la ocupación de sus carrozas.
- Necesidades: visibilidad entre los participantes de una romería, herramienta para gestionar las reservas, la disponibilidad y la recepción del dinero mediante medios digitales y la optimización de la ocupación de sus carrozas.

La aplicación cubre sus necesidades al ofrecer una herramienta de gestión de carrozas en el que pueden indicar el número total de entradas disponibles, dándoles visibilidad a un mayor público objetivo.

2.3.4. ANÁLISIS DEL SEGMENTO: ORGANIZADORES DE ROMERÍAS

Características clave:

- Demografía: entidades locales o ayuntamientos. El tamaño dependerá de la entidad ya que podría ir desde pequeños grupos comunitarios como una asociación de vecinos hasta instituciones gubernamentales.
- Comportamientos: coordinan las romerías, rutas y horarios, así como la disponibilidad de carrozas. Además, se encargan de la promoción del evento.
- Necesidades: una plataforma centralizada en el que se permita tener una visualización global de las próximas romerías con sus fechas y lugares, además de un lugar donde poder poner y actualizar toda la información del evento para facilitar su promoción.

La aplicación cubre sus necesidades al ofrecer una plataforma en la que se encuentran todas las romerías de la isla, pudiendo mejorar la organización para no coincidir con otras, y además optimizar la gestión al conocer la cantidad aproximada de asistentes gracias al número de carrozas participantes y la cantidad de entradas vendidas. Además, a raíz de los comentarios obtienen una retroalimentación para mejorar el próximo evento.

2.3.5. CONCLUSIÓN

Este análisis muestra que la aplicación de romerías posee un mercado potencial en las partes participantes de una romería, cubriendo las necesidades principales de cada uno y aportando un valor único a todos los actores involucrados para mejorar su experiencia.

2.4. ANÁLISIS DE LOS CANALES

2.4.1. IDENTIFICACIÓN DE LOS CANALES

Los canales son los medios a través de los cuales la aplicación se pone en contacto con los usuarios. Estos son imprescindibles para informar a los usuarios, permitirles comprar entradas, e incluso facilitan la promoción de la aplicación para llegar a más clientes.

2.4.2. CANALES DE COMUNICACIÓN

Los canales de comunicación se centran en la promoción de la aplicación, estos canales incluirían:

- Redes sociales: *Instagram, Tiktok, Facebook y Twitter* son imprescindibles hoy día para hacer promociones y llegar a más público, por lo que son canales cruciales para captar más clientes.
- *Marketing* digital: otro canal importante es el *marketing* digital, el realizar campañas de anuncios pagados permite que más gente interesada en este tipo de eventos conozca la aplicación.
- Colaboraciones con entidades locales: otra opción para hacer llegar la aplicación a los potenciales clientes, es establecer colaboraciones con ayuntamientos, empresas de turismo y hoteles que recomendasen y promoviesen el uso de la aplicación entre sus clientes.

2.4.3. CANALES DE DISTRIBUCIÓN

Los canales de distribución son los medios por los que los clientes accederían al producto, que en el caso de 'Romería Viva' sería:

- Aplicación web: la plataforma web permite que cualquier usuario con internet acceda a la aplicación. La web es el canal principal de distribución, al cual se podrá acceder desde un ordenador, una tablet o un teléfono móvil.

2.4.4. CANALES DE SOPORTE

Para mejorar la experiencia del usuario, se necesitan canales de soporte para cualquier problema que pudiera existir. En este caso, sería:

- Número teléfono en *whatsapp*: para consultas rápidas y de fácil resolución, existirían un número de atención al cliente al que enviar un mensaje y recibir ayuda en el uso de la aplicación.
- Correo electrónico: para consultas complejas que requiriesen de una mayor atención, existiría un correo de atención al cliente al que enviar cualquier problema.

2.4.5. CONCLUSIÓN

Los canales son imprescindibles para mejorar la experiencia del cliente. A través de los diferentes medios, se llegará a más público interesado en las fiestas locales de la isla, y evaluando su eficacia, permitirá mejorar la aplicación, posicionándola como una herramienta clave para residentes y turistas.

2.5. ANÁLISIS DE LAS RELACIONES CON EL CLIENTE

2.5.1. IDENTIFICACIÓN DE LAS RELACIONES CON EL CLIENTE

Las relaciones con el cliente son una parte clave para el éxito de la aplicación de romerías, ya que se determina la forma en la que se interactúa y contacta con un usuario en la aplicación. Las claves en las relaciones con el cliente se centran en fidelizar a los usuarios a la aplicación, asegurando que la experiencia dentro de esta sea satisfactoria para retenerlos y crearles valor a largo plazo.

2.5.2. RELACIONES CON LOS PARTICIPANTES DE LAS ROMERÍAS

Los participantes de las romerías son la clave para que el negocio funcione, por lo que el objetivo es crearles una experiencia positiva, incentivando a que continúen usando la aplicación a lo largo del tiempo. Los tipos de relaciones son:

- Autogestión y facilidad de uso: el hecho de poder gestionar la compra de entradas de manera autónoma, además de ver información actualizada de romerías y disponibilidad de carrozas.
- Promociones y descuentos: para los usuarios más activos en la aplicación, existirá un programa de fidelización, en el que se ofrezcan descuentos o reservas sin comisiones tras asistir a un número determinado de romerías.
- Planes *premium*: los participantes podrán suscribirse a un plan premium para comprar entradas sin comisiones. Además, también se podría implementar acceso anticipado a las carrozas más demandadas. De esta forma, se creará una relación a largo plazo entre el cliente y la aplicación.
- Soporte técnico: en caso de cualquier problema, los usuarios contarán con soporte a través de un número de whatsapp y correo electrónico.

El objetivo de la relación es fidelizar al usuario básico mediante incentivos y planes que agreguen valor, asegurando su completa satisfacción.

2.5.3. RELACIONES CON LOS COCHEROS

Los cocheros son indispensables para la aplicación, ya que el servicio clave que se ofrece es la compra-venta de entradas en carrozas. Por ello, la relación con ellos se centrará en mejorar su visibilidad y facilitar la gestión de entradas. Los tipos de relaciones son:

- Soporte técnico personalizado: los cocheros contarán con soporte que les ayude a configurar su perfil y crear las carrozas necesarias para añadir a las distintas romerías. Este soporte se hará a través de tutoriales y una persona que pueda resolver cualquier problema técnico que pudiese surgir.

- Relación de colaboración: se fomentará una relación cercana para conseguir que los cocheros aprecien la plataforma y colaboren de forma activa añadiendo sus carrozas.

El objetivo de esta relación se centra en optimizar el sistema de gestión de reservas y aumentar la rentabilidad de los cocheros, creando una relación de confianza y colaboración para mejorar su experiencia dentro de la aplicación.

2.5.4. RELACIONES CON LOS ORGANIZADORES DE LAS ROMERÍAS

La relación con los organizadores se debe basar en una colaboración estratégica que les ayude a gestionar la logística de las romerías de la manera más eficiente posible. Los tipos de relaciones son:

- Soporte continuo: los organizadores contarán con un soporte técnico personalizado que les ayude a gestionar los datos logísticos para organizar las romerías.
- Colaboración estratégica: se establecerá una relación de colaboración con los organizadores en forma de retroalimentación para mejorar la plataforma y optimizar la organización de las romerías.

El objetivo de esta relación es fomentar el uso de la aplicación por parte de los organizadores como herramienta principal para la gestión logística y la coordinación de las romerías.

2.5.5. EVALUACIÓN DE LA EFICACIA DE LAS RELACIONES

Existen varios indicadores clave para medir el éxito de las relaciones con los clientes:

- Tasa de retención: el uso repetitivo de la aplicación en las distintas romerías.
- Satisfacción del cliente: a través de encuestas o valoraciones tras algunos trámites claves como la compra de una entrada.
- Uso de planes *premium*: número de usuarios que optan por suscribirse a un plan premium.

- Participación en promociones: cuántos usuarios utilizan los descuentos y cómo influyen en el uso de la aplicación.

2.5.6. CONCLUSIÓN

El análisis sobre las relaciones con el cliente destaca las distintas formas en las que la aplicación aportará valor a los usuarios a través de medios como el soporte continuo, relaciones estratégicas o planes de suscripción que fortalecerán el crecimiento de la plataforma.

2.6. ANÁLISIS DEL FLUJO DE INGRESOS

2.6.1. INTRODUCCIÓN AL ANÁLISIS DE LOS FLUJOS DE INGRESOS

El componente más importante dentro del modelo de negocios de la aplicación de romerías, sería el flujo de ingresos, ya que aquí se define como se monetizará la aplicación y cómo se generarán beneficios, asegurando la viabilidad económica en el largo plazo. Para maximizar los ingresos de la plataforma, habrá que combinar distintos modelos de monetización enfocados a los diferentes segmentos de clientes.

2.6.2. INGRESOS POR COMISIONES EN LAS RESERVAS DE LAS CARROZAS

La forma principal en la que la aplicación generará ingresos, será en forma de comisiones por cada compra de entrada de una carroza. De esta forma, se obtendrán ingresos constantes sin que suponga un cargo inicial alto para los clientes. Se aplicará una comisión del 15% sobre el precio de reserva, por lo que siendo el precio promedio de 20€ por entrada, la comisión generaría 3€ de ingreso por cada transacción.

- Segmento objetivo: participantes de la romería que usan la aplicación para comprar entradas de carrozas.
- Estimación de ingresos: se estima que la aplicación la utilicen el primer año un 5-10% de los asistentes (unos 1.000 usuarios iniciales) y tan solo un 50% (unas 500 personas) serán usuarios activos de la aplicación, realizando aproximadamente unas 3 reservas al año, lo que resultará en 1.500 entradas anuales que equivaldrían a unos ingresos aproximados de 4.500€ el primer año.

2.6.3. INGRESOS POR SUSCRIPCIONES *PREMIUM* PARA USUARIOS

La aplicación ofrecerá un plan *premium* para quienes deseen una experiencia mejorada al eliminar las comisiones de las entradas y posibilitando un acceso anticipado a las carrozas. Basándonos en que hay una aproximación de 30 romerías al año en la isla de Gran Canaria, el plan de suscripción mensual será de 5€, y el anual de 30€.

- Segmento objetivo: usuarios que utilicen constantemente la aplicación para comprar entradas y deseen obtener beneficios adicionales, reduciendo sus costes de reserva de carroza, y eliminando los anuncios.
- Estimación de ingresos: se estima de que sólo un 10% de los usuarios activos (unas 50 personas), optarán por el plan *premium*, de los cuales el 70% optará por un plan de suscripción mensual y el otro 30% el plan anual, generando así 2.550€ el primer año.

2.6.4. INGRESOS POR SUSCRIPCIÓN *PREMIUM* PARA COCHEROS

La aplicación ofrecerá un plan *premium* para los cocheros de carrozas, ofreciéndoles mayor visibilidad en la plataforma (por ejemplo, colocándoles al principio en la lista de carrozas de una romería) y dándoles acceso a herramientas más avanzadas como estadísticas de reservas. La suscripción será de 100 € anuales.

- Segmento objetivo: los cocheros que deseen destacar su carroza y optimizar la ocupación de esta.
- Estimación de ingresos: se estima que un 30% de los 50 cocheros registrados opten por el plan *premium*, generando así unos 1500€ el primer año.

2.6.5. INGRESOS POR SUSCRIPCIÓN *PREMIUM* PARA ORGANIZADORES DE ROMERÍAS

La aplicación ofrecerá un plan *premium* para los organizadores de romerías, ofreciéndoles una plataforma de gestión logística de las romerías, incluyendo el control de las carrozas, la venta de entradas, y la comunicación con los participantes. Además, podrán tener una visión general del volumen de participación. La suscripción será de 20€/romería.

- Segmento objetivo: organizadores de romerías, ya sean administraciones públicas o asociaciones de vecinos.
- Estimación de ingresos: estos ingresos se darán a partir del segundo año, pues el primer año se ofrecerá la herramienta de forma gratuita para potenciar su uso.

2.6.6. INGRESOS POR PUBLICIDAD

Otra de las formas de ingreso de la aplicación web, será a través de campañas publicitarias con empresas locales. Esta publicidad se realizará a través de banners y seguirá el famoso modelo de pago por clics.

Para establecer el precio, lo calculamos siguiendo la siguiente formula:

$$CPC = \frac{\text{Coste total}}{\text{Número clics}} = \frac{\text{Ingresos deseados por 100 impresiones}}{CTR \times 100}$$

Así pues, cada 100 visitas se desea recibir un ingreso de 0,50€, y dado que el porcentaje de clics (CTR) se establece en aproximadamente un 1%, quedaría:

$$CPC = \frac{0,50}{0,01 \times 100} = 0,50€$$

Basándonos en que en 2019 había un total de 5.267 establecimientos en Gran Canaria, y que al principio suponemos que solo un 5% estaría interesado en publicitarse en la aplicación web, teniendo en cuenta la estimación de 1.000 usuarios el primer año, recibiríamos una aproximación de 1.315€

2.6.7. CONCLUSIÓN

La aplicación tendrá una estrategia diversificada y flexible en cuanto a las fuentes de ingresos, combinando varias fuentes de monetización. De esta forma, se permite maximizar los ingresos al aprovechar el interés y la participación de los distintos tipos de usuarios.

2.7. ANÁLISIS DE LOS RECURSOS CLAVES

2.7.1. INTRODUCCIÓN

Los recursos claves son los activos y capacidades necesarias para que la aplicación funcione de forma eficiente y tenga éxito, logrando los objetivos propuestos. En los siguientes apartados se expondrán una serie de recursos para permitir su sostenibilidad a largo plazo.

2.7.2. TECNOLOGÍA Y PLATAFORMA DIGITAL

La aplicación necesitará de una plataforma digital sólida, es decir, un backend robusto y un frontend amigable para gestionar toda la interacción con los usuarios. Además, la infraestructura ha de ser escalable para crecer a medida que aumente los usuarios y romerías en la plataforma.

- Desarrollo de *software*: una aplicación web que permita acceder a las funciones principales.
- Base de datos: un sistema de gestión seguro que almacene toda la información de los usuarios, las reservas, las carrozas y las romerías. Es importante mantener la base de datos actualizada con los detalles de las romerías, ya que así se ofrecerá información precisa al cliente.
- Infraestructura en la nube: uso de servidores para asegurar la disponibilidad de la plataforma.

La tecnología es la parte más importante del producto, ya que se trata de una plataforma digital. Su correcta implementación hará que procesos y servicios funcionen adecuadamente y a gran escala a medida que se expanda a otros lugares.

2.7.3. EQUIPO DE DESARROLLO Y SOPORTE TÉCNICO

Para poder construir la aplicación y dar soporte una vez en funcionamiento, se necesita de un desarrollo completo y personal atención al cliente que asegure el buen funcionamiento y una experiencia de usuario satisfactoria.

- Desarrollador de *software*: responsable de la construcción de la plataforma digital y del mantenimiento y actualización constante, dado que se irán implementando distintas funcionalidades a lo largo del tiempo.
- Diseñador *UX/UI*: encargado de crear una interfaz de usuario intuitiva y atractiva que capte la atención del cliente y lo satisfaga.
- Equipo soporte técnico: encargado de atender las dudas y consultas de los usuarios de la aplicación para brindar un soporte rápido y eficiente.

Los distintos roles de trabajo permitirán la optimización continua de la aplicación, asegurando satisfacción y fidelización por parte de los usuarios y cocheros, mejorando también su retención.

2.7.4. INFRAESTRUCTURA DE SEGURIDAD Y PLATAFORMAS DE PAGO

Otro de los puntos más importantes a la hora de realizar la aplicación, es una plataforma de pago segura para gestionar los pagos y suscripciones, siendo esta la propuesta base de la aplicación.

- Integración de pasarelas de pago: se utilizarán plataformas familiares como *stripe* (que ofrece además integración con bizum en España)
- Seguridad y privacidad de los datos: se utilizarán medidas para seguridad y privacidad de los datos de los usuarios como el cifrado de datos, el control de acceso y autenticación o el almacenamiento seguro de contraseñas, cumpliendo así con el Reglamento General de Protección de Datos de Europa (RGPD).

La seguridad y facilidad de uso para los pagos son esenciales para la confianza de los usuarios que utilicen la aplicación, asegurando que las transacciones se efectúen sin problemas. Esto es vital para el éxito financiero de la plataforma.

2.7.5. CONCLUSIÓN

Gracias a los distintos recursos claves analizados, el funcionamiento de la plataforma será sostenible a largo plazo, combinando unos recursos tecnológicos avanzados con un equipo competente que logre el crecimiento estable de la aplicación.

2.8. ANÁLISIS DE LAS ACTIVIDADES CLAVES

2.8.1. DESARROLLO Y MANTENIMIENTO DE LA PLATAFORMA

La actividad principal de la aplicación se basará en la construcción inicial de su infraestructura, así como su mantenimiento y mejora continua a lo largo del tiempo. Se realizarán actualizaciones constantes para mejorar la experiencia de todos los usuarios añadiendo nuevas funcionalidades.

- Desarrollo del *software*: programación y diseño de la aplicación web.
- Mejoras y actualizaciones: incorporación de funcionalidades y corrección de errores.
- Gestión de infraestructura en la nube: administración de servidores para asegurar disponibilidad y seguridad.

Esta actividad es clave para la satisfacción y la retención de los usuarios, ya que habrá que adaptarse a sus necesidades.

2.8.2. GESTIÓN DE SEGURIDAD Y PRIVACIDAD DE DATOS

Establecer medidas y procedimientos para la protección de la información es una de las actividades clave de la plataforma, ya que hay que garantizar la seguridad cumpliendo con las normativas como el RGPD.

- Cifrado de datos: desarrollo de medidas de cifrado para proteger los datos de los usuarios.
- Cumplimiento con normativas: asegurar el cumplimiento del RGPD y otras regulaciones de privacidad.
- Gestión de posibles incidentes de seguridad: establecimiento de protocolos ante la posibilidad alguna brecha de seguridad para actuar de forma rápida.

Los protocolos de seguridad son esenciales para asegurar la integridad de la aplicación y generar confianza en los usuarios.

2.8.3. RELACIONES CON COCHEROS Y RESPONSABLES DE ROMERÍAS

Para que la aplicación funcione, una de las actividades claves consistirá en atraer y mantener tanto romerías como carrozas a través de las relaciones con los responsables. Así, se asegura una oferta variada y de calidad.

- Captación de cocheros: establecer estrategias para atraer a los cocheros locales y que ofrezcan sus carrozas en la aplicación.
- Gestión de relaciones: comunicación constante, soporte y asistencia y resolución de inquietudes para mejorar su experiencia en la aplicación.

La variedad y disponibilidad de carrozas y romerías es el pilar de la aplicación, por lo que se debe asegurar una oferta sólida y atractiva a través de las relaciones con los responsables.

2.8.4. MARKETING Y PROMOCIÓN

Para dar a conocer la aplicación entre potenciales clientes, tanto usuarios como cocheros, es esencial el marketing de esta para llegar a más personas.

- Campañas en redes sociales: publicidad en plataformas como *Facebook*, *Instagram*, *Tiktok* o *Google Ads* facilitará el alcance a los usuarios interesados en romerías.
- Colaboraciones con entidades turísticas: la colaboración con oficinas de turismo facilitará la promoción a los turistas interesados en experimentar las fiestas locales.
- Cuentas de *marketing*: cuentas en redes sociales para realizar publicaciones con novedades o emails de *marketing* con promociones exclusiva, fomentará el uso continuo de la aplicación.

Una buena estrategia de *marketing* es clave para la captación de nuevos usuarios, haciendo llegar a un público mayor la aplicación.

