
Desarrollo de una Plataforma Integral para la Gestión de Eventos

GRADO EN INGENIERÍA INFORMÁTICA

Autora: Mariana Surós Álvarez

TUTORIZADO POR:
Nelson Monzón López

9 de diciembre de 2024

SOLICITUD DE DEFENSA DE TRABAJO DE FIN DE TÍTULO

D^a Mariana Surós Álvarez, autora del trabajo **Desarrollo de una Plataforma Integral para la Gestión de Eventos**, correspondiente a la titulación **Grado en Ingeniería Informática**, plan 41.

SOLICITA

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 tutores

En relación con la presente solicitud, se informa: (firmar donde corresponda)

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

Fdo. _____

Negativamente (justificación en TFT05)

Fdo. _____

DIRECTORA DE LA ESCUELA DE INGENIERÍA INFORMÁTICA

Agradecimientos

A aquellos que han sido fundamentales en la realización de este proyecto.

A quienes me han brindado su apoyo incondicional y sabio consejo,
guiándome con paciencia a lo largo de este camino.

A quienes, con su aliento y creencia en mí, han hecho posible que persiga
mis sueños y alcance mis metas.

A mi familia, por su constante apoyo y por enseñarme los valores de
perseverancia y determinación.

A aquellos que, desde el fondo de mi corazón, me han inspirado con su
ejemplo y me han alentado a superar cualquier obstáculo.

A la SQUAD, por su colaboración y camaradería en cada paso de este
proyecto. Sus aportaciones han sido invaluable.

A todos los amigos y compañeros que han compartido conmigo este viaje,
han tenido paciencia y han brindado ese ruido de fondo que,
paradójicamente, ha contribuido a mi concentración y enfoque en los
momentos más decisivos.

Sus contribuciones y presencia han sido fundamentales para alcanzar el éxito
en este trabajo. A todos ellos, mi mas sincero agradecimiento.

Resumen

Este proyecto de gestión de eventos proporciona una plataforma versátil y confiable para crear eventos a medida de cada usuario. En un momento donde la demanda de eventos es alta, la flexibilidad y escalabilidad garantizadas por la modularidad son esenciales. Además, la robustez del sistema asegura la coherencia y seguridad de los datos en todas las etapas, desde la creación hasta la gestión de eventos. Con un énfasis en la eficiencia y la optimización de recursos, se establece un sólido fundamento para la organización de eventos exitosos y sin contratiempos.

Abstract

This event management project offers a versatile and dependable platform, allowing users to tailor events to their specific needs. In today's high-demand event landscape, the flexibility and scalability provided by its modular design are indispensable. Additionally, the system's robustness guarantees data consistency and security throughout the entire event lifecycle. By prioritizing efficiency and resource optimization, it establishes a sturdy foundation for orchestrating successful and seamless events.

Índice general

1. Introducción	10
1.1. Descripción del proyecto	11
1.2. Objetivos del proyecto	11
1.3. Competencias específicas	12
1.4. Alineamiento con los objetivos de desarrollo sostenible	13
1.5. Legislación vigente	14
2. Estado del arte	16
2.1. Análisis de mercado	16
2.1.1. Tamaño y tendencias del mercado	16
2.1.2. Mercado Objetivo	18
2.1.3. Competidores	18
2.1.4. Ventaja Competitiva	20
2.2. Presupuesto	20
2.2.1. Coste personal	21
2.2.2. Presupuesto final	21
2.3. Modelo de negocio	21
2.4. Conclusiones	23
3. Tecnologías utilizadas	24
3.1. Vue	24
3.2. Vite	25
3.3. Typescript	26
3.4. Nginx	26
3.5. Spring boot	26
3.6. Project Lombok	27
3.7. DataTables	27
3.8. Bootstrap	27
3.9. Postgre SQL	28
3.10. MongoDB	28
3.11. Docker	28
4. Herramientas utilizadas	30
4.1. Visual Studio Code	30

4.2.	DBeaver	30
4.3.	Postman	31
4.4.	GitHub	31
4.5.	ClickUp	31
4.6.	Figma	32
4.7.	Sonar Qube	32
4.8.	Light House	32
4.9.	Dbdiagram	33
4.10.	Lucidchart	33
5.	Metodología y planificación	34
5.1.	Scrum	35
5.1.1.	Ventajas	35
5.1.2.	Roles	35
5.1.3.	Artefactos	36
5.1.4.	Eventos	36
5.2.	Kanban	37
5.2.1.	Definiciones	37
5.3.	Planificación	37
6.	Desarrollo	39
6.1.	Sprint Zero	39
6.1.1.	Evaluación de tecnologías	40
6.1.2.	Definición de requisitos	42
6.2.	Desarrollo de la Pila de Producto	44
6.3.	Etapas de desarrollo	45
6.3.1.	Sprint 1	45
6.3.2.	Sprint 2	48
6.3.3.	Sprint 3	51
6.3.4.	Sprint 4	59
6.3.5.	Sprint 5	61
6.3.6.	Sprint 6	66
6.3.7.	Sprint 7	73
6.3.8.	Sprint 8	75
7.	Conclusiones y trabajo futuro	80

Índice de figuras

2.1.	Tamaño del mercado de software de gestión de eventos ¹	17
2.2.	Formulario de creación de eventos y gestor de Event Planner Sp ²	19
2.3.	Portal de eventos de Zoho Backstage ³	19
2.4.	Listado de eventos de The Knot ⁴	20
3.1.	Fuente: commons.wikimedia.org/wiki/File:Vue.png	25
3.2.	Fuente: commons.wikimedia.org/wiki/File:Vitejs-logo.svg	25
3.3.	Fuente: www.typescriptlang.org/branding	26
3.4.	Fuente: nginx.org/en/	26
3.5.	Fuente: commons.wikimedia.org/wiki/File:Spring_Boot.svg	27
3.6.	Fuente: github.com/projectlombok	27
3.7.	Fuente: commons.wikimedia.org/wiki/File:Datatables_logo_square.png	27
3.8.	Fuente: https://getbootstrap.com/docs/5.0/about/brand/	28
3.9.	Fuente: wiki.postgresql.org/wiki/Logo	28
3.10.	Fuente: www.mongodb.com/company/newsroom/brand-resources	28
3.11.	Fuente: www.docker.com/company/newsroom/media-resources/	29
4.1.	Fuente: code.visualstudio.com/brand	30
4.2.	Fuente: dbeaver.io/	31
4.3.	Fuente: code.visualstudio.com/brand	31
4.4.	Fuente: github.com/logos	31
4.5.	Fuente: clickup.com/brand	32
4.6.	Fuente: https://www.figma.com/community/file/930374612850356203	32
4.7.	Fuente: www.pngwing.com/es/free-png-azlkx	32
4.8.	Fuente: worldvectorlogo.com/logo/google-lighthouse-icon-may-2019-	32
4.9.	Fuente: dbdiagram.io/	33
4.10.	Fuente: cdn-cashy-static-assets.lucidchart.com/lucid/logos/lucidchart.svg	33
5.1.	Planificación inicial del proyecto	38
5.2.	Planificación final del proyecto	38
6.1.	Ejemplo de historia técnica en la plataforma ClickUp.	45
6.2.	Diagrama de base de datos relacional: usuarios.	46
6.3.	Diagrama de base de datos relacional: eventos.	47
6.4.	Base de datos no relacional: logs.	47

6.5. Diagrama de flujo de la aplicación	48
6.6. Paleta de colores de la aplicación.	49
6.7. Logo de la aplicación	49
6.8. Inicio de sesión (Figma)	49
6.9. Registro de usuario (Figma)	49
6.10. Pantalla de inicio (Figma)	49
6.11. Pantalla de gestión de eventos (Figma)	49
6.12. Creación de un evento. Pasos 1-3.	50
6.13. Creación de un evento. Pasos 4-7.	51
6.14. Configuración de los servicios de bases de datos en el <i>docker-compose.yml</i> de la aplicación.	53
6.15. Configuración del servicio de Spring boot en el <i>docker-compose.yml</i> de la aplicación.	54
6.16. Configuración del servicio de Vue en el <i>docker-compose.yml</i> de la aplicación.	54
6.17. Configuración de variables de entorno en Spring Boot.	55
6.18. Ejemplo de mapeador: AnswerFAQMapper.	56
6.19. Ejemplo de servicio: AnswerFAQService.	57
6.20. Ejemplo de servicio: AnswerFAQService.	57
6.21. Implementación del método centralizado para realizar llamadas HTTP utilizando Axios.	58
6.22. Definiciones de <i>DTO</i> en el archivo <i>FaqDto.ts</i>	59
6.23. Definición del enumerado de <i>EventType</i>	59
6.24. Extracto del servicio <i>JwtService</i>	60
6.25. Función de inicio de sesión del servicio <i>AuthenticationService</i>	61
6.26. Extracto del enumerado de roles de Spring Boot.	62
6.27. Extracto del controlador de usuario <i>UserControler</i>	62
6.28. Extracto de router en Vue.	63
6.29. Popup de inicio de sesión	64
6.30. Popup de registro de usuario	64
6.31. Componente de navegación	64
6.32. Página principal del proyecto	64
6.33. Componente de información	65
6.34. Componente de contacto	65
6.35. Selector de cambio de idioma	65
6.36. Módulo de gestión de idiomas <i>i18n.ts</i>	66
6.37. <i>Stepper</i> para la creación de eventos	67
6.38. Selección de tipo de evento durante la creación de un evento.	68
6.39. Formulario de información durante la creación de un evento.	68
6.40. Selección de localización durante la creación de un evento: dirección.	69
6.41. Selección de localización durante la creación de un evento: localización.	69
6.42. Selección de localización durante la creación de un evento: localización.	69
6.43. Creación de una localización nueva.	70
6.44. Formulario financiero en la creación de evento.	70
6.45. Formulario financiero. Listado de códigos de descuento.	70
6.46. Formulario de preguntas frecuentes en la creación de evento.	71

6.47. Formulario de preguntas frecuentes. Listado de preguntas.	71
6.48. Formulario de menú durante la creación de un evento.	72
6.49. Formulario de menú. Listado de preguntas.	72
6.50. Formulario final en la creación de un evento.	72
6.51. Rutas de evento dinámicas en el módulo de enrutamiento.	73
6.52. Ejemplo de creación de evento.	73
6.53. Componente para la edición de un evento.	74
6.54. Gestión de usuarios en un evento.	75
6.55. Mapa con la ubicación de un evento.	75
6.56. Componente de selección de color.	76
6.57. Plantillas para la personalización de la apariencia de un evento.	76
6.58. Filtros de evento.	77
6.59. Filtros de evento con resultados.	77
6.60. Gráfico de asistencia de un evento.	77
6.61. Informe inicial de SonarQube.	78
6.62. Informe final de SonarQube.	78
6.63. Resultados de la aplicación tras los análisis con Lighthouse.	79

Índice de tablas

1.1. Grado de relación del TFG con los Objetivos de Desarrollo Sostenible.	13
2.1. Coste hardware del proyecto.	21
2.2. Coste personal del proyecto.	21
2.3. Presupuesto total del proyecto.	21
6.1. Planificación Sprint 1.	39
6.2. Planificación Sprint 1.	46
6.3. Planificación Sprint 2.	48
6.4. Planificación Sprint 3.	52
6.5. Planificación Sprint 4.	59
6.6. Planificación Sprint 5.	61
6.7. Planificación Sprint 6.	67
6.8. Planificación Sprint 7.	74
6.9. Planificación Sprint 8.	76
7.1. Historia técnica HT1	82
7.2. Historia técnica HT2	83
7.3. Historia técnica HT3	83
7.4. Historia técnica HT4	83
7.5. Historia técnica HT5	84
7.6. Historia técnica HT6	84
7.7. Historia técnica HT7	85
7.8. Historia técnica HT8	85
7.9. Historia técnica HT9	86
7.10. Historia técnica HT10	86
7.11. Historia técnica HT11	87
7.12. Historia de usuario HU1	87
7.13. Historia de usuario HU2	88
7.14. Historia de usuario HU3	88
7.15. Historia de usuario HU4	89
7.16. Historia de usuario HU5	89
7.17. Historia de usuario HU6	90
7.18. Historia de usuario HU7	90
7.19. Historia de usuario HU8	91

7.20. Historia de usuario HU9	91
7.21. Historia de usuario HU10	92
7.22. Historia de usuario HU11	92
7.23. Historia de usuario HU12	93
7.24. Historia de usuario HU13	93
7.25. Historia de usuario HU14	94
7.26. Historia de usuario HU15	95
7.27. Historia de usuario HU16	95
7.28. Historia de usuario HU17	96
7.29. Historia de usuario HU18	96
7.30. Historia de usuario HU19	96
7.31. Historia de usuario HU20	97
7.32. Historia de usuario HU21	97
7.33. Historia de usuario HU22	98

Capítulo 1

Introducción

“Un evento exitoso no ocurre por accidente. Está planeado hasta el último detalle.”

Anónimo

Actualmente, el enorme auge del sector tecnológico y la cantidad de herramientas TIC¹ existentes simplifican el desempeño de actividades de carácter laboral, profesional o de ocio.

Sin embargo, en una sociedad que tiene casi todo al alcance de la mano, surgen necesidades acordes al nuevo estatus socio-tecnológico que se está generando continuamente. Eventos automovilísticos, ferias gastronómicas, eventos culturales, celebraciones familiares o charlas en línea, por mencionar algunos, emplean plataformas en las que se publica un mismo tipo de actividad.

Así, organizar un evento de forma exitosa puede ser una tarea compleja y abrumadora debido a la cantidad de detalles que se deben coordinar. Ya no vale solo experiencia e intuición. El valor de los datos y las herramientas de gestión van de la mano.

En este sentido, la recopilación estratégica, el análisis y la aplicación de datos a través de algunas soluciones informáticas tendencia en el mercado, carecen de flexibilidad y por ende, de adaptación a necesidades específicas.

De ahí que, el desarrollo de una plataforma que centralice la gestión de eventos sin necesidad de consultar individualmente cada uno de ellos, se convierte en una chispa impulsora para el desarrollo del presente proyecto. Es decir, una herramienta de planificación adaptada, personalizada, dinámica y eficiente, facilitando que cada evento bien organizado resulte único.

¹Tecnologías de la Información y las Comunicaciones

1.1. Descripción del proyecto

El proyecto *Event Planner* es un prototipo funcional diseñado para proporcionar una solución versátil que atienda a una amplia variedad de necesidades de planificación de eventos, facilitando la toma de decisiones.

La experiencia de navegación del usuario/usuario potencial comienza con el acceso al contenido abierto al público, que ofrece de forma dinámica, una visión general del servicio y del proceso de toma de decisiones que implica. Así mismo, el motor de búsqueda permite filtrar eventos por año, nombre o tipo de evento además de permitir visualizar aquellos eventos marcados de acceso libre por parte de quien los crea.

A continuación, si el interés despertado se concreta en una necesidad de servicio, el/la usuario/a tendría que registrarse como sucede en cualquier página web, lo que daría acceso a contenidos específicos a delimitarse en función de dicha necesidad identificada, por ejemplo, eventos corporativos, conferencias, festival local, despedida de soltero/a, aniversario de bodas, cumpleaños, etc.

En otras palabras, la plataforma permite a las personas registradas seleccionar el tipo de evento que desean crear, adaptando automáticamente los requerimientos de la información y funcionalidades u opciones a crear, según la naturaleza específica del evento.

Finalmente, dicha personalización asegura el uso de la herramienta en múltiples contextos. Por tanto, podrá gestionar múltiples eventos desde un mismo panel - aquellos creados y en los que se participa - así como gestionar y editar la información de cada evento creado, analizar estadísticas detalladas de su evento, participantes y visualizaciones, entre otros. Además, un mismo evento pueda albergar usuarios/as participantes e introducir, por ejemplo, datos personales para confirmar asistencia o seleccionar un menú.

1.2. Objetivos del proyecto

El proyecto persigue los siguientes objetivos:

- Creación de una plataforma web completa.
- Diseño y desarrollo de módulos personalizables.
- Facilitar la experiencia del usuario.
- Garantizar la seguridad de los datos.
- Proveer una herramienta adaptable.

1.3. Competencias específicas

A continuación, las competencias específicas asociadas a este Trabajo de Fin de Grado[27]:

1. **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.*
2. **CII02.** *Capacidad para planificar, concebir, desplegar y dirigir proyectos, servicios y sistemas informáticos en todos los ámbitos, liderando su puesta en marcha y su mejora continua y valorando su impacto económico y social.*
3. **CII16.** *Conocimiento y aplicación de los principios, metodologías y ciclos de vida de la ingeniería de software.*
4. **CII08.** *Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.*
5. **CII12.** *Conocimiento y aplicación de las características, funcionalidades y estructura de las bases de datos, que permitan su adecuado uso, y el diseño y el análisis e implementación de aplicaciones basadas en ellos.*
6. **CII17.** *Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas.*
7. **TI03.** *Capacidad para emplear metodologías centradas en el usuario y la organización para el desarrollo, evaluación y gestión de aplicaciones y sistemas basados en tecnologías de la información que aseguren la accesibilidad, la ergonomía y usabilidad de los sistemas.*

Competencias como **CII01** y **CII17** fueron fundamentales durante la fase inicial de análisis y diseño, donde se enfocó en asegurar la fiabilidad, seguridad y accesibilidad del sistema mediante la selección de tecnologías adecuadas y la implementación de principios de usabilidad. De manera transversal, competencias como **CII02** y **CII16** se aplicaron a lo largo de todo el proyecto, adoptando la metodología Scrum para la gestión ágil del ciclo de vida del software y garantizando una mejora continua y adaptativa del sistema. Esta metodología permitió no solo la planificación efectiva de actividades, sino también la aplicación de principios de ingeniería del software para mantener altos estándares de calidad en cada etapa del desarrollo. Además, el proyecto ha sido una oportunidad para fortalecer la competencia **CII08**, enfocada en el análisis detallado de requisitos, el diseño robusto y la implementación eficiente de aplicaciones, asegurando así un desarrollo robusto, seguro y eficiente del sistema. Estas competencias se complementaron con el manejo de bases de datos (**CII12**), garantizando un uso óptimo y seguro de la información a través de consultas y procedimientos bien definidos. Finalmente, metodologías centradas en el usuario (**TI03**) aseguraron que la plataforma desarrollada cumpliera con las expectativas y necesidades de los usuarios finales, integrando principios de ergonomía y usabilidad para una experiencia de usuario mejorada.

1.4. Alineamiento con los objetivos de desarrollo sostenible

El proyecto *Event Planner* promueve la sostenibilidad y contribuye a los Objetivos de Desarrollo Sostenible. La siguiente tabla muestra su relación con cada objetivo:

	No procede	Bajo	Medio	Alto
Fin de la Pobreza	X			
Hambre cero	X			
Salud y Bienestar	X			
Educación de calidad	X			
Igualdad de género	X			
Agua limpia y saneamiento	X			
Energía asequible y no contaminante	X			
Trabajo decente y crecimiento económico				X
Industria, Innovación e Infraestructuras				X
Reducción de las desigualdades	X			
Ciudades y comunidades sostenibles	X			
Producción y consumo sostenibles			X	
Acción por el clima	X			
Vida submarina	X			
Vida de ecosistemas terrestres	X			
Paz, justicia e instituciones sólidas	X			
Alianzas para lograr objetivos				X

Tabla 1.1: Grado de relación del TFG con los Objetivos de Desarrollo Sostenible.

El impacto del proyecto se detalla a continuación:

- **Trabajo decente y crecimiento económico:** El proyecto pretende impulsar la creación de empleo y la economía local mediante la organización de eventos como ferias, conferencias, festivales, etc. Además, su diseño facilita la interacción entre diversos sectores, promoviendo la revitalización de la economía.
- **Industria, innovación e infraestructuras:** El uso de tecnologías de desarrollo modernas hace de este proyecto una solución innovadora para la gestión de eventos. Esto no sólo promueve el avance tecnológico sino que también fomenta la colaboración entre diferentes departamentos para optimizar recursos y procesos.
- **Producción y consumo sostenibles:** A través de la personalización y planificación eficiente de los eventos, el proyecto pretende ayuda a optimizar recursos y minimizar desperdicios, fomentando así el uso responsable de los recursos y promoviendo prácticas de gestión de eventos sostenibles.

- **Alianzas para lograr objetivos:** Al permitir la interacción y colaboración entre usuarios, organizadores, y proveedores, promueve sinergias que favorecen el logro de metas comunes relacionadas con los eventos.

1.5. Legislación vigente

Se ha considerado la legislación vigente en materia de seguridad de datos de los/as potenciales usuarios/as, confidencialidad, consentimiento y tratamiento de datos, concretamente:

- Constitución española, Artículo 18.4, en el que se recoge que “La ley limitará el uso de la informática para garantizar el honor y la intimidad personal y familiar de los ciudadanos y el pleno ejercicio de sus derechos”².
- Reglamento (UE) 2016/679 del Parlamento Europeo y del Consejo, de 27 de abril de 2016, relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y a la libre circulación de estos datos y por el que se deroga la Directiva 95/46/CE (Reglamento general de protección de datos)(RGPD)³.
- Ley Orgánica 3/2018, de 5 de diciembre, de Protección de Datos Personales y garantía de los derechos digitales, por la que se adapta al ordenamiento jurídico español al Reglamento (UE) 2016/679 anteriormente mencionado, así como garantizar los derechos digitales de la ciudadanía ⁴.

Los datos personales que los usuarios aporten a través de la página web (nombre y apellidos, teléfono de contacto, dirección de correo electrónico, lugar de celebración del evento), atienden a los siguientes fines de tratamiento:

- Navegar por la página web permitiendo el acceso a la información y a los contenidos de la misma, por lo que resulta obligatorio registrarse, así como la aceptación de sus términos y condiciones.
- Contactar con las personas usuarias en cuanto a la información aportada, a través del apartado de contacto disponible para tal efecto en la página web.
- Adoptar medidas de protección y medidas técnicas de seguridad sobre los datos personales aplicables, de acuerdo a la normativa vigente anteriormente señalada (por ejemplo, cifrado de datos).

Si bien las personas usuarias de la página web tienen derecho a solicitar la rectificación o supresión de los datos, la limitación al tratamiento, u oponerse al mismo, previo conocimiento de la presente política de protección de datos, consienten los fines de tratamiento hasta aquí informados, tanto si acepta dicha política como si continúa navegando en la página web. Los datos obtenidos a través de la página web no serán cedidos a terceros, salvo en el caso de garantizar algún servicio señalado en dicha página, informándose previamente la

²Constitución Española[Const] 29/12/1978. www.boe.es/boe/dias/1978/12/29/pdfs/A29313-29424.pdf

³Reglamento (UE) 2016/679. www.boe.es/buscar/doc.php?id=DOUE-L-2016-80807

⁴BOE-A-2018-16673. www.boe.es/buscar/pdf/2018/BOE-A-2018-16673-consolidado.pdf

naturaleza de los datos concretos y finalidad. Finalmente, se procederá a la anonimización de datos personales cuando no resulten necesarios para los fines de tratamiento hasta aquí especificadas, de conformidad con la normativa aplicable.

Capítulo 2

Estado del arte

En este apartado se detallarán elementos del mercado objeto del presente proyecto, del presupuesto planificado, una descripción del modelo de negocio que mejor se ajusta a las características del proyecto y las conclusiones.

