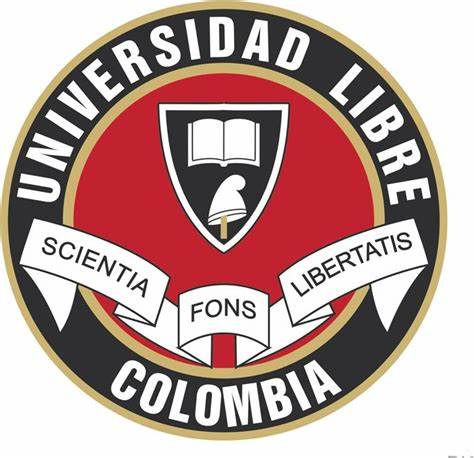
**Desparches**



Castelblanco G. Sara V, Daza C. Carlos E,

Montaña G. Laura G, Pérez J. Laura D. y

Tinoco M. Alison D.

Universidad Libre, Bogotá D.C.

Facultada de Ingeniería

Ingeniero:

Bogotá D.C, Colombia

2 de marzo del 2025

Contenido

[Tabla de ilustraciones 7](#_Toc199444459)

[1. Introducción 7](#_Toc199444460)

[2. Problema 7](#_Toc199444461)

[2.1 Pregunta por resolver: 7](#_Toc199444462)

[3. Objetivos 7](#_Toc199444463)

[3.1 Objetivo general: 8](#_Toc199444464)

[3.2 Objetivos específicos: 8](#_Toc199444465)

[4. Justificación 7](#_Toc199444466)

[4.1 ¿Por qué es necesario este proyecto? 8](#_Toc199444467)

[4.2 ¿Para qué desarrollar Desparches? 8](#_Toc199444468)

[4.3 ¿Cómo se va a desarrollar? 8](#_Toc199444469)

[4.4 ¿Qué aporta este proyecto? 8](#_Toc199444470)

[5. Alcance del proyecto 7](#_Toc199444471)

[5.1 ¿Hasta dónde se llega? 8](#_Toc199444472)

[5.2 ¿Qué no se incluye? 8](#_Toc199444473)

[5.3 ¿Por qué estas limitaciones? 8](#_Toc199444474)

[5.4 Proyección futura 8](#_Toc199444475)

[6. Marco teórico 7](#_Toc199444476)

[6.0 La Naturaleza del Proyecto 7](#_Toc199444477)

[6.1 Análisis de Causa Raíz 8](#_Toc199444478)

[6.1.1 Métodos 8](#_Toc199444479)

[6.1.2 Tecnología 9](#_Toc199444480)

[6.1.3 Materiales 9](#_Toc199444481)

[6.1.4 Mano de Obra 9](#_Toc199444482)

[6.1.5 Medición 10](#_Toc199444483)

[Diagrama Causa raíz de nuestro prototipo 10](#_Toc199444484)

[6.2 Análisis de lluvia de ideas 11](#_Toc199444485)

[6.2.1 Eventos Destacados 11](#_Toc199444486)

[6.2.2 Integración con Redes Sociales 11](#_Toc199444487)

[6.2.3 Sistemas de Calificación y Reseñas 11](#_Toc199444488)

[6.2.4 Alertas en Tiempo Real 11](#_Toc199444489)

[6.2.5 Recomendaciones Personalizadas 12](#_Toc199444490)

[6.2.6 Filtros Avanzados 12](#_Toc199444491)

[6.2.7 Implementación de Técnicas de Optimización 12](#_Toc199444492)

[Diagrama de lluvia de ideas 13](#_Toc199444493)

[6.3 Análisis de la matriz de priorización Impacto vs. Esfuerzo 13](#_Toc199444494)

[6.3.1 Riesgos Críticos (Alta Prioridad) 13](#_Toc199444495)

[6.3.2 Riesgos Significativos (Moderada Prioridad) 14](#_Toc199444496)

[6.3.3 Riesgos Moderados (Baja Prioridad) 14](#_Toc199444497)

[6.3.4 Riesgos Menores (Muy Baja Prioridad) 15](#_Toc199444498)

[Diagrama matriz de priorización Impacto vs. Esfuerzo 16](#_Toc199444499)

[6.5 Análisis del diagrama de flujo 16](#_Toc199444500)

[6.5.1 Puntos positivos 16](#_Toc199444501)

[6.5.2 Oportunidades de mejora 17](#_Toc199444502)

[6.6 Diagrama de flujo 17](#_Toc199444503)

[Diagrama de flujo para los usuarios 17](#_Toc199444504)

[Diagrama de flujo para los administradores 18](#_Toc199444505)

[6.7 Plataforma de diseño del prototipo: Figma 19](#_Toc199444506)

[6.7.1 ¿Qué es Figma? 19](#_Toc199444507)

[6.7.2 Nuestro prototipo 20](#_Toc199444508)

[6.8 Roles 20](#_Toc199444509)

[6.8.1 ¿Que son los roles? 20](#_Toc199444510)

[6.8.2 Roles en nuestro prototipo 20](#_Toc199444511)

[6.9 Criterios cardinales 21](#_Toc199444512)

[9.9.1 ¿Qué son los criterios cardinales? 21](#_Toc199444513)

[6.9.2 Criterios cardinales en nuestro prototipo: 21](#_Toc199444514)

[6.10 Solución de problemas 22](#_Toc199444515)

[6.10.1 ¿Qué significa solucionar problemas en ingeniería de software? 22](#_Toc199444516)

[6.10.2 Aplicación en nuestro prototipo 22](#_Toc199444517)

[Problema: Algunos usuarios pueden tener dificultades para navegar por la plataforma si la interfaz no es lo suficientemente intuitiva. 22](#_Toc199444518)

[6.11 Métodos 23](#_Toc199444519)

[6.11.1 ¿Qué son los métodos en diseño de ingeniería? 23](#_Toc199444520)

[6.11.2 Métodos aplicados en nuestro prototipo 24](#_Toc199444521)

[6.12 Requisitos 24](#_Toc199444522)

[6.12.1 ¿Qué son los requisitos en diseño de ingeniería? 24](#_Toc199444523)

[6.12.2 Requisitos en nuestro prototipo 24](#_Toc199444524)

[6.13 Restricciones 25](#_Toc199444525)

[6.13.1 ¿Qué son las restricciones en diseño de ingeniería? 25](#_Toc199444526)

[6.13.2 Restricciones en nuestro prototipo 25](#_Toc199444527)

[Design ThinKing: 26](#_Toc199444528)

[Empatizar 26](#_Toc199444529)

[Mapa de la Empatía: 27](#_Toc199444530)

[Definir 27](#_Toc199444531)

[User persona: 28](#_Toc199444532)

[Diagrama de afinidad: 28](#_Toc199444533)

[Declaración del problema: 29](#_Toc199444534)

[Idear 30](#_Toc199444535)