2.8.5. GESTIÓN DE SOPORTE Y ATENCIÓN AL CLIENTE.

Para que la experiencia de los clientes sea satisfactoria, hay que brindarles asistencia rápida y eficaz ante cualquier problema o consulta que pudiese surgir.

- Soporte en tiempo real: respuesta rápida a preguntas y problemas mediante correo electrónico o teléfono.
- Gestión de comentarios y *feedback*: analizar la retroalimentación de los clientes para realizar mejoras en la plataforma.
- Resolución de incidencias: solución de problemas dentro de la aplicación que pudiesen experimentar los clientes.

Un buen servicio de atención al cliente asegurará la satisfacción y la permanencia dentro de la aplicación.

2.8.6. CONCLUSIÓN

Las actividades claves son esenciales para el éxito y la sostenibilidad de la plataforma de romerías, manteniendo la aplicación operativa y asegurando una buena experiencia al usuario.

2.9. ANÁLISIS DE LOS ALIADOS

2.9.1. COCHEROS Y ORGANIZADORES DE ROMERÍAS

Además de tener el rol de cliente, los cocheros y organizadores son aliados esenciales para que funcione la aplicación, ya que su colaboración permite ofrecer una variedad de romerías y carrozas al resto de usuarios.

Su rol principal dentro de la aplicación consiste en ofrecer sus propias carrozas a los usuarios a través de la plataforma, permitiendo así disfrutar de las fiestas como un residente más. Además, los organizadores de romerías aportarán la información actualizada en cada fiesta, asegurando que los datos sean precisos para que un número mayor de personas acudan al evento.

En cuanto a los beneficios de esta alianza, se centran en asegurar una oferta de carrozas y una grata experiencia para los usuarios, facilitando la expansión de la aplicación gracias a su variedad.

A cambio de su colaboración, existirá un plan básico gratuito para cocheros a la hora de ofertar las carrozas dentro de la aplicación.

2.9.2. OFICINAS DE TURISMO Y ENTIDADES LOCALES

Para captar el público turístico, una alianza de gran ayuda será con las oficinas de turismo y las entidades locales, pues facilitarán la promoción y se proporcionará un respaldo institucional.

El rol principal de estos aliados consistirá en promover el uso de la aplicación entre turistas y locales que tengan interés en las fiestas locales de la isla.

En cuanto a los beneficios de esta alianza, se dan al obtener una mayor visibilidad entre los turistas y habitantes locales, generando una base de usuarios más amplia que atraerá a su vez a más cocheros que ofertarán sus carrozas dentro de la aplicación.

A cambio de esta colaboración, se realizarán acuerdos de patrocinio y publicidad.

2.9.3. EMPRESAS LOCALES

Las empresas locales como son tiendas de souvenirs, hoteles o restaurantes, serán aliados clave para la monetización de la plataforma, ya que a través de publicidad se podrán obtener beneficios económicos.

El rol principal de estos aliados consistirá en pagar por publicidad dentro de la aplicación, destacando sus servicios entre los usuarios. Además, pueden existir colaboraciones con promociones específicas y descuentos para los usuarios que utilicen la aplicación.

En cuanto a los beneficios de esta alianza, se dan al aumentar los ingresos de la plataforma para así mejorar la sostenibilidad financiera. También se da al ofrecer promociones y descuentos a los usuarios, ya que así enriquecen su experiencia dentro de la plataforma de romerías.

Esta colaboración existirá a través de contratos y patrocinios específicos en los que se establecerán los términos, precios y visibilidad en la aplicación.

2.9.4. CONCLUSIÓN

En conjunto, la variedad de aliados permitirá que la aplicación maximice su público, su rentabilidad, y sea una aplicación frecuente para los residentes que asisten a romerías, además de para los turistas que acuden en vacaciones.

2.10. ANÁLISIS DE LA ESTRUCTURA DE COSTES

2.10.1. COSTES DE DESARROLLO DE LA APLICACIÓN

El desarrollo inicial del proyecto será propio, por tanto, se omiten los costes de contratación de otros desarrolladores externos, dejando únicamente como costes iniciales los gastos de las herramientas y recursos básicos para su implementación.

En cuanto a los gastos para el producto mínimo viable, se encontrarían unos 10€ para la compra del dominio, consultado en la página de *Arsys*. En cuanto a las herramientas de desarrollo, dado que la Universidad de Las Palmas de Gran Canaria otorga licencias, no es necesario ningún gasto inicial en esta parte.

2.10.2. INFRAESTRUCTURA EN LA NUBE

Existen múltiples proveedores en la nube como *AWS*, *Google Cloud*, *Microsoft Azure* o *Firebase*, que ofrecen pagar sus servicios únicamente por el uso. El almacenamiento en la nube es esencial para garantizar la correcta disponibilidad de la aplicación a los usuarios.

Entre todas las opciones, la más económica y sencilla para un proyecto de romerías en una primera fase es *Firebase*, dado que ofrece varias ventajas para una aplicación de tamaño medio, como autenticación integrada con *Google*, notificaciones push, y *Google Analytics* para realizar análisis de los usuarios. El coste estimado para el primer año es de 0€, puesto que ofrece un plan gratuito. A medida que la aplicación creciese, existirían unos costes, aunque seguirán manteniéndose bajos.

2.10.3. SOPORTE AL CLIENTE

El soporte al cliente es vital para que en la primera fase con el MVP se gestionen los problemas iniciales y se gestionen las distintas consultas.

Se estima que el gasto será de 1,500€/brutos al mes para una única persona que trabaje en la atención al cliente. Según páginas como Glassdoor, el salario medio en este sector de trabajo ronda entre los 1.000€ y 1.500€ mensuales dentro de Europa en una jornada laboral estándar. No obstante, en la fase inicial no se requerirá de este coste, dado que, al tener un volumen más bajo en el uso de la aplicación, la gestión y el soporte técnico es asumible internamente.

2.10.4. MARKETING Y PROMOCIÓN

Para captar todo tipo de usuarios dentro de la aplicación (organizadores, cocheros y residentes), se debe realizar una buena campaña de marketing que haga llegar la plataforma a más personas.

Se destinarán unos 300€ al mes en publicidad en redes sociales como *Facebook* e *Instagram* para promocionar la aplicación. Sin embargo, el primer año este coste es opcional, dado que actualmente con el auge de las redes sociales, se puede realizar una buena campaña publicitaria que llegue a la viralización a través de una cuenta gratuita.

2.10.5. TRANSACCIONES BANCARIAS

Actualmente, cuando se utiliza una pasarela de pago, existen unos costes que dependen de la cantidad de transacciones.

En la página web de *Stripe*, podemos comprobar que se aplica una comisión del 1,5% + 0,25€ fijos, por lo que suponiendo que la media del valor de las entradas esté en 20€, se aplicará una comisión de 0,55€/transacción. Suponiendo un volumen de 1.500 compras el primer año, este valor ascenderá a 825€ el primer año.

2.10.6. COSTES LEGALES

Dado que en un principio no es necesario crear una empresa, se opta por darse de alta como autónomo para la salida a producción del proyecto. En España, existe una ayuda

estatal que permite pagar una cuota reducida de 80€/mes durante el primer año siempre y cuando no se haya estado dado de alta previamente, por lo que se cumple con este requisito. Además, se requerirán servicios básicos de contabilidad y presentación de impuestos, lo cual puede costará 70€/mes con la empresa ‘Declarando’, según lo establecido en sus planes de precio en su página web.

Con la suma de ambas cuotas, el coste se eleva a 1.800€ el primer año.

2.10.7. CONCLUSIÓN

A pesar de tener la posibilidad de lidiar con una gran variedad de costes, se han buscado los proveedores y las formas de reducirlos notablemente, ascendiendo a un total de tan sólo 2.635€. No obstante, a medida que la aplicación crezca, los costes también crecerán, ya que se necesitará más ayuda de otro personal, y la infraestructura en la nube podría llegar a sus límites.

2.11. LIENZO DE MODELO DE NEGOCIO

En este apartado, se muestra de forma resumida el modelo de negocio explicado anteriormente, con una representación gráfica en la figura 2.

Modelo Canvas



Figura 2. Modelo Canvas.

Fuente: elaboración propia.

2.12. ANÁLISIS DE LA VIABILIDAD ECONÓMICA Y FINANCIERA

2.12.1. PÚBLICO OBJETIVO

Actualmente, en fecha de junio de 2024, Canarias tiene una población de 2.246.132 personas. Dado que no existe ningún estudio que mencione el porcentaje de asistentes a romerías, se calcula a continuación de forma aproximada:

Tras consultar en múltiples fuentes de noticias como ‘El periódico’ o ‘El Diario’ distintas romerías, de media se podría estimar que en Gran Canaria acuden unas 20.000 personas a cada una, teniendo en total la isla 869.000 habitantes. Por tanto,

$$\text{Habitantes que asisten a romerías} = \frac{20.000}{869.000} \times 100\% = 2,3\%$$

Consecuentemente, en su totalidad asisten 51.661 personas residentes dentro de las Islas Canarias. Además, en Canarias en 2023 hubo 16.706.543 turistas, por lo que

suponiendo que un 0,1% estuviese interesando en las fiestas locales gracias a la información facilitada por la aplicación, esto se traduciría en 16.707 turistas, dejando como público objetivo a un total de 68.368 personas como mínimo. Dadas las ventajas y la facilidad de uso de la herramienta, se estima que en 5 años como máximo sea la herramienta oficial de gestión de romerías dentro de las Islas Canarias, comenzando su implementación en Gran Canaria para meses después expandirse a la totalidad de las islas.



Figura 3. Crecimiento de usuarios.

Fuente: elaboración propia.

Mientras que en 2025-2026 el crecimiento será pequeño ya que los usuarios comenzarán a probar la aplicación, esto se verá acelerado para 2027 mediante la popularidad y visibilidad que vaya cogiendo. Así, para 2028-2029 el crecimiento será exponencial ya que la plataforma será la principal fuente de información de romerías en la región.

2.12.2. ESTIMACIÓN DE INGRESOS

En las Islas Canarias hay como mínimo, 68 romerías en Tenerife y 34 romerías en Gran Canaria, sin contar las islas menores, que serán algunas más, dando con unas 102 romerías anuales. Contando que en cada romería hay de media 20.000 asistentes, de los cuales suponemos que mínimo un 10% va en carroza, nos da un movimiento de 2.000 entradas por romería. Por lo que, asumiendo que en 2029 todo el proceso de compra de entradas se realice por la aplicación, se alcanzaría hasta unos 612.000€ únicamente en comisiones. Así, los ingresos que se estiman, mostrados en la tabla 2, serían:

- $Comisiones = \frac{Usuarios\ actuales}{Usuarios\ totales} * máximo\ comisiones\ €$
- $Suscripciones\ Usuarios = 10\% * 50\% \text{ usuarios actuales} * (70\% * 5€ * 12 \text{ meses} + 30\% * 30€)$
- $Suscripciones\ cocheros = Número\ cocheros * 30\% * 100€$
- $Suscripciones\ organizadores = Número\ romerías * 20€$
- $Publicidad = Usuarios\ actuales * CTR\ del\ 1\% * 0,50€ * Número\ establecimientos\ interesados\ (5\% - 7\% - 10\% - 15\% - 20\%)$

INGRESOS ANUALES

PLANIFICACIÓN EMPRESARIAL

	2025	2026	2027	2028	2029
Comisiones	4.500€	36.000	108.000	270.000€	612.000€
Suscripciones usuarios	2.550€	10.200€	30.600€	76.500€	173.400€
Suscripciones cocheros	1.500€	3.000€	4.500€	6.000€	9.000€
Suscripciones organizadores	0€	200€	500€	1.020€	2.040€
Publicidad	1.315€	7.364€	22.158€	59.175€	178.920€
TOTAL	9.865€	56.764€	165.758€	412.695€	975.360€

Tabla 2. Ingresos anuales.

Fuente: elaboración propia.

2.12.3. ESTIMACIÓN DE GASTOS

Los gastos irán incrementando anualmente en base al crecimiento de la aplicación y de los ingresos percibidos. Para realizar los cálculos, mostrados en la tabla 3, se ha tenido en cuenta distintas fuentes de información.

- Para los cálculos de infraestructura en la nube, se ha utilizado la página web de firebase, en la que se ofrece una calculadora para estimar costes en base a número de usuarios y cantidad de información manejada.
- Para los cálculos de soporte técnico, se ha estimado la contratación de una persona cuando el número de usuarios ha ascendido considerablemente.
- En cuanto al marketing, se ha estimado publicidad mensual en redes sociales pasado un año, para tratar de llegar a más público.

- En cuanto a las transacciones bancarias, aumentan según el crecimiento de usuarios en la aplicación y sus compras dentro de esto.
- En cuanto a los costes legales, se basan en las cuotas de autónomo correspondientes según los ingresos percibidos.



GASTOS ANUALES

PLANIFICACIÓN EMPRESARIAL

	2025	2026	2027	2028	2029
Desarrollo	10€	0€	0€	0€	0€
Infraestructura en la nube	0€	0€	1.000€	3.000€	6.000€
Soporte Técnico	0€	0€	0€	18.000€	18.000€
Marketing	0€	3.600€	3.600€	3.600€	3.600€
Transacciones bancarias	825€	6.600€	19.800€	49.500€	112.200€
Costes legales	1.800€	7.200€	7.920€	7.920€	7.920€
TOTAL	2.635€	17.400€	32.320€	82.020€	147.720€

Tabla 3. Gastos anuales.

Fuente: elaboración propia.

2.12.4. ANÁLISIS ESCENARIOS

Tras todo lo analizado, nos encontramos ante un escenario optimista, en el que se consigue abarcar prácticamente todo el mercado en sólo 5 años, con un crecimiento progresivo de usuarios que se adaptan fácilmente a planes de suscripción, además de unos establecimientos que confían en la plataforma para publicitarse a través de ella, aumentando el número de locales que participan en este espacio publicitario.

Así, a continuación, se muestra un gráfico con los beneficios anuales, que demuestran que nos encontramos ante una idea claramente viable tanto económica como

financieramente, puesto que los ingresos superan los gastos en todo momento, ya que estos son ínfimos debido a las ventajas existentes en su desarrollo.

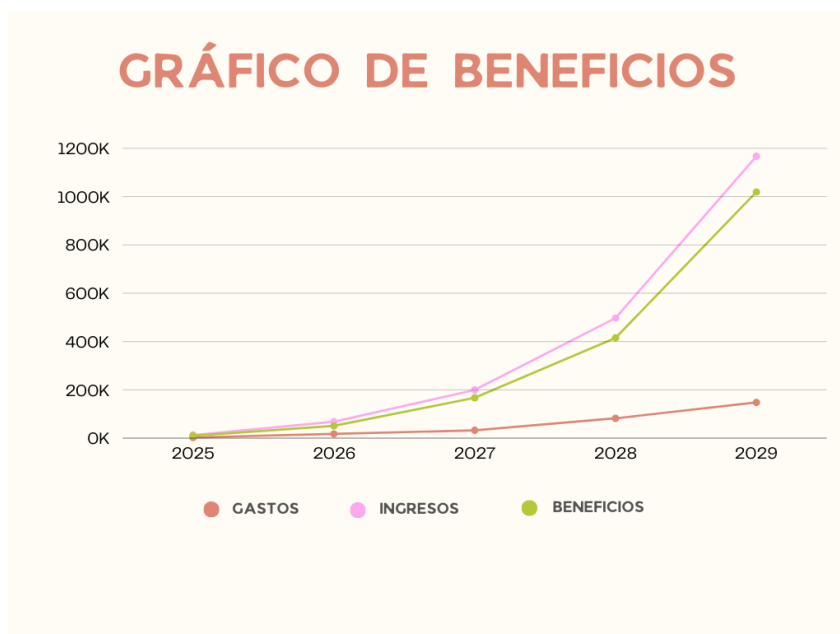


Figura 4. Gráfico de beneficios.

Fuente: elaboración propia.

Sin embargo, no podemos obviar que existe la posibilidad de que se de un escenario más conservador, en cuyo caso seguirían existiendo ingresos, pero en un ritmo más lento, tardando más en llegar al objetivo de usuarios totales. En este caso, mantendremos siempre los ingresos por encima de los gastos, debido a los bajos costes, pero los beneficios serían menores.

3. HERRAMIENTAS Y TECNOLOGÍAS DE DESARROLLO

Las herramientas y lenguajes utilizados para el desarrollo de la aplicación, se listan a continuación, junto con sus beneficios y razones de su elección:

- Editor de código para el *frontend*: *Visual Studio Code*

Es un editor gratuito y de código abierto. Tiene una interfaz sencilla y es fácil de usar. Se integra fácilmente con *Git* y tiene múltiples extensiones para facilitar el desarrollo en el *frontend*, por ello, es de los editores más utilizados por la comunidad *frontend*.

- Editor de código para el *backend*: *IntelliJ*

IntelliJ está diseñado específicamente para *Java*, por tanto, resultó ser el editor ideal para el desarrollo en el *backend*. A pesar de no ser gratuito, la ULPGC ofrece una licencia, por lo que se ha podido utilizar su versión completa. *IntelliJ* ofrece autocompletado inteligente, refactorización, integración con *Maven* e integración con *Git*, lo cual ayuda a agilizar el desarrollo.

- Gestor de proyectos en el *backend*: *Maven*

Maven se utiliza para gestionar proyectos y sus dependencias. A través del archivo *pom.xml* se define la estructura, dependencias y cualquier otra configuración del ciclo de vida del sistema. *Maven* es útil dado que facilita la gestión de dependencias, descargando y actualizando bibliotecas de forma automática.

- Sistema de control de versiones: *Git* y *GitHub*

Git se utiliza de forma local y permite hacer un seguimiento de las versiones del proyecto, revirtiendo cambios en caso de que fuese necesario sin perder todo el trabajo. Además, facilita la resolución de errores. Así, la colaboración con *GitHub* permite mantener el repositorio en la nube, evitando que se pierda parte del proyecto en caso de cambiar a las distintas versiones.

- Base de datos: *MySQL*

MySQL es uno de los *softwares* de gestión de bases de datos más utilizados a nivel mundial, propiedad de la empresa *Oracle*. Resulta fácil de usar y es compatible con múltiples sistemas operativos, para el proyecto en cuestión era la opción más adecuada. Es un sistema *open source*, y ofrece una seguridad robusta que avalan sus varios años de experiencia en el sector.

- Pruebas de *API*: *Postman*

Postman se utiliza para probar el funcionamiento correcto de la *API*. Tiene una interfaz intuitiva y fácil de usar, por lo que el envío y la respuesta de las solicitudes se interpretan fácilmente. Permite realizar todo tipo de solicitudes *REST* y es compatible con los distintos protocolos web.

- Lenguaje de *backend*: *Java*

Java es un lenguaje de programación altamente usado. Se ha escogido principalmente por su uso continuo durante el grado, ya que ha sido uno de los lenguajes más trabajados. Además, ofrece ventajas como su robustez, estabilidad, seguridad las cualidades de un lenguaje orientado a objetos, cruciales para el buen desempeño a la hora de diseñar el *backend*.

- *Framework* de *backend*: *Spring*

El *framework* de *Spring* facilita el desarrollo en aplicaciones *JAVA*. Fue escogido debido a su alta demanda en el mundo laboral, además de sus múltiples ventajas para hacer una aplicación escalable. Entre sus principales características, encontramos la inyección de dependencias, delegando esta responsabilidad a un contenedor que proporciona *Spring*. También es importante destacar su seguridad gracias a *Spring Security*, el cual proporciona autenticación y autorización en las aplicaciones de *Java*. Aun así, su configuración puede ser un tanto tediosa, para ello se ha optado por utilizar la versión de *Spring Boot*, que encapsula toda la configuración de *Spring* para que sea transparente de cara al desarrollador. Por otro lado, se ha usado una herramienta llamada *Spring Initializr* para poner en marcha toda la estructura inicial del proyecto, sin tener mayores preocupaciones sobre esto.

