PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS DE INFORMACIÓN

Trabajo de Titulación

TEMA:

Desarrollo de Prototipos Aplicación Web Responsive e Interfaz de Administrador para Gestión de Eventos Sociales

AUTOR:

Bastidas Moya Adrian Rafael

QUITO DM, 31 DE MAYO

AGRADECIMIENTO

Quisiera expresar mi más profundo agradecimiento a mi director de tesis, PhD Henry Roa, por su invaluable guía en los procesos iniciales de este proyecto. Sus conocimientos y ayuda fueron fundamentales para la culminación exitosa de esta tesis.

Agradezco profundamente a mis padres, cuyo amor y apoyo incondicional han sido cruciales durante todo este proceso de desarrollo e investigación. Su incansable paciencia y motivación constante no solo me sostuvieron en los momentos más difíciles, sino que fueron la base sólida sobre la que pude construir este proyecto. Su ejemplo y dedicación han sido fundamentales en cada paso de mi camino hacia la culminación de mis estudios.

Finalmente, deseo expresar mi profunda gratitud a Julieta Jiménez, cuya amistad y apoyo constante a lo largo de mi carrera universitaria han sido invaluables. Su visión y mentalidad no solo fueron fundamentales en la concepción de este proyecto, sino también en mi crecimiento personal y profesional. Su presencia ha sido un pilar esencial en este viaje.

RESUMEN

El presente trabajo de titulación describe el desarrollo de prototipos de una aplicación web responsive y una interfaz de administrador, ambos diseñados para la gestión de eventos sociales. Este proyecto busca ofrecer una solución tecnológica que permita mejorar la eficiencia en la organización de eventos sociales, optimizando recursos y mejorando la experiencia del usuario, además de facilitar la recopilación de datos esenciales que apoyen la toma de decisiones. El desarrollo se divide en dos componentes principales: el primero es una **plataforma web centralizada** que permite a los usuarios buscar, participar y gestionar eventos, mejorando así la interacción social y la accesibilidad. El segundo componente es la **interfaz de administración**, que facilita la gestión de eventos y la trazabilidad de las participaciones. Se implementará utilizando la metodología ágil SCRUM y aprovechando las tecnologías web modernas para asegurar una implementación efectiva y adaptativa.

CONTENIDO

INDICE DE ILUSTRACIONES

INDICE DE TABLAS

INTRODUCCION

1. MARCO DE REFERENCIA

1.1. Tema

Desarrollo de Prototipos de Aplicación web responsive e Interfaz de Administrador para la gestión de eventos sociales hacia el público en general.

1.2. Justificación

La elección de desarrollar estos prototipos se fundamenta en la creciente penetración de la tecnología web móvil en la vida cotidiana. En un contexto donde la conectividad digital es omnipresente, existe una demanda creciente de soluciones innovadoras que permitan a las personas descubrir y participar de manera eficiente en eventos sociales y actividades de entretenimiento. La propuesta de esta aplicación se destaca al ofrecer una respuesta a esta necesidad imperante, brindando una solución conveniente tanto para los comercios locales como para diversos sectores de la comunidad. Su potencial impacto positivo en la calidad de vida de los individuos y el fortalecimiento de las comunidades locales se evidencian en la capacidad de enriquecer experiencias, impulsar el crecimiento económico a través de la asistencia a eventos y establecer un vínculo entre comercios locales y una audiencia más amplia, promoviendo la visibilidad y los ingresos de estos negocios.

1.3. Planteamiento de problema

Aunque hay muchos eventos y actividades locales, encontrar información sobre ellos puede ser complicado debido a la falta de una plataforma centralizada. Esta dispersión de datos conduce a que las personas se pierdan oportunidades y tengan experiencias limitadas. El problema se agrava al no contar con opciones para la planificación y conexión social durante la participación en eventos, limitando así la experiencia de los usuarios. La tesis se centrará en desarrollar los prototipos de una aplicación web responsive y una interfaz de administrador que simplifiquen la búsqueda y participación en eventos locales, fomentando la interacción social y la exploración en la comunidad. La propuesta busca superar estas limitaciones, permitiendo a los usuarios disfrutar plenamente de distintos tipos de eventos.

1.4. Objetivo General

Desarrollar los prototipos de una página web responsive y una interfaz de administrador para la gestión de eventos sociales, disponibles al alcance de todo público.

1.5. Objetivos Específicos

1.5.1. Diseñar una interfaz de usuario intuitiva y simple que facilite la búsqueda y participación en eventos y actividades locales.

1.5.2. Desarrollar un sistema de gestión de eventos que permita a los comercios locales y público en general cargar y actualizar información sobre sus eventos.

1.5.3. Implementar un sistema de recomendación que sugiera eventos y actividades relevantes para los usuarios.

1.5.4. Crear una interfaz de administrador que permita gestionar la plataforma, visualizar los eventos publicados y obtener estadísticas de uso.

1.6. Alcance

En este proyecto de titulación se centrará en el manejo y creación de los prototipos de una plataforma de gestión de eventos de uso para comercios y público en general, teniendo en cuenta los siguientes módulos y funcionalidades:

1.6.1. **Interfaz de Usuario:** La interfaz será simple de usar e intuitiva para la búsqueda y participación en eventos y actividades locales.

1.6.2. **Sistema de Gestión de Eventos:** Tanto comercios como público registrado podrán crear eventos de tipo social en la plataforma.

1.6.3. **Sistema de Participación en eventos:** El público registrado podrá participar de los distintos eventos creados por los usuarios o empresas dentro de la aplicación.

1.6.4. **Interfaz de Administrador:** Permitirá gestionar la plataforma, visualizar los eventos y obtener estadísticas de uso.

FUNDAMENTACIÓN TEÓRICA

2. Marco Teórico

2.1. Plataforma web para eventos sociales

Una plataforma web para eventos sociales es un sistema en línea diseñado para facilitar la planificación, gestión y participación en diversos eventos sociales, como fiestas, actividades físicas, convenciones, entre otros. Estas plataformas suelen ofrecer una amplia gama de características, tales como la creación de eventos con asistencia masiva, la gestión de listas de participantes, la coordinación de horarios, la provisión de detalles específicos sobre cada evento y, en algunos casos, la repartición automática de los ingresos generados entre el organizador y la plataforma. Estas funcionalidades permiten a los organizadores simplificar el proceso de planificación y comunicación, al tiempo que proporcionan a los asistentes una experiencia más interactiva y cómoda.

2.2. Metodología SCRUM

SCRUM es un marco de trabajo ágil ampliamente reconocido y utilizado en el desarrollo de software y en la gestión de proyectos complejos en diversas industrias. Este método se basa en la iteración y la colaboración, estructurando el trabajo en ciclos cortos y dinámicos conocidos como 'sprints', que suelen durar de 2 a 4 semanas. Durante cada sprint, los equipos multidisciplinarios trabajan para entregar incrementos específicos del producto, asegurando avances regulares y medibles hacia el objetivo final. El ciclo de SCRUM incluye varias reuniones clave, como la planificación del sprint, la revisión del sprint, y las 'Daily Scrums' o reuniones diarias de seguimiento, que son cruciales para mantener al equipo alineado y enfocado en las metas. Este marco no solo fomenta la transparencia y la comunicación continua entre todos los miembros del equipo, sino que también promueve la adaptación rápida a los cambios y la mejora continua, ayudando así a maximizar el valor del producto desarrollado. Según Schwaber y Sutherland, los creadores de SCRUM, este marco de trabajo es efectivo porque permite a los equipos abordar proyectos complejos y adaptarse rápidamente a las realidades cambiantes, garantizando un mayor control del proceso y una entrega de resultados más eficiente (Schwaber, K., & Sutherland, J. (2017). The Scrum Guide).

2.2.1. Roles en SCRUM

En el marco de trabajo SCRUM, se definen varios roles cruciales que contribuyen al éxito del equipo y del proyecto. El Scrum Master es fundamental, actuando como facilitador y asegurando que el equipo siga las prácticas de SCRUM correctamente; ayuda a eliminar obstáculos y actúa como un puente entre el equipo de desarrollo y las demás partes interesadas, garantizando una comunicación fluida y efectiva. El Product Owner es responsable de maximizar el valor del producto que el equipo de desarrollo entrega; prioriza las tareas en el Product Backlog y clarifica las necesidades del cliente para el equipo. Por último, el Equipo de Desarrollo está compuesto por profesionales que realizan el trabajo de entregar el producto final. Este equipo es autoorganizado y multifuncional, combinando y coordinando habilidades cruzadas para cumplir con sus objetivos (Schwaber & Sutherland, 2017).

2.2.2. Eventos en SCRUM

SCRUM organiza el desarrollo utilizando eventos clave que estructuran cada Sprint. La Planificación del Sprint es una sesión donde el equipo selecciona qué trabajo se realizará durante el próximo sprint. La Reunión Diaria, o Daily Scrum, facilita un breve intercambio para actualizar el progreso y coordinar actividades entre los miembros del equipo. La Revisión del Sprint ofrece una demostración del trabajo completado para recibir retroalimentación que pueda mejorar el producto final. Finalmente, la Retrospectiva del Sprint es una reunión enfocada en la mejora continua, donde el equipo reflexiona sobre el sprint anterior y discute qué se puede mejorar en el próximo ciclo (Schwaber & Sutherland, 2017).

2.2.3. Artefactos en SCRUM

Los artefactos en SCRUM son herramientas que permiten a los equipos planificar y medir el progreso de sus proyectos. El Product Backlog es una lista dinámica que contiene todas las características, funciones y requisitos del proyecto, priorizada por el Product Owner. El Sprint Backlog es una selección de tareas del Product Backlog que el equipo se compromete a completar durante un sprint. Finalmente, el Incremento del Producto representa la suma de todos los ítems del Product Backlog completados durante el sprint, más cualquier incremento de sprints anteriores, entregando un paso adelante en la funcionalidad del producto (Schwaber & Sutherland, 2017).

2.3. Tecnología Web Responsiva

La tecnología web responsiva es fundamental en el diseño y desarrollo de sitios web modernos. Esta técnica permite crear interfaces que se adaptan automáticamente a diferentes dispositivos y tamaños de pantalla, incluyendo computadoras de escritorio, tabletas y teléfonos inteligentes, garantizando una experiencia de usuario coherente y accesible en todos ellos. La adaptabilidad se consigue a través de técnicas avanzadas como las consultas de medios (media queries) y rejillas flexibles (flex grids), que permiten que el diseño y contenido de la página se ajusten dinámicamente. Además, se utilizan unidades de medida relativas como los 'em' o 'rem' para la escala de componentes, lo que contribuye a la fluidez y responsividad del sitio web (Marcotte, 2011; Meyer & Weyl, 2016).