2.1. Análisis de mercado

La tecnología y la innovación han impactado dramáticamente en el éxito de cualquier evento en el que hayan tomado parte. No se trata de emplear la última tendencia del mercado, sino de tomar la mejor decisión sobre la herramienta adecuada para alcanzar el efecto deseado en el evento a planificar.

De ahí que un estudio del mercado resulte significativo. Actualmente existe una amplia variedad de opciones para la organización de eventos, que incluye sitios web, aplicaciones móviles, software, herramientas digitales y más. Sin embargo, muchas de estas soluciones están enfocadas en un único sector o bien en un conjunto limitado de sectores.

2.1.1. Tamaño y tendencias del mercado

Respecto al mercado de plataformas de gestión de eventos, ha sido valorado en 30.628,23 millones de dólares en 2022, con previsión de llegar a 59.068 millones de dólares en 2028, considerándose una tasa anual de 11,6 % entre 2022 y 2028¹.

Desde otro punto de vista, según los componentes, el mercado de plataformas de gestión de eventos es dominado por el software respecto al hardware, representando el 76,1 % en 2021, previendo su contribución del 75 % a la cuota general del mercado en 2028. A su vez, dicho mercado está dominado por el sector de agencias de gestión de eventos frente a empresas

¹www.theinsightpartners.com/es/reports/event-management-platform-market

u otras organizaciones, representado casi un 33 % en 2021 alcanzando previsiblemente un 36 % en 2028.

Por otro lado, el incremento de eventos de entretenimiento, deportivos y sociales está potenciando el mercado de software para la planificación, programación, organización, supervisión, seguimiento, control y marketing. Se prevé que de los 13,01 mil millones de dólares estimados para 2024 se llegue a los 21,18 mil millones en 2029, representando una tasa de incremento anual aproximada del 10,24 %, tal y como muestra la figura 2.1.

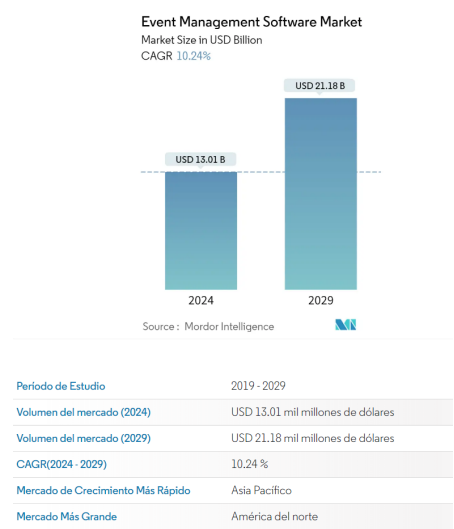


Figura 2.1: Tamaño del mercado de software de gestión de eventos².

A nivel local, a modo de ejemplo, la Consejería de Turismo y Empleo del Gobierno de Canarias ha publicado la primera convocatoria para patrocinar eventos de interés turístico organizado por entidades privadas, con un presupuesto de 6,7 millones de euros, ampliables. Se prevé una convocatoria similar destinada a entidades públicas³.

También, la Jornada de Moda Sostenible y Campaña “Slow Fashion Canaria” a desarrollarse en Octubre de 2024 con el apoyo de los Cabildos Insulares⁴.

En cuanto a las tendencias del mercado de software de gestión de eventos, se espera que el sector de medios y entretenimiento registre un significativo crecimiento, sobre la base de eventos, festivales populares y eventos culturales⁵. En este sentido, análisis de audiencias en eventos en Londres del año en curso, revela la mayoría de usuarios, como millennials⁶ con poder adquisitivo y muy activos en redes sociales[4].

Dado que dichos eventos como cualquier evento público, resultan a su vez complejos de organizar a diferencia de otros sectores, las herramientas de gestión informática constituyen elementos clave, creando perspectivas favorables en el crecimiento del mercado.

²www.mordorintelligence.com/es/industry-reports/event-management-software-market
³www3.gobiernodecanarias.org/noticias/tag/eventos-turisticos/
⁴marketing.proexca.es/jornada-moda-sostenible-canarias
⁵www.eventindustrynews.com/news/four-trends-evolving-the-events-tech-industry
⁶Nacidos entre 1981 y 1996, adaptados a la tecnología, con valores progresistas.

Por tanto, sectores como turismo, gastronomía, ámbito social, cultural, saturados de planificadores y configuradores de eventos centrados en sectores parcelados, constituyen como beneficiarios de la aplicación del presente proyecto. De esta manera, ampliando horizontes no contemplados por otros aplicativos, es capaz de soportar varios formatos (presencial, virtual, híbrido), procurando flexibilidad en las demandas de cada evento, como elemento esencial en el mundo dinámico actual. Además, mejora la experiencia tanto para participantes, personal de marketing, usuarios/as, como para el que diseñe dicha herramienta, contribuyendo a modificar la forma en que un evento es planificado en el ámbito local, en su ejecución y análisis.

2.1.2. Mercado Objetivo

Este proyecto está dirigido a clientes y usuarios, bien PYMES⁷ o particulares, planificadores de eventos, personal de marketing u otros usuarios/as finales, quienes pueden crear, gestionar y administrar cualquier tipo de evento o varios. Desde pequeños encuentros sociales, eventos comunitarios, procesos de formación interna y externa para personal profesional, exhibiciones sobre nuevos proyectos, avances tecnológicos, hasta festividades, eventos culturales, conferencias empresariales, todo en un solo lugar.

En cuanto al enfoque de usuarios, destacan los siguientes:

- Usuario creador: aquel que crea un evento.
- Usuario colaborador: tanto empresas como individuos - permite modificar datos específicos relativos al evento, como por ejemplo, horario de comienzo.
- Usuario participante: persona que acude al evento, proporciona datos para participar y puede valorar la actividad o compartir opiniones.

2.1.3. Competidores

Entre las aplicaciones disponibles, se encuentran aquellas destinadas a eventos de empresa, herramientas digitales para organizar eventos y software específico.

Respecto a las herramientas utilizadas para analizar competidores, destaca Similarweb⁸, una plataforma que genera informes detallados sobre el uso y el alcance de diferentes sitios web. Obtiene sus datos de diversas fuentes, incluida Google Analytics, para proporcionar análisis precisos y útiles.

A continuación, se muestra una selección de competidores, de acuerdo al perfil del presente proyecto.

⁷Empresas compuestas por un número reducido de trabajadores.

⁸www.similarweb.com/

2.1.3.1. Event Planner Sp

Event Planner Sp[8], es un sitio web creado para facilitar la organización de eventos. Fundado en 2002, permite a los usuarios descubrir y reservar lugares para eventos con sus correspondientes proveedores. También ofrece software gratuito de planificación de eventos. El portal cuenta con casi cien mil usuarios y recibe generalmente treinta mil visitas mensuales.

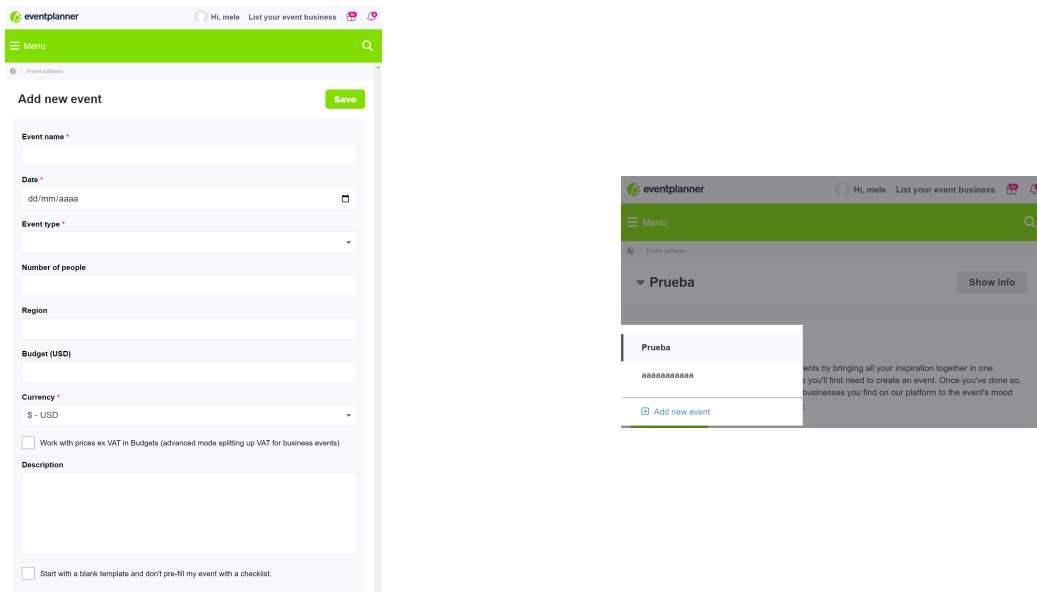


Figura 2.2: Formulario de creación de eventos y gestor de Event Planner Sp⁹.

2.1.3.2. Zoho Backstage

Zoho Backstage[10] es una plataforma de gestión de eventos creada por Zoho Corporation. Fundada en 2018, está dirigida a planificadores de eventos, corporaciones y agencias para facilitar la planificación, programación y ejecución de eventos desde una misma plataforma. Así mismo, puede llevar seguimiento de asistentes y visualizar analíticas de rendimiento, o recibir colaboraciones de las empresas en cuanto a planificación, diseño de contenido y comunicaciones. La plataforma contó el pasado mes de mayo con 150 mil visitas mensuales.

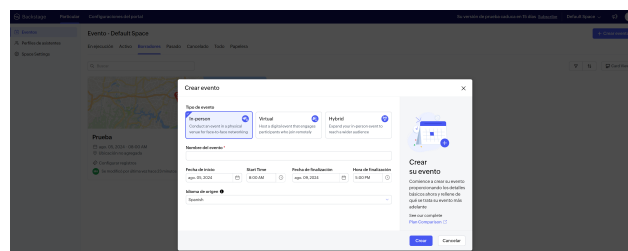


Figura 2.3: Portal de eventos de Zoho Backstage¹⁰.

⁹Fuente: www.eventplanner.net/

2.1.3.3. The Knot Worldwide

The Knot Worldwide[11] es una empresa fundada para facilitar la planificación de celebraciones a nivel global, empleando sitios web líderes en el sector. Fundamentalmente orientada a bodas, ofrece productos, servicios y contenido en el top ranking de aplicaciones móviles, con recursos que ofrecen información, conexiones y servicios más buscados. Además, ofrece herramientas de planificación personalizadas, sitios web de bodas e invitaciones que simplifican la organización de bodas en todo el mundo. Cuenta con más de 4 millones de parejas, cerca de 850 mil proveedores anualmente y más de treinta millones de visitas mensuales, en concreto el último mes de mayo contó con alrededor de 37 millones.

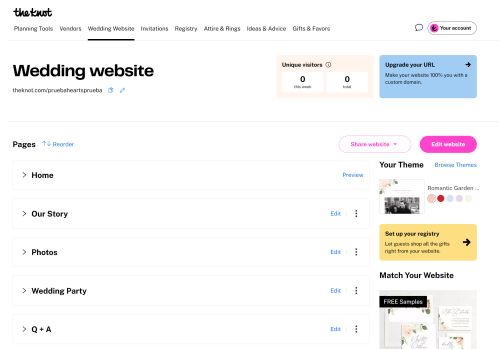


Figura 2.4: Listado de eventos de The Knot¹¹.

2.1.4. Ventaja Competitiva

Tras realizar un análisis de la competencia, se puede determinar que este proyecto ofrece el conjunto de funcionalidades que el resto pero en una única plataforma, con una interfaz más intuitiva, amigable, accesible y flexible. Además, se permite una personalización total de los eventos, incluyendo la posibilidad de añadir secciones específicas como preguntas frecuentes relacionadas con el evento, menús y sistemas de acceso con credenciales, algo que no ofrecen las plataformas existentes.

2.2. Presupuesto

En este apartado se ha contemplado un presupuesto estimado de compra (inversión inicial en hardware y personal) y un presupuesto estimado de operación que incluye el de previsión de venta o modelo de negocio, a saber[24]:

¹⁰Fuente: backstage.zoho.eu

¹¹Fuente: www.theknot.com/

Hardware	Coste
Torre AMD Ryzen 5 5600X 6-Core 32Gb	1800€
Pantalla 27" LG Ultra	390€
Pantalla 24" Acer	240€
Pantalla 22" LG	130€
Pantalla 20" Compaq	110€
Total	2670€

Tabla 2.1: Coste hardware del proyecto.

2.2.1. Coste personal

El costo personal se basa en la tarifa por hora de un desarrollador autónomo, considerando diferentes fuentes de información sobre tarifas del mercado[14, 13, 7].

Personal	Nº Horas	Coste/Hora	Coste
Alumno	300	25€	7.500€
Tutor	50	100€	5000€
Total			12.500€

Tabla 2.2: Coste personal del proyecto.

2.2.2. Presupuesto final

Concepto	Coste
Coste del hardware	2.670€
Coste del personal	12.500€
Presupuesto total	15.170€

Tabla 2.3: Presupuesto total del proyecto.

2.3. Modelo de negocio

Una vez definido el segmento de clientes potenciales a los que está dirigido la página web detallada en este proyecto, las necesidades a cubrir y el servicio que ofrece, procede detallar la captación de clientes y financiación del proyecto.

La inversión inicial se centra en los costes de hardware y desarrollo de la página web a los cuales se ha tenido acceso a través de financiación propia. Sin embargo, la principal ventaja en este apartado de financiación tiene que ver con el negocio online, es decir, el acceso a un

mayor número de clientes. En otras palabras, si bien los clientes acceden de forma automática a través de internet, contribuyen a la financiación de diversas maneras.

Se contactará con empresas locales preferentemente de trayectoria en el sector a través de la estrategia de patrocinio. Así, mediante el marketing de afiliación, se recomendará los productos o servicios de empresas asociadas a los usuarios o visitantes de la página web del proyecto, de forma que, al pinchar cada vez sobre el enlace asociado (por ejemplo, fotografía profesional, *atelier*, consultoras de belleza, hoteles, joyerías, hipermercados, catering y eventos, etc..) se logra una conversión o compra que se traduce en una comisión. Así mismo, al pinchar sobre los enlaces o *banners*¹² o exhibición de logotipos previamente acordados con las empresas asociadas (proveedores), u otras que pudiesen convertirse también en patrocinadores.

Además de atraer potenciales clientes, otra porción de ingresos es generado por los podcasts creados por clientes o usuarios de la página web, con los que se pretende también continuar captando patrocinadores.

A su vez, el lanzamiento de al menos dos concursos semestrales o bien uno anual, de acuerdo a las bases establecidas, cuya temática es seleccionada por los clientes y usuarios, de entre un listado de opciones propuestas (por ejemplo, mejor vestido de novia de la temporada, la situación más incómoda reseñada en el evento que han participado, etc..), obteniendo promoción y beneficios a través del crowdsourcing.

Por otro lado, una vez lanzada la página web, se ofrecerá un acceso gratuito a una versión con funcionalidades limitadas (modelo *freemium*¹³) como el envío de invitaciones personalizadas para el evento seleccionado y conformado por el usuario, como forma de atraer usuarios y dar a conocer la página; posteriormente se accederá a funcionalidades específicas previo pago de suscripción.^{14 15 16 17}

Además se implementará el seguimiento de ingresos como el estudio de otras alternativas de marketing digital y motores de búsqueda (*SEO*¹⁸) para aumentar presencia en línea y venta de entradas de eventos asociados. Respecto al mantenimiento, se emplearán inicialmente herramientas gratuitas para su automatización.

En síntesis, el modelo de negocio propuesto detalla una fuente de ingresos que garantiza la viabilidad del proyecto desde sus inicios, a través de financiación propia, patrocinios empresariales, suscripción, así como aportando ajustes en las estrategias de monetización de acuerdo a los resultados obtenidos y la evolución del proyecto. Además, contribuye al entramado empresarial del sector, tanto local como foráneo, ofreciendo oportunidades de consolidación.

¹²Mensaje publicitario situado en una página web.

¹³Estrategia de negocio en el que se ofrece un producto o servicio gratuito.

¹⁴www.holded.com/es/blog/tipos-modelo-negocio/

¹⁵www.up-spain.com/blog/modelos-negocio-online-mejor-funcionan/

¹⁶www.iebschool.com/blog/que-es-el-modelo-canvas-y-como-aplicarlo-a-tu-negocio-agile-scrum/

¹⁷www.shopify.com/es/blog/17389100-como-elegir-un-modelo-de-negocio-de-comercio-electronico

¹⁸developer.mozilla.org/es/docs/Glossary/SEO

2.4. Conclusiones

Una vez establecido el contexto del mercado y del estado del proyecto, se puede afirmar que este es el momento ideal para su desarrollo, expansión y consolidación, respaldado por las tendencias de mercado, las tendencias tecnológicas actuales y el creciente interés en soluciones personalizables para la creación y gestión de eventos. De esta manera, la experiencia de usuario no resulta tediosa, sino amigable, fresca y adaptada a las características del usuario, seleccionando entre los módulos disponibles. Además, en la fase de prototipo constituye una aplicación gratuita.

En conclusión, este trabajo supone una mejora competitiva tanto en el coste como en el producto que se ofrece, al mismo tiempo, en la transacción que clientes y empresas asociadas pueden realizar al acceder y utilizar el proyecto. Destaca el valor que genera en la solución final y las facilidades y posibilidades que brinda a los usuarios del sistema.

Capítulo 3

Tecnologías utilizadas

En este apartado se abordarán las tecnologías que se han considerado las más idóneas para el desarrollo del prototipo.

3.1. Vue

Vue[28] es un *framework*¹ de *Javascript*² diseñado para la construcción de interfaces de usuario, basado en tecnologías como *HTML*³, *CSS*⁴, entre otras. Además, Vue proporciona un modelo de programación declarativo y basado en componentes que facilita el desarrollo eficiente de interfaces de usuario de cualquier complejidad.

Ventajas del *framework*

- Modularidad: tiene una arquitectura modular basada en componentes que facilita la reutilización de código y la separación de responsabilidades.
- Escalabilidad: permite agregar nuevas funcionalidades sin afectar otras partes de la aplicación.
- Reactividad y rendimiento: detecta automáticamente los cambios de estado de *Javascript*, permitiendo actualizar rápidamente el *DOM*⁵ sin necesidad de optimizaciones manuales.
- Entorno de desarrollo: dispone un entorno de desarrollo robusto y accesible con herramientas como Vue Devtools, Vue CLI y soporte para *TypeScript*⁶.

¹Estructura de desarrollo que incluye herramientas y bibliotecas predefinidas para facilitar el desarrollo de aplicaciones específicas.

²developer.mozilla.org/en-US/docs/Web/JavaScript

³developer.mozilla.org/en-US/docs/Web/HTML

⁴developer.mozilla.org/en-US/docs/Web/CSS

⁵developer.mozilla.org/es/docs/Web/API/Document_Object_Model

⁶developer.mozilla.org/en-US/docs/Glossary/TypeScript

- Versatilidad: Vue es flexible, puede ser utilizado para diversos fines. Por ejemplo:
 - Mejorando *HTML* estático sin una etapa de construcción.
 - Integrando componentes web en cualquier página.
 - Utilizando aplicaciones de una sola página (*SPA*)⁷ con Vue Router y Vuex.



Figura 3.1: Fuente: commons.wikimedia.org/wiki/File:Vue.png

3.2. Vite

Vite⁸ es una herramienta de construcción que mejora significativamente la experiencia de desarrollo front-end⁹. Es conocida por su velocidad y eficiencia en el manejo de grandes proyectos de *Javascript*. Vite utiliza módulos *ES*¹⁰ nativos y aprovecha las capacidades de los navegadores modernos para ofrecer un entorno de desarrollo instantáneo y rápido.

Características destacadas

- Inicio instantáneo del servidor: servidor de archivos bajo demanda sobre *ESM*¹¹ nativo, sin necesidad de empaquetado.
- Reemplazo inmediato de módulos (*HMR*¹²): mantiene la rapidez independientemente del tamaño de la aplicación.
- Integración: soporte nativo para *Typescript*, *JSX*¹³, *CSS* y más.



Figura 3.2: Fuente: commons.wikimedia.org/wiki/File:Vitejs-logo.svg

⁷<https://developer.mozilla.org/en-US/docs/Glossary/SPA>

⁸vitejs.dev/guide/why.html

⁹Práctica de producir *HTML*, *CSS* y *JavaScript*

¹⁰developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Modules

¹¹developer.mozilla.org/es/docs/Web/JavaScript/Guide/Modules

¹²webpack.js.org/concepts/hot-module-replacement/

¹³legacy.reactjs.org/docs/introducing-jsx.html

3.3. Typescript

Typescript[17] se trata de un lenguaje de programación construido sobre *Javascript* que proporciona herramientas mejoradas para el desarrollo de software a cualquier escala.



Figura 3.3: Fuente: www.typescriptlang.org/branding

3.4. Nginx

Nginx¹⁴ es un servidor *HTTP*¹⁵ y *proxy*¹⁶ inverso de alto rendimiento conocido por su robustez y eficiencia en el manejo de cargas de trabajo elevadas. Tiene arquitectura modular, es capaz de manejar múltiples tareas simultáneamente, desde servir contenido estático hasta actuar como proxy inverso para mejorar el rendimiento y la seguridad de las aplicaciones. Soporta protocolos modernos como *HTTP/2*¹⁷ y *HTTP/3*¹⁸.



Figura 3.4: Fuente: nginx.org/en/

3.5. Spring boot

Spring Boot[25] es un *framework* que facilita la creación de aplicaciones *Java*¹⁹ basadas en *Spring*, optimizadas para la producción y listas para ejecutarse con una mínima configuración. Se utiliza ampliamente debido a su capacidad para simplificar el desarrollo de microservicios, aplicaciones web y otras soluciones complejas.

Además, Spring Boot proporciona dependencias de inicio preconfiguradas y un servidor web embebido que permite ejecutar aplicaciones con facilidad y eficiencia, eliminando la necesidad de configuraciones extensas. Esta característica lo hace ideal para proyectos que requieren una implementación rápida y confiable, sin comprometer la escalabilidad ni la seguridad.

¹⁴nginx.org/en/

¹⁵www.cloudflare.com/es-es/learning/ddos/glossary/hypertext-transfer-protocol-http/

¹⁶Proporciona una puerta de enlace entre los usuarios e Internet.

¹⁷Segunda versión principal de *HTTP* que sucede a *HTTP/1.1*

¹⁸Tercera versión principal de *HTTP* que sucede a *HTTP/2*.

¹⁹developer.mozilla.org/es/docs/Glossary/Java



Figura 3.5: Fuente: commons.wikimedia.org/wiki/File:Spring_Boot.svg

3.6. Project Lombok

Project Lombok[15] es una biblioteca *Java* que se integra automáticamente con los *frameworks*, mejorando la experiencia de desarrollo eliminando código repetitivo y tedioso. Lombok automatiza la generación de métodos como *getters*²⁰, *setters*²¹, constructores²² y mucho más, reduciendo la verbosidad del código y ahorrando tiempo para el desarrollador.



Figura 3.6: Fuente: github.com/projectlombok

3.7. DataTables

DataTables[9] es una es una librería de *jQuery*²³ de código abierto centrada en la mejora de la interacción y utilidad de las tablas *HTML* basadas en *Javascript*.