- Lenguajes de *frontend*: *Javascript*, *HTML* y *CSS*

Dentro del desarrollo de *frontend*, se utilizan mayormente estos tres lenguajes, los cuales suelen utilizarse como un “*pack*”. *HTML* existe desde los comienzos del mundo web, mientras que *CSS* surge años más tarde para darle un mejor estilo a las páginas. Por último, *Javascript* se utiliza para las interacciones, animaciones y todo lo relacionado con la lógica de un *frontend*.

- Librería en *frontend*: *React JS*, *Material UI*

React JS es de las librerías más usadas en el *frontend*, con una alta demanda en el mundo laboral, lo cual fortaleció la elección a la hora de usarla para el desarrollo. *React* ofrece múltiples ventajas, entre ellas destacan el uso de componentes, los cuales son reutilizables y ello mejora y agiliza la organización del código, y el uso de un *DOM* virtual, mejorando los recursos ya que sólo se actualizan los necesarios. Dentro de *React JS*, existen múltiples librerías con componentes ya creados. Una de las más populares es *Material UI*. Gracias a *Material UI*, los componentes ya tienen un estilo previo coherente y atractivo, por lo que mejora la experiencia de usuario. Además, también permite modificar sus propios componentes para adaptarlos a necesidades específicas.

4. ANÁLISIS, DISEÑO Y DESARROLLO TECNOLÓGICO

4.1. ANÁLISIS Y DISEÑO

4.1.1. CASOS DE USO

4.1.1.1. ACTORES

Usuario común

Este usuario será aquel que utilice la aplicación web, es decir, cualquiera que tenga un dispositivo tecnológico y desee interaccionar con las funcionalidades básicas de la página.

Usuario cochero

Este usuario será aquel que disponga de una carroza y utilice la aplicación web. Se encargará de ofertar su carroza dentro de una romería. Para poder administrar y gestionar su carroza, debe estar registrado.

Usuario administrador

Este usuario será todo aquel que organice una romería y utilice la aplicación web. Se encargará de administrar las distintas romerías dentro de la página web. Para poder administrar y gestionar su romería, debe estar registrado.

4.1.1.2. CASOS DE USO: USUARIO COMÚN

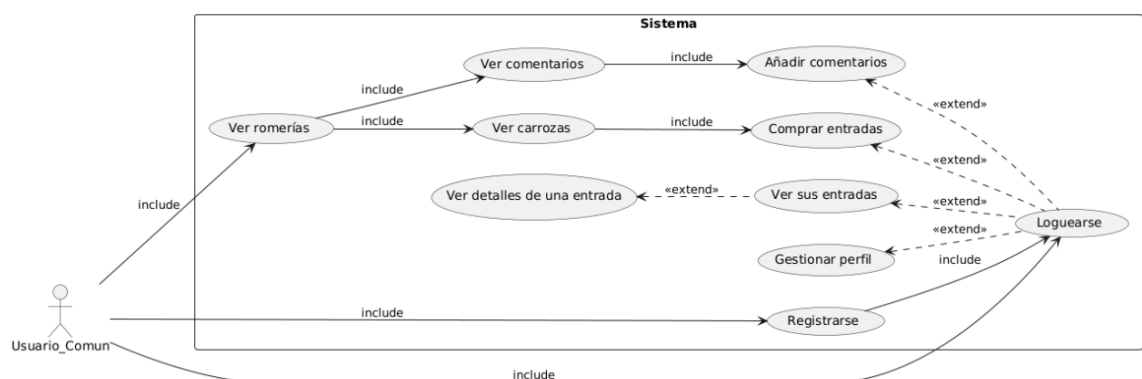


Figura 5. Casos de uso de usuario común.

Fuente: elaboración propia.

A continuación, en la figura 8, se muestra el diagrama entidad-relación.

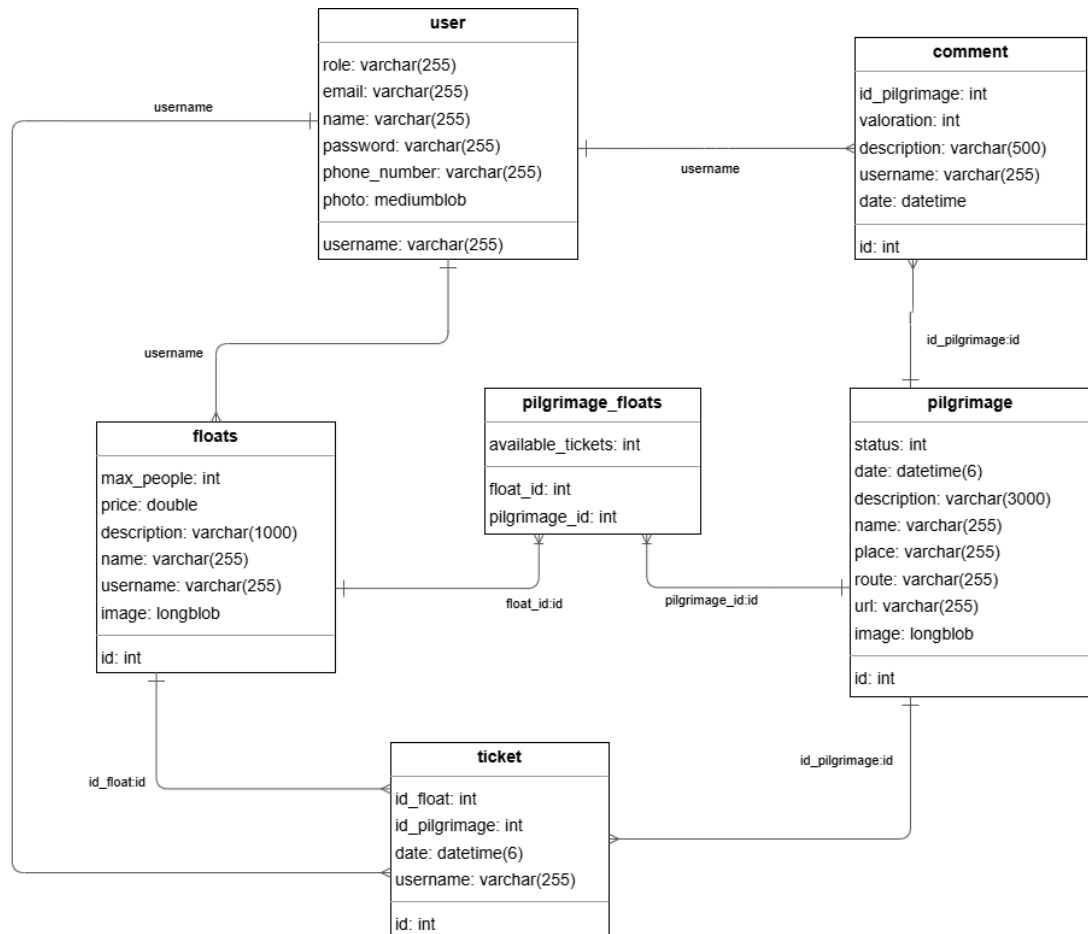


Figura 8. Diagrama entidad-relación.

Fuente: elaboración propia.

Entidades principales:

- **Usuario**: Se almacena la información básica de un usuario registrado, como sería las credenciales correctamente encriptadas, los datos básicos, como correo, nombre y teléfono, y una foto de perfil.
- **Comentario**: Se guardan los comentarios de los distintos usuarios sobre las distintas romerías.
- **Carroza**: Representa la información sobre cada carroza, guardando datos como el propietario, el precio, la capacidad, una foto de la carroza y una breve descripción de su contenido.

- Carroza_Romería: Esta tabla intermedia ayuda a relacionar las carrozas con las romerías, dado que una romería tendrá distintas carrozas para su celebración. Además, almacena las entradas disponibles en cada relación.
- Romería: Contiene los datos principales de la romería, como el lugar y la hora, una descripción, la ruta o una foto de la romería de años anteriores.
- Entrada: Representarían las reservas de una carroza en una romería específica hechas por los usuarios.

Relaciones clave:

- Usuario-Comentario-Romería: Un usuario puede realizar múltiples comentarios en distintas romerías.
- Usuario-Entrada-Carroza-Romería: Un usuario puede comprar múltiples entradas en distintas carrozas en distintas romerías.
- Carroza-Romería: Una carroza puede estar en múltiples romerías, así como una romería puede tener múltiples carrozas.

4.1.3. ARQUITECTURA

A la hora de desarrollar el proyecto, en el *backend* se ha trabajado una arquitectura hexagonal.

La arquitectura hexagonal busca separar la lógica de negocio de las tecnologías externas, permitiendo una mayor flexibilidad, testeo, y mantenimiento del proyecto a lo largo de su vida útil.

En la figura 9 se puede observar cómo la lógica de negocio está separada de la aplicación, mostrando una forma hexagonal en su representación ya que tiene una comunicación bilateral entre las distintas capas.

La arquitectura hexagonal consta del núcleo (o dominio) que es donde se encontraría todo lo que define el comportamiento de la aplicación, centrándose en resolver los problemas que se quieren abordar. También tenemos los puertos que es la interfaz que determina la comunicación con el exterior (ya sea a una base de datos o una *API web*), definiendo los métodos necesarios para interactuar. Además, también tenemos los

adaptadores, que se centran en transformar los datos para que puedan ser usados por el núcleo. Por último, tenemos la infraestructura que contiene los sistemas de almacenamiento, los sistemas de autenticación, etc.

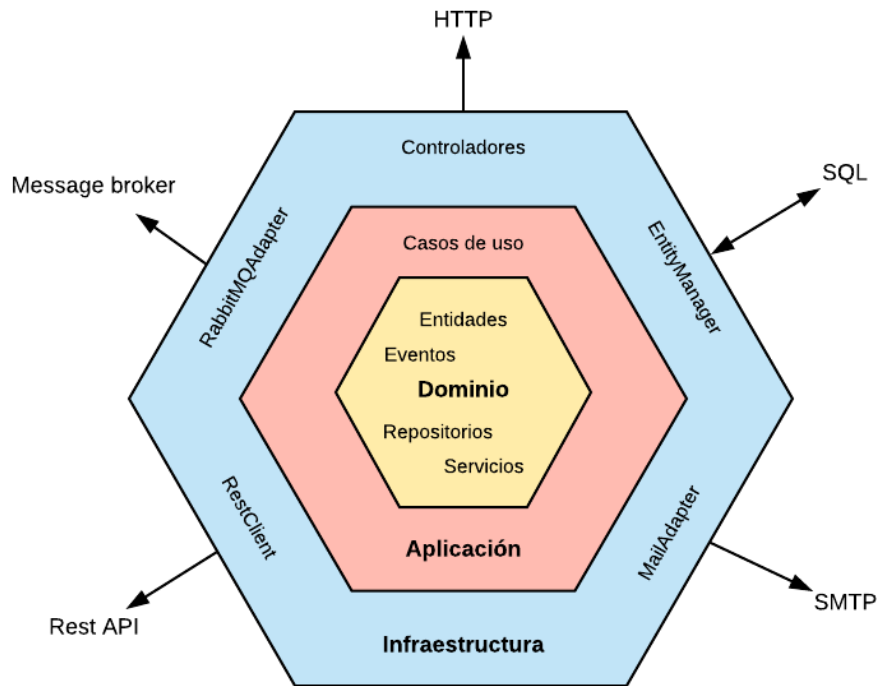


Figura 9. Arquitectura hexagonal.

Fuente: Imágenes de Google.

En la aplicación de romerías, se ha implementado esta arquitectura de forma adaptada, siguiendo algunos de sus principios.

Principalmente, como se ve en la figura 10, existe una separación de responsabilidades. El dominio encapsula las entidades con la lógica de negocio principal, interactuando con el servicio que contiene la lógica del negocio más compleja. El repositorio se encarga de implementar las interacciones con la base de datos a través de *JPA*, y el controlador se centra en manejar las solicitudes *HTTP* y devolver las respuestas al cliente.

En cuanto a la seguridad de la aplicación, se basa en el uso de *tokens JWT*. Estos *tokens* contienen información relevante sobre el usuario, como sus permisos. El cliente se

auténtica, y tras el éxito de esta, se generará un *token* que se enviará de vuelta al cliente. En cada solicitud que el cliente haga a la infraestructura, se deberá enviar el *token* para validarlo a través de *Spring Security* y permitir el acceso a las diferentes funcionalidades. Este *token* tiene una duración máxima de 3.600.000 milisegundos (1 hora), después de lo cual no permitirá realizar más peticiones con este *token*, obligando al usuario a volver a autenticarse para obtener un nuevo *token*, esto se ha hecho así para preservar la seguridad de los usuarios en el entorno de la aplicación.

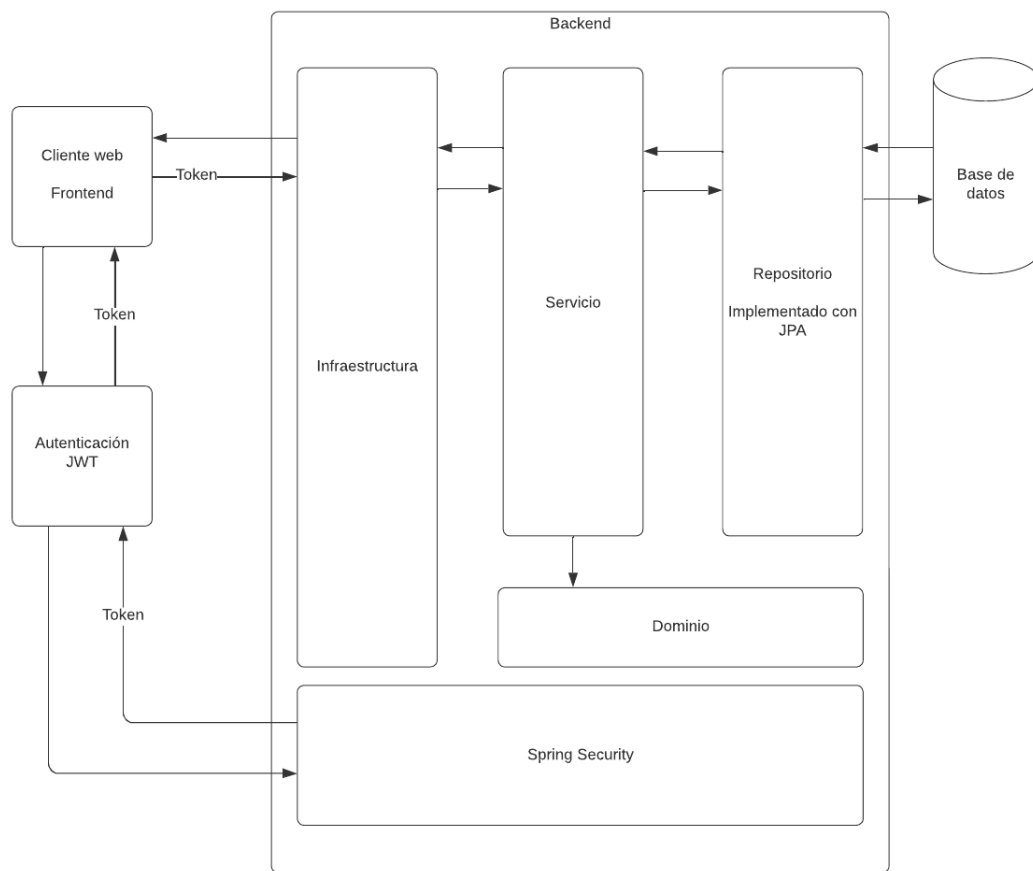


Figura 10. Arquitectura backend.

Fuente: elaboración propia.

Toda la lógica que pertenece al *backend*, debido a su tamaño, es un monolito, ya que está diseñado basando su estructura completa en un único módulo. Sin embargo, debido a su estructura de directorios basados en dominio, mostrados en la figura 11, es fácilmente escalable, pudiendo convertirse en microservicios en un futuro si fuese necesario.

Cada dominio tiene su propio conjunto de directorios que separan las distintas funcionalidades:

- *Controller*: actúa como intermediario entre la lógica de negocio y el *frontend*, manejando las solicitudes y respondiendo a estas.
- *Converter*: se encarga de convertir los modelos a peticiones o respuestas, adaptando su contenido para interactuar con el *frontend*.
- *Model*: en esta carpeta se definen los modelos (entidades) principales, así como las clases de respuesta y petición que representan los datos enviados y recibidos a través de la *API*.
- *Repository*: incluye una interfaz que implementa *JPA* para interactuar con la base de datos exterior.
- *Service*: se incluye una interfaz y su implementación, dónde se resuelve toda la lógica de negocio relacionada con el dominio.

Adicionalmente, existen varias carpetas comunes a todas las entidades:

- *Common*: almacena una validación común a todos los dominios que se utiliza en diferentes ocasiones.
- *Exceptions*: almacena una clase para excepciones comunes a todos los dominios.
- *Security*: almacena toda la seguridad de la aplicación, configurando la autenticación y la autorización del sistema.

Gracias a esta estructura de directorios, el código permanece ordenado, facilitando la implementación de nuevas funcionalidades y asegurando que sea escalable en un futuro, dado que para convertirse en microservicios tan sólo se tendría que extraer una de las carpetas de dominio, y copiar la carpeta commons en el microservicio que se desee crear.

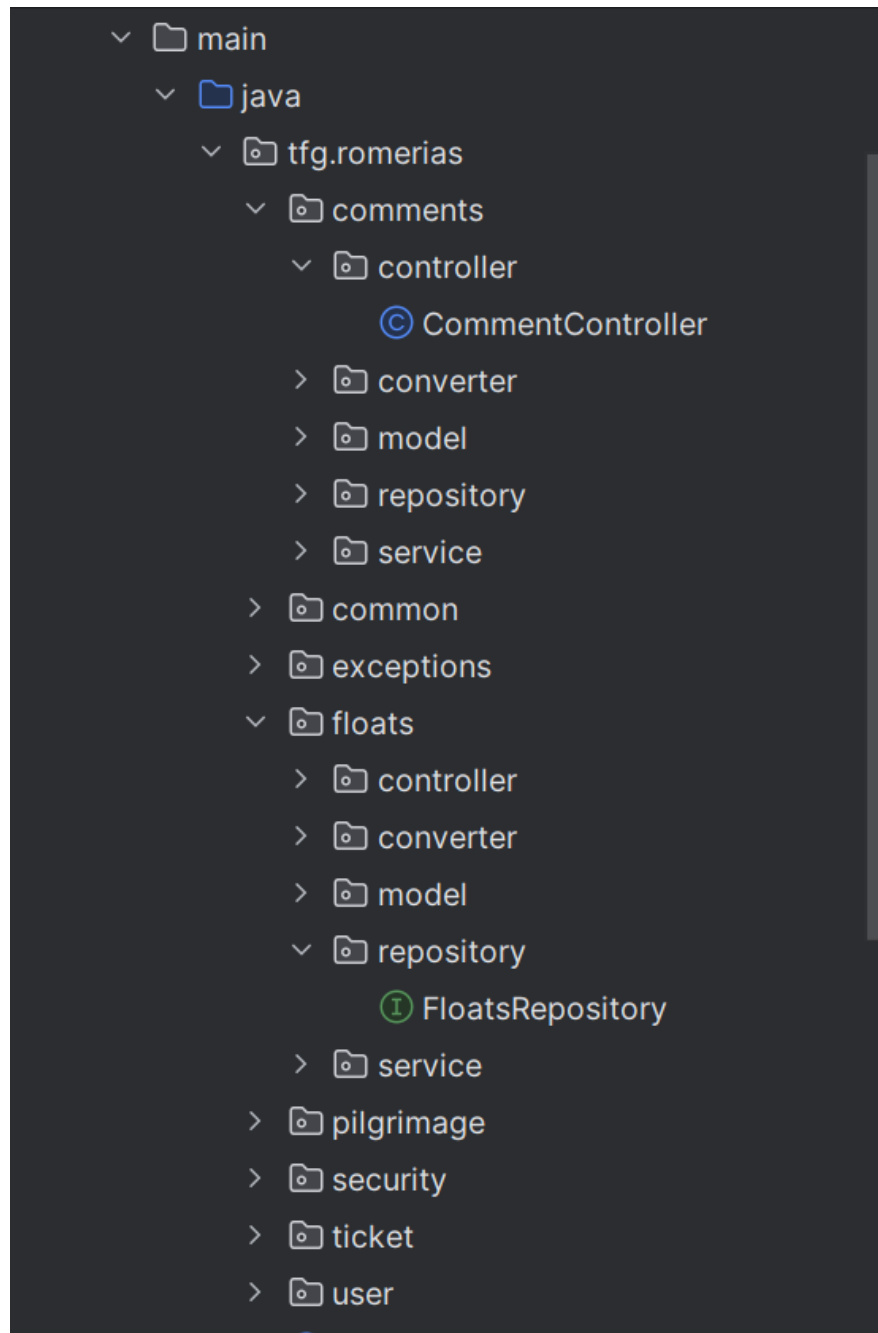


Figura 11. Estructura de directorios.