[Scamper: 30](#_Toc199444536)

[Brainstorming: 31](#_Toc199444537)

[Crazy 8’s: 31](#_Toc199444538)

[Prototipar 32](#_Toc199444539)

[Wireframes 32](#_Toc199444540)

[Prototipos Rápidos 33](#_Toc199444541)

[Link del prototipo en figma en computador 33](#_Toc199444542)

[Link del prototipo en figma en celular 34](#_Toc199444543)

[Testear 34](#_Toc199444544)

[Matriz de retroalimentación: 34](#_Toc199444545)

[User Journey Map 35](#_Toc199444546)

[UX: 36](#_Toc199444547)

[Usabilidad 36](#_Toc199444548)

[Accesibilidad 38](#_Toc199444549)

[Interacción Humano-Computadora (HCI) 42](#_Toc199444550)

[Principios de UX Aplicados 44](#_Toc199444551)

[Diseño centrado en el usuario: 44](#_Toc199444552)

[Jerarquía visual: 45](#_Toc199444553)

[Consistencia: 46](#_Toc199444554)

[Retroalimentación (Feedback): 47](#_Toc199444555)

[Eficiencia: 48](#_Toc199444556)

[Diferencias Relevantes 49](#_Toc199444557)

[UX vs UI: 49](#_Toc199444558)

[Usabilidad vs Accesibilidad: 50](#_Toc199444559)

[Interacción vs Experiencia: 51](#_Toc199444560)

[Arquitectura de la información 51](#_Toc199444561)

[9.1 ¿Qué es y por qué la usamos? 51](#_Toc199444562)

[9.2 Evidencia visual: Capturas del prototipo en Figma 52](#_Toc199444563)

[Diseño de interacción 56](#_Toc199444564)

[- Botones interactivos con retroalimentación 57](#_Toc199444565)

[- Navegación consistente entre pantallas 58](#_Toc199444566)

[-Flujo de interacción: buscar - ver - reservar 58](#_Toc199444567)

[-Control y libertad del usuario 61](#_Toc199444568)

[6.14 Metodologías de Desarrollo de Software 63](#_Toc199444569)

[6.14.1 Rational Unified Process (RUP) 63](#_Toc199444570)

[6.14.2 Metodologías Ágiles - SCRUM 63](#_Toc199444571)

[6.15 Arquitecturas de Software para Plataformas Digitales 63](#_Toc199444572)

[6.15.1 Arquitecturas Multicapa 63](#_Toc199444573)

[6.15.2 Microservicios y Arquitecturas Distribuidas 63](#_Toc199444574)

[6.16 Sistemas de Geolocalización y Computación Espacial 63](#_Toc199444575)

[6.16.1 Tecnologías de Geolocalización y Sistemas de Información Geográfica (GIS) 63](#_Toc199444576)

[6.16.2 Location-Based Services (LBS) 63](#_Toc199444577)

[6.16.3 Algoritmos de Recomendación Espacial 64](#_Toc199444578)

[6.17 Experiencia de Usuario y Diseño de Interfaces 64](#_Toc199444579)

[6.17.1 Principios de Usabilidad Móvil y Diseño Responsivo 64](#_Toc199444580)

[6.17.2 Interacción Táctil 64](#_Toc199444581)

[6.17.3 Comportamiento del Usuario en Plataformas de Eventos 64](#_Toc199444582)

[6.18 Plataformas Digitales y Modelos de Negocio 64](#_Toc199444583)

[6.18.1 Economías de Plataforma y Efectos de Red 64](#_Toc199444584)

[6.18.2 Monetización en Plataformas Locales 64](#_Toc199444585)

[6.19 Tecnologías de Desarrollo Web y Móvil 64](#_Toc199444586)

[6.19.1 Frameworks de Desarrollo Multiplataforma 64](#_Toc199444587)

[6.19.2 Arquitecturas Backend Modernas 65](#_Toc199444588)

[6.20 Seguridad y Privacidad en Aplicaciones Geolocalizadas 65](#_Toc199444589)

[6.20.1 Protección de Datos de Ubicación y Cumplimiento Regulatorio 65](#_Toc199444590)

[7. Metodología 65](#_Toc199444591)

[7.1 Uso de la metodología RUP 65](#_Toc199444592)

[7.2 Ingeniería de Requerimientos en la etapa de Inicio 66](#_Toc199444593)

[7.3 Diagramas UML como apoyo para el análisis y diseño 67](#_Toc199444594)

[7.4 Uso de la metodología Scrum 67](#_Toc199444595)

[8. Resultados 68](#_Toc199444596)

[8.1 Modelo arquitectónico propuesto 69](#_Toc199444597)

[8.2 Prototipos funcionales en Figma 69](#_Toc199444598)

[- Estructura visual y funcional definida 69](#_Toc199444599)

[Link del prototipo en figma en computador 69](#_Toc199444600)

[Link del prototipo en figma en celular 69](#_Toc199444601)

[8.3 Validación del diseño 69](#_Toc199444602)

[9. Conclusiones 69](#_Toc199444603)

[10. Bibliografía 69](#_Toc199444604)

[11. Anexos 69](#_Toc199444605)

# Tabla de ilustraciones

[Ilustración 1 Captura del diagrama causa raíz de la plataforma Desparches en Canva. Fuente: Elaboración propia. 13](#_Toc199363513)

[Ilustración 2Captura del diagrama lluvia de ideas de la plataforma Desparches en Canva. Fuente: Elaboración propia. 15](#_Toc199363514)

[Ilustración 3 Captura del diagrama matriz de priorización de la plataforma Desparches en Canva. Fuente: Elaboración propia. 19](#_Toc199363515)

[Ilustración 4 Captura del diagrama de fujo de usuarios de la plataforma Desparches en Canva. Fuente: Elaboración propia. 21](#_Toc199363516)

[Ilustración 5 Captura del diagrama de flujo de administradores de la plataforma Desparches en Canva. Fuente: Elaboración propia. 22](#_Toc199363517)

[Ilustración 6 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 39](#_Toc199363518)

[Ilustración 7 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 40](#_Toc199363519)

[Ilustración 8 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 41](#_Toc199363520)

[Ilustración 9 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 42](#_Toc199363521)

[Ilustración 10 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 43](#_Toc199363522)

[Ilustración 11 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 44](#_Toc199363523)

[Ilustración 12 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 44](#_Toc199363524)

[Ilustración 13 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 45](#_Toc199363525)

[Ilustración 14 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 46](#_Toc199363526)

[Ilustración 15 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 46](#_Toc199363527)

[Ilustración 16 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 47](#_Toc199363528)

[Ilustración 17 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 48](#_Toc199363529)

[Ilustración 18 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 49](#_Toc199363530)

[Ilustración 19 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 50](#_Toc199363531)

[Ilustración 20Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 51](#_Toc199363532)