2.4. Tecnologías Utilizadas

En este proyecto se utilizarán diversas tecnologías modernas para el desarrollo de la aplicación web responsive y la interfaz de administrador. Estas incluyen:

- **Firebase Storage**

Solución de almacenamiento en la nube desarrollada por Google, que permite a los desarrolladores almacenar y gestionar contenido generado por usuarios de manera segura y escalable. Parte de la suite Firebase, esta plataforma está construida sobre la robusta infraestructura de Google Cloud Storage, lo que garantiza alta disponibilidad y durabilidad del almacenamiento. Firebase Storage está diseñado para integrarse a la perfección con aplicaciones móviles y web, simplificando procesos como la carga, descarga y gestión de contenido multimedia. Además, ofrece características avanzadas como la reanudación automática de transferencias interrumpidas y la generación de URLs seguras para el acceso a los archivos, facilitando así una experiencia de desarrollo más eficiente y productiva (Google, 2021).

- **HTML5**

HTML5 es la quinta revisión del Lenguaje de Marcado de Hipertexto, la cual ha revolucionado la forma en que se desarrollan las páginas web al introducir nuevos elementos semánticos y mejorar el soporte para multimedia. Estos elementos incluyen etiquetas como <article>, <section>, <nav>, y <footer>, que permiten una estructura de página más clara y un mejor SEO. Además, HTML5 facilita la inclusión de contenido multimedia como audio y video directamente en las páginas web sin la necesidad de plugins externos, gracias a las etiquetas <audio> y <video>. Esto ha sido fundamental para el desarrollo de aplicaciones web más ricas e interactivas, accesibles desde cualquier dispositivo. HTML5 también incorpora APIs avanzadas para características como drag-and-drop, geolocalización, y almacenamiento local, extendiendo las capacidades de las aplicaciones web modernas (Pilgrim, 2011; Lawson & Sharp, 2011).

- **Sass**

Sass (Syntactically Awesome Style Sheets) es un preprocesador de CSS potente y eficiente, diseñado para mejorar la forma en que se escriben los estilos CSS en proyectos de desarrollo web. Al introducir funcionalidades avanzadas como variables, anidación, herencia, y mixins, Sass permite a los desarrolladores gestionar estilos complejos con una mayor facilidad y claridad. Estas características facilitan la organización del código y reducen la repetición, mejorando así la mantenibilidad y escalabilidad de las hojas de estilo. Además, Sass soporta la creación de estilos condicionales y bucles con sus estructuras de control, lo que es particularmente útil en proyectos grandes y dinámicos. La capacidad de dividir el código CSS en varios archivos (a menudo llamados "partials") y luego importarlos en un archivo principal también ayuda a mantener los proyectos más organizados y modularizados. Estas ventajas hacen de Sass una herramienta indispensable para los desarrolladores que buscan producir código CSS optimizado y fácil de mantener (Hanson, 2016; Cope, 2015).

- **Typescript**

TypeScript es un superconjunto de JavaScript desarrollado por Microsoft que introduce tipos estáticos opcionales, lo que permite a los desarrolladores escribir código más seguro y fácil de entender y mantener. Al añadir esta capa de tipificación estática, TypeScript mejora significativamente la robustez del código JavaScript, permitiendo la detección de errores en tiempo de compilación antes de que el código sea ejecutado en el navegador o en el servidor. Además, TypeScript proporciona características avanzadas no presentes en JavaScript estándar, como interfaces, tipos genéricos, y decoradores, que facilitan la construcción de aplicaciones grandes y complejas. Estas funcionalidades hacen de TypeScript una herramienta valiosa en el desarrollo de aplicaciones robustas y escalables, especialmente en entornos donde se requiere un alto grado de fiabilidad y mantenibilidad del código (Bierman, G., & Abadi, M., 2014; Palmer, S., 2020).

- **React**

React es una biblioteca de JavaScript desarrollada por Facebook para construir interfaces de usuario interactivas y componentizadas. Es ampliamente utilizada para desarrollar aplicaciones web y móviles que requieren interfaces ricas y dinámicas. Uno de los aspectos más innovadores de React es su uso de un DOM virtual, que optimiza el rendimiento al minimizar las actualizaciones directas en el DOM real del navegador. Esta característica permite a React gestionar eficientemente grandes árboles de componentes, resultando en aplicaciones más rápidas y ágiles. Además, React facilita la gestión del estado y la lógica de la interfaz de usuario, proporcionando un modelo coherente y predecible para el flujo de datos dentro de la aplicación a través de su arquitectura basada en componentes. Los componentes de React pueden ser fácilmente reutilizados y combinados, lo que fomenta la eficiencia en el desarrollo y la mantenibilidad del código. Estas características hacen de React una herramienta esencial para los desarrolladores que buscan crear experiencias de usuario atractivas y de alto rendimiento en múltiples plataformas (Vazquez, 2018; Banks & Porcello, 2017).

- **Redux**

Redux es una biblioteca de JavaScript influyente diseñada para la gestión eficaz del estado en aplicaciones web y móviles. Predominantemente integrada con React, aunque también usable con otras bibliotecas y frameworks, Redux centraliza el estado de la aplicación en un único 'store', lo que simplifica el control y la gestión de estados complejos en aplicaciones de gran escala. Esta centralización facilita el seguimiento y la depuración de cambios en el estado, haciendo más predecibles las mutaciones y la lógica de negocio. Redux promueve principios de arquitectura que aseguran que cada cambio de estado sea explícito y se realice a través de acciones predefinidas y reducers puros, los cuales especifican cómo el estado se transforma en respuesta a estas acciones. Además, al utilizar middleware como Redux Thunk o Redux Saga, Redux puede manejar efectos secundarios como llamadas asíncronas o lógicas complejas de sincronización, extendiendo así sus capacidades más allá de la simple gestión de estado. Estas características hacen de Redux una herramienta valiosa para desarrolladores que buscan mantener la coherencia del estado a través de interfaces de usuario complejas y dinámicas (Eisenberg, 2016; Murray, 2020).

- **Java**

Java es un lenguaje de programación orientado a objetos desarrollado por Sun Microsystems en la década de 1990, que ahora es mantenido por Oracle. Es ampliamente reconocido por su portabilidad, robustez y seguridad, características que lo han consolidado como uno de los lenguajes preferidos para el desarrollo de aplicaciones empresariales, de escritorio y móviles. La clave de su portabilidad reside en la Máquina Virtual de Java (JVM), que permite que un programa Java se ejecute sin modificaciones en cualquier dispositivo que cuente con una JVM compatible, independientemente de la arquitectura subyacente del sistema operativo. Esta capacidad universal lo hace ideal para aplicaciones que necesitan ser distribuidas en una variedad de plataformas hardware y sistemas operativos. Además, Java es conocido por su sólido manejo de memoria y su enfoque en la seguridad, minimizando problemas comunes como las violaciones de memoria que son más prevalentes en otros lenguajes de programación. La naturaleza orientada a objetos de Java facilita la creación de módulos de software reutilizables y sistemas mantenibles, lo que es crucial para aplicaciones empresariales grandes y complejas (Gosling et al., 2014; Horstmann, 2019).

- **Springboot**

Spring Boot es un proyecto dentro del amplio ecosistema Spring que simplifica el proceso de configuración y desarrollo de nuevas aplicaciones Java. Es especialmente valorado por su capacidad de proporcionar configuración automática y basada en convenciones, lo que permite a los desarrolladores centrarse más en las características del negocio y menos en la infraestructura de la aplicación. Spring Boot es crucial para el desarrollo ágil de microservicios, ya que reduce significativamente la complejidad de la configuración requerida para iniciar proyectos basados en Spring. Su enfoque en la "convención sobre configuración" elimina la necesidad de especificaciones detalladas, utilizando en su lugar configuraciones predeterminadas que se pueden personalizar según sea necesario. Además, facilita el despliegue continuo y la integración de aplicaciones, aspectos esenciales para las prácticas modernas de desarrollo de software como la integración continua (CI) y la entrega continua (CD). Gracias a estas características, Spring Boot ha ganado popularidad y se ha convertido en una herramienta indispensable para los desarrolladores que buscan implementar y gestionar servicios de backend de manera eficiente y con menos código boilerplate (Walls, 2016; Long & Gutierrez, 2020).

- **Postgresql**

PostgreSQL es un sistema de gestión de bases de datos relacional y objeto-relacional avanzado, ampliamente reconocido por su robustez y escalabilidad. Este sistema de código abierto es compatible con una variedad de características avanzadas que lo distinguen en el mundo de las bases de datos. Entre estas, las transacciones ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) aseguran que todas las transacciones se procesen de manera fiable, mientras que su soporte para JSON permite integrar datos no estructurados con las operaciones SQL tradicionales. Estas capacidades hacen de PostgreSQL una opción ideal tanto para aplicaciones empresariales de gran envergadura como para nuevas aplicaciones que requieren una gestión flexible y eficiente de diversos tipos de datos. Además, PostgreSQL es extensible por diseño, permitiendo a los desarrolladores introducir y usar sus propios tipos de datos, funciones y extensiones. Su arquitectura y características avanzadas facilitan el desarrollo de aplicaciones que son tanto escalables como resilientes, adaptándose eficientemente a cargas de trabajo intensas y entornos de alta demanda (Obe & Hsu, 2015; Riggs, 2017).

- **JWT (JSON Web Tokens)**

Estándar abierto (RFC 7519) que define una forma compacta y segura de transmitir información entre partes como un objeto JSON. Esta información puede ser verificada y confiable porque está firmada digitalmente. JWT se utiliza comúnmente para la autenticación y la autorización en aplicaciones web y móviles. Un token JWT consta de tres partes: un encabezado, un cuerpo o payload, y una firma. El encabezado típicamente contiene el tipo de token, que es JWT, y el algoritmo de firma utilizado, como HMAC SHA256 o RSA. El cuerpo, también conocido como payload, contiene las reclamaciones que son declaraciones sobre una entidad (usualmente el usuario) y datos adicionales. Estas reclamaciones pueden incluir información de identificación como el ID de usuario, roles, fecha de expiración del token y más. La firma es lo que asegura que el token no haya sido alterado en el tránsito, y se genera aplicando el algoritmo especificado en el encabezado al encabezado y al cuerpo. Los JWT son una herramienta eficaz para manejar la autenticidad de la identidad en aplicaciones descentralizadas, facilitando un método robusto para manejar la seguridad sin necesidad de almacenar sesiones o cookies extensivas (Jones, Bradley & Sakimura, 2015; Parecki & Jones, 2020).