Figura 3.7: Fuente: commons.wikimedia.org/wiki/File:Datatables_logo_square.png

3.8. Bootstrap

Bootstrap[5] es un *framework* de *CSS* que optimiza la construcción de páginas web. Integra configuraciones, diseños y estándares de desarrollo orientados casi exclusivamente hacia la normalización de estilos de navegador cruzado²⁴.

²⁰Métodos que permiten acceder al valor de un atributo.

²¹Métodos que fijan el valor de un atributo.

²²Métodos que inicializan un objeto recién creado y le asignan valores iniciales a sus variables de instancia.

²³jquery.com/

²⁴developer.mozilla.org/es/docs/Learn/Tools_and_testing/Cross_browser_testing



Figura 3.8: Fuente: <https://getbootstrap.com/docs/5.0/about/brand/>

3.9. Postgre SQL

PostgreSQL[26] es un sistema de gestión de base de datos relacional de código abierto de tipo *SQL*²⁵ que permite crear y guardar datos para soluciones software de arquitecturas complejas de forma escalable.



Figura 3.9: Fuente: wiki.postgresql.org/wiki/Logo

3.10. MongoDB

MongoDB[18] es una base de datos de documentos que destaca por su escalabilidad y flexibilidad. A diferencia de las bases de datos relacionales tradicionales que utilizan tablas y filas, MongoDB almacena datos en documentos flexibles similares a *JSON*²⁶, lo que permite que los campos dentro de los documentos puedan variar y que la estructura de datos pueda evolucionar con el tiempo según las necesidades de la aplicación.



Figura 3.10: Fuente: www.mongodb.com/company/newsroom/brand-resources

3.11. Docker

Docker²⁷ es una plataforma abierta diseñada para simplificar el desarrollo, el envío y la ejecución de aplicaciones en entornos consistentes y reproducibles, sin importar la infraestructura subyacente. Además, permite empaquetar aplicaciones y sus dependencias en contenedores ligeros y aislados, lo que facilita la portabilidad y consistencia entre diferentes entornos, resultando ideal para la integración continua y despliegue continuo (*CI/CD*²⁸).

²⁵developer.mozilla.org/es/docs/Glossary/SQL

²⁶developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON

²⁷docs.docker.com/guides/docker-overview/

²⁸www.redhat.com/es/topics/devops/what-is-ci-cd

Esto, ya que acelera el ciclo de desarrollo y permite probar y desplegar cambios de manera rápida y eficiente. También optimiza la gestión de recursos al permitir la ejecución de múltiples contenedores en un mismo *host*²⁹, mejorando la escalabilidad y eficiencia del sistema.



Figura 3.11: Fuente: www.docker.com/company/newsroom/media-resources/

²⁹developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Host

Capítulo 4

Herramientas utilizadas

4.1. Visual Studio Code

Visual Studio Code¹ es un editor de código abierto ampliamente utilizado por su flexibilidad, rendimiento y capacidad de personalización. Permite agregar funcionalidades específicas a través de extensiones como:

- ESLint: proporciona verificación de errores sintácticos y estilísticos en tiempo real.
- Error Lens: resalta y muestra detalles sobre errores directamente en el editor.
- Maven: permite compilar, comprobar y empaquetar aplicaciones de Spring Boot.
- Composer: permite ejecutar comandos que facilitan la integración y gestión de contenedores de Docker.



Figura 4.1: Fuente: code.visualstudio.com/brand

4.2. DBeaver

DBeaver² es una herramienta de gestión de bases de datos universal que permite manipular datos de manera similar a una hoja de cálculo tradicional, simplificando así la creación de informes analíticos basados en registros de diferentes almacenes de datos. Es compatible con más de 80 bases de datos, proporcionando una gran flexibilidad y capacidad e integración en

¹code.visualstudio.com/

²dbeaver.io/about/

diversos entornos. Facilita la migración de datos y esquemas, además de contener un editor de *SQL*.



Figura 4.2: Fuente: dbeaver.io/

4.3. Postman

Postman³ es una plataforma utilizada para construir, utilizar y simplificando cada paso del ciclo de vida de una *API*⁴, proporcionando herramientas para diseñar, probar, documentar y simular. Es esencial para asegurar una comunicación eficiente entre diferentes servicios y aplicaciones.



Figura 4.3: Fuente: code.visualstudio.com/brand

4.4. GitHub

GitHub es una plataforma de alojamiento para proyectos que utilizan el sistema de control de versiones de Git⁵. Esta herramienta proporciona una variedad de características que facilitan la gestión y seguimiento del código fuente, así como la colaboración entre equipos de desarrollo.



Figura 4.4: Fuente: github.com/logos

4.5. ClickUp

ClickUp⁶ es una plataforma diseñada para la gestión y colaboración en proyectos. Permite a los equipos organizar tareas, planificar proyectos y coordinar esfuerzos mediante múltiples

³www.postman.com/product/what-is-postman/

⁴developer.mozilla.org/es/docs/Mozilla/Add-ons/WebExtensions/API

⁵www.git-scm.com

⁶clickup.com/blog/what-is-clickup-used-for/

vistas, como listas, tableros y calendarios.



Figura 4.5: Fuente: clickup.com/brand

4.6. Figma

Figma⁷ es una plataforma que permite realizar tareas de diseño y creación de prototipos para aplicaciones.



Figura 4.6: Fuente: <https://www.figma.com/community/file/930374612850356203>

4.7. Sonar Qube

Sonar Qube⁸ es una herramienta de análisis estáticos que asegura la calidad y mantenibilidad del código y fortalece la seguridad de proyectos de cualquier tamaño.



Figura 4.7: Fuente: www.pngwing.com/es/free-png-azlkx

4.8. Light House

Light House⁹ es una herramienta de código abierto que realiza auditorías de aplicaciones web a través de pruebas exhaustivas con el fin de generar un informe detallado sobre el desempeño de la página, indicando los aspectos a mejorar para optimizar la aplicación, tales como la velocidad de carga, la accesibilidad, optimización para motores de búsqueda (*SEO*¹⁰).



Figura 4.8: Fuente: worldvectorlogo.com/logo/google-lighthouse-icon-may-2019-

⁷www.figma.com/ui-design-tool/

⁸www.sonarsource.com/

⁹www.chrome.google.com/webstore/detail/lighthouse/blipmdconlcpinefehnmjammfjpmphjk?hl=es

¹⁰developer.mozilla.org/es/docs/Glossary/SEO

4.9. Dbdiagram

Dbdiagram¹¹ es una herramienta gratuita y sencilla diseñada para desarrolladores y analistas de datos. Requiere dibujar diagramas de bases de datos (*ERD*¹²) escribiendo el código (*DSL*¹³) mediante el uso del lenguaje de marcado *DBML*¹⁴ para definir la estructura del esquema de la base de datos. Se crean tablas, campos y relaciones, importar y exportar una base de datos concreta en pocos segundos para visualizar luego en otros formatos como *PDF*, *PNG*, *SQL*.



Figura 4.9: Fuente: dbdiagram.io/

4.10. Lucidchart

Lucidchart¹⁵ es una herramienta en línea diseñada para la creación de diagramas, permitiendo visualizar y aclarar ideas, sistemas y procesos complejos.



Figura 4.10: Fuente: cdn-cashy-static-assets.lucidchart.com/lucid/logos/lucidchart.svg

¹¹dbdiagram.io/docs/

¹²Diagrama entidad-relación.

¹³Lenguaje ideado para expresar cierta parte de un sistema.

¹⁴dbml.dbdiagram.io/home

¹⁵www.lucidchart.com/

Capítulo 5

Metodología y planificación

“Una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo” (Avison y Fitzgerald, 1995)[3].

En la evolución de las metodologías y modelos empleados en el desarrollo de software impulsada por las características del mercado para adaptarse y aprovechar oportunidades, destacan “Las metodologías ágiles, más que un marco de procedimientos y herramientas se conciben como una filosofía, una práctica orientada a las personas no a los procesos”(Cadavid, et al. 2013 cp Zumba y León 2018)[30].

En la reciente Conferencia de Desarrolladores de Barcelona (DevBCN, Junio 2024), se invita a los desarrolladores a participar haciendo mención a lo que resulta interesante para cada profesional, apasionado/a del back-end¹ y/o front-end, incluyendo entre otros las metodologías ágiles²³.

Así mismo, en la Conferencia de Agilidad en la Empresa Aragonesa (Caea)⁴ en la que se han empleado dichas metodologías ágiles con éxito, incluso en empresas públicas como el Gobierno de Aragón.

Consecuentemente, como marco de trabajo para el desarrollo de este proyecto se han empleado metodologías de desarrollo ágil que pueden complementarse, concretamente Scrum[23] y Kanban[16], en la estructuración, planificación y control del proceso de desarrollo y gestión del desarrollo del proyecto.

Durante la realización del proyecto, el rol de Product Owner fue desempeñado por un consultor con un amplio conocimiento del sector de la organización de eventos, cuya experiencia

¹Práctica de gestionar la lógica del servidor, bases de datos y su arquitectura.

²agile-spain.org/cas/

³www.devbcn.com/

⁴caearagon.org/#conferencia

y conocimiento del sector aportaron una perspectiva específica y valiosa sobre las necesidades del usuario final. Este profesional, interesado en el prototipo, proporcionó una visión clara y detallada de los requisitos y prioridades del usuario que se transformaron en requisitos para el proyecto, facilitando que cada iteración del desarrollo se alinee con las expectativas del mercado y las demandas reales de los usuarios.

Además, el tutor Dr. Nelson Monzón asumió el rol de Scrum Master, facilitando la metodología ágil y asegurando que los principios de Scrum se aplicaran correctamente. Su liderazgo y orientación permitieron que el equipo de desarrollo, compuesto por la estudiante, pudiera centrarse en un desarrollo eficaz y eficiente de los objetivos del proyecto.

5.1. Scrum

Se define como “un marco de trabajo por el cual las personas pueden abordar problemas complejos adaptativos, a la vez que entregar productos del máximo valor posible productiva y creativamente” (Schwaber & Sutherland, 2017 cp Quesada Reyes, 2020 p.40)[22].

“Como metodología, Scrum prescribe procesos formalizados para el desarrollo y gestión de proyectos informáticos basado en los valores y principios del Manifiesto Ágil, en el empirismo y en los principios Lean”(Reddy, 2015; Permana, 2015; Lei et al., 2017; Schwaber & Sutherland, 2020). Fuentes del Burgo y Sebastián, 2022 p.1662 [12].

Como marco de gestión de equipos, permite mayor flexibilidad y adaptación a las necesidades cambiantes del proyecto, dado que organiza progresivamente y de forma interactiva el desarrollo del software en ciclos cortos (sprints), basándose en dicha adaptación continua, e incluyendo un producto a entregar al final de cada interacción. Las etapas o fases pueden solaparse y repetirse cuantas veces sea necesario; el equipo recibe retroalimentación continua permitiendo ajustes rápidos y eficientes[20].

5.1.1. Ventajas

Algunas ventajas pertinentes para este proyecto tienen que ver con que permite realizar cambios y ajustes en el desarrollo del proyecto según las necesidades del cliente, así como una mejor organización y seguimiento del proyecto en comparación a un desarrollo en cascada o secuencial. Además, permite una revisión constante del progreso y la implementación de mejoras continuas a través de los eventos de Scrum, como las reuniones y las retrospectivas. También fomenta la cooperación entre los distintos roles del equipo, asegurando que cada miembro contribuya al objetivo común.

5.1.2. Roles

- **Equipo de desarrollo:** encargado de crear el incremento de cada Sprint. Sus funciones incluyen la planificación y la pila de Sprint, cumplir con la definición de hecho en cada

tarea y alcanzar los objetivos de cada Sprint.

- **Propietario de producto:** asegura que el trabajo del equipo cumpla con las necesidades del usuario y gestiona la pila de producto. Sus tareas incluyen transmitir el objetivo del producto al equipo, crear y priorizar los elementos de la pila de producto.
- **Scrum Master:** responsable de asegurar que el marco de trabajo se utilice según lo establecido en la Guía de Scrum. Ayuda a los miembros del equipo a crear incrementos de alto valor, elimina impedimentos que afecten al progreso del equipo y asegura que todos los eventos de Scrum se realicen correctamente.

5.1.3. Artefactos

- **Pila de producto:** lista de requisitos del usuario gestionada por el propietario del producto. Incluye historias de usuario con descripciones, subtareas, estimaciones y criterios de validación.
- **Pila de Sprint:** lista de historias de usuario a desarrollar en el próximo Sprint, acordada entre el propietario del producto y el equipo de desarrollo.
- **Incremento:** avance hacia el objetivo identificado propuesto para el producto final. Cada incremento se añade a los anteriores, asegurando compatibilidad y valor conjunto.

5.1.4. Eventos

Los eventos en Scrum se repiten en cada Sprint, permitiendo inspeccionar, adaptar los artefactos y mantener la transparencia del proyecto.

- **Sprint:** período de una a cuatro semanas para lograr un incremento del producto.
- **Sprint Zero:** primer Sprint utilizado para definir principios arquitectónicos, bocetos y preparaciones para futuros Sprints; no se busca funcionalidad, sino preparar el entorno de trabajo.
- **Planificación del Sprint:** reunión para decidir el trabajo a desarrollar en el siguiente Sprint, acorde con los objetivos del propietario del producto y el equipo de desarrollo.
- **Scrum diario:** reunión breve de hasta quince minutos para comentar progresos, resolver dificultades y ajustar la pila de Sprint.
- **Revisión de Sprint:** análisis del resultado del Sprint, evaluando objetivos cumplidos y no cumplidos.
- **Retrospectiva de Sprint:** reunión para sugerir y planear mejoras para futuros Sprints, enfocándose en efectividad y calidad.

5.2. Kanban

Kanban[2] es un método de gestión del trabajo ampliamente utilizado como herramienta de apoyo al marco Scrum. Aunque desarrollado por un Ingeniero de la empresa Toyota en la década de 1950 para mejorar la eficiencia de producción, ha sido utilizado en diferentes industrias como aeronáutica, salud, venta de ropa en tiendas y recursos humanos, entre otras. Fue introducido en el ámbito del desarrollo del software por David J. Anderson (Anderson, 2010).

Su principal característica consiste en utilizar un tablón, ya sea físico o digital, que regula la cadencia de tareas que el equipo tiene que completar, separado por estados y fases en la que se encuentran los trabajos. Por tanto, con una simple revisión rápida del tablón de tareas, dividido en columnas que representen el estado del trabajo, se puedan asignar nuevas tareas de manera muy efectiva (desarrollo y entrega de forma continuada) y entender la situación del proyecto en un momento dado.

Esto ayuda a la mejora de los procesos, es decir, que todos los miembros de un equipo puedan visualizar el progreso del trabajo, conocer el estado de las tareas y organizar el flujo de trabajo, abordando cantidades pequeñas de tareas simultáneamente y con fluidez, maximizando la eficiencia, adaptando continuamente el flujo de trabajo del proyecto [6, 19, 1].

5.2.1. Definiciones

- **WIP:** *Work in progress* es la cantidad de trabajo en la que un equipo puede enfocarse a la vez.
- **Columna:** parte del tablero que se destina a poner las unidades de trabajo, normalmente se agrupan en columnas que representan distintos estados.
- **Tarjeta:** es la representación visual de una tarea.
- **Estado:** identifica la fase del proceso en el que se encuentran las tarjetas: por hacer, en progreso y finalizado.

5.3. Planificación

Al inicio del proyecto, se diseñó una planificación centrada en aspectos corporativos, como la integración con proveedores, colaboraciones entre organizaciones y sistemas de comentarios y reseñas para eventos, ver figura 5.1. Este enfoque inicial reflejaba una visión orientada hacia las necesidades operativas y comerciales del contexto empresarial en el que se desenvolvería el gestor de eventos.

Fases	Duración Estimada (horas)	Tareas (nombre y descripción, obligatorio al menos una por fase)
Creación del <i>Product Backlog</i> inicial	10	Primer acercamiento al entorno del gestor de eventos Maquetación de las primeras historias de usuario y priorización de necesidades
Sprint Zero	70	Estudio de herramientas y familiarización con el problema Investigación del estado del arte en gestores de eventos digitales
Desarrollo de la pila de producto	10	Maquetación de la pila de producto esperada para el gestor de eventos
Release 1 (Sprints 1 y 2)	30	Configuración inicial del entorno de desarrollo Desarrollo del esquema básico de la interfaz de usuario
Release 2 (Sprints 3, 4 y 5)	80	Desarrollo de módulos personalizables para distintos tipos de eventos Implementación del sistema de gestión de presupuestos
Release 3 (Sprints 6, 7 y 8)	60	Facilitar la colaboración entre organizaciones mediante asignación de roles y mensajería integrada Integración de proveedores externos para servicios relacionados con eventos Incorporar la opción de reseñas y sistema de calificación para proveedores
Documentación / Presentación	40	Redacción de la memoria y la presentación conforme a los reglamentos de TFT de la EII

Figura 5.1: Planificación inicial del proyecto

Sin embargo, a medida que avanzaba la fase inicial del proyecto, se identificó la necesidad de ajustar esta planificación para alinearse estrechamente con las expectativas y requisitos de los usuarios finales, es decir, los clientes y participantes de eventos, resultando la siguiente planificación final que suman 300 horas de trabajo en total (figura 5.2).

Fases	Duración Estimada (horas)	Tareas (nombre y descripción, obligatorio al menos una por fase)
Sprint Zero	70	Estudio de herramientas y familiarización con el problema Investigación del estado del arte en gestores de eventos digitales Priorización de necesidades
Desarrollo de la pila de producto	10	Maquetación de la pila de producto esperada para el gestor de eventos
Release 1 (Sprints 1 y 2)	30	Diseño de la base de datos Desarrollo de diagramas de la aplicación Desarrollo del esquema básico de la interfaz de usuario
Release 2 (Sprints 3, 4 y 5)	70	Configuración inicial del entorno de desarrollo Implementación de seguridad en el entorno de desarrollo
Release 3 (Sprints 6, 7 y 8)	80	Desarrollo de módulos personalizables para distintos tipos de eventos Desarrollo de los módulos para usuarios dentro de un evento Testeo de la aplicación
Documentación / Presentación	40	Redacción de la memoria y la presentación conforme a los reglamentos de TFT de la EII

Figura 5.2: Planificación final del proyecto

Capítulo 6

Desarrollo

Dado que cada desarrollo debe configurarse de acuerdo con el producto a obtener, los objetivos, requisitos del mercado y los avances tecnológicos, a continuación, se expondrán la descripción de la secuencia y etapas del diseño que pretende un proceso rentable, visible, eficaz y eficiente[27].

6.1. Sprint Zero

Durante la fase previa al desarrollo se sentaron las bases del proyecto. Concretamente la recopilación y exposición en apartados anteriores, de datos relevantes respecto al mercado destinatario, herramientas a utilizar y aplicaciones similares, posible presupuesto y modelo de negocio. El análisis de todo ello ha permitido identificar cuáles requisitos debería cumplir la página web para luego diseñar una solución o producto con diferenciación en el mercado, pero ante todo, viable desde el punto de vista práctico, técnico y financiero.

A continuación se señalan las historias técnicas diseñadas para llevarse a cabo durante este Sprint:

Id	Nombre	Estimación	Prioridad
HT1	Evaluación de tecnologías	30 horas	Alta
HT2	Análisis competitivo de aplicaciones similares	20 horas	Media
HT3	Definición de requisitos	10 horas	Alta
HT4	Análisis arquitectónico de la aplicación	10 horas	Alta
Estimación total		70 horas	

Tabla 6.1: Planificación Sprint 1.

6.1.1. Evaluación de tecnologías

En primera instancia se debatió sobre el formato de la plataforma, decidiéndose desarrollar una aplicación web frente a las alternativas móviles o de escritorio, teniendo en cuenta las diversas ventajas en cuanto a la elaboración inicial y las futuras[29].

Si bien el resto de opciones también ofrecían potencialidades, se decantó por la idea de entregar un producto con un desarrollo y recursos más centralizados, con características de mayor alcance y proyección comercial. Algunas de estas cualidades serían:

- Accesibilidad y alcance: acceder desde cualquier dispositivo con navegador y conexión a internet permite alcanzar un mayor rango de audiencia sin necesidad de desarrollar aplicaciones específicas para cada sistema operativo.
- Costos y tiempo de desarrollo: su naturaleza multiplataforma permite plazos de desarrollo más cortos al tratarse de una única aplicación, además de los reducidos costes de creación y mantenimiento que implica.
- Compatibilidad y adaptabilidad: las nuevas tecnologías web, junto al diseño *responsive*¹, facilitan que el aplicativo se pueda adaptar a diferentes formatos de pantalla o resolución, obteniendo como resultado una experiencia más consistente para el usuario.

Una vez definidos los lineamientos sobre los que construir la aplicación, se procedió con la selección de las tecnologías más adecuadas para el proyecto, previo análisis de las que mejor se adaptarían al objetivo final. No solo se trata de obtener un producto dotado de robustez, sino también aquellas cualidades que facilitarán un desarrollo más eficiente y modular, dado el potencial para evolucionar desde un prototipo a una solución a mayor escala.

Dado que se pretende ejecutar la aplicación de la manera más sencilla posible, se aprovecha la capacidad de Docker para garantizar la portabilidad y consistencia en entornos de desarrollo y producción. Es decir, permite empaquetar aplicaciones y sus respectivas dependencias en contenedores aislados y ligeros, facilitando un comportamiento uniforme y constante en cualquier entorno, esencial para la plataforma, ya que el despliegue se vuelve más rápido y eficiente.

En cuanto al *front-end*, se emplearon las tecnologías Vue, Vite, Typescript, Nginx, Bootstrap y DataTables, dadas las capacidades y ventajas que presentaban, ya detalladas en el apartado correspondiente.

Por otro lado, se optó por separar la lógica de *front* y la lógica de *back*, considerando los siguientes beneficios ²:

- La posibilidad de dividir el código en dos partes y optimizarlo de forma más rápida además de incrementar recursos a diferentes ritmos.
- El servidor trabaja menos dado que envía datos sin procesar no requiriendo actualizar la página ante cualquier solicitud, generando una mejor experiencia de usuario.

¹developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Responsive_Design

²www.forbes.com/sites/forbestechcouncil/2018/07/19/seven-reasons-why-a-websites-front-end-and-back-end-should-be-kept-separate/

- Fácil actualización ya que se reducen las posibilidades de romper la página web entera tratando de actualizar elementos.
- Facilidad para cambiar a *frameworks* actualizados, dado que la tecnología avanza con rapidez
- Rápida implementación al ser independientes.
- Códigos más fáciles de manejar en la consolidación de las aplicaciones.
- Modularidad que facilita el trabajo simultáneo sin pérdida de trabajo.

Específicamente, Vue es fundamental para el proyecto gracias a su capacidad de aportar modularidad mediante la integración de componentes reutilizables dentro de las vistas. Esto permite evitar la duplicación de código, además de mantener una arquitectura limpia y organizada. Vue también facilita la creación de interfaces de usuario dinámicas y reactivas, sincronizando automáticamente el *DOM* con el estado de la aplicación. A la hora de crear un proyecto de Vue con npm³, se configura con el entorno de desarrollo Vite, que reduce significativamente los tiempos de espera gracias a su servidor de desarrollo rápido y optimización automática de dependencias.