[Ilustración 21 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 52](#_Toc199363533)

[Ilustración 22 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 53](#_Toc199363534)

[Ilustración 23 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 53](#_Toc199363535)

[Ilustración 24 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 54](#_Toc199363536)

[Ilustración 25 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 56](#_Toc199363537)

[Ilustración 26 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 57](#_Toc199363538)

[Ilustración 27 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 58](#_Toc199363539)

[Ilustración 28 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 59](#_Toc199363540)

[Ilustración 29 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 60](#_Toc199363541)

[Ilustración 30 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 61](#_Toc199363542)

[Ilustración 31 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 62](#_Toc199363543)

[Ilustración 32 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 63](#_Toc199363544)

[Ilustración 33 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 64](#_Toc199363545)

[Ilustración 34 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia. 65](#_Toc199363546)

# Introducción

El presente trabajo documenta el desarrollo del proyecto **Desparches**, una plataforma web diseñada para conectar a los usuarios con eventos y promociones relevantes en su entorno inmediato, de forma dinámica, personalizada y en tiempo real. A través de esta plataforma, los usuarios pueden descubrir actividades mediante filtros por ubicación, tipo de evento o intereses, así como explorar un mapa interactivo que muestra opciones disponibles sin necesidad de búsquedas manuales.

Este informe incluye todas las etapas clave del proceso de desarrollo, desde la identificación de necesidades y requisitos, hasta el diseño, construcción y validación del sistema. Se aplican dos metodologías de desarrollo: **Rational Unified Process (RUP)** y **SCRUM**, permitiendo una gestión estructurada, iterativa e incremental del proyecto. Además, se integran herramientas como **diagramas UML, ingeniería de requerimientos**, y prototipos desarrollados en **Figma**, que permiten visualizar y validar la experiencia del usuario antes de su implementación definitiva.

Se espera que, al finalizar el proyecto, se obtenga una aplicación funcional, escalable y centrada en el usuario, que facilite la conexión entre personas y experiencias locales, y que al mismo tiempo proporcione valor a los comercios y organizadores de eventos mediante una herramienta moderna, eficiente y fácil de usar.

# ****Problema****

En la actualidad, existen diversas plataformas que permiten organizar o promocionar eventos (como Eventbrite y Meetup), así como ofrecer promociones locales (como Groupon). Sin embargo, la mayoría de estas soluciones presentan limitaciones importantes en el contexto latinoamericano, y especialmente en ciudades como Bogotá. Estas limitaciones incluyen interfaces poco adaptadas al público local, costos de acceso elevados para pequeños organizadores, escasa integración entre eventos y promociones comerciales, y una baja personalización basada en ubicación y preferencias del usuario.

Además, aunque algunas investigaciones han abordado modelos de negocio digitales o estrategias de promoción en redes sociales, se percibe una brecha entre la teoría y la implementación práctica de soluciones tecnológicas locales que integren experiencias culturales, sociales y comerciales en un solo entorno digital.

Las plataformas actuales tienden a especializarse en un solo tipo de actividad (eventos o promociones) y pocas logran generar una experiencia integrada, dinámica y adaptada al entorno urbano inmediato de los usuarios. A esto se suma la necesidad de visibilidad para pequeños comercios y organizadores, quienes muchas veces carecen de canales eficaces para conectar con su público objetivo.

2.1 Pregunta por resolver:  
¿Cómo diseñar e implementar una plataforma digital que, de forma integrada, geolocalizada y personalizada, permita a los usuarios descubrir eventos y promociones en su entorno inmediato, al tiempo que ofrece a los pequeños comercios y organizadores una herramienta accesible para llegar a su público objetivo? (**Anexo 1.** Detalles adicionales del problema)

# ****3. Objetivos****

3.1 Objetivo general:  
Diseñar e implementar una plataforma digital que integre eventos y promociones geolocalizadas, permitiendo a los usuarios descubrir experiencias cercanas de manera personalizada, y facilitando a pequeños comercios y organizadores la difusión de sus actividades.

## ****3.2 Objetivos específicos:****

1. Identificar y estructurar los requerimientos funcionales y no funcionales necesarios para el desarrollo de la plataforma Desparches.
2. Desarrollar una arquitectura técnica escalable que permita integrar servicios de geolocalización, recomendaciones personalizadas y gestión de contenido.
3. Diseñar e implementar prototipos interactivos que optimicen la experiencia de usuario tanto en dispositivos móviles como en versión web.

# Justificación

## 4.1 ¿Por qué es necesario este proyecto?

En ciudades como Bogotá, muchas personas quieren encontrar planes locales interesantes, pero no tienen una plataforma que reúna todo en un solo lugar. Al mismo tiempo, los pequeños negocios y organizadores de eventos no cuentan con herramientas accesibles para promocionarse. Las plataformas actuales como Eventbrite o Meetup no se adaptan bien al público latinoamericano, son costosas y no ofrecen recomendaciones personalizadas basadas en ubicación.

## 4.2 ¿Para qué desarrollar Desparches?

Desparches busca conectar a personas con eventos, actividades y promociones locales de forma rápida y personalizada, y dar visibilidad a pequeños comercios sin que tengan que pagar altos costos. Esto ayuda a fortalecer la economía local y mejora la vida urbana.

## 4.3 ¿Cómo se va a desarrollar?

El proyecto usará tecnologías como geolocalización, inteligencia artificial y diseño responsivo. Se aplicarán metodologías de desarrollo como RUP y SCRUM para asegurar un proceso ordenado, flexible y eficiente.

## 4.4 ¿Qué aporta este proyecto?

* Crea una solución adaptada al contexto latinoamericano.
* Aplica teorías de ingeniería de software en un caso real.
* Integra tecnologías modernas como IA y microservicios.

# Alcance del proyecto

Este proyecto es de carácter académico, desarrollado en Ingeniería de sofware 1. Se enfoca en el diseño completo de la plataforma desde el análisis hasta la documentación, sin incluir desarrollo en código ni lanzamiento comercial.

## 5.1 ¿Hasta dónde se llega?

* **Ingeniería de requerimientos** detallada, con casos de uso, restricciones y diagramas UML.
* Aplicación de **metodologías RUP y SCRUM**, con documentación para cada fase.
* Diseño de **arquitectura del sistema**: backend, frontend, bases de datos y servicios.
* **Prototipos UX/UI interactivos** en Figma para web y móvil.
* **Modelado de procesos**, flujos de usuario y gestión de eventos/promociones.
* **Entregables finales**: documentación técnica, prototipos validados, plan de implementación y análisis de viabilidad.

## 5.2 ¿Qué no se incluye?

* No se programará la plataforma ni se desplegarán bases de datos.
* No habrá lanzamiento comercial, pruebas reales con usuarios ni marketing.
* No se instalará infraestructura técnica ni servicios en la nube.