- **BCRYOP**

algoritmo de hashing de contraseñas diseñado para ser computacionalmente costoso de modo que pueda resistir ataques de fuerza bruta. Fue creado por Niels Provos y David Mazières basado en el cifrado Blowfish y se presentó en USENIX en 1999. Una característica distintiva de bcrypt es su capacidad de incorporar un factor de trabajo, también conocido como "cost", que regula el tiempo requerido para hash una contraseña. Cuanto mayor es el factor de trabajo, más lento es el proceso, lo que proporciona una defensa adicional contra intentos de hackeo. En la práctica, bcrypt genera un hash de contraseña que comienza con $2a$, $2b$ o $2y$, seguido del factor de costo y un "salt" generado aleatoriamente que se usa para asegurar el hash y evitar ataques de tabla de arco iris. El "salt" y el hash resultante se almacenan juntos, y el mismo proceso debe repetirse exactamente con la contraseña proporcionada por el usuario en futuros intentos de inicio de sesión para verificar la validez. La capacidad de bcrypt para ajustar el factor de costo según la mejora del hardware con el tiempo, junto con el almacenamiento seguro de "salt" integrado en el hash, lo convierte en una opción robusta y preferida para la seguridad de contraseñas en muchas aplicaciones modernas web y móviles (Provos & Mazières, 1999).

DISEÑO DE LAS APLICACIONES

3. Análisis y Diseño de la Propuesta

3.1. Interfaz de usuario y experiencia de usuario

El diseño de la interfaz de usuario (UI) y la experiencia de usuario (UX) son componentes esenciales para garantizar que los usuarios puedan interactuar con la aplicación de manera efectiva y satisfactoria. La UI se enfocará en proporcionar una estructura clara y simple, mientras que la UX se centrará en la facilidad de uso y la eficiencia de la navegación. Los principios clave incluyen:

- **Consistencia**: Uso coherente de colores, tipografía y elementos de diseño en toda la aplicación.

- **Simplicidad**: Interfaces intuitivas que minimicen la carga cognitiva y faciliten la realización de tareas comunes.

- **Accesibilidad**: Diseño inclusivo que garantice la accesibilidad para todos los usuarios que puedan usar una plataforma web.

- **Respuesta rápida**: Interacciones fluidas y rápidas para mejorar la experiencia del usuario.

3.2. Análisis y Levantamiento de requerimientos

Para desarrollar una aplicación efectiva destinada a la gestión de eventos sociales, fue esencial comprender a fondo las necesidades y preferencias de los usuarios potenciales. Este entendimiento se basa en el análisis de requerimientos, un paso crucial para asegurar que las funcionalidades del sistema respondan a las necesidades reales de gestión de eventos, manejo de información y facilitación de la interacción entre los usuarios.

Dada la necesidad de adaptar la plataforma a las especificidades de la gestión de eventos en diferentes contextos sociales y la importancia de incorporar funcionalidades que manejen eficazmente grandes volúmenes de datos, se optó por realizar una entrevista detallada

utilizando Google Forms. Esta entrevista fue diseñada para capturar de manera precisa y estructurada las expectativas y requisitos de los usuarios finales, quienes son organizadores y participantes de eventos sociales.

Herramienta Utilizada: Se empleó Google Forms por su accesibilidad y facilidad de uso, permitiendo a los participantes completar la entrevista en un entorno familiar y flexible.

Participantes: La entrevista fue dirigida a un grupo selecto de organizadores de eventos con experiencia en la gestión de actividades sociales de variada índole.

Objetivos de la Entrevista: Identificar funcionalidades esenciales para la gestión de eventos, entender los desafíos actuales con las plataformas existentes, y recoger sugerencias de características innovadoras que podrían mejorar la experiencia del usuario.

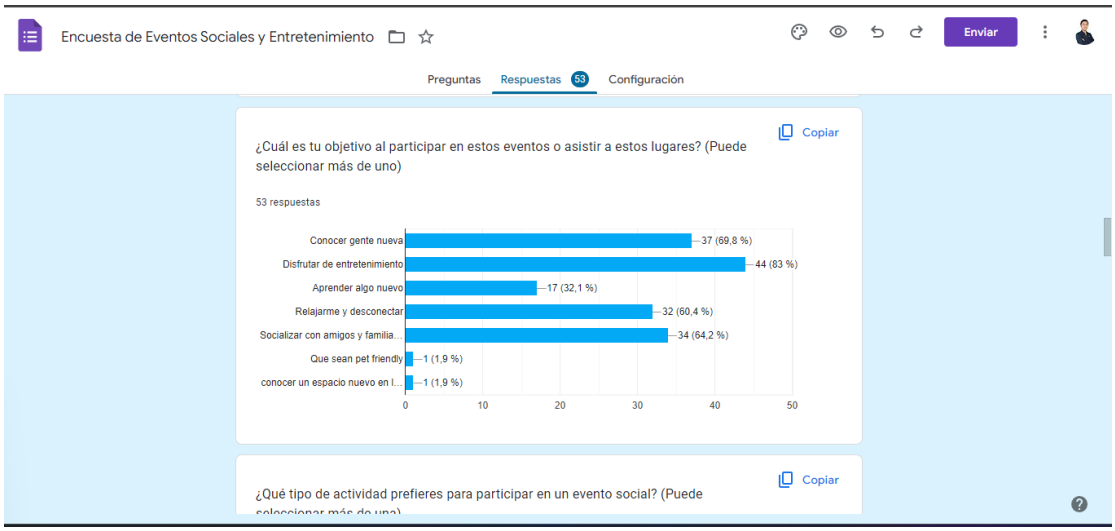

Ilustración 1 Formulario de entrevista - Elaboración Propia

3.3. Requerimientos Funcionales

Tabla 1.
Requerimientos Funcionales

Requerimiento	Descripción
Gestión de eventos (Usuario)	Permitir a los usuarios buscar, visualizar y registrarse en eventos. Incluir detalles como fecha, hora, y ubicación.
Gestión de eventos (Administrador)	Facilitar a los administradores la visualización de los eventos creados
Registro y autenticación (General)	Proporcionar un sistema de registro y autenticación seguro para usuarios y administradores con roles y permisos diferenciados.
Gestión de contenidos (Administrador)	Permitir al usuario administrador poder gestionar los contenidos prioritarios de la plataforma
Gestión de Usuarios (Administrador)	Proporcionar una vista de todos los usuarios registrados en el sistema y sus datos personales
Panel de control (Administrador)	Proporcionar un panel de control para que los administradores puedan ver estadísticas, tendencias y generar informes sobre la participación en eventos.

Nota: Esta tabla muestra los requerimientos funcionales que se tomaron en cuanta para el desarrollo del proyecto.

3.4. Requerimientos No Funcionales

Tabla 2.
Requerimientos No Funcionales

Requerimiento	**Descripción**
Seguridad (General)	Implementar protocolos de seguridad robustos para proteger la información sensible y transacciones en la plataforma.
Escalabilidad (General)	Diseñar la aplicación para que pueda manejar un aumento progresivo en el número de usuarios y carga de datos sin degradar el rendimiento.
Rendimiento (General)	Optimizar la aplicación para garantizar tiempos de respuesta rápidos y manejo eficiente de las peticiones simultáneas.
Disponibilidad (General)	Garantizar que la interfaz de usuario sea intuitiva y fácil de navegar para personas con cualquier nivel de habilidad tecnológica.
Compatibilidad (General)	Asegurar que la aplicación sea compatible con múltiples dispositivos, sistemas operativos y navegadores web.
Accesibilidad (Usuario)	Diseñar la interfaz de usuario cumpliendo con las normas de accesibilidad web WCAG para asegurar que todos los usuarios puedan utilizar la plataforma.

Nota: Esta tabla muestra los requerimientos no funcionales que se tomaron en cuanta para el desarrollo del proyecto.

3.5. Arquitectura del Proyecto

La arquitectura de la aplicación seguirá un enfoque modular y basado en servicios, lo que permite una fácil escalabilidad y mantenimiento. Los componentes principales incluyen:

3.5.1. Arquitectura Microservicios

La arquitectura basada en microservicios es un enfoque de diseño de software donde las aplicaciones se construyen como una colección de servicios independientes, cada uno ejecutándose en su propio proceso y comunicándose a través de mecanismos ligeros, generalmente una API HTTP. Este estilo arquitectónico es ideal para entornos empresariales y aplicaciones web que requieren alta escalabilidad y flexibilidad.

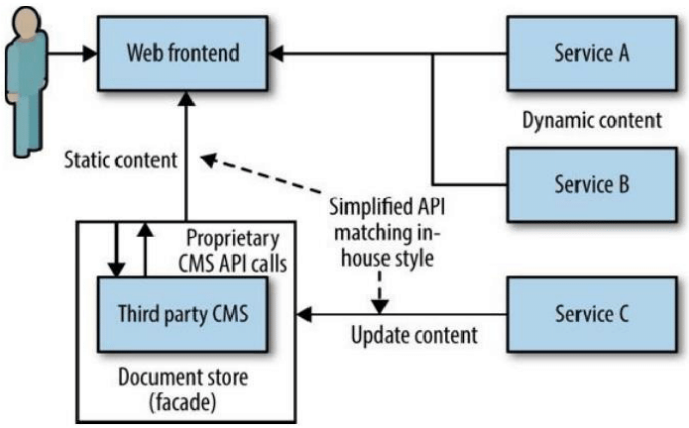

Ilustración 2 Arquitectura Microservicios - ResearchGate

3.5.2. Arquitectura Base de Datos

PostgreSQL opera bajo un modelo de arquitectura cliente-servidor, donde el servidor gestiona los archivos de datos, acepta conexiones de clientes, y realiza acciones sobre la base de datos. Los clientes pueden ser aplicaciones locales o remotas que comunican con el servidor a través de diversas interfaces como libpq para C, JDBC para Java, o psycopg para Python.