Fuente: elaboración propia.

Por la parte del *frontend*, se ha elaborado una arquitectura básica con *React*, en la que la aplicación incluye distintos módulos de componentes, que utilizan la librería de *material UI*, *router* para interactuar con las distintas *URLs*, y funciones con la lógica de negocio que simulan el comportamiento de la capa de servicios (la cual no se ha implementado por el tamaño de la aplicación) para que mediante *axios* comuniquen

las peticiones al *backend*, como se muestra en la figura 12. Además, toda la aplicación está contenida dentro de *AuthContext*, lo cual nos ayuda a manejar la autenticación del usuario de manera global a lo largo de toda la aplicación permitiendo una gestión centralizada y no a nivel de componente.

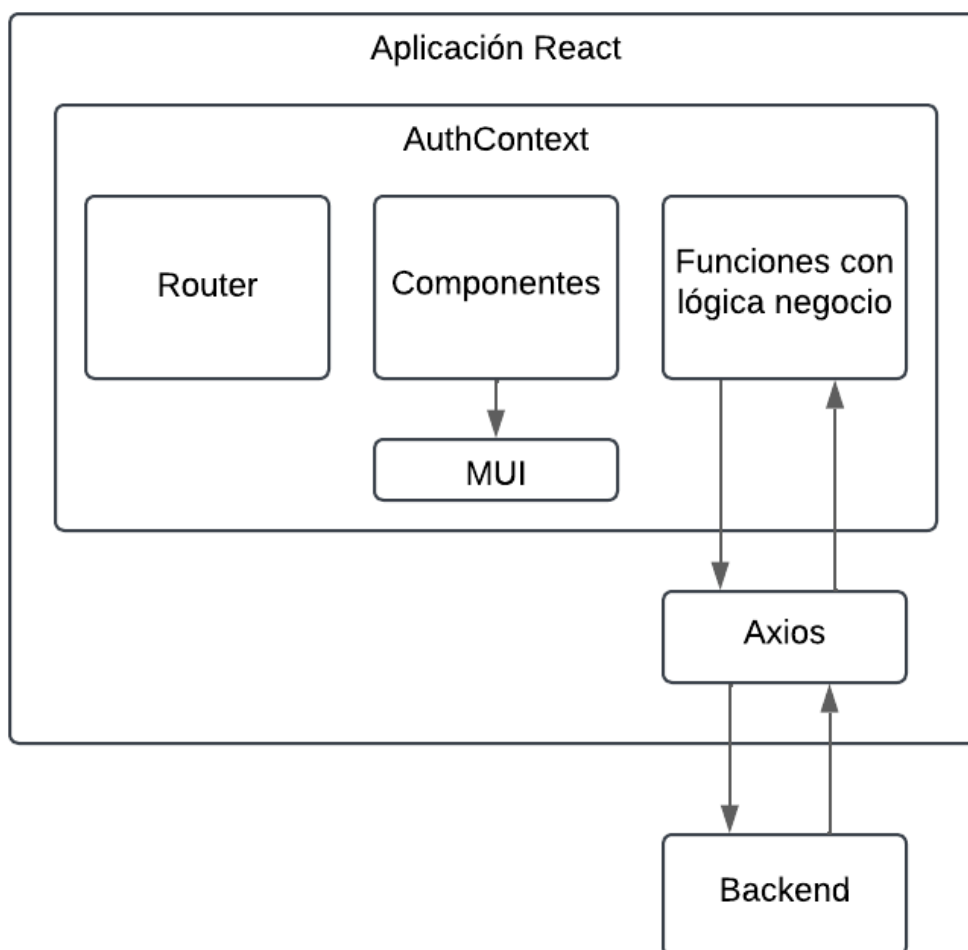


Figura 12. Arquitectura del frontend.

Fuente: elaboración propia.

Para comprender en profundidad la estructura del *frontend* anteriormente presentado, es recomendable conocer las distintas funciones de las distintas capas que componen la aplicación.

En primer lugar, el enrutador, importado desde *react-router-dom*, permite navegar en distintas *urls* sin tener que recargar la página web. Dinámicamente, se gestiona la

navegación, cargando únicamente los componentes específicos asociados a cada *url*. Así, se puede navegar entre distintas vistas, sin tener que recargar elementos comunes, como por ejemplo sería el menú de la aplicación. En la figura 13, se muestra la forma en la que se ha desarrollado en Romería Viva.

```
<Router>
  <AuthProvider>
    <Menu filter={setSearch}/>
    <Container sx={{ mt:"var(--marginTop)", mb:"2rem" }}>
      <Routes>

        <Route path="/" element={<Home filter={search} />}></Route>
        <Route path="/manage-floats" element={<ManageFloats/>}></Route>
        <Route path="/:id" element={<PilgrimagePage/>}></Route>
        <Route path="/buyTicket/:idPilgrimage/:idFloat" element={<BuyTicketPage/>}></Route>
        <Route path="/floats/edit/:id" element={<CreateFloats/>}></Route>
        <Route path="/floats/edit" element={<CreateFloats/>}></Route>
        <Route path="/login" element={<LoginPage/>}></Route>
        <Route path="/myTickets" element={<MyTicketsPage/>}></Route>
        <Route path="/profile" element={<Profile/>}></Route>
      </Routes>

    </Container></AuthProvider>

  </Router>
```

Figura 13. Desarrollo react-router-dom.

Fuente: elaboración propia.

Así, también es necesario hablar del concepto de *Single Page Application* utilizado en la aplicación, que se da cuando una página web utiliza un solo archivo *HTML*, y con el uso de *Javascript* actualiza dinámicamente el contenido de la página. Tal y como se muestra en la figura 14, el archivo *HTML* tiene un elemento `<div>` principal donde se inyecta dinámicamente todo el contenido de la aplicación, lo cual se muestra en la figura 15.

```

<!doctype html>
<html Lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" type="image/svg+xml" href="/vite.svg" />
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Anton&family=Finger+Paint&display=swap" rel="stylesheet">
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>Vite + React</title>
  </head>
  <body>
    <div id="root"></div>
    <script type="module" src="/src/main.jsx"></script>
  </body>
</html>

```

Figura 14. Archivo index.html.

Fuente: elaboración propia.

```

ReactDOM.createRoot(document.getElementById('root')).render(
  <React.StrictMode>
    <ThemeProvider theme={theme}>
      <CssBaseline />
      <App />
    </ThemeProvider>
  </React.StrictMode>
)

```

Figura 15. Archivo main.jsx.

Fuente: elaboración propia.

Además, enlazado con lo anterior, aparece el *Virtual DOM*, un concepto clave en *React*, el cual optimiza el proceso de renderizado. El *Virtual Dom* funciona de forma que, al actualizar un componente, *React* compara el *Virtual Dom* con una copia anterior, aplicando los cambios que fuesen necesarios al *DOM* real, mejorando el rendimiento de la página web.

Otro de los módulos de la aplicación son los componentes, que en este caso podría ir enlazado con los servicios, dado que no se ha implementado un módulo independiente, si no que se ha trabajado la lógica en funciones dentro de los componentes, como se muestra en la figura 16.

```

const loadPilgrimage = async () => {
  const result = await axios.get(`${urlBase}/${id}`)
  setPilgrimage(result.data);
  console.log(result.data);
}

const loadComments = async () => {
  const result = await axios.get(`${urlBaseComments}/pilgrimage/${id}`)
  setComments(result.data);
  console.log(result.data);
  const usernames = result.data.map(comment => comment.username);
  loadUsers(usernames);
}

const loadUsers = async (usernames) => {
  const userRequests = usernames.map(username => axios.get(`${urlBaseUser}/${username}`));
  const usersData = await Promise.all(userRequests);

  const users = usersData.map(user => user.data);
  setUserComment(users);
}

```

Figura 16. Funciones de lógica.

Fuente: elaboración propia.

Los componentes de *React* representan funciones de *Javascript* con una lógica y una presentación enfocadas a la interfaz de usuario. Sus característica clave es que son reutilizables, por lo que se pueden usar en múltiples lugares de la aplicación, como se muestra en la figura 17 y en la figura 18, utilizando el componente *CreatePilgrimage* en distintos lugares y enviándole distintas propiedades según la configuración necesaria, estas propiedades reciben el nombre de “*props*” en el contexto de *React*.

```

export function Home({filter}) {
  return(
    <>
      <Stack spacing={1}>
        {role === 'ROLE_ADMIN' &&(<Button sx={{maxWidth:"15rem", alignSelf:"end"}}
          startIcon={<Add/>} variant="contained" color="primary" onClick={() =>
            {handleModalPilgrimageOpen()}} >
            Nueva romería
          </Button>)}
        <PilgrimageList key={key} filter={filter}/>
      </Stack>
      <CreatePilgrimage open={openModalPilgrimage} close={handleModalPilgrimageClose}
        reload={reloadComponent} />
    </>
  )
}

```

Figura 17. Componente Home.

Fuente: elaboración propia.

```

export function Pilgrimage({ id }) {
  <Modal open={openModalFloat} onClose={handleModalFloatClose}>
  <SelectFloat floats={filteredFloats} selectFloat={setSelectedFloat}>
  </Modal></> : <></></Box>
  <FloatList key={key} id={id} />
</Stack>
</Stack>
<CreatePilgrimage open={openModalPilgrimage} id={id}>
  close={handleModalPilgrimageClose} reload={reloadComponentPilgrimage}/>
  <CreateComment open={openModalComment} close={handleModalPCommentClose}>
  reload={reloadComment} pilgrimageId={id}/>
</Container>
</>
}

```

Figura 18. Componente Pilgrimage.

Fuente: elaboración propia.

Además, hay que añadir que en esta aplicación, se ha hecho uso de la librería *Material UI (MUI)*, que ofrece una colección de componentes ya hechos basados en el *Material Design* de *Google*. Así, se ha optimizado el tiempo dedicado en el desarrollo, ya que se ha podido desarrollar de una forma más rápida. Por otro lado, gracias a *MUI*, se aplica un diseño estandarizado que ofrece una experiencia coherente al usuario. De igual forma, esta librería permite personalizar los componentes para adaptar el diseño, como se muestra en la figura 19, en la que se añaden propiedades *CSS* a los distintos componentes para adaptarlos a los requisitos de la web, por lo que su uso es altamente beneficiabile.

```

<Box sx={{ display: 'flex', gap: '2rem', alignItems: 'center' }}>
  <DirectionsBusOutlined fontSize='large' color='primary' />
  <Typography sx={{ fontWeight: '600' }} variant='h3'>Mis carrozas</Typography>
</Box>

```

Figura 19. Componente Box de MUI.

Fuente: elaboración propia.

Por otra parte, los servicios encapsulan, como ya se ha mencionado, toda la lógica de negocios dentro del *frontend*, así como las comunicación con el *backend*. Los servicios interactúan con el *backend* obteniendo y enviando datos. En el caso de esta aplicación, se han desarrollado unas funciones con esta lógica, tal y como se ha visualizado previamente en la figura 16, donde se muestra como solicita la aplicación los datos al *backend*.

Tal y como se puede apreciar también en la figura 16, la comunicación con el *backend* se hace a través de *axios*, mediante una petición tipo *GET* en este caso. *Axios* es una biblioteca utilizada para realizar solicitudes *HTTP* en el *frontend*, haciendo un manejo automático de los *JSON*.

Por último, es necesario hablar de los *providers* en *React*, ya que permiten gestionar datos y funciones entre los distintos componentes eliminando la necesidad de pasar las propiedades manualmente entre los múltiples niveles y jerarquías de componentes. A través de *Context.Provider* se envuelven todos los componentes que tendrán acceso a los datos, como se puede ver en la figura 20 con el componente *AuthProvider*.

```
<AuthProvider>
  <Menu filter={setSearch} />
  <Container sx={{ mt: "var(--marginTop)", mb: "2rem" }}>
    <Routes>
      <Route path="/" element={<Home filter={search} />}></Route>
      <Route path="/manage-floats" element={<ManageFloats />}></Route>
      <Route path="/:id" element={<PilgrimagePage />}></Route>
      <Route path="/buyTicket/:idPilgrimage/:idFloat" element={<BuyTicketPage />}></Route>
      <Route path="/floats/edit/:id" element={<CreateFloats />}></Route>
      <Route path="/floats/edit" element={<CreateFloats />}></Route>
      <Route path="/login" element={<LoginPage />}></Route>
      <Route path="/myTickets" element={<MyTicketsPage />}></Route>
      <Route path="/profile" element={<Profile />}></Route>
    </Routes>
  </Container>
</AuthProvider>
```

Figura 20. Uso de *AuthProvider*.

Fuente: elaboración propia.

En la aplicación, se ha desarrollado *AuthContext* para manejar el estado de autenticación en la aplicación, como se muestra en la figura 21.

```

// Crea el contexto
const AuthContext = createContext();

// Proveedor del contexto
Tabnine | Edit | Explain
export const AuthProvider = ({ children }) => {
  const [isLoggedIn, setIsLoggedIn] = useState(false);

  const logIn = () => {
    setIsLoggedIn(true);
    localStorage.setItem('isLoggedIn', 'true');
  };
  const logOut = () => {
    setIsLoggedIn(false);
    localStorage.removeItem('isLoggedIn');
  };

  useEffect(() => {
    const savedLoggedState = localStorage.getItem('isLoggedIn');
    if (savedLoggedState === 'true') {
      setIsLoggedIn(true);
    }
  }, []);

  return (
    <AuthContext.Provider value={{ isLoggedIn, logIn, logOut }}>
      {children}
    </AuthContext.Provider>
  );
};

// Hook para usar el contexto
export const useAuth = () => useContext(AuthContext);

```

Figura 21. Desarrollo de AuthContext y AuthProvider.

Fuente: elaboración propia.

Dentro de *AuthProvider*, se han establecido los métodos de inicio y cierre de sesión, gestionando el estado de autenticación del usuario. Además, se ha creado un contexto de *AuthContext*, el cual será accesible mediante la función de *useAuth*, tal y como se muestra en la figura 22, en la que se accede a la variable de *isLoggedIn* mediante esa función.

```

export function Menu({filter}) {
  const { isLoggedIn } = useAuth();
  const [inputText, setInputText] = useState("");
  let inputHandler = (e) => {

```

Figura 22. Uso de useAuth.

Fuente: elaboración propia.

4.1.4. INTERFAZ GRÁFICA

A continuación, se muestra el resultado visual del desarrollo de la aplicación web. Al haber utilizado *Material UI* para su diseño, se ha facilitado la obtención de una experiencia de usuario agradable, siguiendo un diseño lógico, consistente y atractivo en los distintos apartados de la página.

RomeríaViva				
Search...				
Imagen	Nombre	Estado	Fecha	Lugar
	Romería de Disney hola	Finalizada	1 de julio de 2024	Tenor
	Romería de San Juan de Dios	Finalizada	8 de julio de 2024	Aruca
	Romería de las Marías	Próximamente	15 de julio de 2024	Santa María de Guila
	Romería de la Virgen de la Cueva	Finalizada	22 de julio de 2024	Artenara
	Romería de San Roque	Finalizada	29 de julio de 2024	Firgas
< 1 2 3 4 5 6 >				

Figura 23. Página inicio. Fuente: elaboración propia.

RomeríaViva

Search...



Romería de Disney hola

1 de julio de 2024

9:00

Terror

Sitio web oficial

Figura 24. Página romería.

Fuente: elaboración propia.

Romería de Disney hola

📅 1 de julio de 2024 ⌚ 9:00 📍 Teror 🌐 [Sitio web oficial](#)

Descripción

Romería en honor a Nuestra Señora del Pino

Ruta

Desde Arucas hasta Teror

Comentarios

 Maria Rodriguez Jua 29 de noviembre de 2024 19:56 caldcojmad	★★★★☆
 Maria Rodriguez Jua 29 de noviembre de 2024 20:05 hola ejemplo	★★★★★
 Maria Rodriguez Jua 29 de noviembre de 2024 20:06 lalixs	★★★☆☆

Figura 25. Detalles romería y tarjetas opiniones.

Fuente: elaboración propia.

Carrozas disponibles



Carroza Floral
Dueño: float user
Descripción:
Carroza con decoración floral tradicional

Aforo máximo:
40 personas

[COMPRAR ENTRADA](#) **45 €**

Figura 26. Tarjeta de carroza.

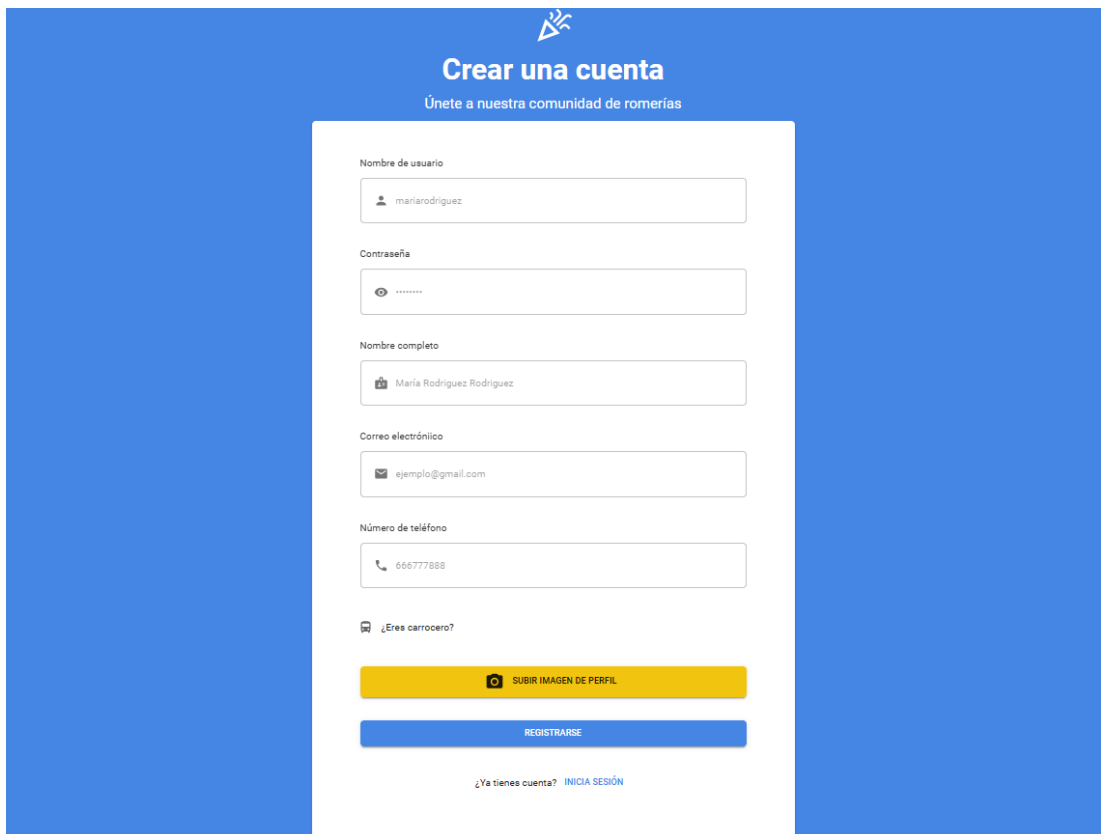
Fuente: elaboración propia.