## 5.3 ¿Por qué estas limitaciones?

* El tiempo y los recursos del ciclo académico son limitados.
* El enfoque es **formativo**, centrado en análisis, diseño y aplicación de metodologías.

## 5.4 Proyección futura

Este trabajo deja bases sólidas para que, en el futuro, pueda desarrollarse el sistema completo. Puede servir como:

* Guía para implementación técnica inmediata.
* Modelo adaptable a otros contextos.
* Proyecto base para tesis o investigaciones.
* Referente para emprendedores o empresas interesadas.

# Marco teórico

## 6.0 La Naturaleza del Proyecto

Es importante aclarar que el desarrollo de Desparches se plantea como un proyecto académico, enfocado en la conceptualización, análisis y diseño teórico de una plataforma digital, sin llegar a la implementación práctica o al desarrollo de código fuente real. Este enfoque responde a la finalidad educativa y de investigación, donde el objetivo principal es aplicar y comprender conceptos de ingeniería de software, diseño de interfaces, arquitectura de sistemas y modelos de negocio en un contexto controlado y limitado.

Según Pressman (2010), los proyectos académicos suelen centrarse en la planificación, diseño y análisis de sistemas para demostrar competencias técnicas y teóricas, sin necesariamente cubrir la etapa de desarrollo y despliegue. Esta práctica permite a los estudiantes y académicos profundizar en el entendimiento del ciclo de vida del software y en la integración de múltiples disciplinas sin la necesidad de recursos extensos o infraestructura real.

Por otra parte, Sommerville (2016) señala que los prototipos conceptuales o proyectos de diseño son fundamentales para validar ideas y explorar soluciones antes de su desarrollo efectivo, favoreciendo la identificación temprana de retos técnicos y funcionales. En este sentido, Desparches se concibe como un ejercicio de diseño teórico que facilita la aplicación práctica de conocimientos, sin realizar la codificación o pruebas de usuario reales.

Este enfoque también evita los costos y riesgos asociados con un proyecto a escala real, como la implementación, mantenimiento, seguridad y escalabilidad, permitiendo concentrarse en la documentación, análisis de requerimientos, diseño arquitectónico y propuesta de estrategias para futuras etapas de desarrollo si fueran requeridas.

## 6.1 Análisis de Causa Raíz

El análisis de causa raíz es una metodología utilizada para identificar el origen de un problema y aplicar soluciones efectivas. En el desarrollo del prototipo de Desparches, se aplicó este análisis para mejorar la funcionalidad y usabilidad de la plataforma.

### 6.1.1 Métodos

El problema se ve agravado por la falta de filtros personalizados para buscar eventos, lo que impide a los usuarios encontrar opciones acordes a sus intereses. Además, el proceso para registrar preferencias es complejo, lo que reduce la participación y fidelización de los usuarios. La ausencia de notificaciones automáticas sobre eventos relevantes disminuye la interacción y la posibilidad de atraer más asistentes. Asimismo, la baja conexión con organizadores y el desconocimiento de tendencias limitan la actualización y variedad de la oferta.

### 6.1.2 Tecnología

El rendimiento ineficiente en dispositivos móviles y la falta de integración con APIs de eventos reducen la usabilidad y actualización del sistema. Esto se traduce en dificultades para proporcionar información en tiempo real a los usuarios. Además, la alta competencia y los cambios constantes en tendencias requieren una adaptación continua que el sistema actual no ha logrado mantener. Problemas de conectividad pueden agravar la experiencia del usuario, causando frustración y desinterés.

### 6.1.3 Materiales

El contenido poco atractivo, como descripciones deficientes y la falta de imágenes o videos de los eventos, genera desinterés en los usuarios. Además, la información desactualizada crea desconfianza y reduce la efectividad de la plataforma. La ausencia de métricas de búsqueda impide optimizar la experiencia del usuario con base en sus necesidades reales.

### 6.1.4 Mano de Obra

La falta de expertos en UX/UI afecta negativamente la usabilidad de la plataforma, haciendo que la navegación no sea intuitiva. La escasa conexión con organizadores también limita la oferta de eventos y dificulta el acceso a información confiable y detallada.

* Medio Ambiente

El ecosistema digital en el que opera la plataforma se caracteriza por una alta competencia y cambios constantes en tendencias de eventos. Estos factores requieren una capacidad de adaptación rápida para no perder relevancia ante otras soluciones del mercado.

### 6.1.5 Medición

La falta de métricas de búsqueda y la ausencia de encuestas de satisfacción dificultan la mejora continua de la plataforma. Sin datos concretos sobre la experiencia del usuario, resulta complicado optimizar el sistema para hacerlo más eficiente y atractivo.

## Diagrama Causa raíz de nuestro prototipo

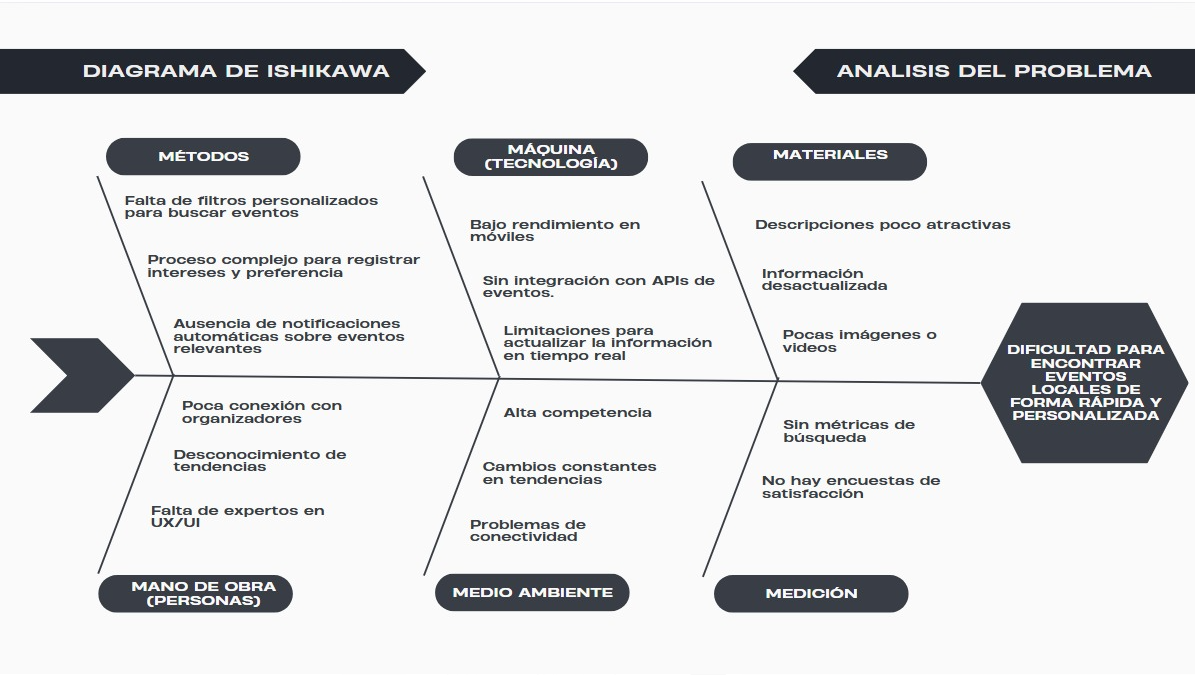


Ilustración 1 Captura del diagrama causa raíz de la plataforma Desparches en Canva. Fuente: Elaboración propia.