Por otro lado, al usar *ESBuild* para la compilación y el *bundling*⁴, acelera el proceso de compilación de los archivos del proyecto.

TypeScript fue seleccionado por sus numerosas ventajas que mejoran la calidad y mantenibilidad del código, principalmente su capacidad para detectar errores comunes mediante análisis estático en tiempo de compilación. Esto reduce significativamente la probabilidad de errores en tiempo de ejecución en producción y permite realizar refactorizaciones de código con mayor confianza en aplicaciones a gran escala.

Para el despliegue y gestión del *front-end* del proyecto se optó por integrar Nginx pues proporciona funcionalidades avanzadas como *proxy* inverso y balanceo de carga, preparando el proyecto para escalar fácilmente en el futuro.

Bootstrap fue elegido para asegurar un diseño receptivo y una interfaz de usuario consistente, pues facilita la creación de interfaces atractivas y funcionales con una variedad de componentes predefinidos.

Para abordar el apartado de administración tanto de la aplicación como de los eventos, se optó por escoger DataTables, pues proporciona funcionalidades avanzadas para manejar y presentar datos en tablas, incluyendo paginación, búsqueda y ordenamiento. Además, está optimizado para manejar grandes conjuntos de datos de manera eficiente, asegurando un rendimiento fluido incluso con miles de filas de datos.

Para el backend, se utilizaron tecnologías como Spring Boot y Lombok. Por su parte, Spring Boot proporciona una configuración predeterminada que facilita la creación de aplicaciones independientes y listos para producción. Una de sus ventajas más significativas es su amplio ecosistema de bibliotecas y herramientas, que permiten desarrollar microservicios,

³docs.npmjs.com/about-npm

⁴Proceso de agrupar todas las dependencias y componentes de una aplicación en un solo archivo.

*APIs RESTful*⁵ y aplicaciones web con facilidad. En el caso de este proyecto, se decidió utilizar una *API RESTful* porque permite una comunicación eficiente y estandarizada entre diferentes componentes del sistema. Además, en el futuro, esta *API* podría ser utilizada por terceros para extraer estadísticas sobre eventos, visualizaciones, crecimiento del mercado, y otros análisis, lo que añade un valor significativo al proyecto al facilitar la integración y el intercambio de datos con otras aplicaciones y servicios.

Lombok se utilizó para reducir el código *boilerplate*⁶ que normalmente se asocia con Java. Al usar anotaciones, Lombok genera automáticamente código como *getters*, *setters*, constructores y métodos *toString*⁷, entre otros, agilizando el desarrollo y mejorando la legibilidad del código. De esta forma, un desarrollador se puede centrar en la lógica del negocio en lugar de en la escritura de código repetitivo.

Para la gestión de datos, se optó por una combinación de bases de datos *SQL* y *NoSQL*⁸, específicamente PostgreSQL y MongoDB, para aprovechar las fortalezas de cada una según las necesidades del proyecto.

PostgreSQL es una base de datos relacional que ofrece robustez, integridad de los datos y soporte para transacciones complejas. Se seleccionó por su capacidad para manejar consultas *SQL* avanzadas y su compatibilidad con estándares *ACID* (Atomicidad, Consistencia, Aislamiento, Durabilidad). PostgreSQL es ideal para la mayoría de las operaciones de la aplicación, donde la consistencia y la integridad de los datos son críticas.

MongoDB, por otro lado, se utilizó específicamente para el manejo de logs y cambios en eventos debido a su naturaleza flexible y escalable. MongoDB es una base de datos *NoSQL* que almacena datos en documentos *JSON*, permitiendo una mayor flexibilidad en el almacenamiento y consulta de datos no estructurados. Esto es especialmente útil para el registro de eventos y cambios, donde la estructura de los datos puede variar y crecer con el tiempo sin necesidad de una redefinición esquemática.

6.1.2. Definición de requisitos

En el desarrollo de software, definir claramente los requisitos es una etapa crítica que determina el éxito del proyecto. Los requisitos de software describen las necesidades y expectativas del usuario y definen lo que el sistema debe hacer y cómo debe funcionar. En este apartado vamos a definir los requisitos funcionales y no funcionales a partir de los cuales hemos iniciado nuestro desarrollo.

Requisitos funcionales

- El sistema permitirá a los usuarios registrados crear eventos especificando detalles como nombre, fecha, ubicación, tipo, entre otros.

⁵developer.mozilla.org/en-US/docs/Glossary/REST

⁶developer.mozilla.org/en-US/docs/Web/API/WebGL_API/By_example/Boilerplate_1

⁷developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Object/toString

⁸Tipos de bases de datos no relacionales que almacenan datos en un formato distinto a las tablas relacionales.

- Durante la creación de eventos, se podrá seleccionar una plantilla de diseño para personalizar la apariencia visual de la página del evento.
- Debe haber una opción para establecer el evento como público o privado.
- Los creadores de eventos podrán asignar un propietario distinto para el evento, delegando responsabilidades de gestión.
- Se permitirá a los gestores de eventos personalizar módulos específicos del evento, menús, y actividades adicionales.
- Los usuarios podrán seleccionar elementos específicos dentro de un evento, según los módulos configurados.
- El sistema implementará técnicas robustas de autenticación y autorización, como *JWT*⁹ u *OAuth2*¹⁰, para proteger la aplicación y gestionar el acceso de usuarios según roles definidos.
- Debe existir un sistema de gestión de roles y permisos de usuario, especificando acciones permitidas para cada tipo de usuario (administrador, organizador, participante, etc.).
- El sistema proporcionará una interfaz intuitiva para la creación, gestión y visualización de eventos.
- Debe haber una interfaz de usuario para la gestión de eventos existentes, permitiendo a los usuarios editar detalles como nombre, fecha, ubicación y tipo de evento.
- Se implementará un sistema de filtrado de eventos que permita a los usuarios encontrar eventos específicos según criterios como visibilidad (público/privado), nombre, tipo, etc.
- Se proporcionará un panel de estadísticas para que los gestores de eventos evalúen el éxito del evento, mostrando datos como visualizaciones y asistentes confirmados.

Requisitos no funcionales

- El sistema deberá cumplir con estándares de seguridad aceptados para proteger los datos sensibles de los usuarios, incluyendo medidas como autenticación robusta, encriptación de datos y protección contra vulnerabilidades conocidas.
- El sistema debe ser capaz de manejar un alto volumen de eventos y usuarios concurrentes sin degradación significativa del rendimiento.
- Las consultas a la base de datos y la carga de páginas deben ser rápidas y eficientes para proporcionar una experiencia de usuario fluida.
- La arquitectura del sistema debe ser escalable, permitiendo añadir recursos y expansión de capacidades para soportar un futuro crecimiento del número de usuarios y eventos.
- La interfaz de usuario debe ser intuitiva y fácil de usar, siguiendo principios de diseño centrado en el usuario y accesibilidad.

⁹Token web *JSON*

¹⁰Estándar diseñado para permitir que un sitio web o una aplicación accedan a recursos alojados por otras aplicaciones web en nombre de un usuario.

- El código fuente y la arquitectura del sistema deben ser mantenibles, permitiendo futuras actualizaciones y modificaciones con facilidad.
- Se debe documentar adecuadamente el código y los procesos de desarrollo para facilitar la comprensión y el mantenimiento por parte de desarrolladores futuros.
- El sistema debe ser compatible con múltiples navegadores web y dispositivos móviles para asegurar una experiencia consistente para todos los usuarios.
- El sistema debe ser localizable y soportar múltiples idiomas, permitiendo que los usuarios seleccionen entre, español, francés, alemán e inglés.

6.2. Desarrollo de la Pila de Producto

La pila de producto presentada en el Anexo I fue desarrollada considerando los requisitos funcionales recopilados. Para esto, se adoptó un enfoque proactivo basado en la experiencia histórica del equipo de desarrollo en proyectos similares. La pila de producto permitió establecer una estructura coherente para la gestión de requisitos, enfocándose especialmente en la elaboración detallada de historias técnicas e historias de usuario.

Durante la fase de definición de dichas historias, se identificaron requisitos esenciales que incluyeron un identificador único para cada historia, un nombre descriptivo que comunicara claramente la funcionalidad requerida, y la asignación de prioridades dentro del contexto del proyecto. La descripción detallada de cada historia de usuario proporcionó un marco claro para comprender su propósito y alcance. Además, se delinearon las subtareas específicas necesarias para completar cada historia, asegurando una descomposición adecuada de las tareas.

Los criterios de validación definidos para cada historia de usuario fueron diseñados para establecer claramente las condiciones bajo las cuales se consideraría que una funcionalidad cumple con los requisitos esperados. Estos criterios proporcionaron un marco objetivo para la aceptación de cada entrega y garantizaron que las expectativas del cliente se alinearan con los resultados obtenidos. Finalmente, se realizó una estimación temporal (preliminar) para cada historia de usuario, basada en la experiencia pasada del equipo y en la complejidad percibida de cada tarea.

Por lo tanto, cada historia técnica y de usuario se compone de los siguientes campos: identificador único, nombre descriptivo, prioridad, descripción, criterios de validación, estimación temporal, subtareas.

Una vez creadas las historias de usuario, estas fueron incluidas en la plataforma ClickUp. Esta herramienta proporciona una vista detallada de cada tarea donde se pueden visualizar e introducir todos los campos mencionados anteriormente, como se muestra en la figura 6.1.

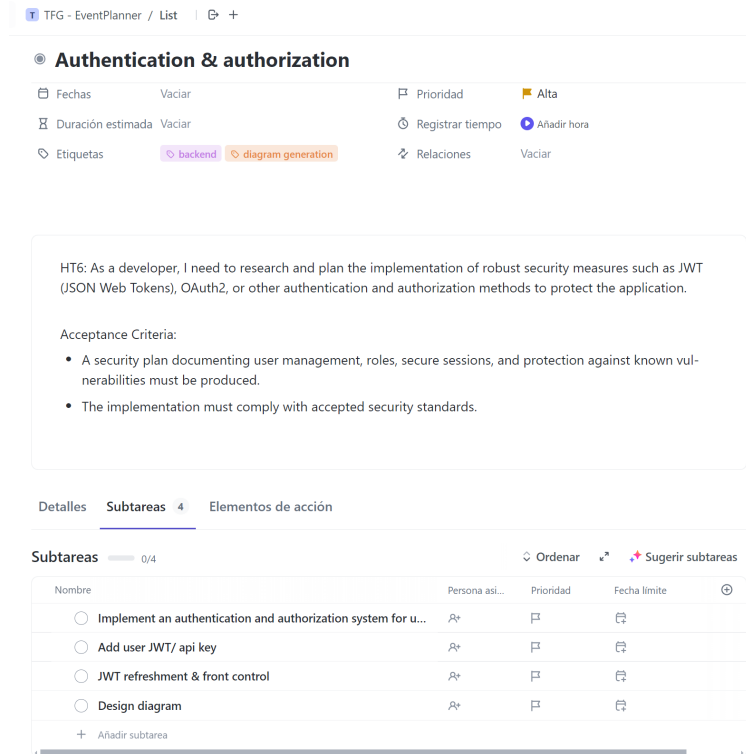


Figura 6.1: Ejemplo de historia técnica en la plataforma ClickUp.

6.3. Etapas de desarrollo

El desarrollo de la aplicación se dividió en ocho Sprints, de una duración de 2 semanas cada una. Cada iteración contó con una primera fase breve de planificación en la cual se definía la pila de Sprint. Durante dicha planificación se añadía el conjunto de historias más prioritarias y cuya estimación en conjunto se acercara lo máximo a las horas de trabajo estimadas para cada Sprint. Cabe mencionar que los Sprints 1, 2 y 3 se enfocaron en establecer las bases fundamentales del proyecto, mediante la implementación de las historia de usuario más críticas y las tareas técnicas esenciales.

Una vez definidas y estimadas las tareas a realizar en el Sprint correspondiente, se procedía con el desarrollo. Durante esa fase, se trabajó en los objetivos planificados para cada Sprint. Posteriormente, en la fase de revisión, se evaluaban las tareas realizadas, se valoraba el incremento de valor añadido y en caso que de ser necesario se actualizaba la pila de producto. A continuación, se detallará el trabajo realizado durante cada Sprint.

6.3.1. Sprint 1

Durante el primer sprint del proyecto, se enfocó en el diseño del modelo de base de datos, tanto para la implementación relacional en PostgreSQL como para la no relacional

en MongoDB. Un paso fundamental para asegurar, que la estructura de la base de datos pudiera soportar eficientemente las necesidades de almacenamiento y consulta del sistema, manteniendo la integridad de los datos.

Id	Nombre	Estimación	Prioridad
HT5	Diseño del modelo de base de datos	15 horas	Alta
Estimación total		15 horas	

Tabla 6.2: Planificación Sprint 1.

En el caso de PostgreSQL, se dividió el diagrama en dos secciones principales, uno dedicado a la gestión de usuarios y otro orientado a la administración de eventos. Esta separación facilitó la comprensión y el mantenimiento del modelo, permitiendo un enfoque más específico en cada área funcional del sistema.

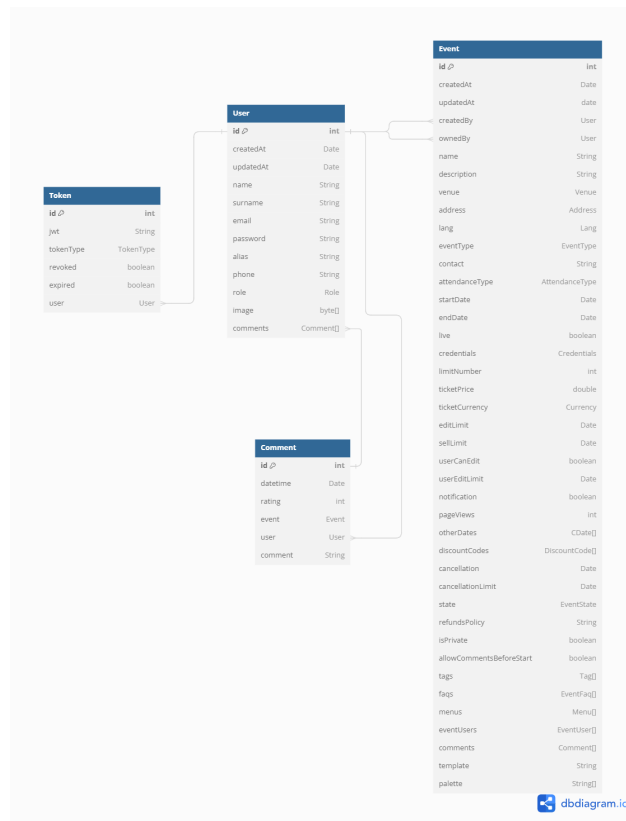


Figura 6.2: Diagrama de base de datos relacional: usuarios.

6.3.2. Sprint 2

En este sprint se abordaron dos historias técnicas importantes orientadas al diseño y estructura de la aplicación:

Id	Nombre	Estimación	Prioridad
HT8	Diseño de la interfaz de usuario	10 horas	Media
HT10	Diseño de diagrama de flujo	5 horas	Media
Estimación total		15 horas	

Tabla 6.3: Planificación Sprint 2.

Primero, se desarrolló el diagrama de flujo, una representación visual que describe la secuencia de acciones y decisiones dentro de la aplicación. Este diagrama es fundamental para comprender y comunicar cómo interactúan los distintos elementos y funcionalidades básicas del sistema, proporcionando una guía clara para el desarrollo y la implementación.

El diagrama de flujo no solo facilita la comprensión del flujo de trabajo y la navegación dentro de la aplicación, sino también sirve como referencia para identificar posibles mejoras y optimizaciones en la experiencia del usuario, asegurando coherencia y eficiencia en el diseño.

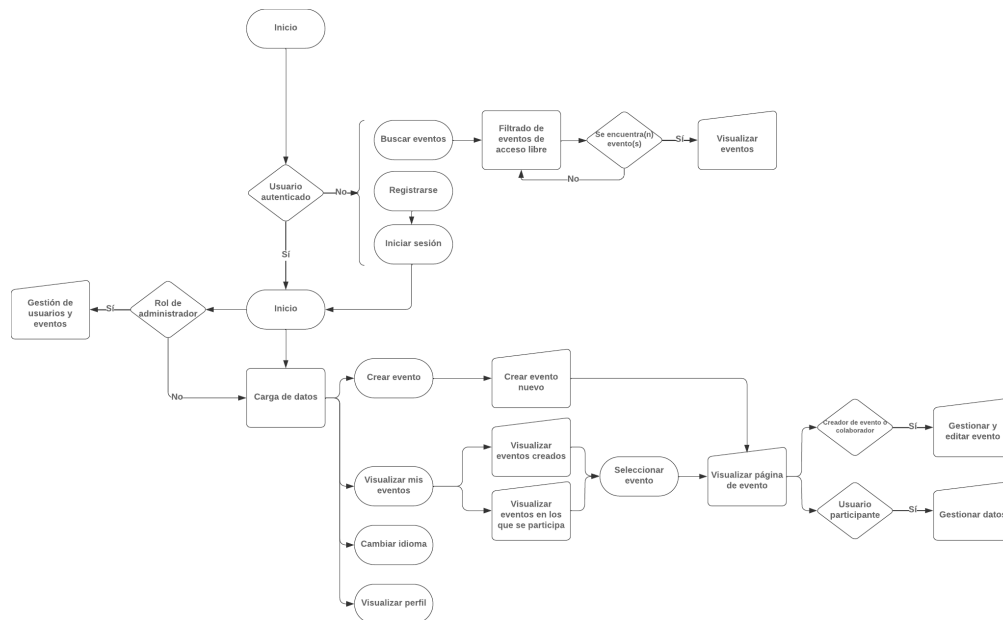


Figura 6.5: Diagrama de flujo de la aplicación

Además, en este sprint se realizaron los *mockups*¹¹, que son representaciones estáticas de la interfaz de usuario utilizando la paleta de colores definida para el proyecto. Estos *mockups* permitirán visualizar cómo se verá y funcionará la aplicación final, proporcionando una vista previa de las pantallas y elementos clave. A saber:

¹¹Representación visual similar a un prototipo que simula el aspecto visual de un diseño.



Figura 6.6: Paleta de colores de la aplicación.



Figura 6.7: Logo de la aplicación

Login

Username:

Password:

Login

Figura 6.8: Inicio de sesión (Figma)

Sign Up

Name:

Username:

Email:

Phone:

Password:

☐ Accept Terms & Conditions

Sign Up

Figura 6.9: Registro de usuario (Figma)

EVENT PLANNER

Welcome to EventPlanner

Create Events Discover Events Connect with Others

Search for an event

Event name

Year

Event type

Contact Us Follow Us

Figura 6.10: Pantalla de inicio (Figma)

EVENT PLANNER Home About Contact

New event

My events

Owned Events

CONCERT

Participated Events

SOCCER

TRAVEL

Figura 6.11: Pantalla de gestión de eventos (Figma)

(a) Creación de un evento. Paso 1 (Figma)

(b) Creación de un evento. Paso 2 (Figma)

(c) Creación de un evento. Paso 3.1 (Figma)

(d) Creación de un evento. Paso 3.2 (Figma)

Figura 6.12: Creación de un evento. Pasos 1-3.

(a) Creación de un evento. Paso 4 (Figma)

(b) Creación de un evento. Paso 5 (Figma)

(c) Creación de un evento. Paso 6 (Figma)

(d) Creación de un evento. Paso 7 (Figma)

Figura 6.13: Creación de un evento. Pasos 4-7.

6.3.3. Sprint 3

Durante este Sprint se llevó a cabo la configuración inicial del proyecto, sentando las bases tanto para el desarrollo del front-end como del back-end.

Id	Nombre	Estimación	Prioridad
HT11	Configuración del entorno de desarrollo	25 horas	Alta
Estimación total		25 horas	

Tabla 6.4: Planificación Sprint 3.

Una de las decisiones clave fue la implementación de una infraestructura basada en contenedores utilizando Docker. Esta elección estratégica aseguró un despliegue eficiente y escalable del sistema, facilitando la gestión de entornos de desarrollo, pruebas y producción de manera consistente.

En el lado del *back-end*, se ha configurado un entorno que incluye los servicios esenciales para el despliegue del proyecto. Para los servicios de bases de datos PostgreSQL y MongoDB, se han utilizado imágenes oficiales de Docker, asegurando así la mantenibilidad y estabilidad de los datos en el entorno de desarrollo y producción.

El resto de los servicios, como la aplicación Spring Boot y la interfaz de usuario Vue, se desplegaron a través de imágenes creadas de forma local. Esta decisión se fundamentó en la necesidad de personalizar y optimizar cada entorno de ejecución específicamente para las particularidades del proyecto. Crear imágenes locales permitió tener un control preciso sobre las dependencias y configuraciones presentes en cada servicio, garantizando coherencia y reproducibilidad a lo largo de todo el ciclo de desarrollo y despliegue. Además, aporta la flexibilidad de subir las imágenes así creadas al Docker Hub¹² o a cualquier otro repositorio de imágenes en producción, si así se requiriera. Esto facilita la distribución y el despliegue en diferentes entornos, asegurando que las versiones de las aplicaciones estén debidamente controladas y sean fácilmente accesibles para la implementación en nuevos servidores o actualizaciones en el entorno de producción.

Por su parte, el contenedor del servidor, que alberga la *API*, se ha desplegado mediante un contenedor dedicado que utiliza variables de entorno, como se muestra en la figura 6.14. El uso de variables de entorno asegura que la configuración sensible, como contraseñas de bases de datos y claves secretas de tokens *JWT*, se mantenga fuera del código fuente y de las imágenes del contenedor. Esta práctica es esencial para garantizar la seguridad y protección de los datos sensibles.

Al separar la configuración confidencial en variables de entorno, se evita la exposición accidental de información sensible durante el desarrollo, el despliegue y la operación del sistema. Además, esto facilita la gestión de diferentes entornos, permitiendo configuraciones específicas para desarrollo, pruebas y producción sin necesidad de modificar el código fuente.

¹²Registro para repositorios de software en la nube.

```

db:
  image: postgres:16.3
  container_name: evplannerdb
  ports:
    - "${DB_PORT}:5432"
  restart: always
  environment:
    - POSTGRES_DB=${DB_DBNAME}
    - POSTGRES_USER=${DB_USER}
    - POSTGRES_PASSWORD=${DB_PASS}
  volumes:
    - pgdata:/var/lib/postgresql/data
  networks:
    - evplanner-network
mongodb:
  image: mongo:4.4.26
  container_name: evplmongo
  restart: always
  environment:
    MONGO_INITDB_DATABASE: ${MONGO_DBNAME}
    MONGO_INITDB_ROOT_USERNAME: ${MONGO_USER}
    MONGO_INITDB_ROOT_PASSWORD: ${MONGO_PASS}
  networks:
    - evplanner-network
mongo-express:
  image: mongo-express:latest
  restart: always
  ports:
    - "${MONGO_PORT}:8081"
  environment:
    ME_CONFIG_MONGODB_ADMINUSERNAME: ${MONGO_USER}
    ME_CONFIG_MONGODB_ADMINPASSWORD: ${MONGO_PASS}
    ME_CONFIG_MONGODB_SERVER: ${MONGO_SERVER}
  networks:
    - evplanner-network

```

Figura 6.14: Configuración de los servicios de bases de datos en el *docker-compose.yml* de la aplicación.