The login page features a blue background with a white central form. At the top, there is a logo of a hand with a star, followed by the heading "Bienvenido de nuevo" and the subtext "Inicia sesión para continuar". The form contains two input fields: "Usuario" with the text "mariaRodriguez" and "Contraseña" with masked characters. Below these is a blue "INICIAR SESIÓN" button. At the bottom, a link reads "¿No tienes cuenta? [REGÍSTRATE](#)".

Figura 27. Página inicio de sesión.

Fuente: elaboración propia.



The registration page has a blue background with a white central form. It starts with a logo of a hand with a star, followed by the heading "Crear una cuenta" and the subtext "Únete a nuestra comunidad de romerías". The form includes several input fields: "Nombre de usuario" (mariaRodriguez), "Contraseña" (masked), "Nombre completo" (María Rodríguez Rodríguez), "Correo electrónico" (ejemplo@gmail.com), and "Número de teléfono" (666777888). There is a checkbox labeled "¿Eres carrotero?" and a yellow button "SUBIR IMAGEN DE PERFIL". Below these is a blue "REGISTRARSE" button. At the bottom, a link reads "¿Ya tienes cuenta? [INICIA SESIÓN](#)".

Figura 28. Página registro.

Fuente: elaboración propia.

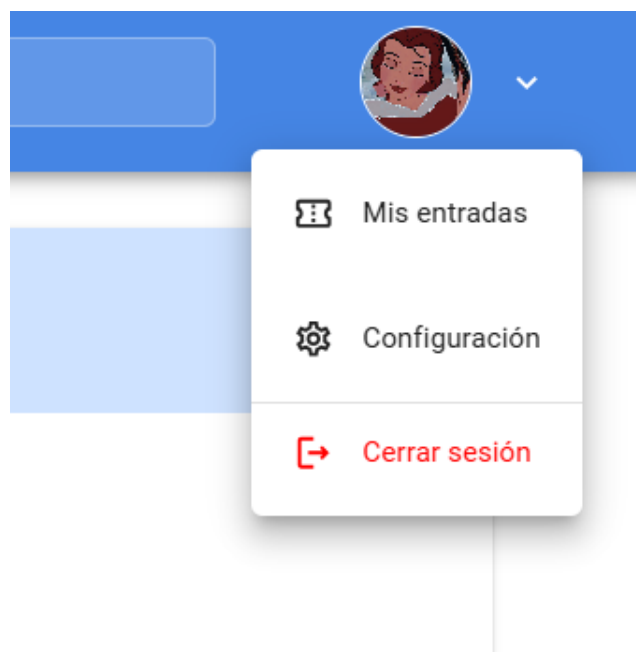


Figura 29. Caja usuario común.

Fuente: elaboración propia.

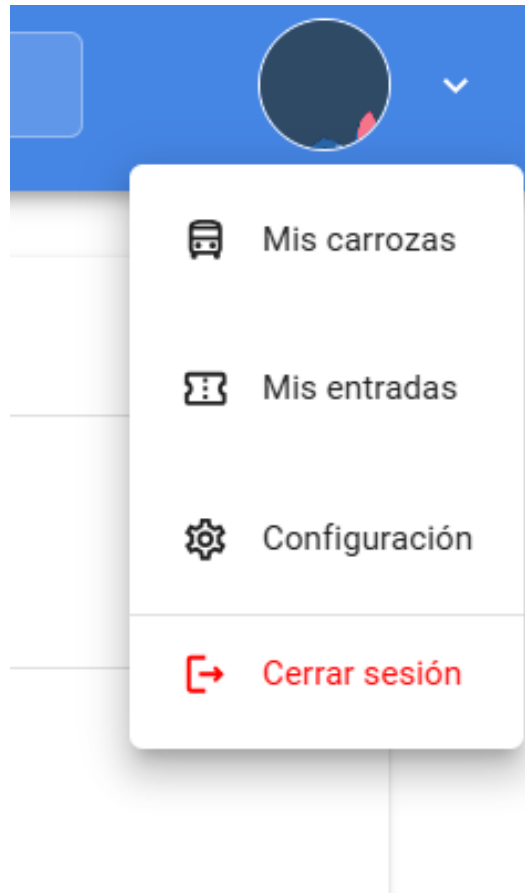


Figura 30. Caja usuario cochero.
Fuente: elaboración propia.

A screenshot of a 'Editar perfil' (Edit profile) form in a web application. The form is white with a light blue header bar. It features a profile picture placeholder and four input fields: 'Nombre completo' (Full name) with the placeholder text 'Nombre ewwUsuario', 'Nombre de usuario' (Username) with the placeholder text 'admin', 'Correo electrónico' (Email) with the placeholder text 'usuario@ejemplo.com', and 'Número de teléfono' (Phone number). The form is set against a background of a blurred application interface.

Figura 31. Configuración perfil.
Fuente: elaboración propia.

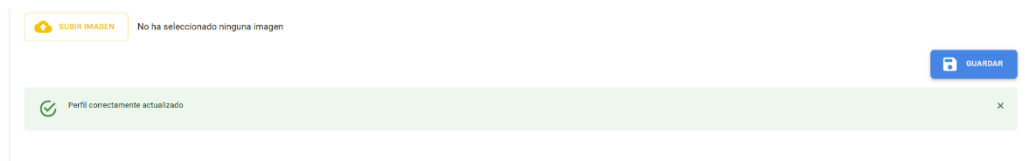


Figura 32. Notificación perfil editado.

Fuente: elaboración propia.

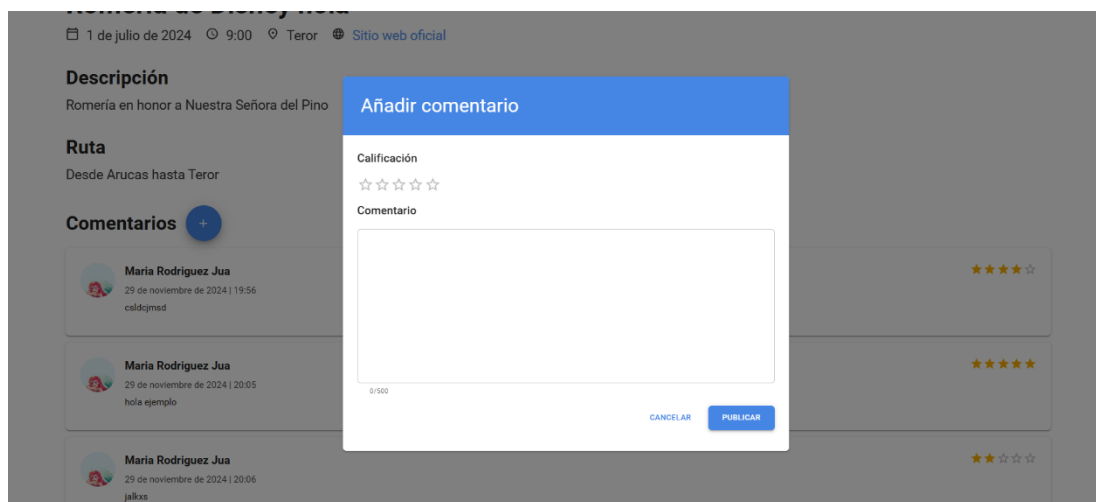


Figura 33. Modal añadir comentario.

Fuente: elaboración propia.

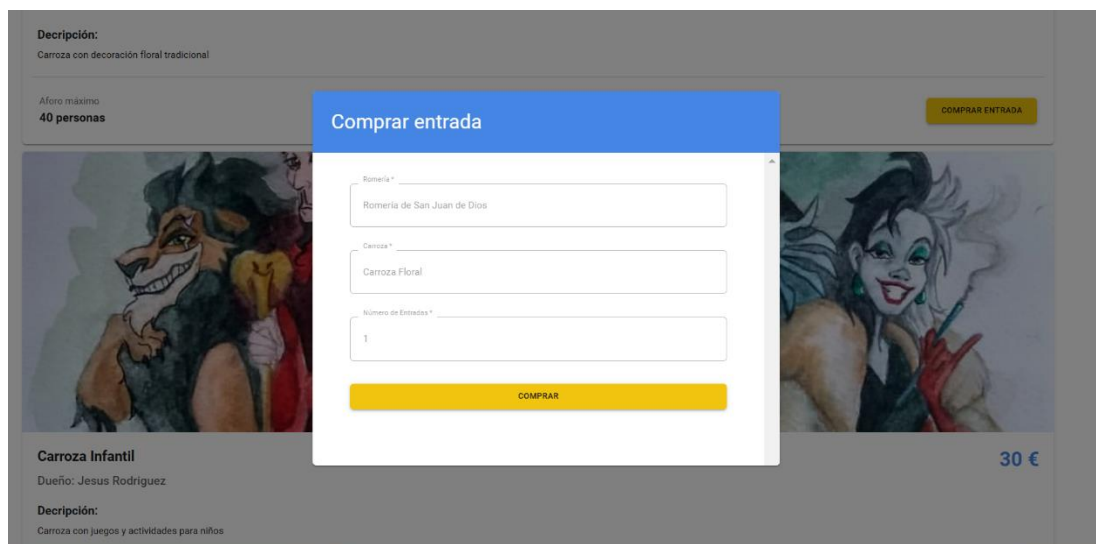


Figura 34. Modal comprar entrada.

Fuente: elaboración propia.

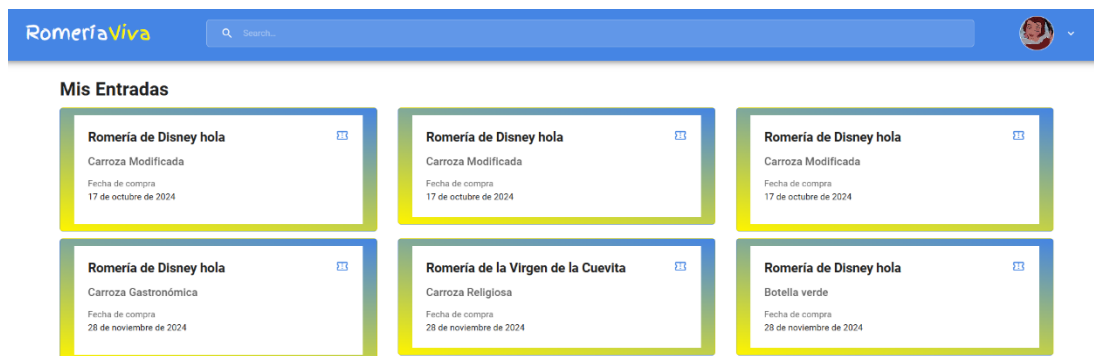


Figura 35. Página mis entradas.

Fuente: elaboración propia.



Figura 36. Modal detalles entrada.

Fuente: elaboración propia.

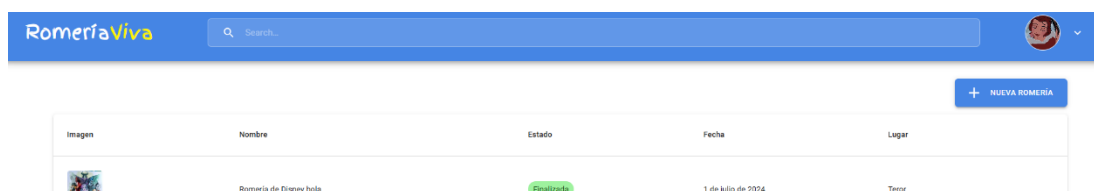


Figura 37. Botón añadir romería.

Fuente: elaboración propia.

Editar romería

Nombre *
Romería de Disney hola

Estado *
Finalizada

Fecha *
01/07/2024

Hora *
09:00

Municipio *
Teror

Página web (opcional)
http://example.com/romeria-pino

Descripción
Romería en honor a Nuestra Señora del Pino

42/1000
Ruta (ej. Desde la Iglesia hasta el edificio del cabildo)
Desde Arucas hasta Teror

SUBIR IMAGEN Imagen seleccionada

1 de julio de 2024 9:00 Teror Sitio web oficial

Figura 38. Modal editar romería.

Fuente: elaboración propia.

Añadir carroza

Nombre *

Precio (en euros) *

Descripción

Propietario

Efectivo

Aforo máximo *

5/1000

SUBIR IMAGEN No se ha seleccionado ninguna imagen

CANCELAR GUARDAR

Figura 39. Modal añadir carroza.

Fuente: elaboración propia.

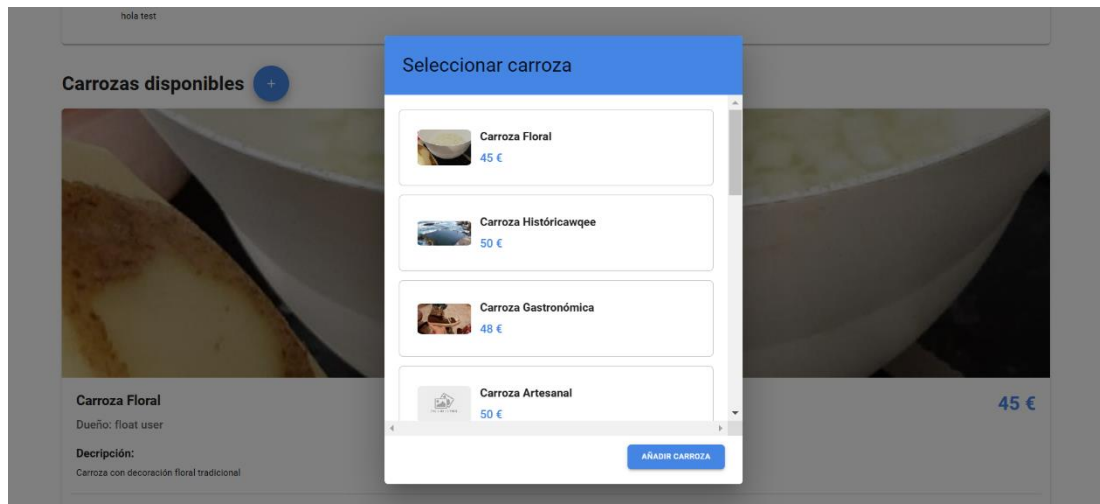


Figura 40. Modal seleccionar carroza.

Fuente: elaboración propia.

4.2. DESARROLLO

A pesar de que la aplicación conste de un *frontend* y un *backend*, se ha profundizado en el desarrollo del *backend*, dado que en un sistema informático lo ideal es que la lógica de negocio esté al 100% en la capa del *backend*. Esto es así dado que el cliente ideal simplemente debería cargar datos y renderizarlos de cara al usuario. De esta forma, cualquier cliente desarrollado a posteriori, véase versión móvil o tablet, se comportarán exactamente igual, ya que compartirán la misma lógica de negocio.

Dentro de los aspectos claves del *backend*, se encuentra el sistema de autenticación mediante *tokens*, el cual garantiza la seguridad en las sesiones. Como se muestra en la figura 41, se ha realizado conteniendo el *username* del cliente logueado, el rol, y el tiempo de expiración, que se ha decidido que sea de 1 hora.

```

public String generateToken(String username, String role) {
    return Jwts.builder()
        .setSubject(username)
        .claim("role", role)
        .setIssuedAt(new Date())
        .setExpiration(new Date(System.currentTimeMillis() + EXPIRATION_TIME))
        .signWith(secretKey)
        .compact();
}

```

Figura 41. Generador de token.

Fuente: elaboración propia.

Como se muestra en la figura 42, a través del controlador, el usuario envía sus credenciales al *endpoint* específico, siendo este validado por *Spring Security*. Tras ello, se genera un *token* que se devuelve al cliente, junto con el *username* y el rol, el cual deberá ser usado en cada solicitud posterior al *backend* para poder acceder a los distintos recursos.

```

@PostMapping("/login")
public ResponseEntity<Map<String, String>> login(@RequestBody LoginRequest loginRequest) {
    try {
        Authentication authentication = authenticationManager.authenticate(
            new UsernamePasswordAuthenticationToken(loginRequest.getUsername(), loginRequest.getPassword())
        );
        User user = userService.getUserByUsername(loginRequest.getUsername());
        String token = tokenService.generateToken(user.getUsername(), user.getRole());
        Map<String, String> response = new HashMap<>();
        response.put("token", token);
        response.put("username", user.getUsername());
        response.put("role", user.getRole());
        return ResponseEntity.ok(response);
    } catch (AuthenticationException e) {
        return ResponseEntity.status(401).body(Map.of("error", "Login failed"));
    }
}

```

Figura 42. Método login.

Fuente: elaboración propia.

Para poder restringir las funciones según los roles de usuario, es necesario configurarlo a través de un filtro de seguridad, como se muestra en la figura 43, para especificar a qué *endpoint* tiene acceso cada usuario, según el tipo de solicitud *REST*.


```

.authorizeHttpRequests(auth -> auth
    .requestMatchers(HttpMethod.POST, "/pilgrimages").hasAuthority("ROLE_ADMIN")
    .requestMatchers(HttpMethod.PUT, "/pilgrimages/**").hasAuthority("ROLE_ADMIN")
    .requestMatchers(HttpMethod.DELETE, "/pilgrimages/**").hasAuthority("ROLE_ADMIN")
    .requestMatchers(HttpMethod.GET, "/pilgrimages/*/addFloat/*").hasAuthority("ROLE_FLOATS")
    .requestMatchers(HttpMethod.POST, "/floats").hasAuthority("ROLE_FLOATS")
    .requestMatchers(HttpMethod.PUT, "/floats/**").hasAuthority("ROLE_FLOATS")
)

```

Figura 43. Filtro de seguridad.

Fuente: elaboración propia.

Además, estas restricciones se han añadido también a los métodos, con la etiqueta `@PreAuthorize`, como se muestra en la figura 44.

```

@PostMapping()
@PreAuthorize("hasAuthority('ROLE_ADMIN')")
public PilgrimageResponse addPilgrimage(@RequestBody PilgrimageRequest pilgrimageRequest){
    System.out.println("PilgrimageRequest: " + pilgrimageRequest);
    Pilgrimage pilgrimage = converter.convertFromRequest(pilgrimageRequest);
    return converter.convertToResponse(pilgrimageService.savePilgrimage(pilgrimage));
}

```

Figura 44. Método `addPilgrimage`.

Fuente: elaboración propia.

Otra de las implementaciones interesantes que se han realizado dentro del *backend* es la paginación, optimizando así el manejo de datos y reduciendo así la carga tanto en el cliente como en el servidor.