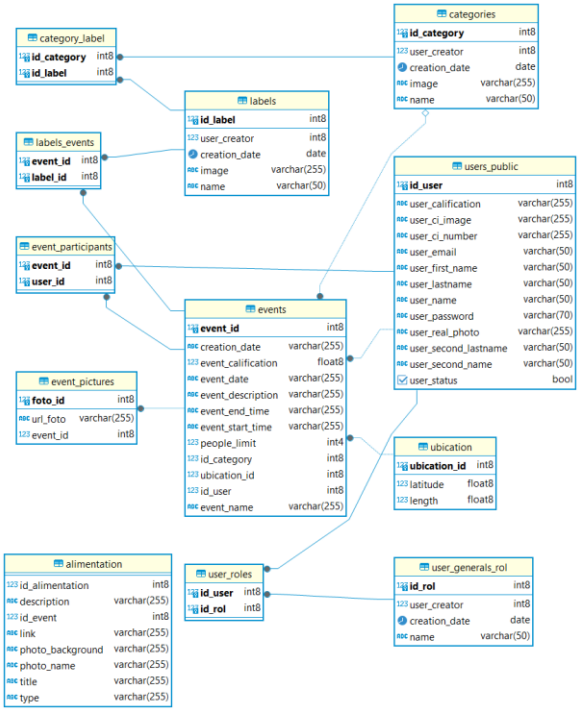

Ilustración 3 Base de datos - Elaboración Propia

3.5.3. Almacenamiento Multimedia

Firebase funciona como una plataforma Backend-as-a-Service (BaaS) que permite a los desarrolladores centrarse en crear experiencias de usuario front-end mientras maneja el backend y la infraestructura automáticamente. La plataforma proporciona APIs accesibles a través de SDKs para plataformas populares como iOS, Android y web.

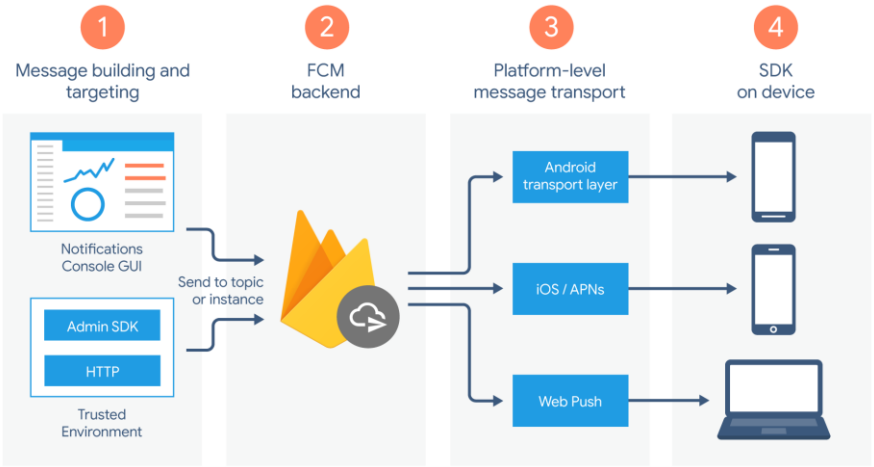

Ilustración 4 Arquitectura Firebase – Firebase

3.5.4. Caso de Uso UML

En el desarrollo del sistema de gestión de eventos sociales, se ha empleado el lenguaje de modelado UML para representar los casos de uso correspondientes a los diferentes niveles de acceso de dos tipos de usuarios: el administrador y el usuario organizador/asistente.

El usuario administrador tiene acceso a un conjunto limitado de funciones administrativas esenciales para el mantenimiento del sistema. Este rol está diseñado principalmente para gestionar cuentas de usuario, acceder a reportes generales y análisis de la actividad del sistema, y realizar configuraciones básicas. Sin embargo, no tiene capacidad para gestionar directamente los detalles específicos de los eventos, enfocando su rol más en tareas de supervisión general y aseguramiento del buen funcionamiento del sistema.

El usuario organizador/asistente, por otro lado, combina las funciones de organización y participación en eventos. Como organizador, este usuario tiene la capacidad de:

- **Crear y gestionar eventos:** Puede configurar nuevos eventos, editar detalles, manejar las inscripciones, y cargar documentación relevante.

- **Interactuar con los eventos:** Como asistente, puede inscribirse en eventos, recibir actualizaciones y notificaciones, y acceder a materiales y contenidos relacionados con los eventos a los que está suscrito.

Este diseño dual del rol permite a los usuarios tener un control completo sobre los eventos que organizan mientras disfrutan de la experiencia de participación en otros eventos, optimizando la interacción con la plataforma para ambas funciones.

Los diagramas UML elaborados para este sistema reflejan claramente estas capacidades y restricciones de acceso, garantizando que las funcionalidades implementadas en el software cumplan con las expectativas y necesidades de los usuarios, al tiempo que mantienen una clara separación de las responsabilidades administrativas y de usuario dentro del sistema.