El servicio del servidor depende directamente de dos servicios de base de datos: db (PostgreSQL) y mongodb (MongoDB), garantizando que ambos estén disponibles y operativos antes de que el servicio del servidor se inicie. Esta dependencia asegura que la *API* pueda interactuar correctamente con las bases de datos desde el principio, evitando errores de conexión y garantizando un funcionamiento estable del sistema.

Para la creación del contenedor del lado del servidor, se ha empleado un archivo Dockerfile que especifica la construcción y ejecución de la aplicación Java. Este archivo copia el archivo ‘.jar’ generado previamente con Maven y configura el entorno para ejecutar la aplicación de manera eficiente y segura. A través de este enfoque, se optimiza el desempeño de la *API*, minimizando tiempos de carga y consumos excesivos de recursos.

```

server:
  image: evplanner-api:latest
  container_name: evplanner-sv
  ports:
    - "${API_PORT}:3004"
  volumes:
    - "./server:/api"
  environment:
    - DB_URL=jdbc:postgresql://evplannerdb:${DB_PORT}/${DB_DBNAME}
    - MONGO_URL=mongodb://evplmongo:${MONGO_PORT}/${MONGO_DBNAME}
    - DB_USER=${DB_USER}
    - DB_PASS=${DB_PASS}
    - ALLOWED_ORIGINS=${ALLOWED_ORIGINS}
    - JWT_SECRET=${JWT_SECRET}
    - JWT_EXPIRATION_TIME=${JWT_EXPIRATION_TIME}
    - JWT_REFRESH_TOKEN_EXPIRATION=${JWT_REFRESH_TOKEN_EXPIRATION}
  depends_on:
    - db
    - mongodb
  networks:
    - evplanner-network

```

Figura 6.15: Configuración del servicio de Spring boot en el *docker-compose.yml* de la aplicación.

En cuanto al *front-end*, se ha optado por utilizar Nginx como servidor web para servir la aplicación Vue. Nginx ofrece un rendimiento excepcional y una configuración flexible que permite manejar el enrutamiento y la distribución de carga de manera eficiente. Este nuevo servicio llamado client depende del servicio de Spring boot, server, garantizando que la API esté disponible antes de que se inicie el *front-end*, para así asegurar una integración fluida entre el *front-end* y el *back-end* de la aplicación.

```

client:
  image: evplanner-client:latest
  container_name: evplanner-app
  environment:
    - CHOKIDAR_USEPOLLING=true
  ports:
    - "${APP_PORT}:82"
  volumes:
    - "./client:/app"
    - "/app/node_modules"
  depends_on:
    - server
  networks:
    - evplanner-network

```

Figura 6.16: Configuración del servicio de Vue en el *docker-compose.yml* de la aplicación.

El Dockerfile para el *front-end* está diseñado para copiar las dependencias especificadas en el package.json, construir la aplicación Vue y desplegarla dentro del contenedor Nginx, permitiendo gestionar fácilmente el ciclo de vida del *front-end*, desde el desarrollo hasta la puesta en producción.

Una vez configurados y creados los contenedores, se procedió a verificar la creación y conexión a las bases de datos de MongoDB y PostgreSQL utilizando DBeaver. Esta herramienta

facilitó la verificación de la correcta configuración y operatividad de ambas bases de datos en los contenedores, asegurando que las conexiones estuvieran establecidas correctamente y que las bases de datos estuvieran listas para recibir y almacenar datos.

A continuación, se procedió con la creación de la *API*. Este proceso incluyó varios pasos importantes, cada uno de los cuales fue esencial para garantizar una integración fluida y un funcionamiento eficiente de la aplicación.

El primer paso fue la instalación de las dependencias necesarias para interactuar con las bases de datos. Se instalaron las bibliotecas de PostgreSQL y MongoDB, así como Lombok. Estas dependencias son fundamentales para establecer una conexión estable y eficiente con las bases de datos y para mejorar la mantenibilidad del código.

Posteriormente, se configuró el archivo ‘application.properties’ para que pudiera recibir adecuadamente las variables de entorno definidas en el contenedor, tal y como muestra la figura 6.17.

```
# Datasource connection
spring.datasource.url=${DB_URL}
spring.datasource.username=${DB_USER}
spring.datasource.password=${DB_PASS}
spring.data.mongodb.uri=${MONGO_URL}
spring.sql.init.mode=always
spring.datasource.initialization-mode=always

# Hibernate
spring.jpa.hibernate.ddl-auto=update
spring.jpa.hibernate.show-sql=true

# Env variables
allowed.origins=${ALLOWED_ORIGINS}
security.jwt.secret-key=${JWT_SECRET}
security.jwt.expiration-time=${JWT_EXPIRATION_TIME}
security.jwt.refresh-token=${JWT_REFRESH_TOKEN_EXPIRATION}
```

Figura 6.17: Configuración de variables de entorno en Spring Boot.

Una vez realizada la configuración, se probó la conexión con las bases de datos utilizando Postman. Esta herramienta permitió verificar que la API podían comunicarse correctamente con PostgreSQL y MongoDB, asegurando que las consultas y las operaciones de almacenamiento funcionaran según lo esperado.

En el siguiente paso, se definieron los dominios (entidades, *requests*, *responses* y enumerados) necesarios. Las entidades representan las tablas de la base de datos en el código Java, facilitando la interacción con los datos. Los objetos *requests* y *responses* se utilizan para la comunicación entre el cliente y el servidor. En lugar de enviar directamente las entidades de la base de datos, se emplearon objetos *request* para recibir datos de entrada y objetos *response* para enviar datos de salida. Esto proporcionó una capa adicional de seguridad y control, permitiendo validar y transformar los datos antes de almacenarlos o enviarlos al cliente.

Los mapeadores son componentes esenciales en esta arquitectura. Se encargan de convertir las entidades en objetos *responses* y viceversa. Esto es especialmente útil para mantener

la separación de preocupaciones, asegurando que los datos que se envían al cliente estén en el formato correcto y contengan solo la información necesaria. Por ejemplo, el AnswerFAQMapper de la figura 6.18 convierte una entidad AnswerFAQ en un AnswerFAQResponse que puede ser fácilmente consumido por el *front-end*.

```
@NoArgsConstructor
public class AnswerFAQMapper {
    public static AnswerFAQResponse generateResponse(AnswerFAQ answerFAQ) {
        return AnswerFAQResponse.builder()
            .id(answerFAQ.getId())
            .answer(answerFAQ.getAnswer())
            .build();
    }

    public static AnswerFAQ createAnswerFAQ(AnswerFAQRequest request) {
        return AnswerFAQ.builder()
            .id(null)
            .answer(request.getAnswer())
            .build();
    }

    public static AnswerFAQ updateAnswerFAQ(AnswerFAQ answerFAQ, AnswerFAQRequest request) {
        return AnswerFAQ.builder()
            .id(answerFAQ.getId())
            .answer(request.getAnswer())
            .build();
    }
}
```

Figura 6.18: Ejemplo de mapeador: AnswerFAQMapper.

Los servicios y los controladores son componentes esenciales en una aplicación Spring Boot. Los servicios encapsulan la lógica de negocio y gestionan las operaciones con los datos, mientras que los controladores manejan las solicitudes *HTTP* y coordinan la interacción entre el *front-end* y los servicios.

Por ejemplo, en el servicio AnswerFAQService visto en la figura 6.19, se encapsulan todas las operaciones relacionadas con la entidad AnswerFAQ. Este servicio utiliza el repositorio AnswerFAQRepository para acceder a la base de datos y el mapeador AnswerFAQMapper para convertir entre las entidades y los objetos de respuesta. El mapeador se utiliza para mantener la separación de responsabilidades, asegurando que los datos enviados al cliente estén en el formato correcto y contengan solo la información necesaria. De esta manera, se evita exponer directamente las entidades de la base de datos al *front-end*, lo que mejora la seguridad y la flexibilidad del sistema.

```

@RequiredArgsConstructor
@Service
public class AnswerFAQService {
    public List<AnswerFAQResponse> getAll() {
        Iterable<AnswerFAQ> answerFAQs = repository.findAll();

        List<AnswerFAQResponse> answerFAQResponses = new ArrayList<>();
        answerFAQs.forEach(answerFAQ -> answerFAQResponses.add(AnswerFAQMapper.generateResponse(answerFAQ)));
        return answerFAQResponses;
    }

    public AnswerFAQResponse createAnswerFAQ(AnswerFAQRequest request) {
        final AnswerFAQ answerFAQ = repository.save(AnswerFAQMapper.createAnswerFAQ(request));
        return AnswerFAQMapper.generateResponse(answerFAQ);
    }

    // Resto de métodos
    // ..
}

```

Figura 6.19: Ejemplo de servicio: AnswerFAQService.

Como ejemplo de controlador, el AnswerFAQController maneja las solicitudes HTTP relacionadas con AnswerFAQ. Este controlador expone varios *endpoints*¹³ que permiten realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre la entidad AnswerFAQ. Cada método del controlador se corresponde con una operación específica y delega la lógica de negocio al servicio correspondiente. Por ejemplo, el método createAnswerFAQ recibe una solicitud POST con los datos de un AnswerFAQRequest, los pasa al servicio para crear una nueva entidad y devuelve una respuesta con los datos del AnswerFAQ creado.

Al emplear este enfoque modular y bien estructurado, se logra una mayor claridad en el código, facilitando su mantenimiento y evolución. Además, esta arquitectura permite una mejor gestión de los datos y una mayor seguridad al evitar la exposición directa de las entidades de la base de datos al cliente.

```

@RestController
@RequiredArgsConstructor
@RequestMapping("/api/answer-faqs")
public class AnswerFAQController {
    @GetMapping
    public ResponseEntity<List<AnswerFAQResponse>> getAll() {
        return ResponseEntity.ok(service.getAll());
    }

    @PostMapping
    public ResponseEntity<AnswerFAQResponse> createAnswerFAQ(@RequestBody AnswerFAQRequest request) {
        return ResponseEntity.ok(service.createAnswerFAQ(request));
    }

    // Resto de métodos
    // ..
}

```

Figura 6.20: Ejemplo de servicio: AnswerFAQService.

¹³ URL específica que se utiliza para acceder a un recurso proporcionado por una aplicación web desde una API.

Una vez configurado el *back-end*, se procedió con la configuración del *front-end*. Este proceso implicó varios pasos clave para asegurar que la aplicación Vue estuviera completamente integrada con el *back-end* y que ofreciera una experiencia de usuario coherente y fluida.

Primero, se crearon los servicios necesarios para realizar las llamadas al *back-end*. Estos servicios son módulos en Vue que encapsulan las llamadas *HTTP* a la *API*, utilizando herramientas como Axios¹⁴ para gestionar las solicitudes. Para evitar la repetición de código y simplificar la gestión de las solicitudes *HTTP*, se implementó un archivo centralizado que manejara las llamadas a la *API*. Este enfoque facilita la reutilización del código, reduce la duplicación y asegura que todas las solicitudes *HTTP* sigan un patrón consistente. Al probar las llamadas de tipo GET¹⁵ desde los servicios, se verificó la correcta conexión con el *back-end*, asegurando que la comunicación entre el *front-end* y el *back-end* funcionara sin problemas. Esta fase de pruebas es crucial para identificar y solucionar problemas de conexión o de configuración antes de avanzar con el desarrollo de funcionalidades más complejas.

```
const BASE_URL = import.meta.env.VITE_LOC_URL
const TOKEN_VAL = import.meta.env.VITE_AUTH_TOKEN
const TOKEN_HEADER = import.meta.env.VITE_AUTH_TOKEN_HEADER

async function sendRequest(method: string, url: string, data?: any) {
  const config: AxiosRequestConfig = {
    method,
    url: BASE_URL + url,
    headers: {
      [TOKEN_HEADER]: TOKEN_VAL,
    },
    data,
  };

  // Bloque de gestión
  // ...
}
export { sendRequest }
```

Figura 6.21: Implementación del método centralizado para realizar llamadas HTTP utilizando Axios.

A continuación, se configuró el enrutamiento utilizando Vue Router. Esto permitió definir las rutas de la aplicación, asegurando que cada vista se cargara correctamente según la *URL* solicitada.

Seguidamente, se configuró el almacenamiento del estado global usando Vuex. Vuex permite gestionar el estado de la aplicación de manera centralizada, facilitando la comunicación entre componentes y la gestión de datos persistentes, como la información de usuario.

También se crearon las interfaces correspondientes, conocidas como DTO¹⁶, que son estructuras de datos diseñadas para transferir información entre el *front-end* y el *back-end*. Usar *DTOs* en el *front-end* permite definir claramente los datos esperados y simplificar la

¹⁴Librería de *JavaScript* que puede ejecutarse en el navegador permitiendo operaciones como cliente *HTTP*.

¹⁵developer.mozilla.org/en-US/docs/Web/HTTP/Methods/GET

¹⁶Patrón de diseño utilizado para transferir datos entre subsistemas de una aplicación de *software*.

manipulación de estos datos, garantizando que el formato sea consistente y válido tanto al enviar como al recibir información de la *API*.

```
export interface AnswerFaqDto {
  id: number;
  answer: string;
}

export interface QuestionFaqDto {
  id?: number;
  question: string;
}

export default interface EventFaqDto {
  id?: number;
  eventId: number;
  questionId: number;
  answerId: number;
}
```

Figura 6.22: Definiciones de *DTO* en el archivo `FaqDto.ts`

Asimismo, se crearon los enumerados necesarios para representar estados y tipos específicos dentro de la aplicación. Los enumerados facilitan la gestión de valores constantes y predefinidos, mejorando la legibilidad y el mantenimiento del código. Por ejemplo:

```
export enum EventType {
  BIRTHDAY = "BIRTHDAY",
  PARTY = "PARTY",
  MUSICAL = "MUSICAL",

  // Resto de tipos
  // ..
}
```

Figura 6.23: Definición del enumerado de `EventType`.

Por último, para mejorar la estética y la coherencia visual de la aplicación, se añadió una paleta de colores y se configuraron estilos básicos. Esto incluyó la definición de variables de colores y la creación de estilos globales para asegurar una apariencia uniforme en toda la aplicación.

6.3.4. Sprint 4

Durante el Sprint 4, se llevó a cabo la historia técnica enfocada en la investigación e implementación de medidas avanzadas de seguridad para la plataforma.

Id	Nombre	Estimación	Prioridad
HT6	Investigación de técnicas de autenticación y seguridad	20 horas	Alta
Estimación total		20 horas	

Tabla 6.5: Planificación Sprint 4.

Como resultado de esta investigación, se optó por implementar *JSON Web Token (JWT)* como la medida principal de seguridad para la *API* del servidor. Esta decisión fue crucial debido a las características y ventajas que ofrece *JWT* como estándar abierto (RFC 7519) para la autenticación y transferencia segura de información entre el cliente y el servidor en aplicaciones web modernas.

La elección de *JWT* se fundamentó en varias razones estratégicas y técnicas. *JWT* proporciona un método compacto y seguro para transmitir información entre dos partes como un objeto *JSON* firmado digitalmente. Esto permite la verificación de la integridad de los datos y la autenticación del usuario sin necesidad de almacenar información en el servidor, lo cual es ideal para arquitecturas distribuidas y escalables.

La implementación de *JWT* en el sistema se realizó utilizando bibliotecas estándar y prácticas recomendadas. Primero, se configuró un servicio dedicado (*JwtService*) para la generación, validación y extracción de tokens *JWT*. Este servicio utiliza una clave secreta y tiempos de expiración configurables mediante variables de entorno para firmar y verificar los tokens, asegurando que solo el servidor pueda emitir y verificar la autenticidad de los tokens, tal y como puede verse en la figura 6.24.

En el controlador de autenticación (*AuthenticationController*), se definieron *endpoints* para el registro, inicio de sesión y actualización de tokens *JWT*. Cuando un usuario se registra o inicia sesión, se genera un nuevo par de tokens *JWT*: un token de acceso y un token de actualización. El token de acceso se utiliza para autenticar las solicitudes del cliente al servidor, mientras que el token de actualización se utiliza para obtener un nuevo token de acceso sin necesidad de autenticación adicional.

Además, se implementó un mecanismo de manejo de tokens en el servicio de autenticación (*AuthenticationService*) de forma que, cuando un usuario realice acciones como el inicio de sesión o el registro, se generen y almacenen nuevos tokens *JWT* en la base de datos asociados al usuario. Para garantizar la seguridad y la gestión adecuada de estos tokens, se implementaron métodos para revocar todos los tokens activos de un usuario cuando inicia sesión nuevamente o solicita un nuevo token de acceso.

```
@Service
public class JwtService {
    private String buildToken(Map<String, Object> extraClaims, UserDetails userDetails, long expiration) {
        return Jwts
            .builder()
            .setClaims(extraClaims)
            .setSubject(userDetails.getUsername())
            .setIssuedAt(new Date(System.currentTimeMillis()))
            .setExpiration(new Date(System.currentTimeMillis() + expiration))
            .signWith(getSignInKey(), SignatureAlgorithm.HS256)
            .compact();
    }

    // Otros métodos
    // ...
}
```

Figura 6.24: Extracto del servicio *JwtService*.

Para mejorar la seguridad del sistema, se configuraron filtros (`JwtAuthenticationFilter`) para validar los tokens *JWT* en cada solicitud entrante antes de permitir el acceso a los recursos protegidos. Estos filtros interceptan las solicitudes, verifican la validez del token y autentican al usuario basándose en la información contenida en el token *JWT*.

```
@Service
@RequiredArgsConstructor
public class AuthenticationService {

    // Definiciones y otros métodos
    // ...

    public AuthenticationResponse login(LoginRequest request) {
        authenticationManager.authenticate(
            new UsernamePasswordAuthenticationToken(request.getEmail(), request.getPassword()));

        User user = userRepository.findByEmail(request.getEmail()).orElseThrow();
        String jwtToken = jwtService.generateToken(user);
        String refreshToken = jwtService.generateRefreshToken(user);

        revokeAllUserTokens(user);
        saveUserToken(user, jwtToken);
        return AuthenticationResponse.builder()
            .accessToken(jwtToken)
            .refreshToken(refreshToken)
            .build();
    }

    // Métodos relacionados con JWT
    // ..
}
```

Figura 6.25: Función de inicio de sesión del servicio `AuthenticationService`

6.3.5. Sprint 5

Durante el Sprint 5, se llevaron a cabo varias historias técnicas y de usuario clave para fortalecer tanto la seguridad como la funcionalidad de la aplicación.

Id	Nombre	Estimación	Prioridad
HT7	Gestión de roles y permisos de usuario	5 horas	Media
HU1	Registro de un usuario	4 horas	Alta
HU2	Inicio de sesión	4 horas	Alta
HU22	Gestión de usuarios	12 horas	Baja
Estimación total		25 horas	

Tabla 6.6: Planificación Sprint 5.

Una de las iniciativas principales fue la implementación de gestión de roles y permisos en el *back-end*. Esta tarea implicó la configuración detallada de roles específicos, como usuario estándar, administrador y proveedor, cada uno con permisos definidos que regulan el acceso a diferentes partes del sistema. Por ejemplo, se otorgaron permisos solo a los administradores

para realizar operaciones sensibles como la eliminación de usuarios o la actualización de roles, tal y como muestra la figura 6.26.

```
@RequiredArgsConstructor
public enum Role {
    ADMIN("Admin",
        Set.of(
            Permission.ADMIN_READ,
            Permission.ADMIN_UPDATE,
            Permission.ADMIN_CREATE,
            Permission.ADMIN_DELETE,
            Permission.EVENT_READ,
            Permission.EVENT_UPDATE,
            Permission.EVENT_CREATE,
            Permission.EVENT_DELETE,
            Permission.EVENT_ASSIGN_ROLES,
            Permission.USER_READ,
            Permission.USER_UPDATE,
            Permission.USER_UPDATE_OWN,
            Permission.VENDOR_READ,
            Permission.VENDOR_UPDATE,
            Permission.VENDOR_UPDATE_OWN,
            Permission.VENDOR_CREATE,
            Permission.VENDOR_DELETE
        )
    ),
    // Otros roles de aplicación y de evento
    // ..
}
```

Figura 6.26: Extracto del enumerado de roles de Spring Boot.

Simultáneamente, se procedió a limitar las rutas de acceso necesarias en el *back-end* mediante el uso de anotaciones `PreAuthorize` en los controladores, asegurando que solo los usuarios autenticados y autorizados pudieran acceder a recursos específicos. Un ejemplo claro es el controlador `UserController`, en la figura 6.27 donde se implementaron restricciones para garantizar que solo los administradores y los usuarios autenticados con roles específicos pudieran realizar acciones como actualizar información de usuario o modificar contraseñas.

```
@RestController
@RequiredArgsConstructor
@RequestMapping("/api/users")
public class UserController {

    // Otros métodos
    // ...

    @PatchMapping("/change-role")
    @PreAuthorize("hasRole('ADMIN')")
    public ResponseEntity<String> changeRole(ChangeRoleRequest request) {
        userService.changeRole(request);
        return ResponseEntity.ok().build();
    }
}
```

Figura 6.27: Extracto del controlador de usuario `UserController`.

En el frontend, se integraron estas mejoras mediante la configuración de restricciones basadas en roles dentro del enrutador. Se estableció una lógica robusta para manejar la autenticación y la autorización, redirigiendo automáticamente a los usuarios no autenticados hacia la página de inicio de sesión al intentar acceder a áreas protegidas de la aplicación. Además, se implementaron acciones y mutaciones para gestionar de manera efectiva el inicio y cierre de sesión, manteniendo actualizado el estado global y sincronizándolo con el almacenamiento local mediante el uso de `localStorage`.

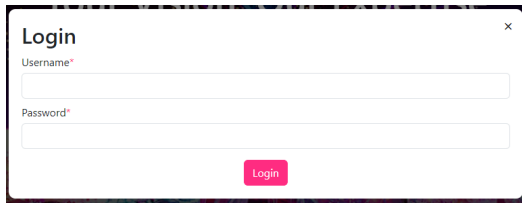
```
router.beforeEach((to, from, next) => {
  const role = localStorage.getItem('role')
  const isLoggedIn = localStorage.getItem('isLoggedIn')
  const requiresAuth = to.meta.requiresAuth

  if (requiresAuth && !isLoggedIn) {
    next('/login') // User is not authenticated, redirect to login
  } else if (requiresAuth && to.path === '/admin' && role !== 'ADMIN') {
    next('/')
  } else {
    next() // Allow navigation to other routes
  }
})
```

Figura 6.28: Extracto de router en Vue.

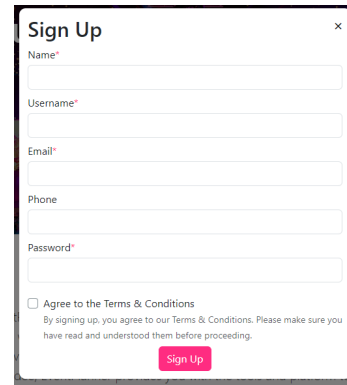
Se optó por el uso de `localStorage` en lugar de otras formas de almacenamiento como `sessionStorage` por varias razones clave. Primero, `localStorage` proporciona una capacidad de almacenamiento mayor en comparación con `sessionStorage`, lo cual es beneficioso para aplicaciones que requieren mantener datos persistentes incluso después de que el usuario cierra y vuelve a abrir el navegador. Esto es especialmente útil para mantener la sesión de usuario activa entre sesiones y garantizar una experiencia continua y sin interrupciones para los usuarios. Por otro lado, `localStorage` es accesible a través de la propiedad global `window.localStorage`, lo que lo hace compatible con diferentes navegadores y entornos de ejecución. Por último, por su simplicidad de uso y su capacidad para almacenar datos de manera estructurada, como objetos JSON, lo cual facilita la manipulación y recuperación de información relacionada con el estado de la sesión de usuario y los roles asignados.