## 6.2 Análisis de lluvia de ideas

Para el desarrollo del prototipo de Desparches, se realizó una lluvia de ideas con el objetivo de definir las principales funcionalidades y mejorar la experiencia del usuario. Durante este proceso, se identificaron las siguientes propuestas clave:

### 6.2.1 Eventos Destacados

Sección especial donde se mostrarán los eventos más populares y exclusivos, ayudando a los usuarios a descubrir opciones atractivas sin necesidad de realizar búsquedas manuales extensas.

### 6.2.2 Integración con Redes Sociales

Permitir a los usuarios compartir eventos con facilidad y ver qué eventos están siguiendo o a cuáles asistirán sus amigos, fomentando la interacción social y la participación.

### 6.2.3 Sistemas de Calificación y Reseñas

Los usuarios podrán calificar y dejar opiniones sobre los eventos a los que asisten. Esto proporcionará referencias valiosas para futuros asistentes y contribuirá a la mejora continua de los organizadores.

### 6.2.4 Alertas en Tiempo Real

Se implementarán notificaciones automáticas para avisar a los usuarios sobre eventos cercanos a su ubicación, permitiendo que descubran actividades interesantes en el momento adecuado.

### 6.2.5 Recomendaciones Personalizadas

Gracias a algoritmos de inteligencia artificial, Desparches podrá sugerir eventos en función del historial de asistencia de cada usuario, optimizando la experiencia de búsqueda y aumentando la relevancia de las sugerencias.

### 6.2.6 Filtros Avanzados

Los usuarios podrán realizar búsquedas personalizadas según categorías, precios y tipos de eventos (cultural, deportivo, familiar, entre otros), facilitando la localización de actividades que se ajusten a sus preferencias.

### 6.2.7 Implementación de Técnicas de Optimización

* Para mejorar la experiencia de usuario, se contemplan varias mejoras técnicas, como:
* Reemplazar búsquedas manuales por sugerencias automáticas basadas en la ubicación.
* Integrar eventos y promociones en una sola vista personalizada.
* Incorporar funciones como “eventos similares”, que recomendarán actividades en función del historial del usuario.

## Diagrama de lluvia de ideas



Ilustración 2Captura del diagrama lluvia de ideas de la plataforma Desparches en Canva. Fuente: Elaboración propia.

## 6.3 Análisis de la matriz de priorización Impacto vs. Esfuerzo

Para optimizar el desarrollo del prototipo de Desparches, se utilizó una matriz de priorización Impacto vs. Esfuerzo, la cual permitió organizar y seleccionar las funcionalidades clave según su importancia y la dificultad de implementación (Project Management Institute, 2021).

### 6.3.1 Riesgos Críticos (Alta Prioridad)

* Los riesgos de mayor impacto y mayor probabilidad deben abordarse de inmediato, ya que pueden comprometer la operatividad y la seguridad de la plataforma. Entre ellos destacan:
* Caída total de la plataforma en eventos clave (Mayor - Casi Seguro): Este escenario representa un alto riesgo para la confiabilidad del sistema, afectando la experiencia del usuario y la reputación del servicio.
* Pérdida de datos de usuarios y eventos (Catastrófico - Casi Seguro): La seguridad de los datos es un aspecto crítico. Una fuga de información puede generar problemas legales y pérdida de confianza por parte de los usuarios.
* Brechas de seguridad y fuga de datos (Catastrófico - Muy Probable): Una vulnerabilidad en la plataforma podría ser explotada por atacantes, comprometiendo datos sensibles y generando problemas de cumplimiento normativo.

### 6.3.2 Riesgos Significativos (Moderada Prioridad)

* Estos riesgos pueden afectar la funcionalidad y experiencia del usuario, aunque su impacto es manejable con planes de contingencia adecuados:
* Errores en la integración con APIs de terceros (Moderado - Muy Probable): La incapacidad de integrar servicios externos puede limitar la funcionalidad del sistema.
* Problemas de escalabilidad con alto tráfico (Mayor - Muy Probable): Si la plataforma no es capaz de manejar una gran cantidad de usuarios simultáneamente, puede presentar fallos operativos en momentos críticos.
* Desinformación por datos erróneos en la plataforma (Alto - Posible): Datos incorrectos pueden afectar la toma de decisiones de los usuarios y generar una mala experiencia.

### 6.3.3 Riesgos Moderados (Baja Prioridad)

* Estos riesgos pueden impactar la experiencia del usuario, pero su resolución no es urgente:
* Curva de aprendizaje de usuarios y adopción (Menor - Posible): Puede resolverse con una mejor experiencia de usuario (UX) y tutoriales accesibles.
* Errores en pagos y cobros (Moderado - Posible): Afecta la confianza de los usuarios, pero se puede mitigar con pruebas constantes y soporte efectivo.
* Problemas de compatibilidad con distintos dispositivos (Mayor - Posible): Se puede reducir con pruebas en múltiples plataformas y versiones.

### 6.3.4 Riesgos Menores (Muy Baja Prioridad)

* Son problemas que no afectan significativamente la operatividad y pueden abordarse con mejoras incrementales:
* Dudas frecuentes sobre uso, solucionadas con FAQs y tutoriales (Insignificante - Poco Probable): Este tipo de inconvenientes se resuelve fácilmente con documentación clara y accesible.
* Dificultades en el servicio de soporte (Menor - Poco Probable): Un servicio de atención bien estructurado puede minimizar este riesgo.
* Fallas en la moderación de eventos (Moderado - Poco Probable): Se pueden mitigar con una combinación de herramientas automáticas y revisión manual.

## Diagrama matriz de priorización Impacto vs. Esfuerzo



Ilustración 3 Captura del diagrama matriz de priorización de la plataforma Desparches en Canva. Fuente: Elaboración propia.

## 6.5 Análisis del diagrama de flujo

El diagrama de flujo del prototipo Desparches muestra los pasos clave en la interacción de los usuarios y administradores con la plataforma.

### 6.5.1 Puntos positivos

* Claridad y estructura lógica: El diagrama sigue un flujo claro y ordenado, lo que facilita la comprensión del proceso.
* Opciones flexibles para el usuario: Se permite explorar sin necesidad de registrarse o reservar, ofreciendo una experiencia más libre.
* Decisiones bien definidas: Se presentan preguntas clave que guían al usuario de manera intuitiva.