Para ello, se ha hecho uso de las interfaces de *Page* y *Pageable* ofrecidas por Spring *Data JPA*, estableciendo un método en el servicio que maneje los datos según la página, como se muestra en la figura 45.

```

@Override
public Page<Pilgrimage> getPilgrimages(int pageNo, int pageSize){
    Pageable pageable = PageRequest.of(pageNo, pageSize);
    return pilgrimageRepository.findAll(pageable);
}

```

Figura 45. Método `getPilgrimages`.

Fuente: elaboración propia.

Además, dentro del controlador, mostrado en la figura 46, se ha establecido un método para la paginación, estableciendo unos valores por defecto para la página en la que se

encuentra y el tamaño de datos devuelto, por si no viniesen en la petición. De esta forma el cliente recibirá únicamente 5 romerías en cada petición, pudiendo así manejar más fácilmente los detalles de estas.

```
@GetMapping()
public ResponseEntity<Page<PilgrimageResponse>> getPilgrimagesPagination(
    @RequestParam(defaultValue = "0") int pageNo,
    @RequestParam(defaultValue = "5") int pageSize){
    Page<Pilgrimage> pilgrimages = pilgrimageService.getPilgrimages(pageNo, pageSize);
    return ResponseEntity.ok(pilgrimages.map(converter::convertToResponse));
}
```

Figura 46. Método getPilgrimagesPagination.

Fuente: elaboración propia.

Por último, es interesante mostrar el desarrollo de los casos de uso de las entidades. Para ello, se presentará a continuación el desarrollo de la entidad de comentarios, dado que es una entidad sencilla y ello facilitará su entendimiento.

Dentro de la clase modelo, mostrada en la figura 47, se establecen las propiedades que se han establecido en la base de datos. Se identifica como una clase entidad gracias a la etiqueta `@Entity`. Además, para acortar el desarrollo, con la ayuda de la librería de *Lombok*, se añade la etiqueta `@Data` para los métodos *getter*, *setter*, *toString*, y otros. Con las etiquetas `@Builder`, `@NoArgsConstructor` y `@AllArgsConstructor`, permitimos generar fácilmente la construcción de la clase, permitiendo no tener ningún parámetro o todos. Con las etiquetas `@Id`, y `@GeneratedValue`, se establece la clave primaria que se encuentra en la base de datos y la forma en la que se genera. Con `@ManyToOne` y `@JoinColumn`, relacionamos la propiedad de una clave foránea con la columna en la base de datos.

```

@Entity
@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class Comment {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer id;
    private String description;
    private Integer valoracion;
    @ManyToOne
    @JoinColumn(name = "username")
    private User user;
    @ManyToOne
    @JoinColumn(name = "id_pilgrimage")
    private Pilgrimage pilgrimage;
    @Column(nullable = false)
    private LocalDateTime date;
}

```

Figura 47. Clase Comment.

Fuente: elaboración propia.

A continuación, en las figuras 48, 49 y 50, se muestran los modelos de respuesta y petición, además de los métodos dentro del convertidor del modelo. Como se puede apreciar, dentro de la respuesta y la petición no se envían clases de entidad, si no que se sustituye por la propiedad primaria, evitando así enviar un objeto dentro de un objeto, además de la posibilidad de un bucle infinito al contenerse entre ambos objetos (es decir, un comentario tiene un usuario, y ese usuario tiene ese comentario, por lo que se daría una situación no deseada al repetirse una y otra vez).

```
@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class CommentRequest {

    private Integer id;
    private String description;
    private Integer valoration;
    private String username;
    private Integer pilgrimageId;
    private LocalDateTime date;

}
```

Figura 48. Clase CommentRequest.

Fuente: elaboración propia.

```

@Data
@Builder
@NoArgsConstructor
@AllArgsConstructor
public class CommentResponse {
    private Integer id;
    private String description;
    private Integer valoracion;
    private String username;
    private Integer pilgrimageId;
    private LocalDateTime date;
}

```

Figura 49. Clase CommentResponse.

Fuente: elaboración propia.

```

└─ laura
public CommentResponse convertToResponse(final Comment comment){
    return new CommentResponse(comment.getId(),
        comment.getDescription(), comment.getValoration(), comment.getUser().getUsername(),
        comment.getPilgrimage().getId(), comment.getDate());
}

2 usages └─ laura *
public Comment convertFromRequest(final CommentRequest commentRequest){
    return new Comment(commentRequest.getId(),
        commentRequest.getDescription(), commentRequest.getValoration(),
        userService.getUserByUsername(commentRequest.getUsername()),
        pilgrimageService.getPilgrimageById(commentRequest.getPilgrimageId()), commentRequest.getDate());
}

```

Figura 50. Métodos convertToResponse y convertFromRequest.

Fuente: elaboración propia.

Dentro de la clase de repositorio, se hace uso de *JPA*, siendo este el encargado de toda la comunicación con la base de datos.

Además, para establecer funciones que no se encuentran por defecto en la interfaz, se utilizan los métodos con nombres de acciónEntidadValor, como se muestra en la figura 51 con *findByUserUsername* y *findByPilgrimageId* y así *JPA* lo interpreta y realiza el desarrollo por debajo, sin tener que realizarse manualmente mediante una consulta, aunque también ofrece esta opción para casos más complejos en los que sea necesario.

```
3 usages  👤 laura
public interface CommentRepository extends JpaRepository<Comment,Integer> {
    1 usage  👤 laura
    List<Comment> findByUserUsername(String username);

    1 usage  👤 laura
    List<Comment> findByPilgrimageId(Integer idFloat);
}
```

Figura 51. Interfaz CommentRepository.

Fuente: elaboración propia.

Para la clase de servicio, se ha desarrollado una interfaz y una clase que la implementa, como se muestran en la figura 52 y en la figura 53.

Dentro de la interfaz, se establecen todos los métodos que serán necesarios para la lógica de negocio.

```
10 usages  1 implementation  👤 laura
public interface ICommentService {
    1 usage  1 implementation  👤 laura
    List<Comment> getCommentsByPilgrimage(Integer idPilgrimage);
    1 usage  1 implementation  👤 laura
    List<Comment> getCommentsByUser(String username);

    2 usages  1 implementation  👤 laura
    Comment getCommentById(Integer id);
    2 usages  1 implementation  👤 laura
    Comment saveComment(Comment comment);
}
```

Figura 52. Interfaz ICommentService.

Fuente: elaboración propia.

Dentro de la clase, se realiza el desarrollo de los métodos previamente definidos, haciendo uso del repositorio para la conexión con la base de datos e implementar toda la lógica necesaria en el trato de los datos.

En la figura 53, se muestran dos implementaciones muy sencillas, en las que se devolverá una lista de todos los comentarios existentes según la romería y según el usuario.

```
1 usage  👤 laura
@Override
public List<Comment> getCommentsByPilgrimage(Integer idPilgrimage) {
    return commentRepository.findByPilgrimageId(idPilgrimage);
}

1 usage  👤 laura
@Override
public List<Comment> getCommentsByUser(String username) {
    return commentRepository.findByUserUsername(username);
}
```

Figura 53. Métodos `getCommentsByPilgrimage` y `getCommentsByUser` de `CommentService`.

Fuente: elaboración propia.

Para finalizar, la clase de controlador maneja los *endpoint*, devolviendo un resultado u otro según el *endpoint* establecido. En la figura 54 se muestran los *endpoint* de los métodos vistos anteriormente. Se establece que al recibir una solicitud de tipo *GET* a los *endpoints* específicos, se devolverá una lista con el modelo comentario en formato respuesta al cliente.

```

@laura *
@GetMapping(Ⓜ"user/{username}")
public List<CommentResponse> getCommentsByUsername(@PathVariable String username){
    return commentService.getCommentsByUser(username).stream()
        .map(converter::convertToResponse)
        .collect(Collectors.toList());
}

@laura *
@GetMapping(Ⓜ"pilgrimage/{id}")
public List<CommentResponse> getCommentsByPilgrimage(@PathVariable Integer id){
    return commentService.getCommentsByPilgrimage(id).stream()
        .map(converter::convertToResponse)
        .collect(Collectors.toList());
}

```

Figura 54. Métodos `getCommentsByUsername` y `getCommentsByPilgrimage` de `CommentController`.

Fuente: elaboración propia.

4.3. PRUEBAS DEL SISTEMA

Uno de los aspectos más importantes en el desarrollo del *backend* es la elaboración de pruebas del sistema para comprobar el correcto funcionamiento de la aplicación. Así, en este proyecto se han elaborado pruebas unitarias y de integración, realizando así las distintas pruebas para garantizar un análisis exhaustivo de los distintos componentes del sistema, promoviendo su calidad y su robustez.

Pruebas unitarias

En cuando a las pruebas unitarias, se han implementado con *JUnit* y *Mockito*, verificando así todas las funcionalidades individuales de las clases de repositorio, servicio y controlador. En estas pruebas se ha buscado probar los diferentes elementos de forma aislada, por lo que se han utilizado “*mocks*” de las diferentes dependencias de cada elemento que se estaba testando, esto se consigue mediante *Mockito* que se encarga de generar este “*mock*”, que no es más que una copia sin datos de dicha clase. De esta forma se evitan posibles errores como *NullPointerException* y se centra simplemente en verificar que se hacen las llamadas correctas entre clases con los datos correctos mediante métodos como “*verify()*”.

Además, para el desarrollo de test en la clase de repositorio, se ha elaborado una base de datos en memoria con *H2*, como se muestra en la figura 55. Este enfoque evita efectuar cambios en la base de datos real, asegurando que las pruebas sean reproducibles en cualquier entorno. La configuración inicial incluye la creación de una estructura simulada con datos esenciales para los casos de prueba.

```
@TestConfiguration
public class TestDatabaseConfig {

    @laura
    @Bean
    @Primary
    public DataSource dataSource() {
        return new EmbeddedDatabaseBuilder()
            .setType(EmbeddedDatabaseType.H2)
            .build();
    }
}
```

Figura 55. Configuración test base de datos.

Fuente: elaboración propia.

En la figura 56, se presenta uno de los *test* de la clase servicio, “*shouldSavePilgrimageTest()*”, dejando explícito en su nombre lo que debería hacer mencionado *test*. En este caso, se utiliza la función “*when()*” para establecer el resultado simulado cuando llame a la función contenida dentro del método en el servicio, que en este caso sería la respuesta de la acción de guardar la romería dentro del método del servicio por lo que debería devolver, que sería la misma romería. Tras ello, llamamos al método que se quiere probar, que sería guardar romería, y con el “*verify()*” verificamos que la respuesta de este método sea la misma romería. Así, se comprueba el funcionamiento de guardar una romería sin tener que llamar la función externa de guardado de “*pilgrimageRepository*”. Este enfoque garantiza que las pruebas no dependan del entorno real, favoreciendo el enfoque de la lógica interna del sistema.

```

@Test
void shouldSavePilgrimageTest(){

    when(pilgrimageRepository.save(pilgrimage)).thenReturn(pilgrimage);

    pilgrimageService.savePilgrimage(pilgrimage);

    verify(pilgrimageRepository).save(pilgrimage);
}

```

Figura 56. Test *shouldSavePilgrimageTest*.

Fuente: elaboración propia.

Pruebas de integración

Por otro lado, se ha hecho el desarrollo de test de integración, los cuales verifican el funcionamiento completo del *backend* desde la llamada al controlador hasta su interacción final con la base de datos. Estas pruebas garantizan la correcta cohesión entre los componentes principales del sistema.

En la figura 57, se ilustra el test “*when_you_call_pilgrimage_with_put_id_then_update*”, un ejemplo en el que se actualiza una romería, como se ha dejado claro en el nombre del propio *test*, en el cual se indica qué debe hacer este. Primero, se establece una *string* con la solicitud en formato *JSON*. Posteriormente, se envía a través de un *mock* una solicitud de tipo *PUT* al *endpoint* específico, incluyendo el contenido necesario para su petición.

Tras ello, comprobamos que el estado de la respuesta sea satisfactorio, y además vamos comprobando uno a uno los campos de la respuesta que se han modificado, verificando que la actualización se ha hecho correctamente y no ha habido ningún fallo en la ejecución.

```

@Test
@Order(4)
void when_you_call_pilgrimage_with_put_id_then_update() throws Exception{
    String jsonRequest = "{ \"id\": \"+idAuxPilgrimage+\", \"name\": \"Modificada Romeria\", \"place\": \" +
        \"Ubicación modificada\", \"description\": \"Descripción modificada\", \"date\": \" +
        \"2023-02-28T19:00:00\", \"status\": \"2\"}";

    mockMvc.perform(put( urlTemplate: "/pilgrimages/{id}", idAuxPilgrimage).contentType(MediaType.APPLICATION_JSON)
        .content(jsonRequest))
        .andExpect(status().isOk())
        .andExpect(jsonPath( expression: "$.name").value( expectedValue: "Modificada Romeria"))
        .andExpect(jsonPath( expression: "$.place").value( expectedValue: "Ubicación modificada"))
        .andExpect(jsonPath( expression: "$.description").value( expectedValue: "Descripción modificada"))
        .andExpect(jsonPath( expression: "$.url").doesNotExist())
        .andExpect(jsonPath( expression: "$.date").value( expectedValue: "2023-02-28T19:00:00"))
        .andExpect(jsonPath( expression: "$.route").doesNotExist())
        .andExpect(jsonPath( expression: "$.image").doesNotExist())
        .andExpect(jsonPath( expression: "$.status").value( expectedValue: 2));
}

```

Figura 57. Test when_you_call_pilgrimage_with_put_id_then_update.

Fuente: elaboración propia.

5. RESULTADOS, CONCLUSIONES Y TRABAJOS FUTUROS

5.1. RESULTADOS

La aplicación de romerías tiene como objetivo la digitalización de un mercado tradicional, pero tras el análisis del modelo de negocio, se muestra un enfoque fuerte y rentable. Se propone satisfacer las necesidades de todos los actores en una romería, otorgando así a la aplicación el potencial para convertirse en la herramienta principal que se utilice en estos eventos.

Además, se asegura una monetización efectiva gracias a la variedad de ingresos junto con los bajos costes iniciales, respaldando así la rentabilidad de la plataforma desde el primer año.

En los próximos años, se espera un aumento de usuarios y la expansión a todas las islas, e incluso se plantea la posibilidad de incluir otro tipo de eventos similares, disponiendo de un alto potencial de crecimiento.

Entre los resultados del producto mínimo viable desarrollado, se destaca la centralización de la información, la digitalización del proceso de compra-venta y la facilitación de la logística.

En definitiva, se presenta una idea de negocio sostenible, escalable y adaptable, que además cuenta con un claro camino hacia la mejora tecnológica, garantizando que la aplicación se posicione como referente en el ámbito de las romerías.

5.2. CONCLUSIONES

El desarrollo de este proyecto me ha permitido obtener aprendizajes relevantes tanto en el ámbito de la informática como en el ámbito empresarial.

Por un lado, la elaboración del modelo de negocio ha sido una experiencia extremadamente útil y enriquecedora, ya que me ha supuesto un reto convertir una idea tecnológica en una propuesta rentable y viable. Además, analizar los distintos aspectos del modelo *canvas*, me ha dado una idea más clara de como conectar la aplicación con el mundo real, estableciendo estrategias comerciales para ello. Este

proceso también me ha permitido desarrollar otras habilidades necesarias como son las decisiones estratégicas, la evaluación de mercados y la identificación de oportunidades, herramientas que son clave para cualquier proyecto que quiera montar en el futuro.

Por otro lado, el uso de tecnologías como *Spring Boot* y *React JS* ha sido una experiencia formativa de gran valor, ya que me ha permitido acercarme al desarrollo de *software* profesional, usando tecnologías modernas y que hoy día se solicitan en múltiples ofertas laborales. Además, el uso de una arquitectura hexagonal, así como el uso de *Spring Security* en el *backend* ha ampliado en gran medida mis conocimientos, preparándome mejor para afrontar retos más complejos en mi carrera profesional.

Finalmente, el desarrollo de este proyecto me ha ayudado a consolidar mi confianza como profesional, mostrándome que soy capaz de enfrentarme a problemas reales y proponer soluciones efectivas e innovadores.

En definitiva, este trabajo me ha aportado una visión de lo que es tener una idea y llevarla a cabo, desarrollando una solución tecnológica y empresarial.

5.3. TRABAJOS FUTUROS

A posteriori, se prevé el desarrollo de una versión móvil de la aplicación, creando una app móvil nativa para sistemas *iOs* y *Android*, utilizando *React Native* como base para su desarrollo.

También se establece la idea de implementar un sistema de mensajería entre usuarios, facilitando así la comunicación entre los asistentes, los cocheros y los administradores.

Además, también se desarrollarán notificaciones *push* para tener avisos sobre cualquier cambio que pudiese haber en una carroza o una romería, además de recordatorios días antes de los eventos y cualquier otra actualización que fuese relevante.

Otro desarrollo interesante sería la implementación de mapas dentro de la aplicación en la que se muestren las romerías en los distintos puntos de la isla, así como las rutas específicas que se llevarán a cabo en cada una.

Por último, una de las ideas a futuro sería el desarrollo de un apartado para administradores y cocheros en el que hubiese estudios estadísticos sobre la cantidad de asistentes, la retroalimentación o el volumen de compra de entradas en carrozas.