Además de las mejoras en la gestión de roles y en la seguridad del acceso, se desarrollaron nuevas vistas clave durante este sprint, incluyendo páginas dedicadas al registro de usuarios, inicio de sesión y gestión avanzada de perfiles. La correcta implementación de estas vistas no solo aseguró una experiencia de usuario más intuitiva y segura, sino que también validó la funcionalidad completa de autenticación desde el frontend hacia el backend, garantizando que todas las interacciones fueran consistentes y confiables.



A login popup form with a title bar 'Login' and a close button 'x'. It contains two input fields: 'Username*' and 'Password*'. Below the fields is a red 'Login' button.

Figura 6.29: Popup de inicio de sesión



A sign up popup form with a title bar 'Sign Up' and a close button 'x'. It contains five input fields: 'Name*', 'Username*', 'Email*', 'Phone', and 'Password*'. Below the fields is a checkbox for 'Agree to the Terms & Conditions' with a small text block below it. At the bottom is a red 'Sign Up' button.

Figura 6.30: Popup de registro de usuario

Por otro lado, se crearon componentes reutilizables para las vistas, lo cual incluyó la implementación de generadores. Estos generadores, como CheckboxGenerator, DatePickerGenerator, PopupGenerator y TooltipHintGenerator, permiten modularizar y estandarizar la creación de elementos interactivos y descriptivos en las interfaces de usuario.

Además, se implementó el componente de navegación (navbar), diseñado para mejorar la experiencia de navegación. Este componente ofrece acceso rápido a varias secciones esenciales del sitio, como la página principal, información del proyecto, contacto, autenticación, cambio de idioma y creación y gestión de eventos para usuarios autenticados. Estas funcionalidades fueron incorporadas con el objetivo de optimizar la accesibilidad y la usabilidad general del sitio, asegurando una navegación intuitiva y eficiente para todos los usuarios.

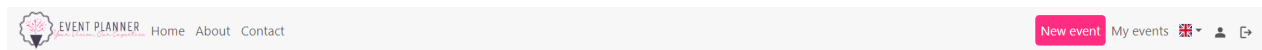


Figura 6.31: Componente de navegación

En el sprint también se desarrolló la página principal del proyecto, diseñada como un punto focal para los visitantes y usuarios del sitio. Esta página fue cuidadosamente estructurada para ofrecer una visión clara y atractiva de los servicios y funcionalidades disponibles.

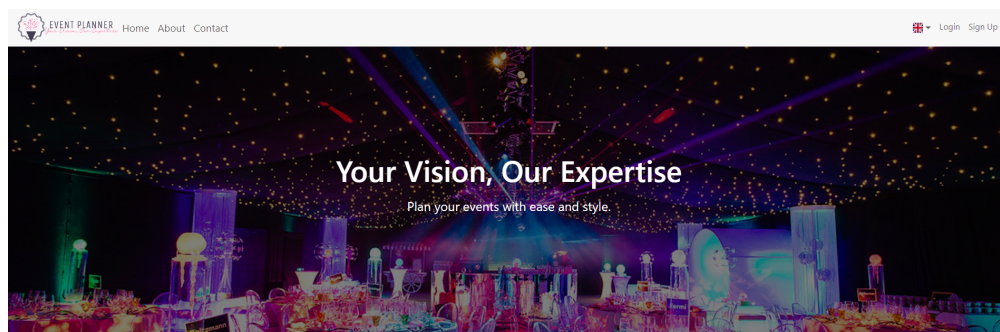


Figura 6.32: Página principal del proyecto

Se diseñaron los componentes de información y de contacto. El botón del componente de información lleva a la creación siempre y cuando haya un usuario con la sesión iniciada, caso contrario, abre el *Pop-up*¹⁷ de registro.

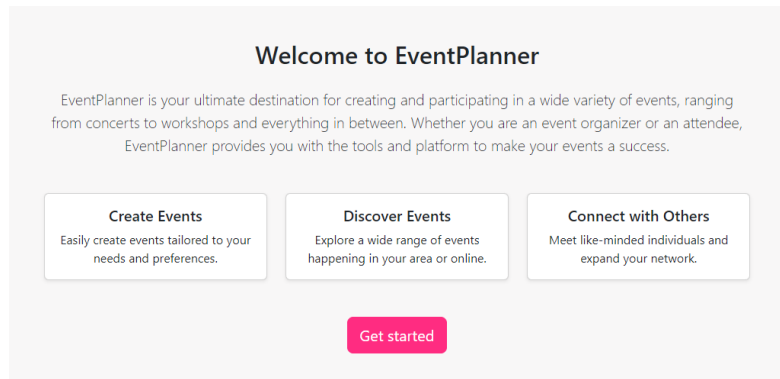


Figura 6.33: Componente de información

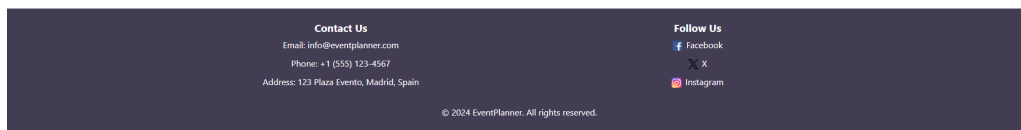


Figura 6.34: Componente de contacto

Por último, se implementó un módulo para la gestión de idiomas, ofreciendo a los usuarios la posibilidad de elegir entre varios idiomas disponibles. Este módulo se apoya en un archivo centralizado que almacena y controla las traducciones en diferentes idiomas, lo que garantiza una experiencia plurilingüe coherente y adaptable.

El archivo de gestión de idioma (i18n) juega un papel crucial en esta funcionalidad, empleando `vue-i18n` para manejar las traducciones en diferentes idiomas. Este archivo permite cargar constantes de idioma desde archivos *JSON* separados para cada módulo del proyecto, asegurando una internacionalización efectiva y flexible.

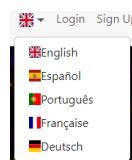


Figura 6.35: Selector de cambio de idioma

¹⁷Ventana emergente.

```

const locales = ['en', 'es', 'de', 'pt', 'fr'];

function getInitialLocale() {
  const storedLocale = localStorage.getItem('lang');
  return storedLocale && locales.includes(storedLocale) ? storedLocale : 'en';
}

const i18n = createI18n({
  legacy: false,
  locale: getInitialLocale(),
  messages: { en, es, de, pt, fr }
});

export async function loadLangConstants(modules: string | string[]) {
  if (!Array.isArray(modules)) modules = [modules];

  for (const module of modules) {
    for (const locale of locales) {
      const messages = await import(`@/locales/${module}/${locale}.json`);
      i18n.global.setLocaleMessage(locale, {
        ...i18n.global.getLocaleMessage(locale),
        ...messages.default
      });
    }
  }
}

export function changeLanguage(i18n: any, locale: any) {
  i18n.locale = locale
  localStorage.setItem('lang', locale)
}

export default i18n;

```

Figura 6.36: Módulo de gestión de idiomas i18n.ts

6.3.6. Sprint 6

El Sprint 6 se centró en la creación de eventos, por tanto, se desarrollaron las siguientes historias de usuario:

Id	Nombre	Estimación	Prioridad
HU3	Crear un evento	14 horas	Alta
HU7	Personalizar módulos del evento	4 horas	Alta
HU8	Establecer fecha límite de edición del evento	1 hora	Baja
HU9	Establecer un propietario distinto del creador del evento	1 hora	Media
HU10	Establecer datos financieros para el evento	1 hora	Media
HU12	Asignar fechas adicionales a un evento	4 horas	Baja
HU13	Añadir códigos de descuento	1 hora	Baja
HU17	Establecer eventos como públicos o privados	1 hora	Alta
HU19	Seleccionar elementos específicos dentro de un evento	1 hora	Baja
HU20	Establecer fechas de lanzamiento del evento	1 hora	Alta
HU21	Asignar una ubicación al evento	1 hora	Media
Estimación total		30 horas	

Tabla 6.7: Planificación Sprint 6.

Se implementó la funcionalidad para crear eventos, que permite a los usuarios autenticados configurar y personalizar eventos según sus necesidades específicas. Esta característica se distingue por su enfoque para facilitar el proceso de creación mediante un *stepper*¹⁸ que guía al usuario a través de varias fases. Esto reduce la complejidad y el tiempo requerido, permitiendo a los usuarios ingresar una amplia gama de información para personalizar sus eventos de manera integral.

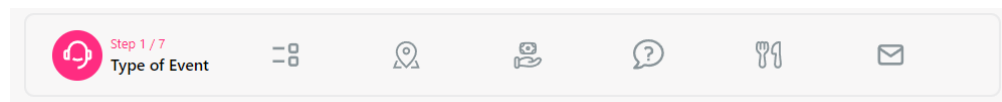


Figura 6.37: *Stepper* para la creación de eventos

El proceso de creación de eventos se divide en varios pasos, cada uno diseñado para facilitar la configuración detallada y personalizada del evento. Comienza con la selección del tipo de evento, tal y como muestra la Figura 6.38, seguido por la introducción de datos generales como el nombre, descripción, idioma, propietario, fechas de inicio y fin, tipo de asistencia, límite de participantes, y configuraciones de acceso como credenciales, notificaciones y etiquetas. Figura 6.39.

¹⁸Elemento que indica pasos a realizar o por los que pasar.

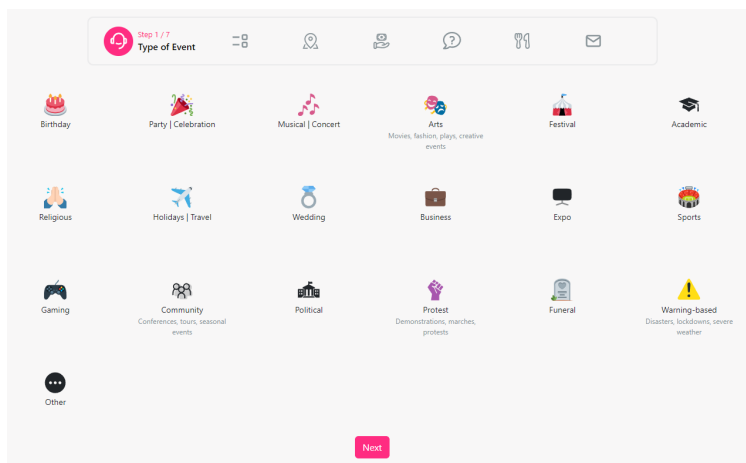


Figura 6.38: Selección de tipo de evento durante la creación de un evento.

Step 2 / 7 General data

Event data

Name*

Description*

Language*

☐ I am not the owner of this event

☐ This is a live event

Event dates

Start date*

End date*

Event attendance

Type of attendance *

Event attendance limit

Leave empty for no limit.

Event date limits

☐ The event has an edit deadline

☐ The event has a cancellation deadline

☐ Event users have an edit deadline

Event access

☐ This event requires credentials

Event notifications

☐ Notifications will be sent

Whenever the event undergoes significant changes, notifications will be sent to all relevant parties. This includes, but is not limited to:

- Event Update: Any updates to the event details, such as changes in the schedule, venue, or description, will trigger a notification to all attendees.
- Event Cancellation: If the event is canceled, immediate notifications will be sent to all registered participants and stakeholders.
- Event Reminder: Reminders will be sent out a day before the event to all participants.
- RSVP Confirmation: Confirmation emails will be sent upon successful RSVP by the participants.

Keep your attendees informed and engaged by enabling these notifications.

Event tags

+

Back Next

Figura 6.39: Formulario de información durante la creación de un evento.

El tercer paso se centra en la localización, ofreciendo vistas específicas dependiendo de si se elige una localización exacta, donde se pueden visualizar datos y fotos del lugar, como se observa en la Figura 6.40, o bien se opta por una dirección alternativa, figuras 6.41 y 6.42.

Step 3 / 7 Location

Event location

Is this event hosted on a venue?*

☐ Yes ☒ No

Address details

Country* State* City*

Street* Number* Postal code*

Other details

Back Next

Figura 6.40: Selección de localización durante la creación de un evento: dirección.

Step 3 / 7 Location

Event location

Is this event hosted on a venue?*

☒ Yes ☐ No

Venue details

Choose a venue*

- Conference Hall
- Expo Center
- Private estate
- Outdoor Park
- Downtown Plaza
- Beachside Resort

Back Next

Figura 6.41: Selección de localización durante la creación de un evento: localización.

Step 3 / 7 Location

Event location

Is this event hosted on a venue?*

☒ Yes ☐ No

Venue details

Choose a venue*

Is the venue not on the list?

Create venue

Back Next

Step 3 / 7 Location

Event location

Is this event hosted on a venue?*

☒ Yes ☐ No

Venue details

Choose a venue*

Conference Hall

A large hall suitable for conferences and events.

1234567890

Main St 123, Los Angeles, CA, 12345, United States.

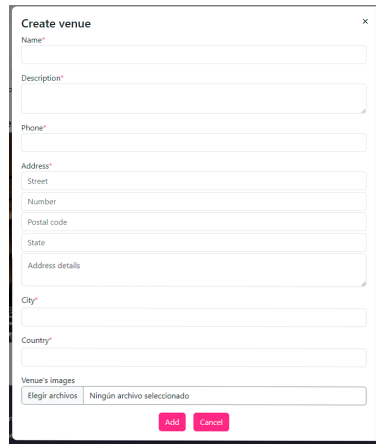
Is the venue not on the list?

Create venue

Back Next

Figura 6.42: Selección de localización durante la creación de un evento: localización.

De no existir una localización entre el listado, se puede crear una nueva, tal y como se muestra en la siguiente figura.



Create venue

Name*

Description*

Phone*

Address*

Street

Number

Postal code

State

Address details

City*

Country*

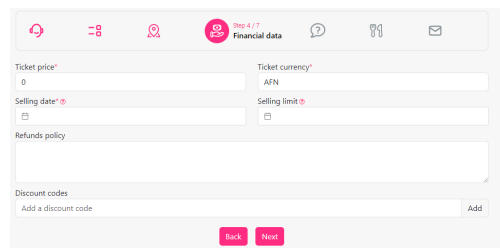
Venue's images

Elegir archivos Ningún archivo seleccionado

Add Cancel

Figura 6.43: Creación de una localización nueva.

En el cuarto paso se manejan los aspectos financieros del evento, como el precio de la entrada, la moneda, fechas de venta, límites de venta, políticas de reembolso y códigos de descuento (figura 6.44).



Step 4 / 7 Financial data

Ticket price* 0

Ticket currency* AFN

Selling date* 25/07/2024 23:59

Selling limit* 25/08/2024 17:00

Refunds policy

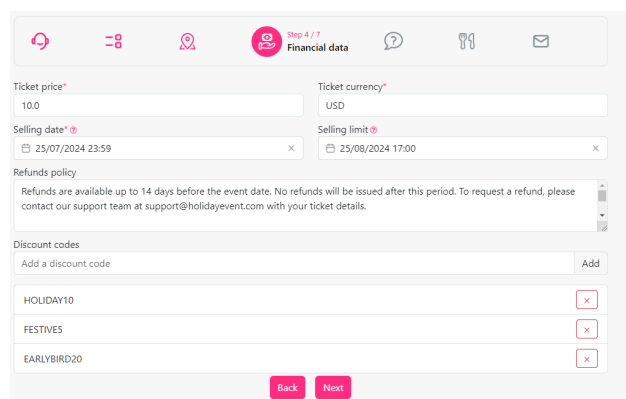
Discount codes

Add a discount code Add

Back Next

Figura 6.44: Formulario financiero en la creación de evento.

Se pueden añadir tantos códigos de descuento como se considere, siendo el resultado final el mostrado en la siguiente figura.



Step 4 / 7 Financial data

Ticket price* 10.0

Ticket currency* USD

Selling date* 25/07/2024 23:59

Selling limit* 25/08/2024 17:00

Refunds policy

Discount codes

Add a discount code Add

HOLIDAY10

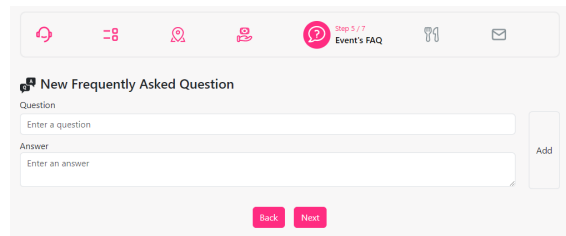
FESTIVES

EARLYBIRD20

Back Next

Figura 6.45: Formulario financiero. Listado de códigos de descuento.

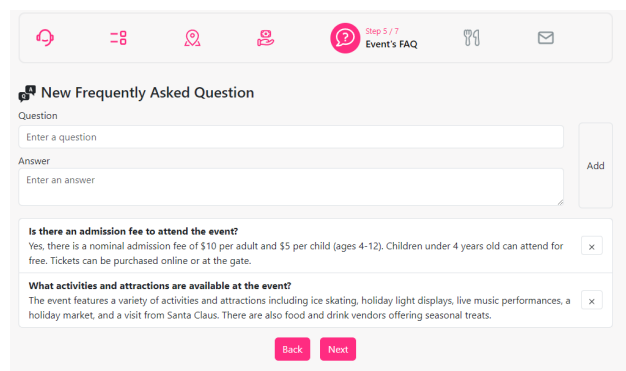
El quinto paso aborda las preguntas frecuentes del evento, proporcionando un espacio para gestionar las consultas más comunes de manera organizada (figura 6.46).



The screenshot shows a web interface for creating an event FAQ. At the top, a navigation bar includes icons for home, menu, location, people, a highlighted 'Event's FAQ' (Step 5 of 7), and other settings. Below the navigation bar, the form is titled 'New Frequently Asked Question'. It contains two text input fields: 'Question' with the placeholder 'Enter a question' and 'Answer' with the placeholder 'Enter an answer'. To the right of the 'Answer' field is an 'Add' button. At the bottom of the form are 'Back' and 'Next' buttons.

Figura 6.46: Formulario de preguntas frecuentes en la creación de evento.

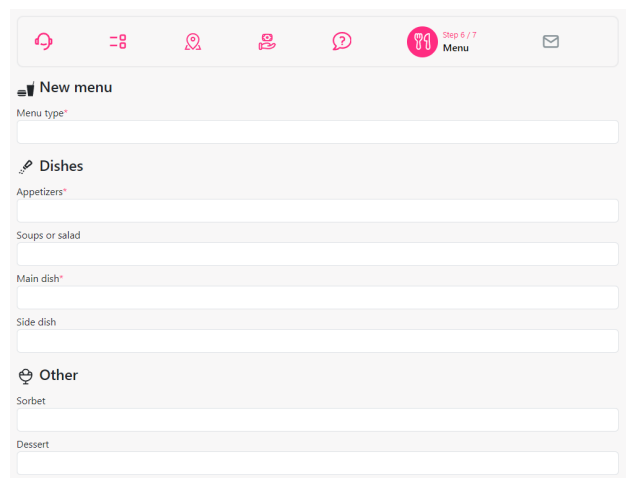
Se pueden añadir tantas preguntas frecuentes como se considere, siendo el resultado final el mostrado en la siguiente figura.



This screenshot shows the same 'New Frequently Asked Question' form as Figure 6.46, but with a list of added questions below the input fields. The first question is 'Is there an admission fee to attend the event?' with the answer 'Yes, there is a nominal admission fee of \$10 per adult and \$5 per child (ages 4-12). Children under 4 years old can attend for free. Tickets can be purchased online or at the gate.' The second question is 'What activities and attractions are available at the event?' with the answer 'The event features a variety of activities and attractions including ice skating, holiday light displays, live music performances, a holiday market, and a visit from Santa Claus. There are also food and drink vendors offering seasonal treats.' Each question and answer pair is enclosed in a box with a close button (x) on the right. The 'Back' and 'Next' buttons are still at the bottom.

Figura 6.47: Formulario de preguntas frecuentes. Listado de preguntas.

En el sexto paso, se configura el menú del evento, permitiendo la adición de platos, bebidas y otra información relevante para la experiencia gastronómica, tal y como puede verse en la figura 6.48.



The screenshot shows a web interface for creating an event menu. The navigation bar at the top is similar to the previous steps, but the highlighted item is 'Menu' (Step 6 of 7). The form is titled 'New menu'. It starts with a 'Menu type*' field. Below this, there is a section for 'Dishes' which includes three sub-sections: 'Appetizers*', 'Soups or salad', and 'Main dish*', each followed by a text input field. There is also a 'Side dish' section with a text input field. Below the 'Dishes' section is an 'Other' section with two sub-sections: 'Sorbet' and 'Dessert', each followed by a text input field.

Beverages

Cocktail

Drink*

Extra

Important information

Menu's warnings

Add

Back Next

Figura 6.48: Formulario de menú durante la creación de un evento.

Se pueden añadir tantos menús como se considere, siendo el resultado final el mostrado en la siguiente figura.

Menu's warnings

Add

Menu type: Formal Dinner
Appetizers: Shrimp Cocktail
Soups or salad: Lobster Bisque
Main dish: Prime Rib with Horseradish Cream
Side dish: Garlic Mashed Potatoes
Sorbet: Lemon Basil Sorbet
Dessert: Chocolate Fondant with Raspberry Coulis
Cocktail: Classic Martini
Drink: Sparkling Water
Important information: Please inform us of any dietary restrictions or allergies at least 48 hours in advance.
Menu's warnings: The menu contains shellfish, dairy, and gluten.

Menu type: Buffet Style
Appetizers: Smoked Salmon Canapés
Soups or salad: Mixed Green Salad with Balsamic Vinaigrette
Main dish: Honey-Glazed Ham
Side dish: Sautéed Green Beans with Almonds
Sorbet: Mango Sorbet
Dessert: Assorted Holiday Cookies
Cocktail: Mimosa
Drink: Freshly Squeezed Orange Juice
Important information: Vegetarian options are available upon request.
Menu's warnings: The menu contains nuts, dairy, and gluten.

Back Next

Figura 6.49: Formulario de menú. Listado de preguntas.

Finalmente, el séptimo paso es la confirmación, donde se solicita un correo electrónico de contacto, se selecciona la plantilla y la paleta de colores que representarán visualmente el evento. La figura 6.50 expone la selección de color.

Event contact

Email*

Event template

Select a template*

Template 1 Template 2 Template 3 Template 4 Template 5

Select a palette*

Back Submit

Figura 6.50: Formulario final en la creación de un evento.

Por último, se implementó una mejora significativa en el módulo de enrutamiento de la aplicación. Se ha agregado una ruta dinámica que asigna un subdominio único a cada evento creado, tal y como puede verse en la figura 6.51. Esta configuración permite que, una vez creado satisfactoriamente un evento, los usuarios puedan acceder directamente a la página dedicada de dicho evento.

```
const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),
  routes: [

    // Otras rutas
    // ...

    // Dynamic route for events
    { path: '/event/:eventType/:eventId/:eventName', component: EventView, meta: { requiresAuth: true } },
  ],

  // Resto de código
  // ..
});
```

Figura 6.51: Rutas de evento dinámicas en el módulo de enrutamiento.