### 6.5.2 Oportunidades de mejora

* Falta de validaciones: No se contempla qué sucede si el usuario introduce datos incorrectos al registrarse o iniciar sesión.
* Salida abrupta tras la reserva: Tras confirmar la reserva, el usuario solo tiene la opción de salir de la web. Se podría añadir la opción de ver el estado de la reserva o explorar más eventos.
* Exploración sin necesidad de inicio de sesión: No está claro si el usuario puede explorar eventos sin necesidad de registrarse, lo que podría ser un punto a aclarar.

## 6.6 Diagrama de flujo

### Diagrama de flujo para los usuarios

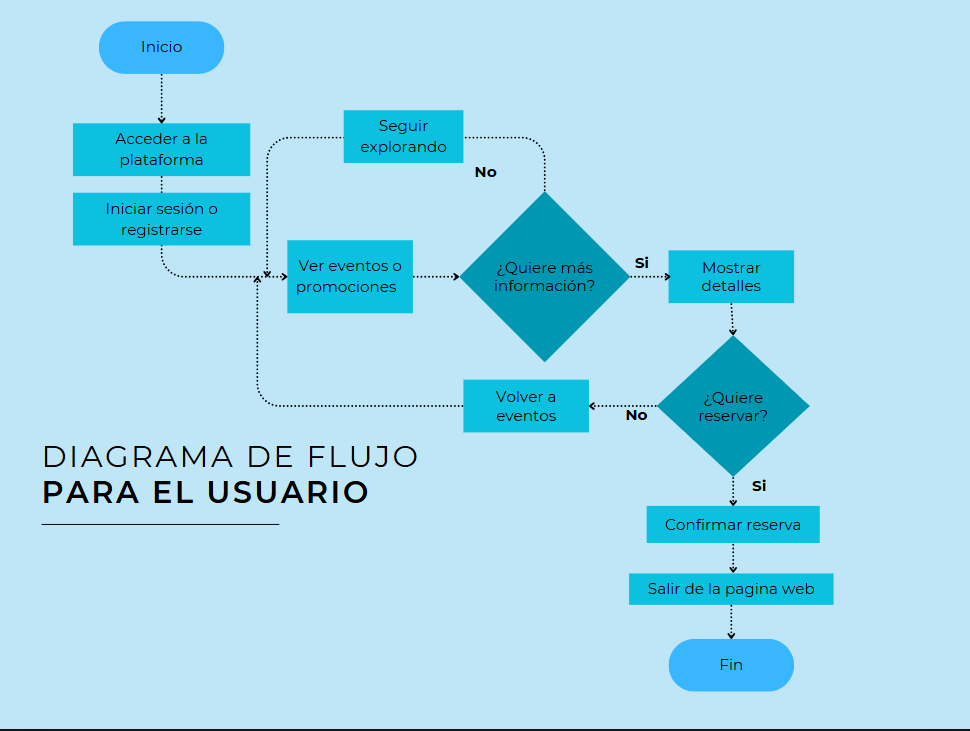


Ilustración 4 Captura del diagrama de fujo de usuarios de la plataforma Desparches en Canva. Fuente: Elaboración propia.

### Diagrama de flujo para los administradores

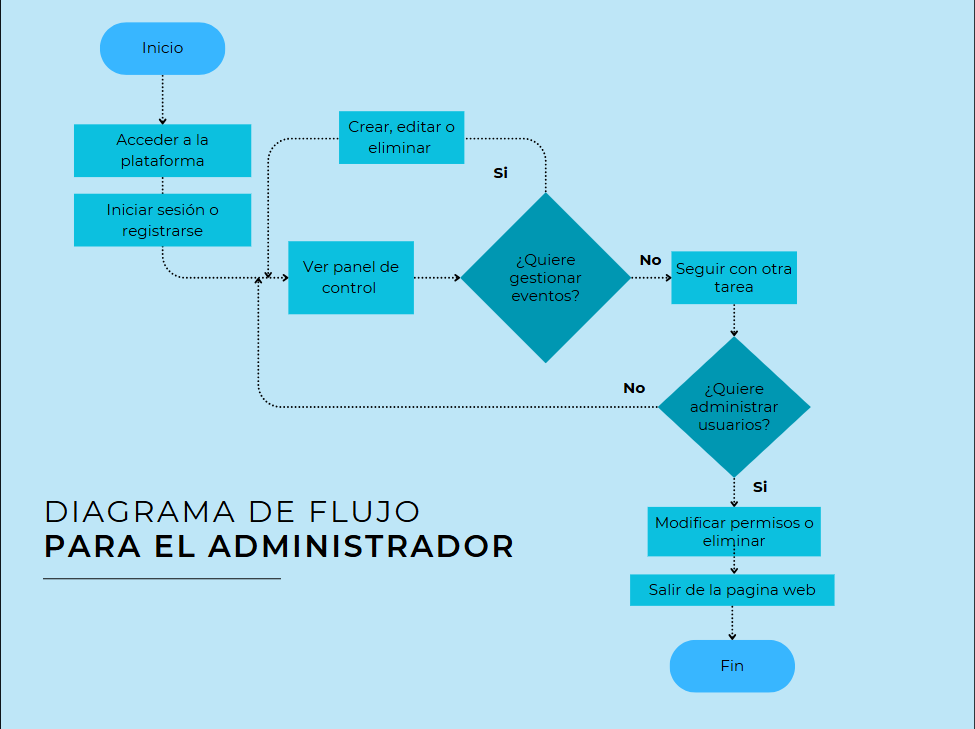


Ilustración 5 Captura del diagrama de flujo de administradores de la plataforma Desparches en Canva. Fuente: Elaboración propia.

## 6.7 Plataforma de diseño del prototipo: Figma

### 6.7.1 ¿Qué es Figma?

Figma es una plataforma de edición gráfica y diseño de interfaces. Además, es una plataforma online y colaborativa. Con Figma se puede hacer un poco de todo a nivel de diseño gráfico, desde diseñar páginas web e interfaces gráficas de aplicaciones, o crear publicaciones para redes sociales, hasta la posibilidad de poder crear presentaciones. Por este motivo, es una de las herramientas más valoradas por empresas y estudios de diseño gráfico. (Pixartprinting, s.f.)

### 6.7.2 Nuestro prototipo

En este caso utilizamos la aplicación de diseño Figma para el desarrollo de Desparches, se utilizó Figma para diseñar el prototipo de la plataforma permitir visualizar la experiencia del usuario antes de su implementación. Gracias a esta herramienta, se logró estructurar la interfaz, definir la disposición de los elementos y aunque inicialmente es un prototipo, se han integrado botones interactivos para mejorar la navegación (que en realidad son hipervínculos). Con el tiempo, se agregarán más funcionalidades para optimizar la experiencia del usuario y ponerle código.