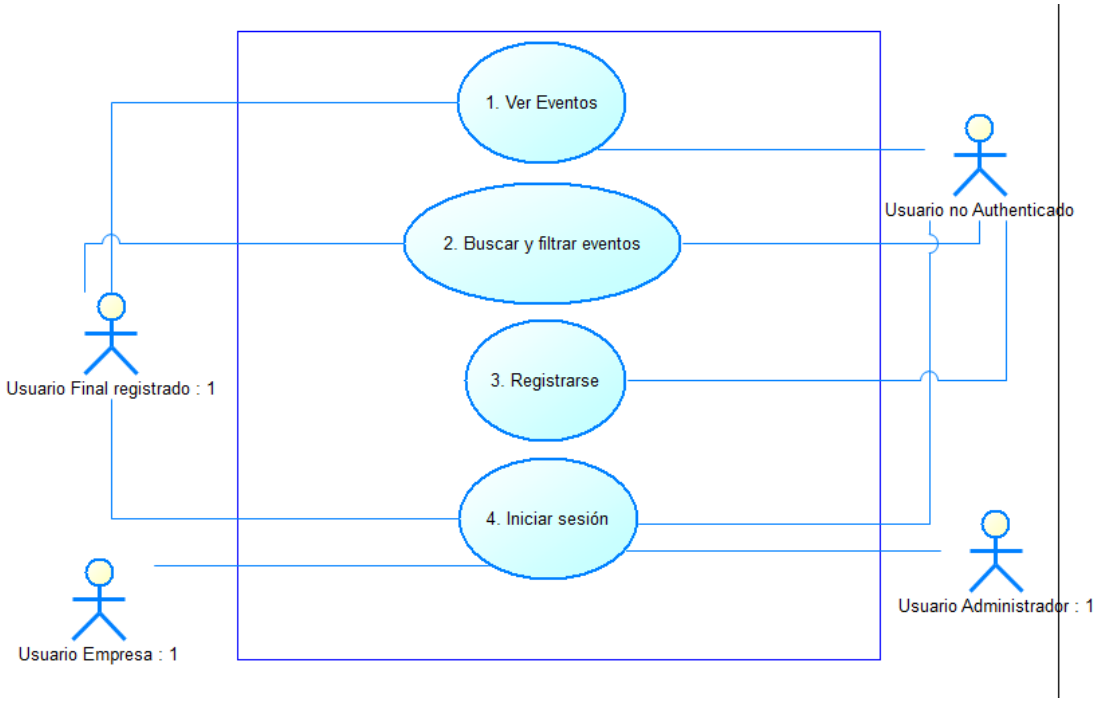

Ilustración 5 Diagrama UML1 - Elaboración Propia

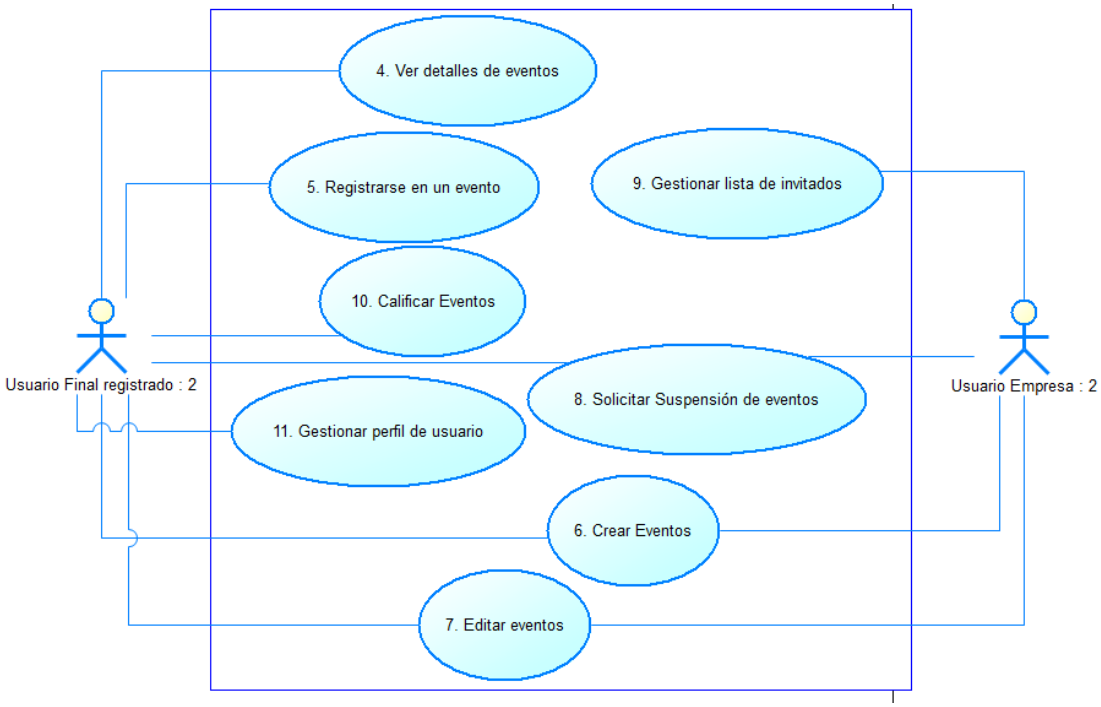

Ilustración 6 Diagrama UML2 - Elaboración Propia

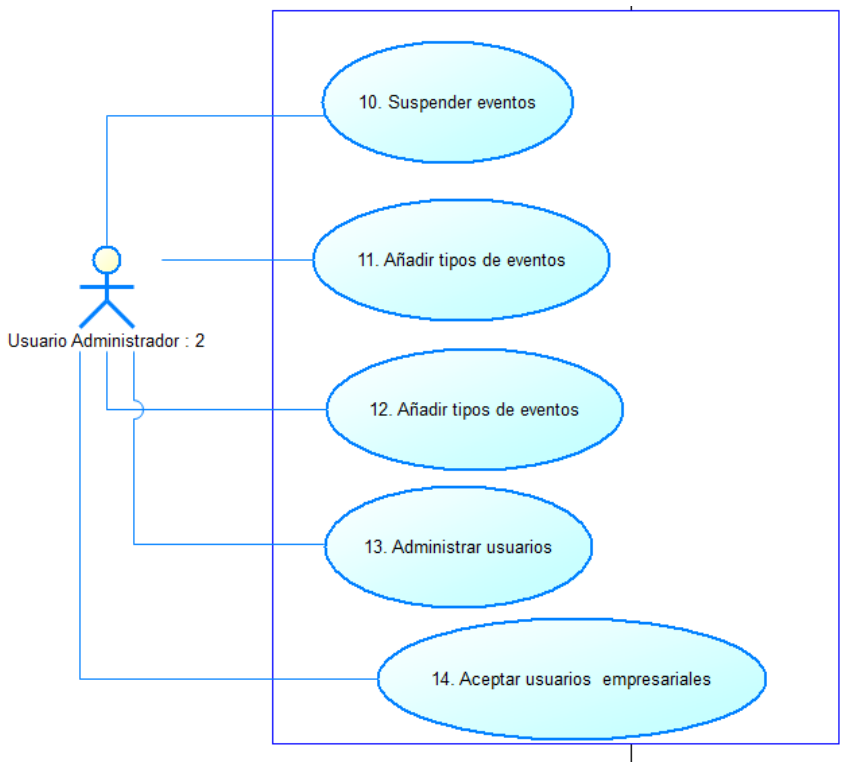

Ilustración 7 Diagrama UML3 - Elaboración Propia

3.6. Seguridad y privacidad

La seguridad y privacidad fueron aspectos fundamentales en el diseño y desarrollo de la aplicación. Las medidas adoptadas incluyeron:

- **Encriptación de Datos**: Todos los datos sensibles se transmitieron utilizando HTTPS para protegerlos de interceptaciones y ataques man-in-the-middle.

- **Encriptación de contraseñas**: Al ser datos de acceso se utilizó bcrypt para guardar las contraseñas en la base de datos, esta herramienta usa una encriptación de un solo lado y tiene un costo computacional alto para descifrarse, asegurando así las claves de acceso

- **Gestión de Sesiones**: JWT se utilizó para la autenticación y autorización de usuarios, proporcionando un método seguro y eficiente para gestionar sesiones.

- **Políticas de Acceso**: Se definieron roles y permisos claros para controlar el acceso a diferentes partes de la aplicación, garantizando que solo los usuarios con las autorizaciones necesarias pudieran realizar ciertas acciones.

- **Protección contra Ataques**: Se implementaron técnicas para prevenir ataques comunes como Cross-Site Scripting (XSS) y Cross-Site Request Forgery (CSRF).

DESARROLLO DE LAS APLICACIONES

5. Desarrollo

5.1. Tecnologías utilizadas en el desarrollo

El desarrollo de la aplicación utilizó una variedad de tecnologías modernas para garantizar un producto robusto y eficiente. Estas incluyeron:

- **Frontend**
  - **HTML5:** Usado en la estructura del contenido web
  - **Sass:** Usado como preprocesador de CSS para escribir estilos más organizados y eficientes.
  - **Typescript:** Superconjunto de JavaScript que añade tipos estáticos opcionales, mejorando la robustez y mantenibilidad del código, usado como lenguaje de programación.
  - **React:** Usado para construir interfaces de usuario basadas en componentes reutilizables y reactivos.
  - **Redux:** Librería para la gestión del estado de la aplicación, asegurando una comunicación eficiente entre los diferentes componentes.
- **Backend**
  - **Java:** Lenguaje de programación orientado a objetos utilizado por su robustez y portabilidad.
  - **Spring Boot:** Framework que facilita la creación de aplicaciones Java basadas en Spring, utilizado para desarrollar microservicios que manejan funciones específicas de la aplicación.

- **Base de Datos**
  - **PostgreSQL**: Sistema de gestión de bases de datos relacional y objeto-relacional conocido por su robustez y escalabilidad. Ofrece características avanzadas como transacciones ACID y soporte para JSON.

  - **FireBase Storage**: Proveedor Cloud de Google que nos facilitará la carga y descarga de imágenes en la plataforma de administrador y la plataforma del usuario común, evitando así el tener que almacenar grandes volúmenes de imágenes en base 64 dentro de la base de datos.

- **Autenticación y Autorización**
  - **JSON Web Tokens (JWT)**: Estándar para transmitir información de manera segura entre partes como un objeto JSON. Utilizado para la autenticación y autorización en la aplicación.
  - **Bcrypt**: Algoritmo de hashing robusto para contraseñas, diseñado para ser computacionalmente costoso y resistir ataques de fuerza bruta. Utilizado para la seguridad en la autenticación de usuarios.

5.2. Creación del sistema de gestión de eventos

El sistema de gestión de eventos permite a los usuarios crear, editar y eliminar eventos, así como gestionar la participación de los asistentes. Este sistema incluyó:

- **Formulario de Creación de Eventos**: Se desarrolló una interfaz intuitiva para la entrada de datos relevantes como el nombre del evento, fecha, ubicación y detalles adicionales. Este formulario fue implementado utilizando React y TypeScript para asegurar una experiencia de usuario fluida y reactiva.

- **Gestión de Participantes**: Recibir confirmaciones y mantener una lista actualizada de asistentes. Los usuarios pueden ver y gestionar las listas de participantes desde la interfaz del evento, utilizando componentes React conectados al backend mediante llamadas API RESTful.

- **Integración de Calendario**: Se añadió la funcionalidad para que el usuario pueda tener sus eventos creados o a los que va a asistir siempre organizados en una vista que le permite conocer de forma rápida todas sus actividades en el mes.

5.3. Sprints de SCRUM

Sprint 1: Configuración inicial de servicios, levantamiento de base de datos en Docker y diseño de logo principal

- **Objetivo:** Establecer la estructura base de la aplicación

- **Actividades:**
  - Configuración del entorno de desarrollo.
  - Configuración del entorno Docker.
  - Diseño y creación de logo principal

- **Resultados:**
  - Base de datos funcional.
  - Configuración de servidor Netflix eureka

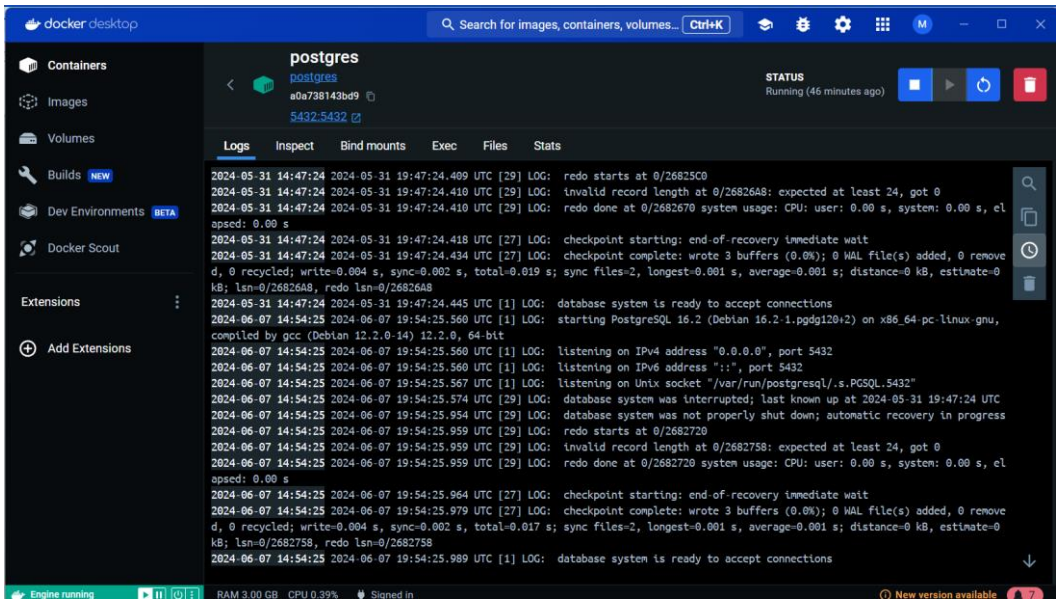

Ilustración 8 Docker con Postgresql -Elaboración Propia

Ilustración 9 Logo y Nombre Principal - Elaboración Propia

Ilustración 10 Logo Principal -Elaboración Propia

Sprint 2: Autenticación y log-in

- **Objetivo:** Creación de seguridad y página de log-in

- **Actividades:**
  - Implementación de la autenticación usando JWT.
  - Implementación de bcrypt para contraseñas
  - Interfaz de log-in de usuario.

- **Resultados:**
  - Sistema de login funcional.

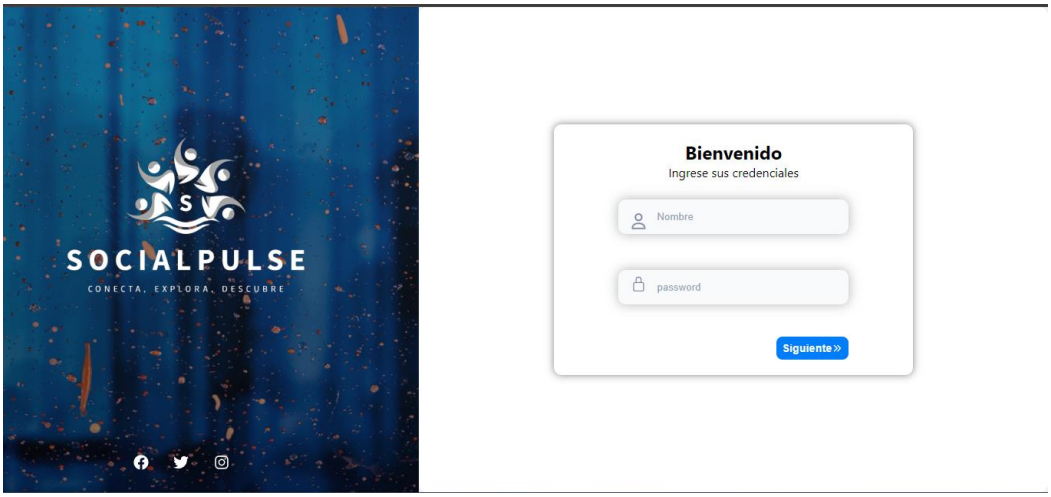

Ilustración 11 Login funcional - Elaboración Propia

Sprint 3: Creación de Dashboard para toma de decisiones de administrador

- **Objetivo:** Creación de vista para Administrador de dashboard y conexión de sidebar

- **Actividades:**
  - Creación de APIS para dashboard de administrador.
  - Interfaz de dashboard para administrador.