Un ejemplo de ruta de evento puede ser: ‘wedding_1/a-&m-wedding’ tal y como muestra la siguiente figura:

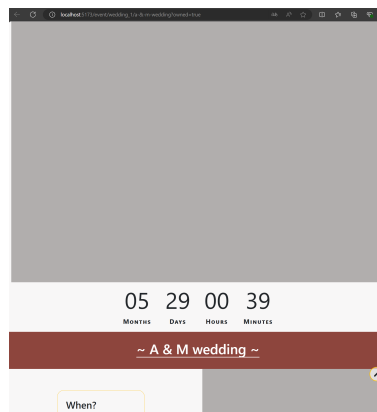


Figura 6.52: Ejemplo de creación de evento.

6.3.7. Sprint 7

Durante el Sprint 7, se enfocó en el desarrollo de tareas relacionadas con la gestión de usuarios y eventos dentro del sistema. Se implementaron las siguientes historias de usuario:

Id	Nombre	Estimación	Prioridad
HU5	Gestionar eventos existentes	15 horas	Alta
HU6	Gestionar usuarios asociados a un evento	10 horas	Media
HU11	Edición de datos de usuarios	1 hora	Media
HU14	Modificar el estado de un evento	1 hora	Media
HU18	Asociar roles a usuarios asociados al evento	3 horas	Alta
Estimación total		30 horas	

En primer lugar, se realizó una modificación en el proceso de creación de eventos, donde ahora, al crear un nuevo evento, los usuarios son redirigidos directamente al *dashboard*¹⁹ que muestra todos sus eventos existentes. Además, simultáneamente se abre una nueva pestaña que lleva al usuario a la página del evento recién creado. Esta actualización ha simplificado y agilizado el acceso a la gestión global de eventos, proporcionando una experiencia más fluida y organizada para los usuarios.

Es importante mencionar que las imágenes mostradas en esta sección corresponden a un evento creado para ilustrar los módulos y componentes introducidos. Estas imágenes permiten visualizar de manera práctica y detallada las nuevas funcionalidades implementadas.

Adicionalmente, se introdujeron nuevos componentes clave diseñados para mejorar la gestión detallada de eventos y usuarios asociados. El primero de ellos es un componente dedicado a la edición de datos de un evento, permitiendo a los administradores y organizadores actualizar información crucial como fechas, descripciones y configuraciones específicas del evento. Este componente se accede desde un elemento *sticky*²⁰ que se mantiene visible en la vista del evento, moviéndose según el usuario se desplaza hacia arriba o abajo en la página. Esto asegura un acceso rápido y constante a las opciones de edición, facilitando una gestión eficiente y sin interrupciones del evento.

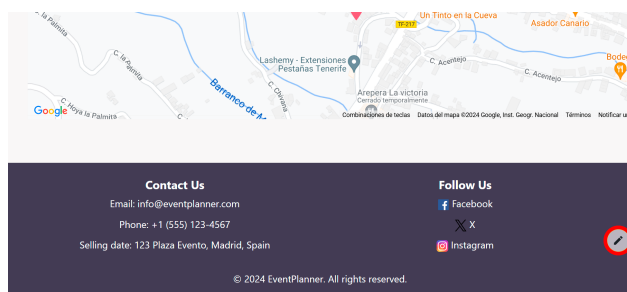


Figura 6.53: Componente para la edición de un evento.

Otro componente fundamental desarrollado fue el diseñado para la gestión de usuarios asociados a un evento. Este componente permite gestionar de manera centralizada los usuarios

¹⁹Herramienta de gestión de la información.

²⁰Híbrido de los posicionamientos relativo y fijo, donde el elemento se mantiene fijo en la pantalla pero relativo a su contenedor principal o a la ventana del navegador.

que están involucrados en un evento específico. Los administradores pueden asignar roles personalizados, gestionar permisos y realizar ajustes en la información de los usuarios según sea necesario.

Este componente fue implementado y demostrado utilizando datos y ejemplos reales de un evento modelo, asegurando que las nuevas funcionalidades se integraran de manera efectiva y coherente dentro del sistema global de gestión de eventos. Estas mejoras están orientadas a optimizar la administración y personalización de eventos, mejorando así la eficiencia y la experiencia del usuario final.



El screenshot muestra la interfaz 'Gestión de usuarios' con una barra de herramientas superior que incluye botones para 'Copiar', 'Excel', 'Imprimir' y un botón 'Filtrar'. Debajo de la barra, hay un campo de búsqueda y un selector de 'Mostrar: 10 registros por página'. La tabla principal contiene los siguientes datos:

Nombre	Apellido	Asiste	Menú	Alergias	Cancelaciones	Rol	Administrar
Adri						Admin	☒
Adri						User	☒
Airam						User	☒
Alberon						User	☒
Alejandro						User	☒
Alex						User	☒
Alex						User	☒

Figura 6.54: Gestión de usuarios en un evento.

Por último, se añadió un componente que muestra el mapa de la localización del evento. Este componente proporciona una visualización clara y precisa de la ubicación seleccionada a la hora de crear el evento, mejorando la experiencia del usuario al facilitar la identificación del lugar del evento. En la siguiente figura se muestra un ejemplo de la localización de un evento prueba.

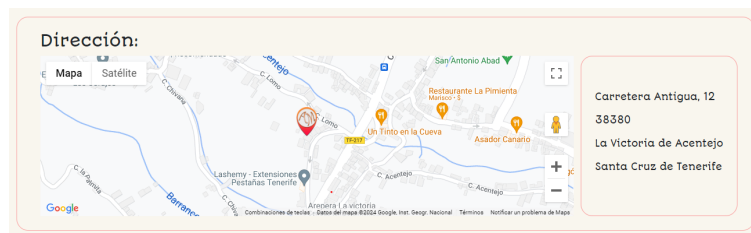


Figura 6.55: Mapa con la ubicación de un evento.

6.3.8. Sprint 8

En el Sprint 8 se abordaron varias historias de usuario clave enfocadas en la creación de *templates* (plantilla) y paletas de colores para la personalización de los eventos, así como en la gestión de estadísticas de los mismos. Además, se llevaron a cabo pruebas de calidad y rendimiento utilizando herramientas como SonarQube y Lighthouse.

Id	Nombre	Estimación	Prioridad
HT9	Análisis de accesibilidad y usabilidad	2 horas	Media
HU4	Seleccionar la plantilla del evento	10 horas	Media
HU15	Filtro de eventos	1 horas	Alta
HU16	Control de estadísticas del evento	2 horas	Baja
HU23	Gestión de asistencia a un evento	5 horas	Media
Estimación total		20 horas	

Tabla 6.9: Planificación Sprint 8.

Durante este sprint, se desarrolló un componente específico para facilitar la selección de colores, proporcionando a los usuarios una herramienta intuitiva para personalizar la apariencia de sus eventos. Este componente de selección de colores permite una personalización detallada y adaptable a las necesidades específicas de los organizadores de eventos.

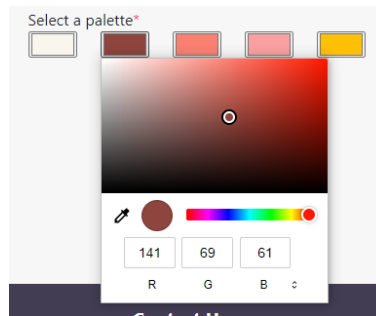


Figura 6.56: Componente de selección de color.

Además, se crearon un total de cinco plantillas diferentes entre las cuales los usuarios pueden elegir para personalizar la apariencia del evento. Estas plantillas ofrecen flexibilidad y permiten a los organizadores adaptar la presentación visual del evento según sus preferencias y necesidades específicas. A continuación, se presentan algunas de las plantillas disponibles, destacando la variedad y personalización que se pueden lograr con ellas.



Figura 6.57: Plantillas para la personalización de la apariencia de un evento.

También se añadieron componentes para el filtrado de eventos, mejorando significativamente la experiencia de búsqueda y gestión. El componente de búsqueda de eventos permite a los usuarios filtrar eventos por nombre, tipo y año, facilitando el acceso a la información relevante y agilizando la administración de múltiples eventos.



The screenshot shows a search form titled "Search for an event" with a red asterisk and the instruction "Select at least one search field to search for an event". It contains three input fields: "Event name", "Year" (with "YYYY" as a placeholder), and "Event type". A "Search" button is located below the fields. At the bottom, it says "No results found".

Figura 6.58: Filtros de evento.



This screenshot shows the same search form, but with the "Year" field filled with "2025". Below the form, a "Search results" section displays a result for a "Gaming Tournament". The result includes the description "A competitive gaming event.", the "Start date: 16/8/2025", and the "Event type: GAMING".

Figura 6.59: Filtros de evento con resultados.

En el componente de gestión de eventos, se añadieron varios elementos estadísticos esenciales para los organizadores, como un gráfico de asistencia creado utilizando la librería Chart.js y sus componentes de vue-chartjs, y una biblioteca de *JavaScript* utilizada para crear gráficos interactivos y personalizables. Este gráfico proporciona una visualización clara y concisa de la participación de los usuarios en el evento, facilitando el seguimiento y análisis de la asistencia. Además, se implementaron otras visualizaciones estadísticas que permiten a los organizadores monitorizar diversos aspectos del evento en tiempo real, tomando decisiones informadas basadas en los datos recopilados, como por ejemplo, número de usuarios participantes y número de visualizaciones totales de un evento.

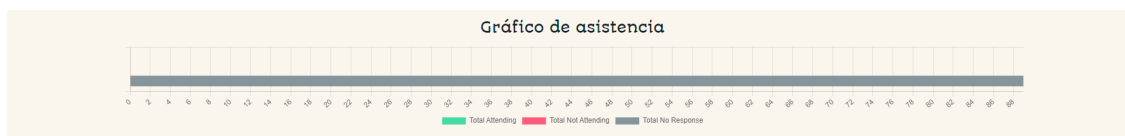


Figura 6.60: Gráfico de asistencia de un evento.

Dado que “la prueba de validación proporciona la garantía final de que el software cumple con todos los requerimientos informativos, funcionales, de comportamiento y de rendimiento”(Pressman 2010, pp.387)[21], se realizaron pruebas usando las herramientas de SonarQube y Lighthouse para garantizar la accesibilidad, uso y rendimiento de la aplicación. Estas aseguran que la plataforma cumple con los estándares de calidad y ofrece una experiencia óptima a los usuarios.

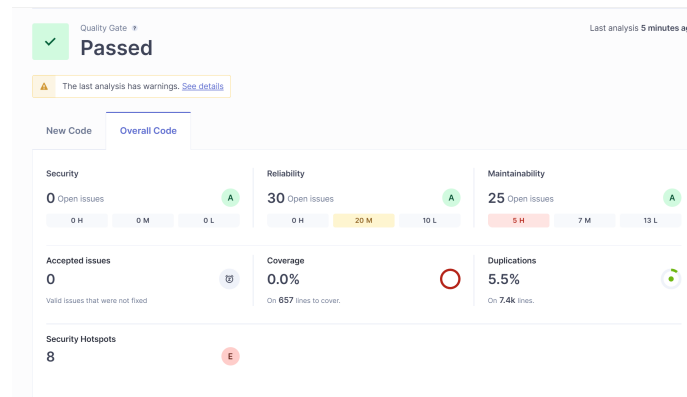


Figura 6.61: Informe inicial de SonarQube.

Para mejorar los resultados obtenidos en SonarQube, se realizaron varios cambios significativos en el código base. Estos cambios incluyeron la refactorización de funciones complejas para mejorar su legibilidad y mantenimiento, la corrección de errores de sintaxis y lógica, y la optimización de la gestión de recursos para reducir posibles puntos de falla.

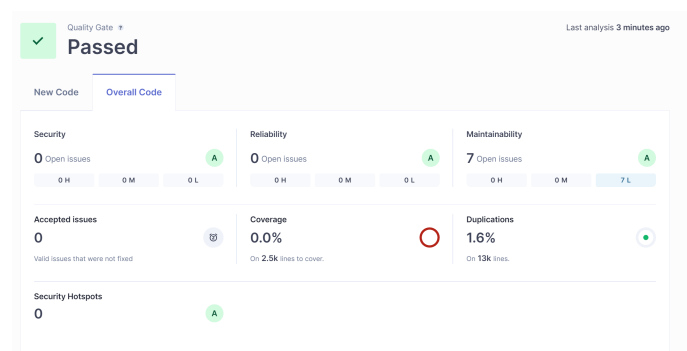
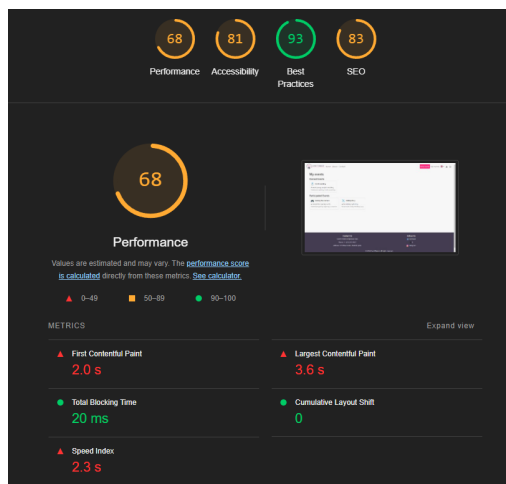
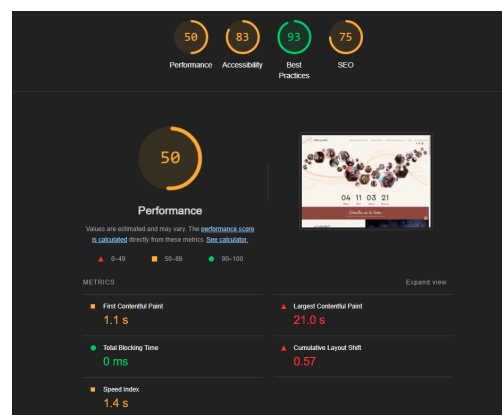


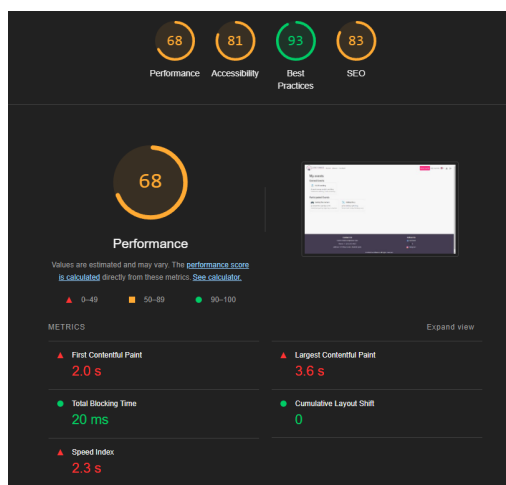
Figura 6.62: Informe final de SonarQube.



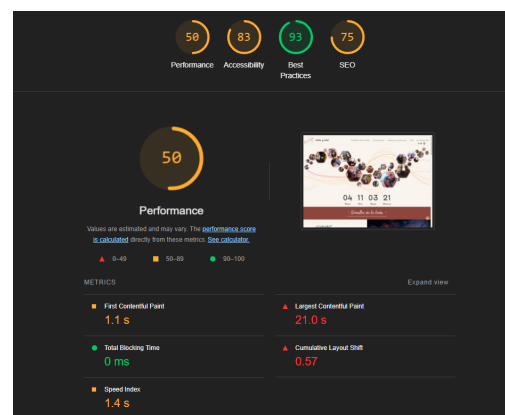
(a) Resultado de análisis: panel de eventos



(b) Resultado de análisis: evento prueba



(c) Resultado de análisis: panel de eventos



(d) Resultado de análisis: evento prueba

Figura 6.63: Resultados de la aplicación tras los análisis con Lighthouse.

Capítulo 7

Conclusiones y trabajo futuro

El proyecto asociado a este Trabajo de Fin de Grado ha formalizado actividades relacionadas con el desarrollo de una plataforma para la gestión de eventos, contribuyendo con un producto eficiente y coherente con las demandas del mercado actual.

Este documento describe dichas actividades y su implementación mostrando las características y potencialidades del prototipo. La finalización del proyecto enfatiza la necesidad de conocimientos previos sobre una variedad de factores del mercado y potenciales usuarios, así como componentes a nivel técnico y competencial, tanto en la etapa inicial de recopilación de información; como en la etapa media de estudio de viabilidad y planificación; en la etapa de diseño y codificación propiamente dicha de un producto a medida con entregas específicas, hasta la etapa de prueba, integración e implementación, para finalmente ocuparse del seguimiento, actualización y mantenimiento. A su vez, llevar a cabo dicha tarea con éxito resulta beneficiada por el cúmulo de experiencias y competencias técnicas disponibles, elevada motivación de logro, entrenamiento técnico y estilo de gestión que ha permitido adaptarse a las circunstancias, a obtener soluciones ante el desafío que implica el diseño y creación de software.

Se ha efectuado un análisis de las herramientas y tecnologías disponibles y se ha justificado la elección con ajuste a las características del proyecto, concluyendo que la adaptabilidad, creatividad y calidad a través de la metodología ágil y de escabilidad, han constituido un factor decisivo. De hecho, el desarrollo de módulos no solo permite a los usuarios diversas funcionalidades como las descritas a lo largo del presente proyecto, sino garantiza la viabilidad de modificaciones de acuerdo a necesidades específicas de usuarios y del mercado, es decir, complejidad y actualización.

De los análisis de la plataforma efectuadas en contextos reales, se concluye un resultado positivo en el desempeño de sus funcionalidades, además de los resultados obtenidos a través del desarrollo y aplicación de pruebas de SonarQube y Lighthouse, señalando que cumple con los estándares de calidad y ofrece una experiencia óptima a los usuarios. En otras palabras, se han cumplido los objetivos planteados en este proyecto.

En conclusión, este trabajo supone una mejora competitiva tanto en el coste como en el

producto que se ofrece, al mismo tiempo, en la transacción que clientes y empresas asociadas pueden realizar al acceder y utilizar la plataforma. Destaca el valor que genera en la solución final y las facilidades y posibilidades que brinda a los usuarios del sistema. A su vez, ha supuesto una oportunidad para poder continuar aplicando y/o fortalecer competencias relativas al presente Trabajo de Fin de Grado.

Sin embargo, una combinación de propuestas de mejora complementaría la aproximación práctica hasta aquí expuesta. Así, en el futuro se puede enriquecer el catálogo incluyendo el desarrollo de otros campos de personalización, como listas de reproducción de canciones, confirmación de asistencia, solicitud de necesidades especiales (transporte asistido, intérprete de lenguaje de signos, por mencionar algunas), venta de entradas a través de portales de pago, entre otras.

Otra mejora consistiría en planificar e implementar un período limitado de prueba gratuita, capaz de soportar varios formatos de eventos de marketing como testeo, ya que, como afirma Pressman (2010), no se termina de probar por completo un software, es decir “nunca se termina de probar; la carga simplemente pasa de usted (el ingeniero de software) al usuario final” pp.388 [21].

También, incluir elementos de fácil acceso y navegación para personas con diversidad funcional, como reconocimiento de voz (*voice management*) en el compromiso con una sociedad accesible, equitativa e inclusiva.

La adopción de herramientas y aplicaciones emergentes como una forma de mantener la plataforma actualizada constituye otra propuesta de mejora.

Finalmente, desarrollar un plan de seguridad que incluya estrategias que consideren al menos un listado de posibles ataques, los requisitos bajo los cuales pueden ser ejecutados y las pruebas que puedan replicarlos, con la finalidad de mejorar posibles debilidades, como podrían ser proyectos de código abierto.

Si tuviese que expresar una valoración más personal, como estudiante, sin duda destacaría la comprensión de la enorme complejidad y desafíos inherentes al proceso de desarrollo de software que he tenido oportunidad de experimentar en el desarrollo del presente proyecto de Trabajo de Fin de Grado. El análisis, la creatividad, la integración de conocimientos teóricos y prácticos y la búsqueda constante de soluciones técnicas a lo largo de todas las fases ya descritas. Por tanto, un fortalecimiento de las competencias técnicas y habilidades de resolución de problemas, esenciales para enfrentar desafíos presentes y futuros en el campo de la Ingeniería Informática.