## 6.8 Roles

### 6.8.1 ¿Que son los roles?

Un rol es lo que asignas a los usuarios en el sitio para permitirles realizar ciertas acciones. Estas acciones serán determinadas por las capacidades dadas al rol, y qué nivel de permiso se ha otorgado a cada capacidad.

### 6.8.2 Roles en nuestro prototipo

En Desparches, se identificaron diferentes roles dentro del sistema, como usuarios y administradores cada uno cumple una función específica dentro de la plataforma (EDteam, s.f.)

* Rol de usuario: Este rol está destinado a quienes acceden a la plataforma para descubrir eventos y promociones. Los usuarios pueden explorar el mapa interactivo, recibir recomendaciones según su ubicación y guardar eventos o descuentos de interés.
* Rol de administrador: Los administradores tienen un mayor nivel de acceso y son responsables de gestionar el contenido de la plataforma. Pueden agregar, modificar o eliminar eventos y promociones, verificar la autenticidad de las ofertas y supervisar el correcto funcionamiento del sistema para garantizar una experiencia óptima a los usuarios.

## 6.9 Criterios cardinales

### 9.9.1 ¿Qué son los criterios cardinales?

Los criterios cardinales son principios fundamentales en el diseño de ingeniería. Estos guían la funcionalidad, eficiencia y experiencia del usuario en un sistema, asegurando que cumpla con sus objetivos de manera óptima.

### 6.9.2 Criterios cardinales en nuestro prototipo:

En el desarrollo del prototipo de Desparches, se han considerado varios criterios cardinales del diseño en ingeniería:

* Usabilidad: El diseño del prototipo es intuitivo y permite una navegación sencilla, asegurando que los usuarios puedan encontrar eventos y promociones sin dificultad.
* Interactividad: Se han incorporado botones interactivos que simulan la navegación real dentro de la plataforma, brindando una experiencia cercana a la versión final.
* Funcionalidad: Aunque es un prototipo, representa fielmente la estructura y operatividad de la plataforma, permitiendo visualizar cómo será su funcionamiento en la etapa de desarrollo.
* Escalabilidad: El diseño ha sido estructurado para permitir futuras mejoras y nuevas funcionalidades sin comprometer la organización de la interfaz.
* Accesibilidad: El prototipo está diseñado para ser visualizado en distintos dispositivos, asegurando una correcta adaptación a diferentes tamaños de pantalla.

## 6.10 Solución de problemas

### 6.10.1 ¿Qué significa solucionar problemas en ingeniería de software?

En el desarrollo de cualquier software o sistema digital, pueden surgir desafíos relacionados con la usabilidad, rendimiento, seguridad, escalabilidad y otros aspectos clave. Solucionar problemas en ingeniería de software implica identificar estos obstáculos y aplicar estrategias para mejorar el código y la arquitectura, asegurando que el sistema funcione bien y brinde una buena experiencia a los usuarios. (Rodríguez, 2015)

### 6.10.2 Aplicación en nuestro prototipo

* Usabilidad:

### Problema: Algunos usuarios pueden tener dificultades para navegar por la plataforma si la interfaz no es lo suficientemente intuitiva.

Solución: Se implementaron botones interactivos bien distribuidos y un diseño limpio para facilitar la exploración (Nielsen, Usability 101: Introduction to Usability, 2012).

* Interactividad:

Problema: Al ser un prototipo en Figma, no cuenta con funciones dinámicas reales.

Solución: Se usaron hipervínculos simulados para que el usuario pueda visualizar la experiencia de navegación.

* Funcionalidad:

Problema: Aún no se pueden realizar búsquedas reales de eventos en el prototipo.

Solución: Se diseñó una estructura de filtros que podrá implementarse en la versión programada.

* Escalabilidad:

Problema: La plataforma debe permitir futuras expansiones sin perder orden en la interfaz.

Solución: Se diseñó con una estructura modular, facilitando la incorporación de nuevas funciones.

* Accesibilidad:

Problema: La plataforma debe ser accesible en distintos dispositivos.

Solución: Se diseñó con una interfaz adaptable a diferentes tamaños de pantalla para mejorar la experiencia del usuario.

## 6.11 Métodos

### 6.11.1 ¿Qué son los métodos en diseño de ingeniería?

Los métodos en diseño de ingeniería son estrategias y procesos utilizados para desarrollar soluciones eficientes y funcionales. Estos permiten estructurar el diseño, evaluar opciones y optimizar el rendimiento del producto o sistema.

### 6.11.2 Métodos aplicados en nuestro prototipo

Para el desarrollo del prototipo de Desparches, se emplearon varios métodos de diseño y desarrollo:

* Investigación y análisis de necesidades: Se identificaron las necesidades del usuario y las funcionalidades clave a través de la observación de plataformas similares y encuestas exploratorias.
* Prototipado con Figma: Se utilizó esta herramienta para diseñar la interfaz y simular la experiencia de usuario antes de la implementación final.
* Pruebas de usabilidad: Se realizaron pruebas internas para evaluar la navegación y funcionalidad del prototipo, asegurando una experiencia intuitiva.

## 6.12 Requisitos

### 6.12.1 ¿Qué son los requisitos en diseño de ingeniería?

Los requisitos en diseño de ingeniería son las condiciones y características que debe cumplir un producto o sistema para garantizar su funcionalidad, eficiencia y usabilidad. Se dividen en requisitos funcionales y no funcionales. (Visure Solutions, 2023)

### 6.12.2 Requisitos en nuestro prototipo

* Requisitos funcionales (definen lo que el sistema debe hacer):
* Permitir a los usuarios visualizar eventos y promociones en su área geográfica.
* Incluir un mapa interactivo con la ubicación de eventos y descuentos.
* Ofrecer una interfaz intuitiva para explorar actividades cercanas.
* Implementar botones interactivos para simular la navegación en la plataforma.
* Requisitos no funcionales (definen cómo debe funcionar el sistema):
* Diseño adaptable para distintos dispositivos (PC, y móvil).
* Interfaz sencilla y atractiva para una experiencia de usuario óptima.
* Capacidad de escalabilidad para futuras mejoras y nuevas funciones.
* Eficiencia en la carga y respuesta de la plataforma.

## Restricciones

### 6.13.1 ¿Qué son las restricciones en diseño de ingeniería?

Las restricciones son los límites o condiciones que deben considerarse en el desarrollo de un proyecto. Estas pueden ser técnicas, económicas, legales o de usabilidad, y afectan la planificación, diseño y ejecución del sistema (Martins, 2024).