6. FUENTES DE INFORMACIÓN

30 Seconds of Code. (2024, febrero 7). *Remove accents in JavaScript*. Recuperado de <https://www.30secondsofcode.org/js/s/remove-accents/>

Arsys. (s.f.). Búsqueda de dominios. Recuperado de <https://www.arsys.es/dominios/buscar/vivaromer%C3%ADa>

Arsys. (2024, octubre 30). ¿Qué es *Visual Studio Code* y cuáles son sus ventajas? Recuperado de <https://www.arsys.es/blog/que-es-visual-studio-code-y-cuales-son-sus-ventajas>

Asana. (2024, febrero 3). Ejemplo de análisis competitivo. Recuperado de <https://asana.com/es/resources/competitive-analysis-example>

Baeldung. (s.f.). *Spring Boot Tutorials*. Recuperado de <https://www.baeldung.com/>

Bluuweb. (2023, agosto 2). *Material UI en React*. Recuperado de <https://bluuweb.dev/05-react/material-ui.html>

Canva. (s.f.). Herramientas de diseño gráfico. Recuperado de <https://www.canva.com/>

Calzados Canarias. (2024). Romerías en Gran Canaria 2024. Recuperado de https://calzadoscanarias.com/romerias-en-gran-canaria-2024/?srsltid=AfmBOoo2bH47c_1ueGbg4H77AjIHnpqJLEZyA7cZGx0ySOFT6e8I3-Pm

Casa de los Balcones. (2024, enero 10). Calendario de Romerías en Tenerife 2024. Recuperado de <https://casa-balcones.com/calendario-de-romerias-en-tenerife-2024/>

Consejo de la Unión Europea. (s.f.). Reglamento General de Protección de Datos. Recuperado de <https://www.consilium.europa.eu/es/policies/data-protection/data-protection-regulation/>

Datosmacro Expansión. (2024). Demografía de Canarias. Recuperado de <https://datosmacro.expansion.com/demografia/poblacion/espana-comunidades-autonomas/canarias#:~:text=Canarias%20cerr%C3%B3%20junio%20de%202024,cuanto%20a%20poblaci%C3%B3n%20se%20refiere>

Dev.to. (2022, enero 3). *Search Bar in React.js*. Recuperado de <https://dev.to/salehmubashar/search-bar-in-react-js-5451>

Diagrams.net. (s.f.). Herramienta de diagramación. Recuperado de <https://app.diagrams.net/>

Docs Spring. (s.f.). Documentación oficial de *Spring*. Recuperado de <https://docs.spring.io/>

Dongee. (2024, marzo 31). Características, ventajas y desventajas de *MySQL*. Recuperado de <https://www.dongee.com/tutoriales/mysql-caracteristicas-ventajas-y-desventajas/>

Ebac. (2023, julio 17). ¿Qué es *React*?. Recuperado de <https://ebac.mx/blog/que-es-react>

ElDiario.es. (2024, septiembre 7). La romería del Pino congrega a 35,000 personas. Recuperado de https://www.eldiario.es/canariasahora/sociedad/romeria-ofrenda-pino-congrega-teror-35-000-personas_1_11636719.html#:~:text=Multitudinaria%20romer%C3%ADa%20ofrenda%20en%20Teror

EUR-Lex. (2016). Reglamento general de protección de datos (RGPD). Recuperado de <https://eur-lex.europa.eu/ES/legal-content/summary/general-data-protection-regulation-gdpr.html>

Facebook. (s.f.). Plataforma oficial. Recuperado de <https://www.facebook.com/>

Firebase. (s.f.). Precios de *Firebase*. Recuperado de <https://firebase.google.com/pricing?hl=es-419>

Formadores IT. (2023, agosto 24). *Postman* y sus principales ventajas. Recuperado de <https://formadoresit.es/que-es-postman-cuales-son-sus-principales-ventajas/>

Glassdoor. (s.f.). Sueldos en atención al cliente. Recuperado de https://www.glassdoor.es/Sueldos/atencion-al-cliente-sueldo-SRCH_KO0,19.htm

Gobierno de Canarias. (2019). Estadísticas de Restauración en Canarias. Recuperado de

https://www.gobiernodecanarias.org/cmsgobcan/export/sites/turismo/downloads/estadistica/restauracion/Totales_Canarias_2000-2019.xls

GoDaddy. (2024, agosto 26). ¿Qué es *IntelliJ IDEA*?. Recuperado de <https://www.godaddy.com/resources/es/crearweb/intellij-idea-que-es>

Hola Islas Canarias. (s.f.). Portal turístico oficial. Recuperado de <https://www.holaislascanarias.com/>

HubSpot. (2023, mayo 10). ¿Qué es arquitectura hexagonal?. Recuperado de <https://blog.hubspot.es/website/que-es-arquitectura-hexagonal#:~:text=La%20arquitectura%20hexagonal%2C%20tambi%C3%A9n%20conocida,que%20tiene%20como%20objetivo%20principal>

Infoautónomos. (2024, mayo 22). Tarifa Plana para Autónomos. Recuperado de <https://www.infoautonomos.com/seguridad-social/tarifa-plana-autonomos/>

Infoautónomos. (2024, noviembre 5). Cuota de autónomos: ¿Cuánto se paga?. Recuperado de https://www.infoautonomos.com/seguridad-social/cuota-de-autonomos-cuanto-se-paga#:~:text=El%20nuevo%20sistema%20establece%20las,y%20m%C3%A1xima%20de%20590%20euros*

Innowise. (s.f.). Ventajas y desventajas de *Java*. Recuperado de <https://innowise.com/es/blog/benefits-and-drawbacks-of-java/>

Kinsta. (2023, agosto 21). *Git* vs. *GitHub*. Recuperado de <https://kinsta.com/es/base-de-conocimiento/git-vs-github/>

KeepCoding. (2024, marzo 15). ¿Qué son los *Providers* en *React*?. Recuperado de <https://keepcoding.io/blog/que-son-los-providers-en-react#:~:text=Los%20providers%20en%20React%20son,de%20la%20jerarqu%C3%ADa%20de%20componentes>

La Provincia. (2024, agosto 29). La vara de Pescao llena Arinaga. Recuperado de <https://www.laprovincia.es/gran-canaria/2024/08/29/vara-pescao-llena-arinaga-36-107517569.html>

La Provincia. (2024, septiembre 22). Valsequillo echa las calles para ofrecer sus frutos. Recuperado de <https://www.laprovincia.es/gran-canaria/2024/09/22/valsequillo-echa-calles-ofrecer-frutos-108411247.html>

La Provincia. (2024, octubre 27). La Romería de San Rafael gana la UD. Recuperado de <https://www.laprovincia.es/gran-canaria/2024/10/27/romeria-san-rafael-gana-ud-110518112.html>

LinkedIn. (s.f.). ¿Por qué usar Maven?. Recuperado de <https://www.linkedin.com/advice/3/why-should-you-use-maven-java-development-vodbe?lang=es&originalSubdomain=es>

Local Guide Gran Canaria. (2024, agosto 10.). Romerías en Gran Canaria. Recuperado de <https://localguidegrancanaria.com/romerias-gran-canaria/>

Lucid. (s.f.). Herramienta de diagramación. Recuperado de <https://lucid.app/>

Medium. (2023, noviembre 17). *Simplifying Pagination in Spring Boot*. Recuperado de <https://medium.com/@rathoreabhi1990/simplifying-pagination-in-spring-boot-4e33c9ff4d07>

Meetup. (s.f.). Plataforma para eventos. Recuperado de <https://www.meetup.com/es-ES/>

MUI. (s.f.). Material UI. Recuperado de <https://mui.com/>

Osterwalder, A., & Pigneur, Y. (2010). *Business Model Generation: A Handbook for Visionaries, Game Changers, and Challengers*.

PlantUML. (s.f.). Herramienta UML. Recuperado de <https://www.plantuml.com/>

RD Station. (2024, julio 15). Modelo Canvas: ¿Qué es?. Recuperado de <https://www.rdstation.com/blog/es/modelo-canvas-que-es/>

Sensedia. (2020, agosto 27). Uso del patrón de arquitectura hexagonal. Recuperado de <https://www.sensedia.com.es/post/uso-del-patron-de-arquitectura-hexagonal#:~:text=Soluci%C3%B3n%20de%20servicios%20externos%20independientes,F%C3%A1cil%20de%20probar%20la%20aplicaci%C3%B3n>

Spring IO. (s.f.). *Spring Security*. Recuperado de <https://spring.io/projects/spring-security>

Spring Initializr. (s.f.). Configura tu proyecto Spring Boot. Recuperado de <https://start.spring.io/>

Statista. (2024, julio 25). Población de Canarias por Isla. Recuperado de <https://es.statista.com/estadisticas/474029/poblacion-de-canarias-por-isla/#:~:text=La%20isla%20de%20Tenerife%20era,unas%20869.000%20personas%20en%202024>

Stripe. (s.f.). Precios de Stripe. Recuperado de <https://stripe.com/es/pricing>

StubHub. (s.f.). Compra de entradas. Recuperado de <https://www.stubhub.es/>

Ticketmaster. (s.f.). Compra de entradas. Recuperado de <https://www.ticketmaster.es/>

Turistas Canarias. (2024, noviembre 6). Afluencia turística en Canarias. Recuperado de <https://turistascanarias.com/#:~:text=El%20a%C3%B1o%202023%20marc%C3%B3%20un,de%20un%207%2C19%20%25>

W3Schools. (s.f.). Tutoriales y referencias. Recuperado de <https://www.w3schools.com/>

YoPagoPorClick. (s.f.). Definición de CPC, CTR, CPA, CPM, CPL. Recuperado de <https://yopagoporclic.com/cpc-ctr-cpa-cpm-cpl/>

ANEXO I: MANUAL DE USUARIO

Este manual de usuario tiene como objetivo ayudar a comprender el funcionamiento de la aplicación “RomeríaViva”.

Ver romerías

Al entrar en la página web, lo primero que se muestra en el inicio es una lista de romerías con información relevante: imagen, nombre, estado, fecha y lugar.

Esta página es accesible al entrar, o pulsando en el logo “RomeríaViva”.

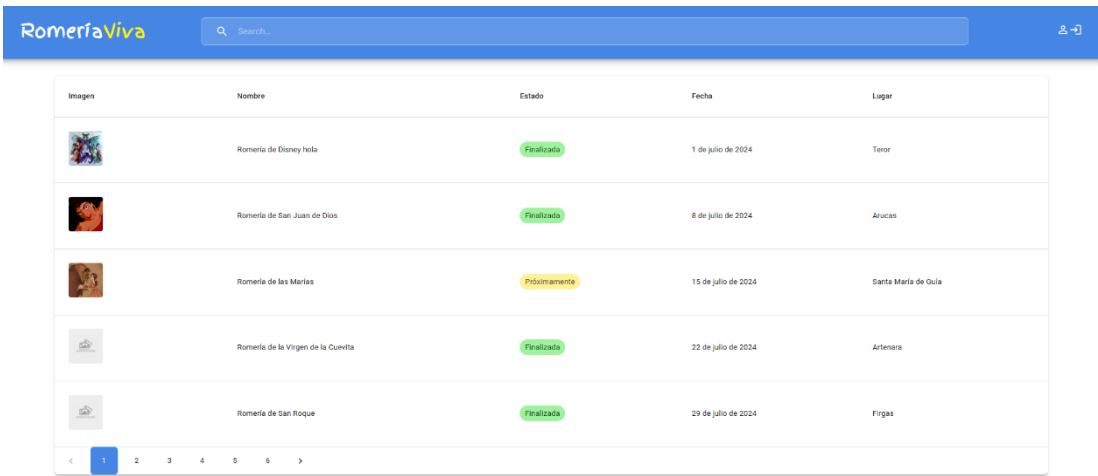


Ilustración 1. Página inicio.

Fuente: elaboración propia.

Dentro de esta página, se puede filtrar en la búsqueda por nombre, estado, fecha y lugar, haciendo más fácil el encuentro de una búsqueda en específico.

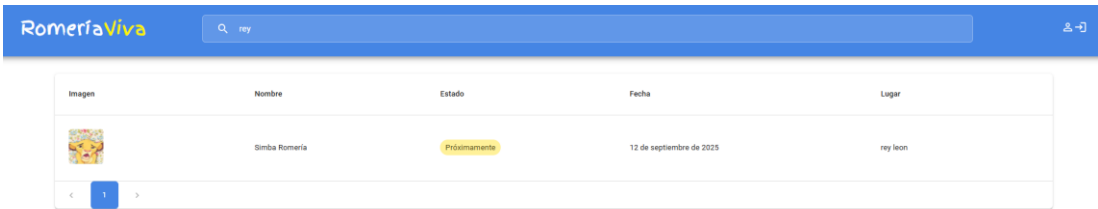


Ilustración 2. Búsqueda de romería.

Fuente: elaboración propia.

Detalles romerías

Al pulsar una de las romerías, esta se abrirá para mostrar todos sus detalles. Está dividido en varias secciones: detalles de la romería, descripción, ruta, comentarios, carrozas disponibles.

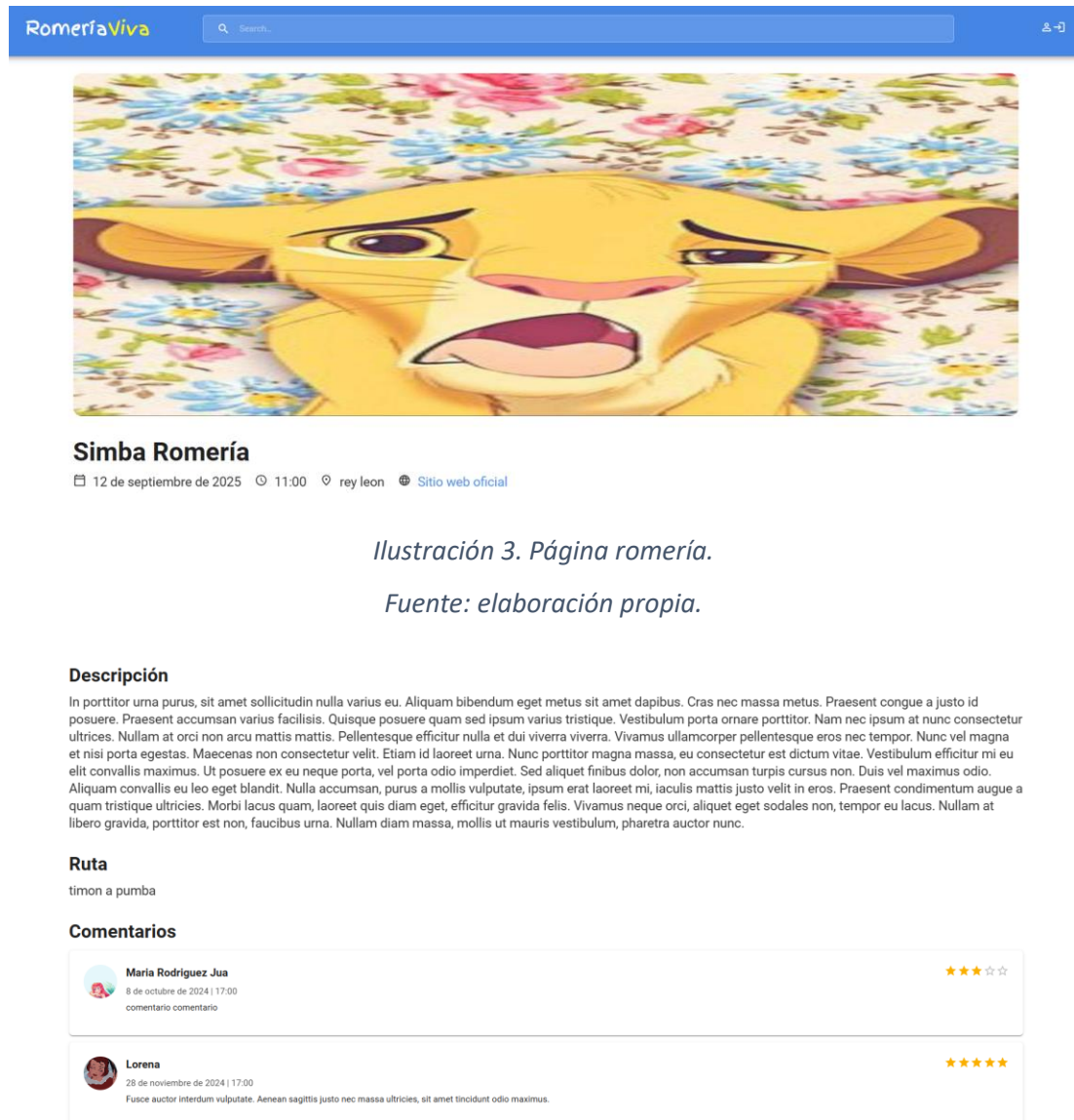


Ilustración 3. Página romería.

Fuente: elaboración propia.

Ilustración 4. Detalles romería y opiniones.

Fuente: elaboración propia.

Carrozas disponibles



Botella verde 12 €

Dueño: float user

Descripción:

Nullam quis diam gravida, tempor turpis a, ornare ligula. Suspendisse consequat maximus purus sit amet dictum. Aenean dapibus ante nec lectus hendrerit, nec pulvinar orci facilisis. Aliquam ut enim eu massa scelerisque vulputate. Vestibulum viverra dui id eros convallis, in viverra nibh venenatis. Nunc elementum risus sed convallis bibendum. Integer dictum ex scelerisque euismod volutpat. Vestibulum maximus, odio nec trincidunt vulputate, libero urna sollicitudin nisi, non commodo diam tellus vel nisi. Vestibulum eleifend massa sem, sed faucibus dolor efficitur ut. Sed mattis neque at nibh gravida, sit amet venenatis est scelerisque. Aenean vitae mauris a velit dapibus molestie sit amet eu arcu. Donec ac justo vitae est aliquet bibendum quis aliquet metus. Vivamus vehicula odio a blandit malesuada. Vestibulum non lacrima odio. Morbi sit amet rhoncus nibh. Vivamus vel commodo orci. Etiam posuere mauris et sollicitudin porttitor. Maecenas sed purus tortor.

Aforo máximo
200 personas

[COMPRAR ENTRADA](#)

Ilustración 5. Detalles carrozas disponibles.

Fuente: elaboración propia.

Inicio de sesión

Para iniciar sesión, es necesario pulsar en el icono del menú que se encuentra en la esquina superior derecha. Cualquiera que se haya registrado previamente, podrá iniciar sesión en la aplicación.

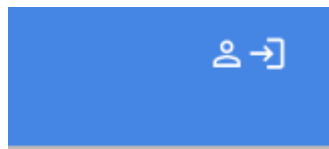


Ilustración 6. Botón inicio sesión.

Fuente: elaboración propia.

En el inicio de sesión, se solicitará el nombre de usuario y la contraseña, utilizando un texto de ayuda dentro de cada campo para facilitar el entendimiento al usuario.

The image shows a login interface on a blue background. At the top center is a white icon of a hand with three fingers pointing up. Below it, the text 'Bienvenido de nuevo' is written in white, followed by 'Inicia sesión para continuar' in a smaller font. The login form is a white rectangle with a thin border. It contains two input fields: the first is labeled 'Usuario' with a person icon and contains the text 'mariaRodriguez'; the second is labeled 'Contraseña' with an eye icon and contains masked characters '*****'. Below these fields is a blue button with the text 'INICIAR SESIÓN' in white. At the bottom of the form, there is a link that says '¿No tienes cuenta? REGÍSTRATE'.

Ilustración 7. Inicio sesión.

Fuente: elaboración propia.

Registro

Un usuario que no tenga cuenta, podrá registrarse dentro de la aplicación ingresando sus datos personales como nombre completo, teléfono o correo electrónico. Además, podrá registrarse como un usuario común o como gestor de carrozas.

Ilustración 8. Registro.

Fuente: elaboración propia.

Menú sesión iniciada

Una vez el usuario haya iniciado sesión, tendrá acceso a distintas páginas según su rol dentro de la aplicación.

Un usuario común tendrá acceso a una página para ver entradas, la página de configuración, y el cierre de sesión.

Un usuario que gestione carrozas tendrá acceso a una página para gestionar sus carrozas, a la página de ver entradas, a la página de configuración y al cierre de sesión.

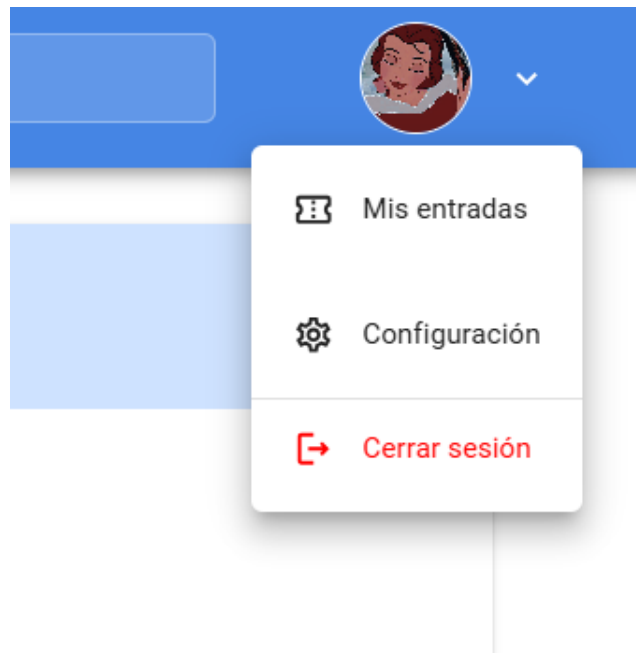


Ilustración 9. Menú usuario común.

Fuente: elaboración propia.