- **Resultados:**
  - Sistema de dashboard funcional.

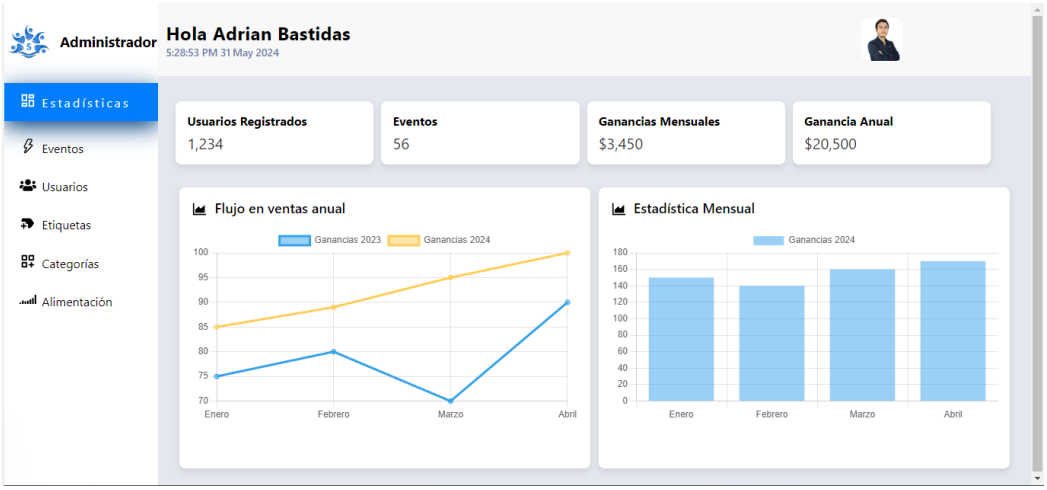

Ilustración 12 dashboard - Elaboración Propia

Sprint 4: Funcionalidades de gestión de contenido para la plataforma

- **Objetivo:** Creación de múltiples pestañas para gestión de alimentación de la plataforma

- **Actividades:**
  - Creación de APIS para páginas de administrador.
  - Interfaz de páginas de alimentación para administrador.

- **Resultados:**
  - Páginas de alimentación de la plataforma funcionales.

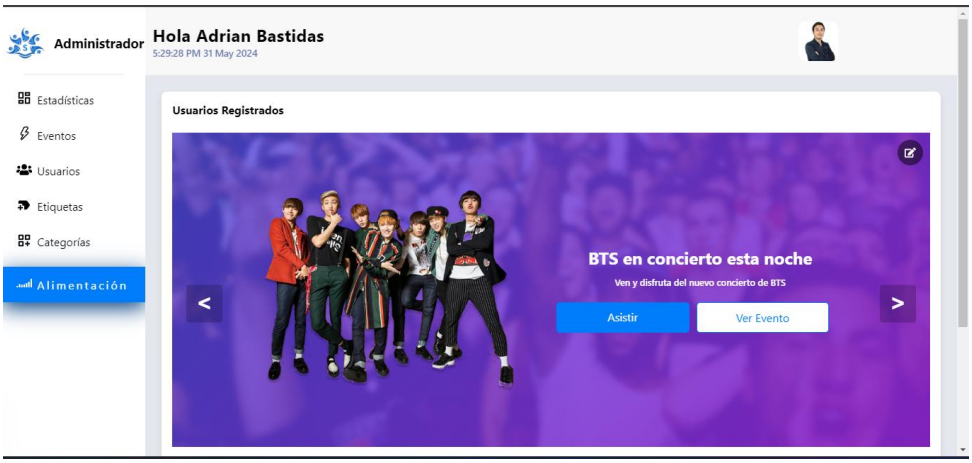

Ilustración 13 Alimentación de administrador - Elaboración Propia

Sprint 5: Diseño de plataforma de usuario

- **Objetivo:** Creación de diseño a seguir para la plataforma de eventos

- **Actividades:**
  - Creación de diseño para página web responsive.

- **Resultados:**
  - Diseño completo de página web.

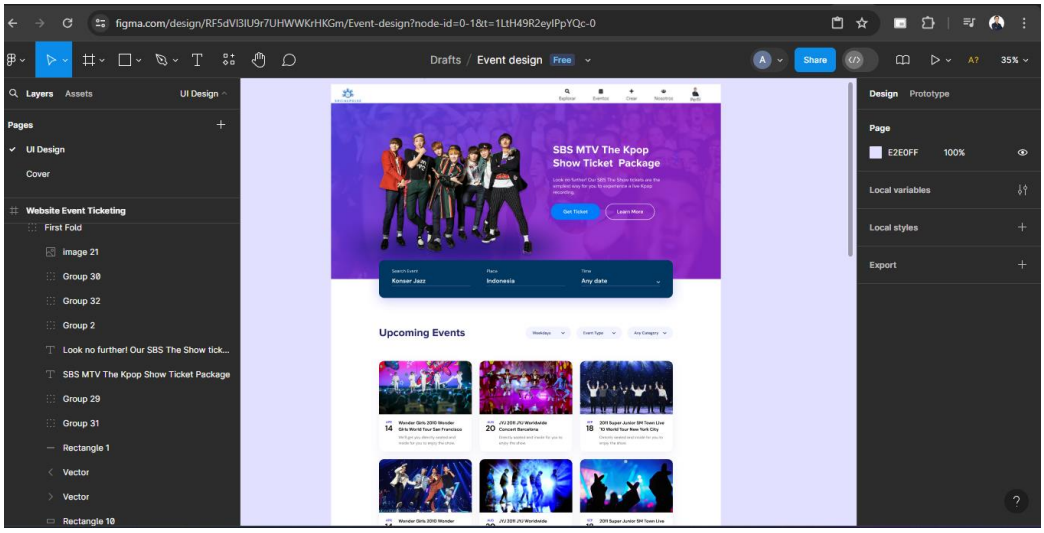

Ilustración 14 figma - Elaboración Propia

Sprint 6: Creación de plataforma inicial pt1

- **Objetivo:** Creación de banner y navegación por plataforma

- **Actividades:**
  - Conexión de múltiples pestañas de la página.
  - Creación del banner principal
- **Resultados:**
  - Banner y navbar completos

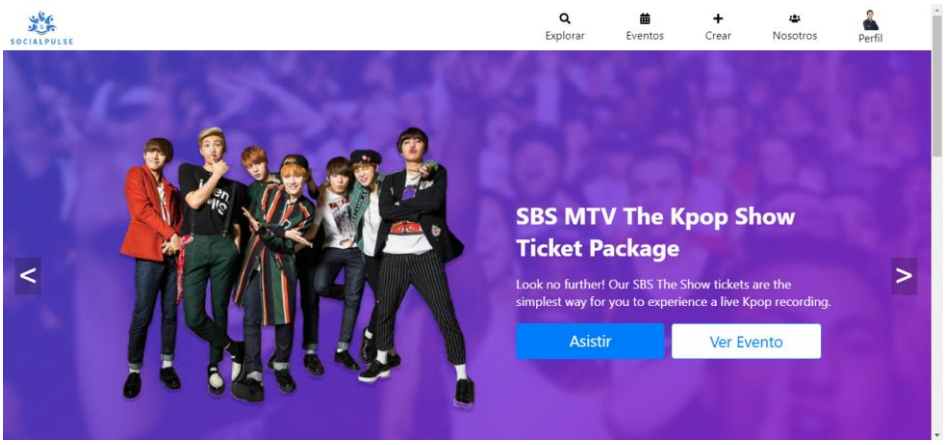

Ilustración 15 Banner- navbar - Elaboración Propia

Sprint 7: Creación de plataforma inicial pt2

- **Objetivo:** Creación de tarjetas de eventos y filtros

- **Actividades:**
    - Creación de componentes reusables para tarjetas de eventos y su visualización.
    - Creación y funcionalidad de filtros para eventos
- **Resultados:**
    - Tarjetas y visualización de eventos completos

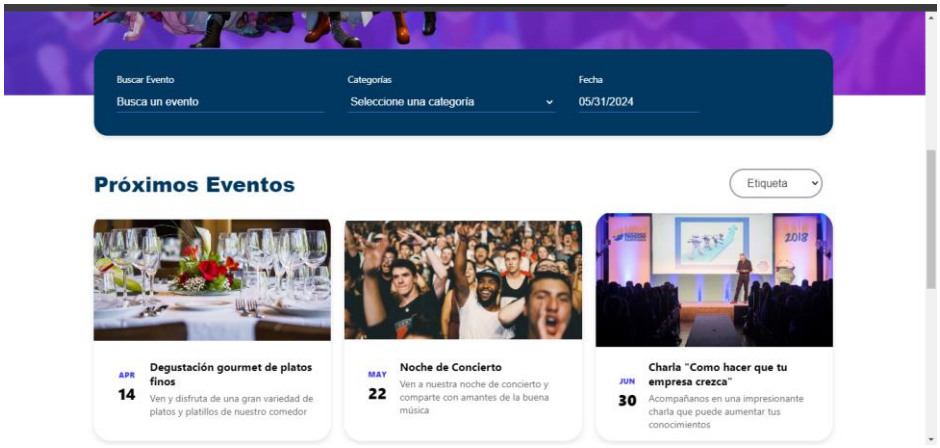

Ilustración 16 filtros-eventos - Elaboración Propia

Sprint 8: Eventos agendados

- **Objetivo:** Creación de visualización de eventos a los que el usuario debe asistir