Anexo I

Pila de Producto

Historias técnicas

Evaluación de tecnologías	
Código	HT1
Historia Técnica	Como desarrollador, necesito investigar y evaluar diversas tecnologías para seleccionar las más adecuadas que cumplan con los requisitos del proyecto.
Prioridad	Alta
Criterios de Validación	- Se debe realizar un análisis comparativo de las tecnologías basado en criterios como rendimiento, escalabilidad, facilidad de mantenimiento y soporte comunitario. - Se debe seleccionar y justificar las tecnologías elegidas en función de los requisitos del proyecto y las capacidades del equipo.
Estimación	30 horas

Tabla 7.1: Historia técnica HT1

Análisis competitivo de aplicaciones similares	
Código	HT2
Historia Técnica	Como desarrollador, necesito estudiar detalladamente aplicaciones existentes para comprender sus características, funcionalidades, ventajas y limitaciones con el objetivo de diferenciar el prototipo.
Prioridad	Media
Criterios de Validación	- Se debe identificar elementos diferenciadores mediante características únicas y mejoras significativas. - Se debe documentar el resultado del análisis comparativo para guiar las decisiones de diseño y desarrollo.
Estimación	20 horas

Tabla 7.2: Historia técnica HT2

Definición de requisitos	
Código	HT3
Historia Técnica	Como desarrollador, necesito identificar los requisitos específicos del sistema, tanto funcionales como no funcionales.
Prioridad	Alta
Criterios de Validación	- Se debe elaborar un documento detallado que liste los requisitos funcionales y no funcionales del sistema.
Estimación	10 horas

Tabla 7.3: Historia técnica HT3

Análisis arquitectónico de la aplicación	
Código	HT4
Historia Técnica	Como desarrollador, necesito investigar y analizar diferentes diseños arquitectónicos para seleccionar el más adecuado que cumpla con las necesidades del proyecto.
Prioridad	Alta
Criterios de Validación	- Se debe realizar un análisis detallado de las arquitecturas consideradas, destacando ventajas, desventajas y escalabilidad. - Se debe seleccionar una arquitectura que garantice la eficiencia y la escalabilidad.
Estimación	10 horas

Tabla 7.4: Historia técnica HT4

Diseño del modelo de base de datos	
Código	HT5
Historia Técnica	Como desarrollador, necesito diseñar un modelo de bases de datos que soporte eficientemente las necesidades de almacenamiento y consulta del sistema.
Prioridad	Alta
Criterios de Validación	- Se debe crear un esquema de base de datos completo. - El diseño debe optimizar la velocidad de acceso y mantener la integridad de los datos.
Estimación	15 horas

Tabla 7.5: Historia técnica HT5

Investigación de técnicas de autenticación y seguridad	
Código	HT6
Historia Técnica	Como desarrollador, necesito investigar y planificar la implementación de medidas de seguridad robustas como JWT (JSON Web Tokens), OAuth2 u otros métodos de autenticación y autorización para proteger la aplicación.
Prioridad	Alta
Criterios de Validación	- Se debe documentar un plan de seguridad que cubra la gestión de usuarios, roles, sesiones seguras y protección contra vulnerabilidades conocidas. - La implementación debe cumplir con los estándares de seguridad aceptados.
Estimación	20 horas

Tabla 7.6: Historia técnica HT6

Gestión de roles y permisos de usuario	
Código	HT7
Historia Técnica	Como desarrollador, necesito definir roles de usuario y permisos de acceso para garantizar la seguridad y la integridad de los datos según el rol de cada usuario.
Prioridad	Media
Criterios de Validación	- Se debe crear una matriz de permisos que especifique las acciones permitidas para cada tipo de usuario (administrador, organizador de eventos, participante, etc.). - Los permisos deben ser configurados y probados para asegurar su efectividad y consistencia en diferentes escenarios de uso.
Estimación	5 horas

Tabla 7.7: Historia técnica HT7

Diseño de la interfaz de usuario	
Código	HT8
Historia Técnica	Como desarrollador, necesito crear mockups que representen el diseño visual del proyecto, asegurando una experiencia de usuario intuitiva y eficiente.
Prioridad	Media
Criterios de Validación	- Se deben generar mockups para las vistas y funciones principales del sistema, basados en las necesidades y expectativas de los usuarios finales. - El diseño debe ser validado mediante pruebas de usabilidad y la retroalimentación de los usuarios para garantizar una interfaz efectiva y atractiva.
Estimación	10 horas

Tabla 7.8: Historia técnica HT8

Análisis de accesibilidad y usabilidad	
Código	HT9
Historia Técnica	Como desarrollador, necesito estudiar las mejores prácticas de accesibilidad para crear una interfaz de usuario inclusiva y fácil de usar.
Prioridad	Media
Criterios de Validación	- Se debe realizar un análisis de interfaces para identificar elementos que mejoren la accesibilidad del sistema. - Las recomendaciones de accesibilidad deben ser implementadas para asegurar que el sistema sea accesible para todos los usuarios.
Estimación	2 horas

Tabla 7.9: Historia técnica HT9

Diseño de diagrama de flujo	
Código	HT10
Historia Técnica	Como desarrollador, necesito diseñar un diagrama de flujo que represente visualmente los procesos y la lógica del sistema, facilitando la comprensión y la implementación del proyecto.
Prioridad	Media
Criterios de Validación	- Se debe crear un diagrama de flujo detallado que incluya todos los procesos principales y subprocesos del sistema. - El diagrama debe ser actualizado conforme se realicen cambios en los requisitos o en la lógica del sistema.
Estimación	5 horas

Tabla 7.10: Historia técnica HT10

Configuración del entorno de desarrollo	
Código	HT11
Historia Técnica	Como desarrollador, necesito configurar el entorno de desarrollo utilizando las tecnologías seleccionadas.
Prioridad	Alta
Criterios de Validación	- Se debe configurar el despliegue automático de la aplicación. - El entorno de desarrollo debe permitir la ejecución del frontend y backend de manera simultánea. - La configuración debe incluir un proceso claro y reproducible para la instalación y ejecución del entorno de desarrollo. - Se debe asegurar que la comunicación entre el frontend y el backend funcione correctamente en el entorno configurado.
Estimación	25 horas

Tabla 7.11: Historia técnica HT11

Historias de usuario

Registro de un usuario	
Código	HU1
Historia de Usuario	Como nuevo usuario, quiero poder registrarme en la plataforma para crear y gestionar eventos.
Prioridad	Alta
Criterios de Validación	- Se debe proporcionar un formulario de registro donde los usuarios puedan ingresar su nombre, correo electrónico, contraseña y otros datos relevantes. - La plataforma debe asegurar que los datos ingresados son válidos y que el correo electrónico no esté ya registrado.
Estimación	4 horas

Tabla 7.12: Historia de usuario HU1

Inicio de sesión	
Código	HU2
Historia de Usuario	Como usuario registrado, quiero poder iniciar sesión en la plataforma para acceder a mis eventos y configuraciones.
Prioridad	Alta
Criterios de Validación	- Se debe proporcionar un formulario de inicio de sesión donde los usuarios puedan ingresar su correo electrónico y contraseña. - El sistema debe validar las credenciales y permitir el acceso a la cuenta si son correctas. - Se debe proporcionar una opción para restablecer la contraseña en caso de que el usuario la haya olvidado.
Estimación	4 horas

Tabla 7.13: Historia de usuario HU2

Crear un evento	
Código	HU3
Historia de Usuario	Como usuario registrado, quiero poder crear un evento en la plataforma para gestionar todos los aspectos relacionados con la planificación y organización del evento.
Prioridad	Alta
Criterios de Validación	- Se debe proporcionar un formulario para ingresar detalles del evento, como nombre, fecha, ubicación y tipo, entre otros. - La plataforma debe permitir la selección de módulos personalizables para adaptarse a diferentes tipos de eventos.
Estimación	14 horas

Tabla 7.14: Historia de usuario HU3

Seleccionar la plantilla del evento	
Código	HU4
Historia de Usuario	Como usuario que crea un evento, quiero poder seleccionar una plantilla para el evento que determine cómo se verá visualmente la página del evento, para personalizar la apariencia según mis necesidades.
Prioridad	Media
Criterios de Validación	- Durante el proceso de creación del evento, debe haber una opción para seleccionar entre varias plantillas de diseño pre-definidas. - Los usuarios deben poder previsualizar cada plantilla antes de seleccionarla para ver cómo se verá el evento. - Una vez seleccionada, la plantilla debe aplicarse a la página del evento y ser visible para todos los usuarios asociados al evento
Estimación	10 horas

Tabla 7.15: Historia de usuario HU4

Gestionar eventos existentes	
Código	HU5
Historia de Usuario	Como usuario registrado, quiero poder acceder a la plataforma para gestionar los eventos existentes, ya sea para revisarlos o modificarlos.
Prioridad	Alta
Criterios de Validación	- Se debe proporcionar una página de inicio de sesión para que los usuarios registrados accedan a sus eventos. - Los usuarios deben poder ver una lista de eventos creados por ellos o en los que están involucrados como colaboradores. - Se debe proporcionar una interfaz fácil de usar para editar los detalles del evento, como nombre, fecha, ubicación, tipo, etc.
Estimación	15 horas

Tabla 7.16: Historia de usuario HU5

Gestionar usuarios asociados a un evento	
Código	HU6
Historia de Usuario	Como administrador de un evento, quiero poder gestionar los usuarios asociados al evento.
Prioridad	Media
Criterios de Validación	- Los administradores deben poder ver la lista completa de usuarios asociados al evento. - Debe haber opciones claras y accesibles para editar y eliminar usuarios asociados al evento. - Debe ser posible cambiar los roles de los usuarios. - Los datos editables de los usuarios deben incluir al menos su nombre, menú (si aplica), lista de música (si aplica), etc. - Las acciones de gestión deben ser intuitivas y seguras, garantizando que solo los administradores puedan realizar cambios significativos en los usuarios asociados al evento.
Estimación	10 horas

Tabla 7.17: Historia de usuario HU6

Personalizar módulos del evento	
Código	HU7
Historia de Usuario	Como gestor de un evento, quiero poder personalizar los módulos del evento, seleccionando los elementos específicos que se mostrarán en la página del evento.
Prioridad	Alta
Criterios de Validación	- Los gestores deben poder seleccionar los módulos relevantes para su evento, como lista de canciones, lista de preguntas frecuentes, menú, etc. - Debe haber opciones para editar y actualizar los módulos seleccionados durante la planificación del evento.
Estimación	4 horas

Tabla 7.18: Historia de usuario HU7

Establecer fecha límite de edición del evento	
Código	HU8
Historia de Usuario	Como gestor de un evento, quiero poder establecer una fecha límite de edición, después de la cual los detalles del evento no puedan ser modificados.
Prioridad	Baja
Criterios de Validación	- Los gestores deben poder establecer una fecha límite de edición para su evento durante el proceso de planificación. - Los gestores deben poder editar la fecha límite de edición del evento. - Después de la fecha límite, los usuarios no podrán editar los detalles del evento ni de sus usuarios.
Estimación	1 hora

Tabla 7.19: Historia de usuario HU8

Establecer un propietario distinto del creador del evento	
Código	HU9
Historia de Usuario	Como creador de un evento, quiero poder asignar un propietario distinto para el evento para delegar responsabilidades de gestión.
Prioridad	Media
Criterios de Validación	- Los creadores de eventos deben tener la opción de asignar un propietario diferente durante o después del proceso de creación del evento. - La plataforma debe permitir seleccionar al nuevo propietario de una lista de usuarios registrados. - Se debe notificar al nuevo propietario sobre su asignación al evento.
Estimación	1 hora

Tabla 7.20: Historia de usuario HU9

Establecer datos financieros para el evento	
Código	HU10
Historia de Usuario	Como gestor de un evento, quiero poder establecer datos financieros opcionales, como precio de entrada, fecha de venta y límite de venta, para gestionar la venta de entradas.
Prioridad	Media
Criterios de Validación	- Debe haber opciones para agregar el precio de la entrada, la fecha de inicio de la venta y el límite de venta durante la configuración del evento. - Los datos de pago deben ser editables antes de la fecha de venta establecida. - La plataforma debe mostrar claramente la información de pago en la página del evento.
Estimación	1 hora

Tabla 7.21: Historia de usuario HU10

Edición de datos de usuarios	
Código	HU11
Historia de Usuario	Como gestor de un evento, quiero permitir que los usuarios participantes editen ciertos datos, como menú o preferencias, hasta una fecha límite específica.
Prioridad	Media
Criterios de Validación	- Los gestores deben poder establecer qué datos pueden ser editados por los usuarios y hasta qué fecha. - La plataforma debe permitir a los usuarios editar los datos especificados hasta la fecha límite. - Después de la fecha límite, los usuarios no podrán realizar cambios en los datos especificados.
Estimación	1 hora

Tabla 7.22: Historia de usuario HU11

Asignar fechas adicionales a un evento	
Código	HU12
Historia de Usuario	Como gestor de un evento, quiero poder asignar fechas adicionales, para mayor flexibilidad en la planificación.
Prioridad	Baja
Criterios de Validación	- Los gestores deben poder agregar fechas adicionales durante la configuración del evento. - Las fechas adicionales deben estar claramente especificadas y ser editables antes de su cumplimiento. - La plataforma debe mostrar todas las fechas relevantes en la página del evento.
Estimación	1 hora

Tabla 7.23: Historia de usuario HU12

Añadir códigos de descuento	
Código	HU13
Historia de Usuario	Como gestor de un evento, quiero poder añadir códigos de descuento que sean aplicables dentro de una fecha específica o de manera permanente para incentivar la venta de entradas.
Prioridad	Baja
Criterios de Validación	- Los gestores deben poder crear y asignar códigos de descuento con condiciones específicas, como fechas de validez y porcentaje de descuento. - Los códigos de descuento deben ser aplicables durante el proceso de compra de entradas. - La plataforma debe validar los códigos de descuento y mostrar los descuentos aplicados correctamente.
Estimación	1 hora

Tabla 7.24: Historia de usuario HU13

Modificar el estado de un evento	
Código	HU14
Historia de Usuario	Como gestor de un evento, quiero poder modificar el estado del evento para mantener a los asistentes informados. Además, el sistema debe poder cambiar automáticamente el estado del evento según ciertas condiciones predefinidas.
Prioridad	Media
Criterios de Validación	- Los gestores deben tener la opción de cambiar manualmente el estado del evento en cualquier momento. - El sistema debe poder cambiar automáticamente el estado del evento a “en curso” cuando la fecha de inicio del evento sea alcanzada y a “finalizado” cuando la fecha de finalización del evento sea alcanzada. - Los cambios en el estado del evento, tanto manuales como automáticos, deben ser notificados a todos los usuarios asociados al evento. - La plataforma debe mostrar claramente el estado actual del evento en la página del evento.
Estimación	1 hora

Tabla 7.25: Historia de usuario HU14

Filtro de eventos	
Código	HU15
Historia de Usuario	Como usuario, quiero poder filtrar la lista de eventos para encontrar eventos concretos de forma sencilla.
Prioridad	Alta
Criterios de Validación	- Se debe permitir filtrar los eventos por su visibilidad, mostrando solo aquellos marcados como públicos por su creador. - Los filtros deben incluir opciones para buscar por año, nombre del evento y tipo de evento. - Al aplicar un filtro, se debe mostrar una lista actualizada de eventos que cumplan con los criterios especificados. - La interfaz de usuario debe ser intuitiva y permitir la combinación de múltiples filtros. - Los resultados de la búsqueda deben cargarse rápidamente y mostrar información relevante de cada evento.
Estimación	1 horas

Tabla 7.26: Historia de usuario HU15

Control de estadísticas del evento	
Código	HU16
Historia de Usuario	Como gestor de un evento, quiero tener acceso a estadísticas del evento, para evaluar el éxito del evento.
Prioridad	Baja
Criterios de Validación	- La plataforma debe proporcionar un panel de estadísticas donde se muestren datos como número de visualizaciones, asistentes confirmados, entradas vendidas, entre otros. - Las estadísticas deben ser actualizadas en tiempo real y ser exportables para análisis posterior.
Estimación	2 horas

Tabla 7.27: Historia de usuario HU16

Establecer eventos como públicos o privados	
Código	HU17
Historia de Usuario	Como gestor de un evento, quiero poder establecer si mi evento será público o privado.
Prioridad	Alta
Criterios de Validación	- Debe haber una opción durante la creación del evento para especificar si será público o privado. - Si el evento es privado, se deben proporcionar herramientas para invitar a usuarios específicos asociados al evento.
Estimación	1 hora

Tabla 7.28: Historia de usuario HU17

Asociar roles a usuarios asociados al evento	
Código	HU18
Historia de Usuario	Como administrador de un evento, quiero poder asignar roles específicos a los usuarios asociados al evento.
Prioridad	Alta
Criterios de Validación	- Los administradores deben poder asignar roles como gestor, organizador, colaborador o usuario normal a los usuarios asociados al evento. - Los usuarios deben tener permisos y restricciones según el rol asignado.
Estimación	3 horas

Tabla 7.29: Historia de usuario HU18

Seleccionar elementos específicos dentro de un evento	
Código	HU19
Historia de Usuario	Como usuario de un evento, quiero poder seleccionar elementos específicos dentro del evento, como menú, lista de canciones, etc.
Prioridad	Baja
Criterios de Validación	- Debe haber opciones para que los usuarios seleccionen elementos específicos dentro del evento, según los módulos configurados por el gestor. - Los usuarios deben poder ver las opciones disponibles y hacer sus selecciones de forma fácil y clara.
Estimación	1 hora

Tabla 7.30: Historia de usuario HU19

Establecer fechas de lanzamiento del evento	
Código	HU20
Historia de Usuario	Como gestor de un evento, quiero poder establecer una fecha de lanzamiento inicial y una fecha de lanzamiento final para mi evento.
Prioridad	Alta
Criterios de Validación	- Debe haber opciones para especificar tanto la fecha de lanzamiento inicial como la fecha de lanzamiento final. - La plataforma debe mostrar claramente estas fechas en la página del evento para que los invitados estén informados.
Estimación	1 hora

Tabla 7.31: Historia de usuario HU20

Asignar una ubicación al evento	
Código	HU21
Historia de Usuario	Como gestor de un evento, quiero poder asignar una ubicación al evento para que los invitados puedan acceder fácilmente a la información.
Prioridad	Media
Criterios de Validación	- Debe haber opciones para agregar y editar la información de la ubicación del evento, incluyendo detalles como nombre, dirección, información de contacto, fotos, etc. - Los usuarios deben poder ver la ubicación del evento en un mapa integrado en la página del evento.
Estimación	1 hora

Tabla 7.32: Historia de usuario HU21

Gestión de usuarios	
Código	HU22
Historia de Usuario	Como administrador de la plataforma, quiero poder gestionar los usuarios registrados, sus datos y permisos.
Prioridad	Baja
Criterios de Validación	- Los administradores deben tener acceso a una interfaz de administración donde puedan ver y editar la información de los usuarios registrados. - Deben poder realizar acciones como editar perfiles de usuario, restablecer contraseñas, revocar permisos, editar eventos, etc.
Estimación	12 horas

Tabla 7.33: Historia de usuario HU22

Anexo II

Glosario de acrónimos

- **API:** *Application Programming Interface* o Interfaz de Programación de Aplicaciones.
- **API RESTful:** Interfaz de programación de aplicaciones que utiliza el protocolo HTTP.
- **CI/CD:** *Continuous integration and continuous delivery* o Integración y distribución continua.
- **CSS:** *Cascading Style Sheets* u Hojas de estilo en cascada.
- **DBML:** *Database Markup Language* o Lenguaje de marcado de bases de datos.
- **DOM:** *Document Object Model* o Modelo de objeto de documento.
- **DSL:** *Digital Subscriber Line* o Línea de suscriptor digital.
- **DTO:** *Data Transfer Object* u Objeto de transferencia de datos.
- **ERD:** *Entity relationship diagram* o Diagrama de entidad-relación.
- **ES:** *Expert system* o Sistema experto.
- **ESBuild:** *Extremely fast bundler* o Agrupador extremadamente rápido.
- **ESLint:** *Static code analysis* o Análisis de código estático.
- **FAQ:** *Frequently asked question* o Preguntas frecuentes.
- **GET:** Solicitud de obtención de información de una fuente.
- **HMR:** *Hot Module Replacement* o Reemplazo de Módulo en Caliente.
- **HTML:** *HyperText Markup Language* o Lenguaje de Marcado de Hipertexto.
- **HTTP:** *Hypertext Transfer Protocol* o Protocolo de Transferencia de Hipertexto.
- **HTTPS:** *Hypertext Transfer Protocol Secure* o Protocolo de Transferencia de Hipertexto Seguro.
- **HT:** Historia técnica.

- **HU:** Historia de usuario.
- **JSON:** *JavaScript Object Notation* o Notación de objetos JavaScript.
- **JWT:** *JSON Web Token* o Token web de JSON.
- **PDF:** *Portable Document Format* o Formato de documento portátil.
- **PNG:** *Portable Network Graphic* o Gráfico de red portátil.
- **POST:** Solicitud de envío de datos a un servidor.
- **PYMES:** Pequeñas y medianas empresas.
- **SEO:** *Search Engine Optimization* u Optimización para Motores de Búsqueda.
- **SPA:** *Single-Page Application* o aplicación de una sola página.
- **SQL:** *Structured Query Language* o Lenguaje de Consulta Estructurado.
- **TFG:** Trabajo de Fin de Grado.
- **TIC:** Tecnologías de la Información y la Comunicación.
- **URL:** *Uniform Resource Locator* o Localizador de Recursos Uniforme.
- **WIP:** *Work in Progress* o Trabajo en progreso.

Bibliografía

- [1] Anderson, D. *Kanban: Successful Evolutionary Change for Your Technology Business*. Blue Hole Press, 2010.
- [2] APD. ¿En qué consiste la metodología kanban y cómo utilizarla? <https://www.apd.es/metodologia-kanban/>, 2021.
- [3] Avison, D. and G. Fitzgerald. *Information Systems Development: Methodologies, Techniques, and Tools*. McGraw-Hill, 1995.
- [4] Análisis de participación y tamaño del mercado de software de gestión de eventos, tendencias de crecimiento y pronósticos, (2024-2029). www.mordorintelligence.com/es/industry-reports/event-management-software-market, October 2023.
- [5] Bootstrap introduction. getbootstrap.com/docs/4.1/getting-started/introduction/.
- [6] Castillo Anzules, M. y Guaña Moya, M. Kanban: una metodología ágil para la gestión eficiente del flujo de trabajo en el desarrollo de software, una revisión sistemática. *Revista Ingenio Global*, volumen 3, pp.17-2, Mayo 2024.
- [7] Sueldo DAW 2024: ¿cuánto cobra este desarrollador? www.ceac.es/formacion-profesional/grados-superiores/informatica-tecnologia/desarrollo-aplicaciones-web/sueldo, 2024.
- [8] Event Planner Sp. Contact. www.eventplanner.net/contact.php, n.d.
- [9] Datatables vue. datatables.net/manual/vue.
- [10] Zoho. Acerca de nosotros. www.zoho.com/es-xl/aboutus.html?ireft=nhome&src=es-xl-footer, n.d.
- [11] The Knot. About Us. www.theknotww.com/about-us/, n.d.
- [12] Fuentes del Burgo, J. y Sebastián, M. Comparative analysis of the board tool in the agile methodologies scrum, kanban and scrumban in software projects. conference: 26th international congress on project management and engineering cidip. <https://www.researchgate.net/publication/364302850>, 2022.

- [13] Sueldos para el puesto de Freelance Software Developer en España. www.glassdoor.es/Salaries/freelance-software-developer-salary-SRCH_KO0,28.htm?countryRedirect=true, 2024.
- [14] How Much can a Freelance Programmer Make? lancebase.io/how-much-can-a-freelance-programmer-make/, n.d.
- [15] Lombok features. projectlombok.org/features/.
- [16] Martins, J. ¿qué es la metodología kanban y cómo funciona? <https://asana.com/es/resources/what-is-kanban>, 2022.
- [17] Microsoft. Typescript for the new programmer. www.typescriptlang.org/docs/handbook/typescript-from-scratch.html, 2023.
- [18] MongoDB documentation. www.mongodb.com/docs/.
- [19] Ovais Ahmad, M; Dennehy, D; Conboy, K; Oivo, M . Kanban in software engineering: A systematic mapping study. *Journal of Systems and Software, volume (137),pp.96-113*, 2018.
- [20] Palacio, J. Scrum manager i. las reglas del juego. *Scrum Manager*, 2015.
- [21] Pressman, R., Ph.D. *Ingeniería del software, un enfoque práctico*. McGraw-Hill Interamericana Editores, S.A, 2010.
- [22] Quesada Reyes, N.T. Estudio sobre metodologías Ágiles en los proyectos software. propuesta de plan de implantación para pymes. trabajo fin de master. Master's thesis, Universidad de Sevilla, España, 2020.
- [23] Schwaber, K and Sutherland, J . La guía de scrum. la guía definitiva de scrum:las reglas del juego. 2017.
- [24] M. Shepperd. *Rethinking Success in Software Projects: Looking Beyond the Failure Factors, pp.27-49. Software Project Management in a Changing World*. Ruhe Wohlin, 2014.
- [25] Spring framework. spring.io/projects/spring-framework, 2023.
- [26] The PostgreSQL Global Development Group. PostgreSQL: About. www.postgresql.org/about/.
- [27] Universidad de Las Palmas de Gran Canaria. Grado en ingeniería informática guía docente. https://www2.ulpgc.es/aplicaciones/proyectosdocentes/pdf.php?id_proyecto=68714&NUEVA=1, 2023/24. 40991 - TRABAJO FIN DE GRADO.
- [28] Vue Quick Start. vuejs.org/guide/quick-start.html.
- [29] Web Accessibility Initiative (WAI). Introduction to web accessibility. <https://www.w3.org/WAI/fundamentals/accessibility-intro/#context>, 2022.
- [30] Zumba Gamboa, J.P y León Arreaga, C.A. Evolución de las metodologías y modelos utilizados en el desarrollo de software. innova. *Research Journal, volumen (3), No. 10, pp.20-33*, 2018.