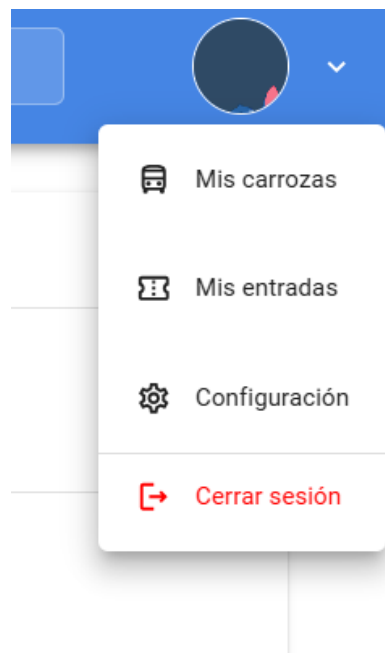


Ilustración 10. Menú usuario cochero.

Fuente: elaboración propia.

Configuración

Dentro de la página de configuración se podrá cambiar cualquier dato personal introducido.

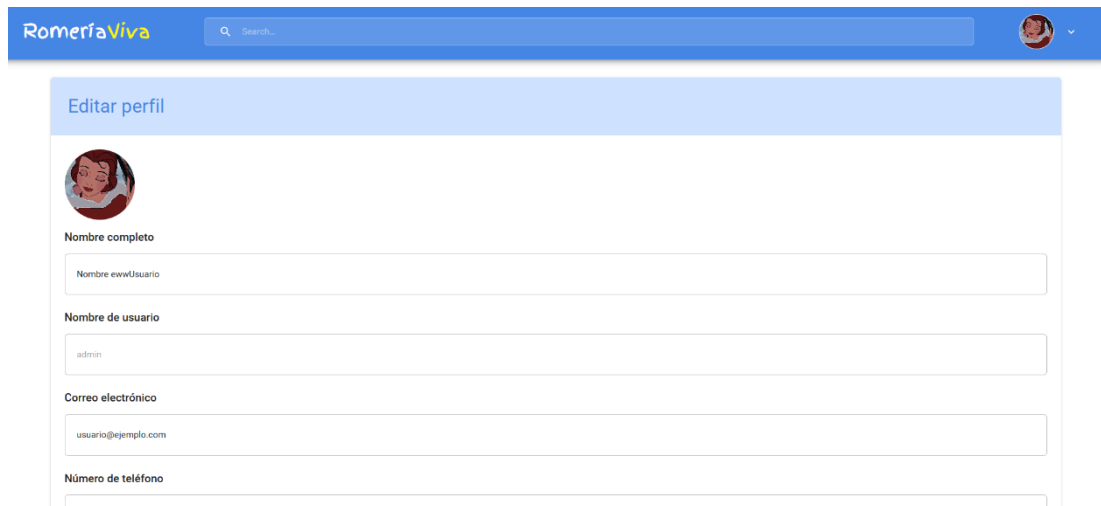


Ilustración 11. Configuración perfil.

Fuente: elaboración propia.

Una vez se guarde el cambio de dato, aparecerá una notificación de perfil correctamente actualizado.

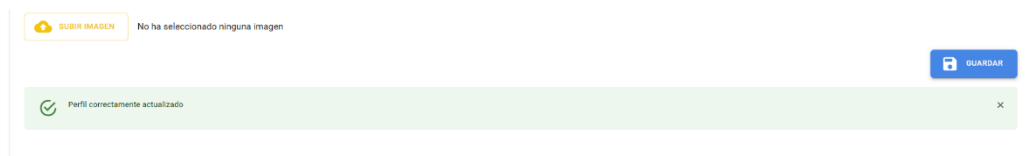


Ilustración 12. Notificación actualización.

Fuente: elaboración propia.

Añadir comentario

Cualquier usuario con sesión iniciada podrá añadir un comentario a una romería, escogiendo una valoración numérica mediante un sistema de calificación de estrellas y escribiendo un comentario.

El acceso a añadir comentario se hace a través de la sección de comentarios de detalles de una romería.

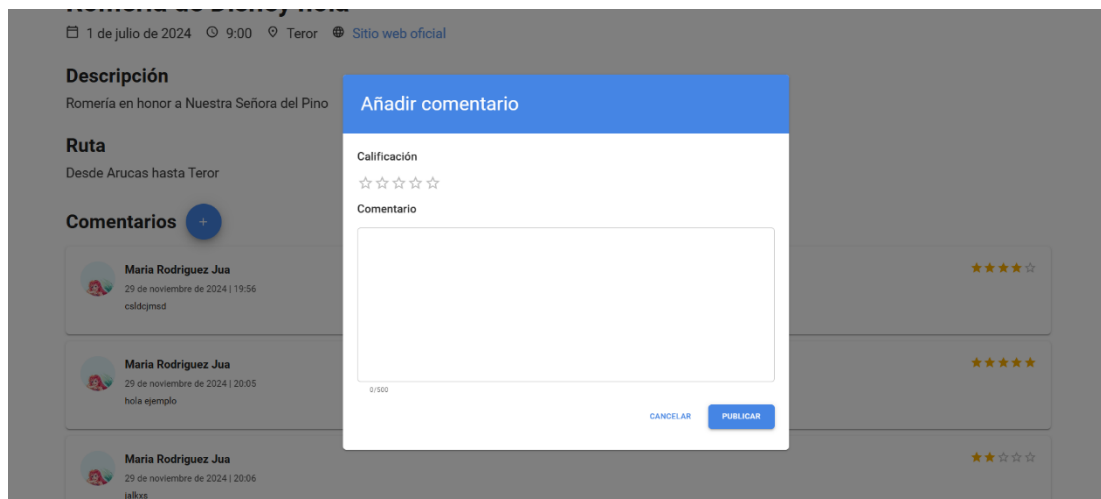


Ilustración 13. Añadir comentario.

Fuente: elaboración propia.

Comprar entradas

Cualquier usuario con sesión iniciada podrá comprar entradas de una carroza para una romería concreta.

El acceso a comprar entradas se hace a través de la sección de carrozas de detalles de una romería, en un botón específico dentro de cada carroza para comprar la entrada.

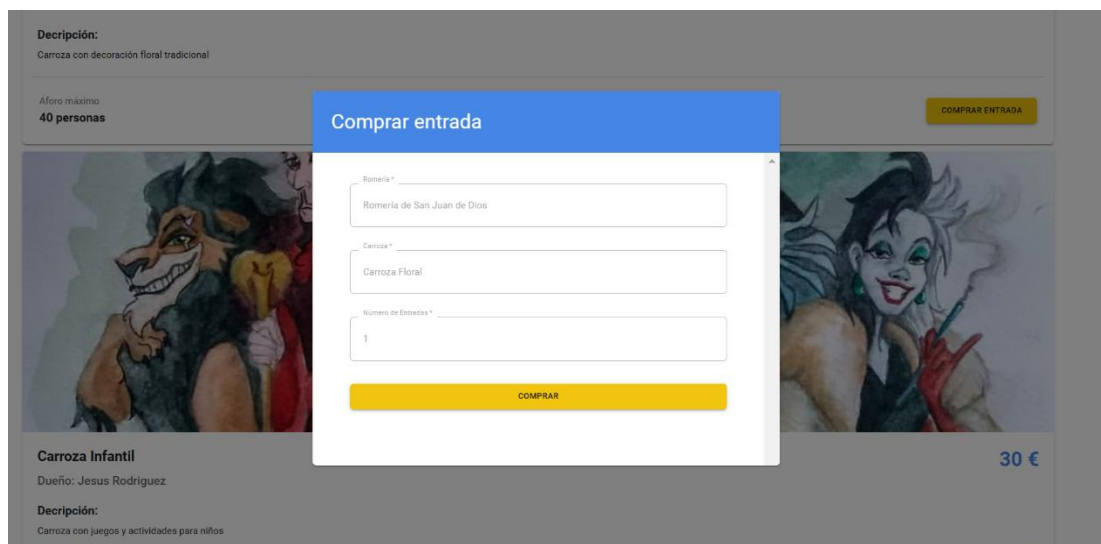


Ilustración 14. Comprar entrada.

Fuente: elaboración propia.

Mis entradas

Cualquier usuario con sesión iniciada podrá ver sus entradas compradas, con detalles de la romería, la carroza, y la fecha de compra.

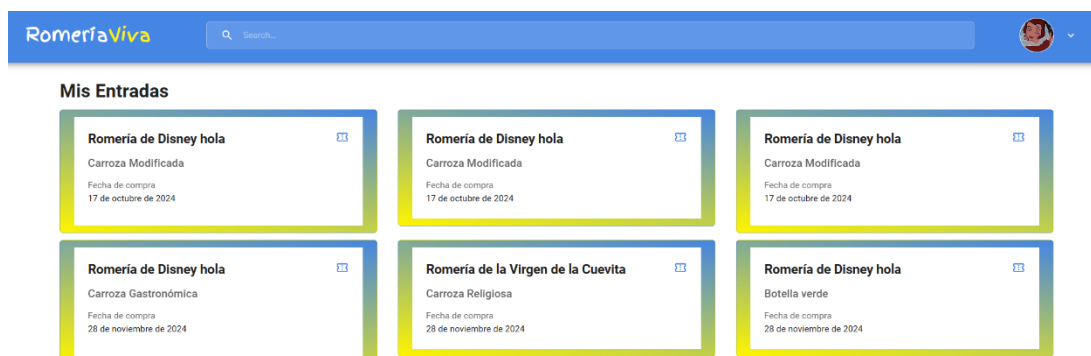


Ilustración 15. Página mis entradas.

Fuente: elaboración propia.

Para acceder al código QR de la entrada, tan sólo se tendrá que pulsar en una de las entradas. Una vez abierto el modal, se mostrará código QR, el titular de la entrada, la romería, la carroza y la fecha de compra.



Ilustración 16. Detalles entrada.

Fuente: elaboración propia.

Mis carrozas

Cualquier usuario con un rol de gestor de carrozas, podrá acceder a sus carrozas.

Se mostrará una lista con todas sus carrozas, mostrando detalles como imagen, nombre, precio, aforo y acciones.

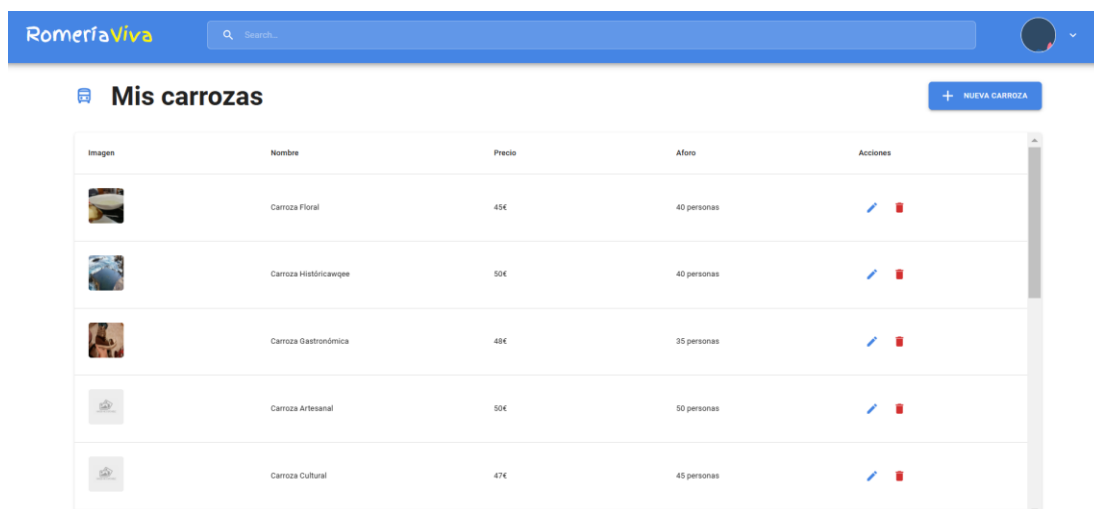


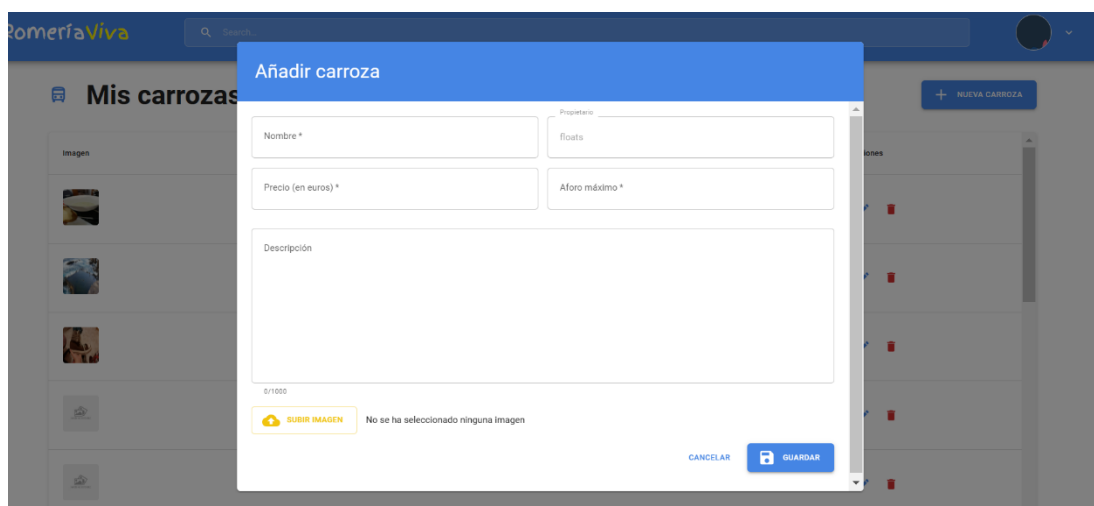
Imagen	Nombre	Precio	Aforo	Acciones
	Carroza Floral	45€	40 personas	 
	Carroza Histórica	50€	40 personas	 
	Carroza Gastronómica	48€	35 personas	 
	Carroza Artesanal	50€	50 personas	 
	Carroza Cultural	47€	45 personas	 

Ilustración 17. Página mis carrozas.

Fuente: elaboración propia.

Dentro de esta página, el usuario podrá añadir y editar carrozas, pulsando el botón correspondiente.

Al pulsarlo, se abrirá un modal con los datos necesarios para añadir o modificar una carroza.



Añadir carroza

 SUBIR IMAGEN

No se ha seleccionado ninguna imagen

CANCELAR

GUARDAR

Ilustración 18. Añadir carroza.

Fuente: elaboración propia.

Añadir carrozas a romerías

Cualquier usuario con un rol de gestor de carrozas, podrá añadir carrozas a una romería.

Para ello, tendrá que acceder a los detalles de una romería, a la sección de carrozas disponibles, donde se mostrará un botón para realizar esta acción.

Al pulsarlo, se abrirá un modal para seleccionar una de las carrozas, añadiéndolas así a la romería en cuestión.

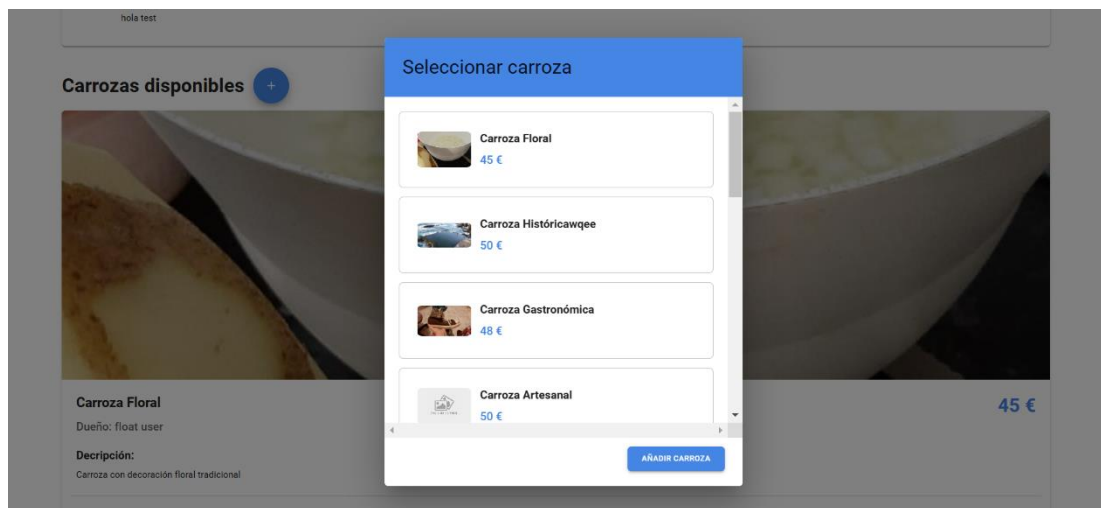


Ilustración 19. Seleccionar carroza.

Fuente: elaboración propia.

Añadir/Editar romerías

Para añadir y editar romerías, se deberá tener un usuario de tipo administrador, que se insertará manualmente en la base de datos ya que no cualquiera debe poder crear romerías.

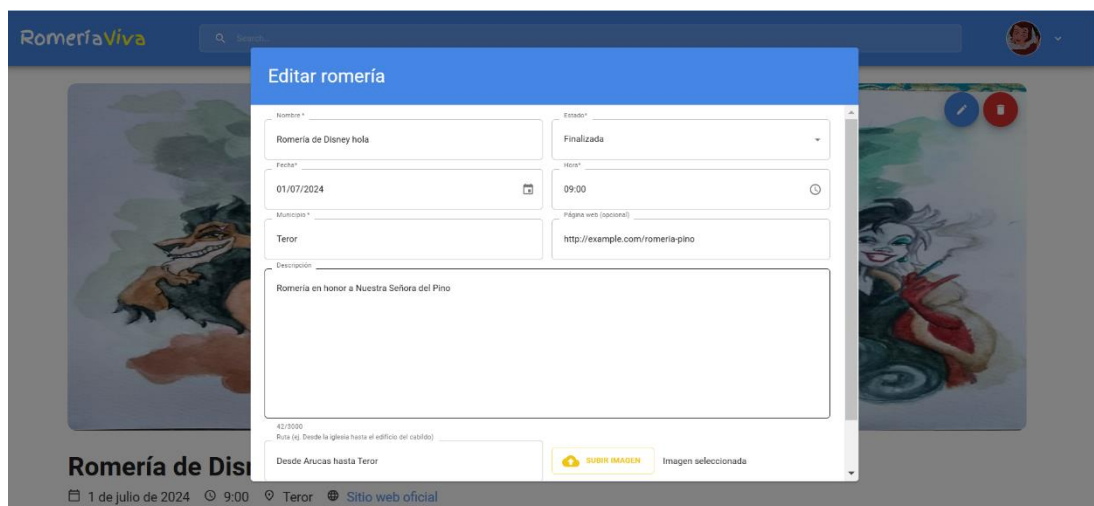
Un usuario de tipo administrador, tendrá la posibilidad de añadir una nueva romería mediante un botón en la pantalla de inicio, y podrá editar romerías mediante un botón dentro del detalle de una romería.

Pulsado cualquiera de los dos botones, se abrirá un modal con los datos necesarios para la creación/edición de una romería.

RomeríaViva				
Search				
+ NUEVA ROMERÍA				
Imagen	Nombre	Estado	Fecha	Lugar
	Romería de Disney hola	Finalizada	1 de julio de 2024	Teror

Ilustración 20. Botón nueva romería.

Fuente: elaboración propia.



Editar romería

Nombre *
Romería de Disney hola

Estado *
Finalizada

Fecha *
01/07/2024

Hora *
09:00

Municipio *
Teror

Página web (opcional)
http://example.com/romeria-pino

Descripción
Romería en honor a Nuestra Señora del Pino

Etiquetas
Etiqueta (puede ser más de una)

Desde Arucas hasta Teror

SUBIR IMAGEN Imagen seleccionada

Romería de Disney hola

1 de julio de 2024 9:00 Teror Sitio web oficial

Ilustración 21. Editar romería.

Fuente: elaboración propia.