- **Actividades:**
    - Implementación de calendario para mostrar eventos
    - Funcionalidad en APIS para eventos que debe asistir el usuario
- **Resultados:**
    - Visualización he interacción con eventos que el usuario ha elegido asistir

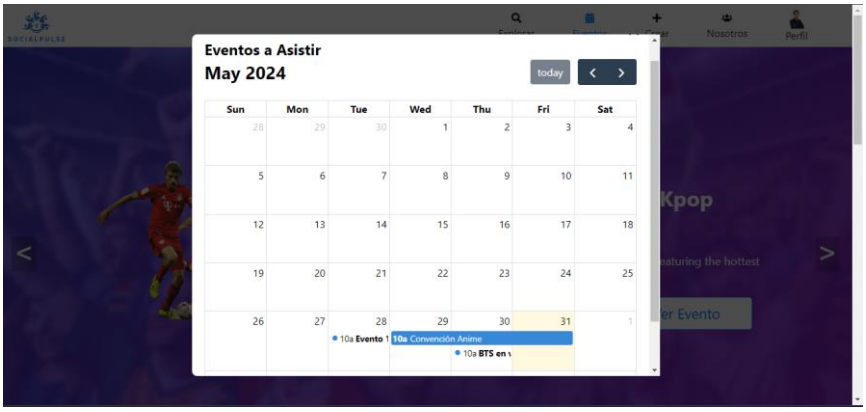

Ilustración 17 Calendario de eventos - Elaboración Propia

Sprint 9: Gestión de eventos

- **Objetivo:** Creación de funcionalidad y estilo para gestión de eventos

- **Actividades:**
  - Implementación de formulario para gestión de eventos
  - Implementación de mapa y direcciones para ubicación de eventos
  - Creación de API para funcionamiento
- **Resultados:**
  - Sistema de gestión de eventos integrado con mapa interactivo

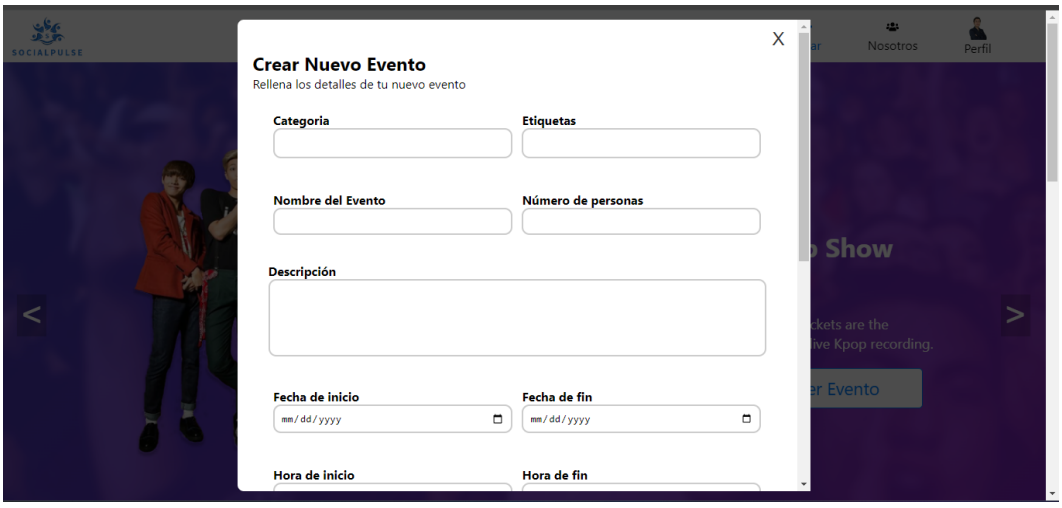

Ilustración 18 formulario evento - Elaboración Propia

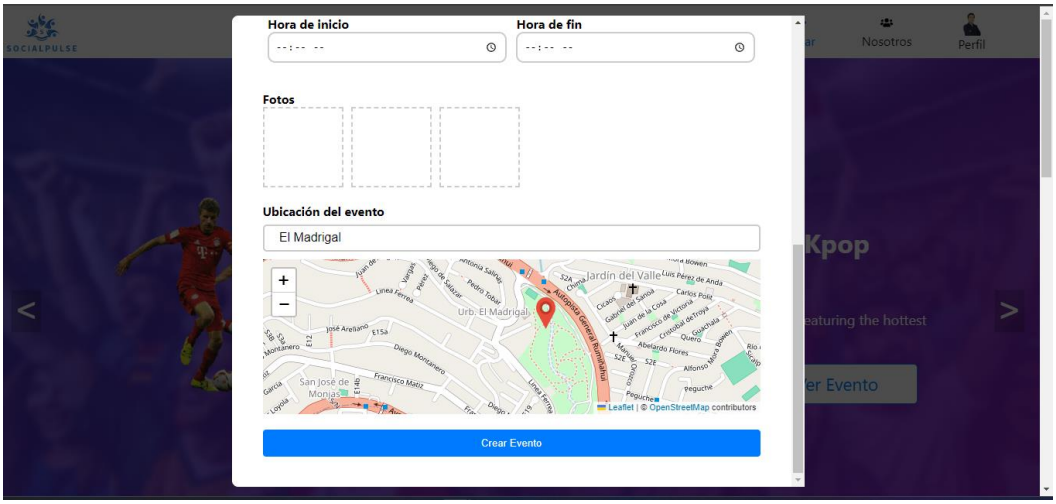

Ilustración 19 formulario mapa - Elaboración Propia

Sprint 10: Responsividad de la plataforma

- **Objetivo:** Convertir la plataforma web en una plataforma de uso para dispositivos móviles

- **Actividades:**
  - Uso de MediaQueris para control de responsividad de plataforma
  - Adaptación de código HTML para plataforma responsiva
- **Resultados:**
  - Plataforma responsiva

Ilustración 20 responsive 1 - Elaboración Propia

Ilustración 21 Responsive 2 - Elaboración Propia

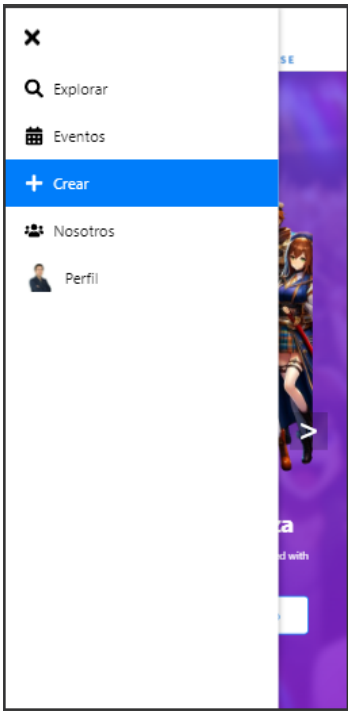

Ilustración 22 Responsive 3 - Elaboración Propia

Sprint 11: Documentación de Endpoints

- **Objetivo:** Documentación de Endpoints del back para su mejor entendimiento

- **Actividades:**
  - Uso de librería para documentación web de APIs
  - Mapeo de endpoints para documentación
- **Resultados:**
  - APIs documentadas y funcionales

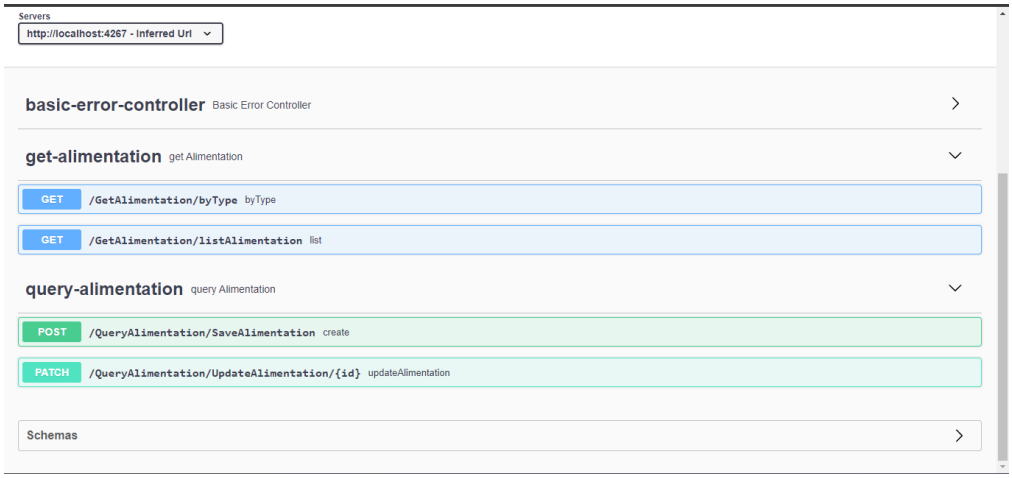

Ilustración 23 docuemntación endpoints 1 - Elaboración Propia

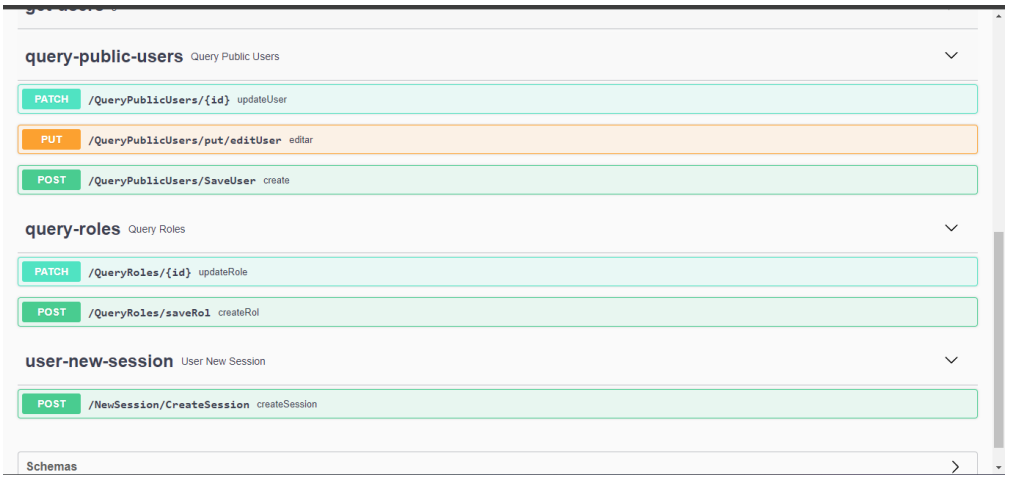

Ilustración 24 Documentación Endpoints 2 - Elaboración Propia

5.4. Pruebas y resolución de problemas

El proceso de desarrollo incluyó pruebas rigurosas para garantizar la calidad y funcionalidad de la aplicación. Esto incluyó:

Tabla 3.
Pruebas y Resolución de Problemas

Cod_Historia de usuario	Historia de usuario	Sistema	Cod_Prueba	Descripción de pruebas	Resultado esperado	Evaluación de la prueba
H1	Configuración inicial de servicios y levantamiento de base de datos en Docker	Backend	P1	- **Prueba de configuración de Docker:** Verificar que el entorno Docker se haya configurado correctamente y que la base de datos PostgreSQL esté operativa. - **Prueba de configuración de Netflix Eureka**: Verificar que el servidor Eureka esté configurado y funcionando correctamente para el descubrimiento de servicios.	Base de datos y servidor Eureka funcionales	100%
H2	Autenticación y log-in	Seguridad	P2	- **Prueba de autenticación exitosa o fallida:** Verificar que los usuarios registrados puedan iniciar sesión correctamente con las credenciales válidas. - **Prueba de recuperación de contraseña:** Verificar que los usuarios puedan restablecer su contraseña en caso de olvido y recibir un enlace de restablecimiento válido. - **Prueba de cierre de sesión**: Verificar que los usuarios puedan cerrar sesión correctamente y que se les redirija a la página de inicio de sesión.	Sistema de login funcional	100%

Cod_Historia de usuario	Historia de usuario	Sistema	Cod_Prueba	Descripción de pruebas	Resultado esperado	Evaluación de la prueba
				- **Prueba de diseño y usabilidad:** Verificar que la interfaz de inicio de sesión sea intuitiva y fácil de usar.		
H3	Creación de Dashboard para toma de decisiones de administrador	Backend	P3	- **Prueba de integración de APIs de dashboard:** Verificar que las APIs desarrolladas para el dashboard del administrador funcionen correctamente y devuelvan los datos esperados. - **Prueba de visualización de datos:** Verificar que los datos del dashboard se muestren correctamente y que la interfaz sea intuitiva.	Sistema de dashboard funcional	100%
H4	Funcionalidades de gestión de contenido para la plataforma	Backend	P4	- **Prueba de integración de APIs de administración:** Verificar que las APIs para las páginas de administración funcionen correctamente y permitan la gestión del contenido. - **Prueba de visualización y usabilidad:** Verificar que las interfaces de administración sean intuitivas y funcionales.	Páginas de administración funcionales	100%
H5	Diseño de plataforma de usuario	Frontend	P5	- **Prueba de diseño responsivo:** Verificar que el diseño de la página web sea responsivo y se adapte correctamente a diferentes tamaños de pantalla. - **Prueba de usabilidad:** Verificar que el diseño sea intuitivo y fácil de navegar.	Diseño completo de página web	100%

Cod_Historia de usuario	Historia de usuario	Sistema	Cod_Prueba	Descripción de pruebas	Resultado esperado	Evaluación de la prueba
H6	Creación de plataforma inicial pt1	Frontend	P6	**- Prueba de conexión de pestañas:** Verificar que las pestañas de la página estén correctamente conectadas y funcionen como se espera. - **Prueba de visualización del banner:** Verificar que el banner principal se muestre correctamente.	Banner y navbar completos	100%
H7	Creación de plataforma inicial pt2	Frontend	P7	- **Prueba de componentes reusables:** Verificar que los componentes para las tarjetas de eventos sean reusables y se muestren correctamente. - **Prueba de filtros:** Verificar que los filtros para eventos funcionen correctamente y filtren los eventos según los criterios especificados.	Tarjetas y visualización de eventos completos	100%
H8	Eventos agendados	Frontend	P8	- **Prueba de implementación del calendario:** Verificar que el calendario muestre correctamente los eventos a los que el usuario debe asistir. - **Prueba de integración de APIs:** Verificar que las APIs devuelvan correctamente los eventos del usuario.	Visualización e interacción con eventos	100%
H9	Gestión de eventos	Backend	P9	- **Prueba de formulario de gestión de eventos:** Verificar que el formulario permita crear, editar y eliminar eventos correctamente.	Sistema de gestión de eventos funcional	100%

Cod_Historia de usuario	Historia de usuario	Sistema	Cod_Prueba	Descripción de pruebas	Resultado esperado	Evaluación de la prueba
				- **Prueba de integración del mapa:** Verificar que el mapa interactivo funcione correctamente para la ubicación de eventos.		
H10	Responsividad de la plataforma	Frontend	P10	- **Prueba de responsividad:** Verificar que la plataforma se adapte correctamente a dispositivos móviles utilizando Media Queries. - **Prueba de usabilidad en móviles:** Verificar que la interfaz sea intuitiva y fácil de usar en dispositivos móviles.	Plataforma responsiva	100%
H11	Documentación de Endpoints	Backend	P11	- **Prueba de documentación:** Verificar que todos los endpoints de la API estén correctamente documentados utilizando una librería de documentación como Swagger. - **Prueba de mapeo de endpoints:** Verificar que todos los endpoints estén correctamente mapeados y documentados.	APIs documentadas y funcionales	100%

Nota: Esta tabla muestra las distintas pruebas y evaluaciones realizadas durante el desarrollo y al finalizar este mismo

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

El desarrollo de los prototipos de una aplicación web responsive y una interfaz de administrador para la gestión de eventos sociales ha demostrado ser una solución eficaz para centralizar la información sobre eventos y mejorar la experiencia del usuario en la planificación y participación en eventos locales. Las principales conclusiones obtenidas del proyecto son las siguientes:

- **Centralización de Información**: La plataforma proporciona un punto centralizado para la creación, gestión y búsqueda de eventos sociales, lo que facilita a los usuarios encontrar y participar en eventos relevantes.

- **Uso de Tecnologías Modernas**: La implementación de tecnologías modernas como React, TypeScript, Java, Spring Boot y PostgreSQL aseguró un desarrollo ágil, robusto y escalable, cumpliendo con los requisitos de funcionalidad y rendimiento.

- **Mejora en la Experiencia del Usuario**: La interfaz de usuario intuitiva y el diseño responsivo mejoraron significativamente la experiencia del usuario, permitiendo una navegación fácil y accesible desde diversos dispositivos.

El proyecto ha cumplido satisfactoriamente con todos los objetivos planteados inicialmente, lo que se evidencia en la implementación efectiva de los prototipos de la aplicación web y la interfaz de administrador. Los objetivos específicos, que incluían el desarrollo de una interfaz de usuario intuitiva, la creación de un sistema de gestión de eventos eficiente y la configuración de una interfaz administrativa completa, fueron alcanzados plenamente. Cada uno de estos componentes ha sido desarrollado y probado con éxito, demostrando su funcionalidad en el contexto de mejorar la organización y participación en eventos sociales. Además, la adaptabilidad y respuesta del sistema en diferentes dispositivos corroboran la efectividad del diseño responsive y la correcta utilización de las tecnologías seleccionadas. Por tanto, se puede afirmar que el proyecto no solo cumplió con los objetivos propuestos, sino que también estableció una sólida base para futuras mejoras y ampliaciones del sistema.

6.2. Recomendaciones

Para futuros desarrollos y mejoras en la plataforma, se recomienda lo siguiente:

- **Mejorar la Personalización**: Implementar algoritmos avanzados de recomendación para sugerir eventos basados en los intereses y comportamientos previos de los usuarios, personalizando aún más la experiencia del usuario.

- **Ampliar Funcionalidades**: Incluir características adicionales como la búsqueda de eventos por ubicación en el mapa, la integración de chat y comunidades con temáticas de nuestras etiquetas o categorías y la posibilidad de seguir y recibir notificaciones de organizadores específicos.

- **Escalabilidad y Desempeño**: Implementar mejoras en la arquitectura para manejar un mayor volumen de usuarios y eventos sin comprometer el desempeño. Esto podría incluir el uso de tecnologías de balanceo de carga y uso de microfronts para futuros desarrollos multi lenguaje.

- **Mejora de diseño:** Rediseñar la plataforma con un diseñador para mejorar la respuesta del usuario a esta y volverla más atractiva o llamativa a la vista, manteniendo la facilidad de uso para que todo el mundo pueda acceder a ella.

- **Actualización y Mantenimiento**: Mantener una estrategia de actualización regular para las tecnologías utilizadas en la plataforma, asegurando la compatibilidad y beneficiándose de las últimas mejoras y parches de seguridad ofrecidos por las herramientas y frameworks empleados.

- **Nuevas implementaciones:** Considerar la creación de una app para la parte del usuario, de esta forma se podría llegar a un mercado todavía mayor que solo la plataforma responsiva.

REFERENCIAS BIBLIOGRAFICÁS

- Allen, J. (2009). *Event Planning: The Ultimate Guide to Successful Meetings, Corporate Events, Fundraising Galas, Conferences, Conventions, Incentives, and Other Special Events.* John Wiley & Sons.
- Brook, R. (2019). *Mastering Identity and Access Management with Microsoft Azure: A Comprehensive Guide to IAM.* Packt Publishing.
- Cohn, M. (2005). *Agile Estimating and Planning.* Prentice Hall.
- Cooper, A. R. (2007). *About Face 3: The Essentials of Interaction Design.* Wiley.
- Creswell, J. W. (2013). *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches.* Sage Publications.
- Flanagan, D. &. (2012). *The Mobile Application Hacker's Handbook.* Wiley.
- Getz, D. (2008). *Event Studies: Theory, Research, and Policy for Planned Events.* Routledge.
- Norman, D. A. (2013). *The Design of Everyday Things.* Basic Books.
- Rubin, J. &. (2008). *Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests.* Wiley.
- Schwaber, K. &. (2013). *Scrum Guide.* Scrum.org.
- Smith, C. D. (2017). *Responsive Web Design in Practice.* Addison-Wesley.
- Tondreau, B. (2017). *Lean UX: Designing Great Products with Agile Teams.* O'Reilly Media.
- Viega, J. &. (2002). *Building Secure Software: How to Avoid Security Problems the Right Way.* Addison-Wesley.
- West, D. (2017). *Privacy and Security in Mobile Apps: Developments and Strategies.* O'Reilly Media.