### 6.13.2 Restricciones en nuestro prototipo

* Restricciones técnicas:
* El prototipo se desarrolla en Figma, por lo que aún no tiene una funcionalidad programada real.
* Los botones interactivos son hipervínculos simulados y no funcionalidades reales.
* El mapa interactivo es una representación estática en esta etapa inicial.
* Restricciones económicas:
* Se cuenta con recursos limitados para el desarrollo e implementación inicial.
* No se han integrado servicios pagos como bases de datos en la versión del prototipo.
* Restricciones de usabilidad:
* El diseño debe ser intuitivo para usuarios con distintos niveles de experiencia tecnológica.
* La interfaz debe ser clara y accesible en diferentes tamaños de pantalla.
* Restricciones de tiempo:
* El desarrollo del prototipo debe completarse en el tiempo estipulado para la entrega del proyecto.
* Las futuras mejoras dependen de la disponibilidad de tiempo para el desarrollo completo.

## 6.14 Design ThinKing:

Etapas:

## 6.15 Empatizar

En la etapa de Empatía del proceso de Design Thinking para el proyecto Desparches, el equipo se centró en comprender profundamente a los usuarios, dejando de lado suposiciones para escuchar y observar con atención sus emociones, comportamientos y experiencias reales. Esta fase permitió ver el mundo desde la perspectiva del usuario, entendiendo no solo lo que necesitan, sino lo que sienten y lo que verdaderamente los motiva a buscar eventos o actividades en su entorno.

Herramienta utilizada:

### 6.15.1 Mapa de la Empatía:

Hecho en Canva, el equipo utilizó esta herramienta visual para recolectar y organizar la información clave sobre el usuario en seis dimensiones: lo que piensa y siente, lo que ve, lo que oye, lo que dice y hace, sus frustraciones (dolores) y lo que lo motiva (ganancias). Esta estructura permitió detectar patrones comunes entre los usuarios de Desparches, facilitando la comprensión de sus necesidades tanto emocionales como prácticas. (**Anexo 2.** Detalles adicionales del mapa de empatía)

**Link canva:** [Mapa de la empatía ( Desparches)](https://www.canva.com/design/DAGkGBO4HYU/sLIG3x42ZrRM4EYgnF28Xg/view?utm_content=DAGkGBO4HYU&utm_campaign=designshare&utm_medium=link2&utm_source=uniquelinks&utlId=hf02750d210)

## 6.16 Definir

En la etapa de Definir del proceso de Design Thinking para el proyecto Desparches, se consolidó la información obtenida en la fase de empatía mediante herramientas clave como el User Persona, que permitió perfilar detalladamente al usuario objetivo, comprendiendo sus motivaciones, frustraciones y necesidades reales frente a la búsqueda de eventos y promociones locales. A partir de estas características, se elaboró un Diagrama de Afinidad, agrupando hallazgos en categorías significativas que revelaron patrones comunes entre los usuarios, como la necesidad de inmediatez, la geolocalización precisa y la preferencia por experiencias personalizadas.

Finalmente, se aplicó la técnica de ¿Cómo podríamos...? para transformar esos desafíos en oportunidades de diseño, con preguntas orientadoras como “¿Cómo podríamos conectar a los usuarios con eventos que se adapten a sus intereses en tiempo real?” o “¿Cómo podríamos facilitar el descubrimiento de promociones exclusivas en su entorno cercano?”. Estas herramientas guiaron la definición clara del problema central, sentando las bases para idear soluciones efectivas y centradas en el usuario.

Herramientas utilizadas:

### 6.16.1 User persona:

Hecho en Canva, la creación de arquetipos de usuarios nos permitió representar con detalle a las personas que utilizarían Desparches, incluyendo sus datos demográficos, motivaciones y frustraciones al buscar eventos o promociones locales. Esta herramienta nos ayudó a mantenernos enfocados en las verdaderas necesidades de nuestros usuarios, asegurando que las soluciones diseñadas respondieran a sus expectativas y hábitos de comportamiento digital. (**Anexo 3.** Detalles adicionales del user persona)

Link Canva: <https://www.canva.com/design/DAGjBxaqn-Y/hOvNgFk0Kx5EgGFkf9f35Q/edit?utm_content=DAGjBxaqn-Y&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

### 6.16.2 Diagrama de afinidad:

Hecho en Canva, Utilizamos el diagrama de afinidad para organizar y analizar la información recolectada durante la fase de investigación del proyecto Desparches. Agrupamos las ideas y hallazgos en seis categorías principales: Descubrimiento de Planes, Tecnología, Modelo de Negocio, Comunidad, Visualización y Confirmación de Eventos. Esta estructuración nos permitió identificar patrones en el comportamiento de los usuarios y detectar oportunidades clave para el diseño de nuestra solución. Gracias a esta herramienta, pudimos enfocar nuestros esfuerzos en aspectos críticos como la mejora de la experiencia de búsqueda, la integración de la comunidad y la claridad en la presentación de eventos y promociones. (**Anexo 4.** Detalles adicionales del diagrama de afinidad)

Link Canva: <https://www.canva.com/design/DAGlUutg87s/9Z4sbK25t-t9mRfk5M_atQ/edit?utm_content=DAGlUutg87s&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

### 6.16.3 Declaración del problema:

Hecho en Canva, Durante la etapa de definir del proyecto Desparches, empleamos la herramienta "Cómo podríamos..." para transformar los insights recopilados en preguntas que impulsaran soluciones creativas. Clasificamos las ideas en seis áreas clave: Personalización Avanzada, Interacción en Tiempo Real, Seguridad y Confianza, Experiencia Offline, Visualización y Confirmación de Eventos, generando preguntas como: “¿Cómo podríamos evitar la saturación de notificaciones sin perder relevancia?” o “¿Qué elementos visuales o datos clave debería tener la vista previa de un evento para facilitar la decisión de asistir?”. Esta técnica fue fundamental para enfocar los retos del proyecto en necesidades reales de los usuarios y facilitar una lluvia de ideas más dirigida y estratégica.(**Anexo 5.** Detalles adicionales de la declaración del problema)

Link Canva: <https://www.canva.com/design/DAGlUudXu8A/tE9yd-IrLSonQbHkkH97EA/edit?utm_content=DAGlUudXu8A&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

## 6.17 Idear

En la etapa de Idear, se buscó generar una amplia gama de ideas innovadoras y creativas para resolver los problemas identificados en la fase de empatía. Utilizando diversas técnicas de ideación, el equipo del proyecto Desparches exploró soluciones disruptivas que pudieran responder a las necesidades y deseos de los usuarios de manera efectiva. Se alentó la creatividad y la colaboración entre los miembros del equipo para asegurar que las propuestas fueran lo más completas posibles. Herramientas como SCAMPER, Brainstorming y Crazy 8’s fueron clave en este proceso, permitiendo la generación de ideas visuales y funcionales que luego se refinarían para crear una experiencia de usuario excepcional. Al finalizar la fase, las mejores ideas fueron seleccionadas para continuar con el proceso de diseño y prototipado, siempre manteniendo el enfoque en mejorar la experiencia del usuario.

Herramientas utilizadas:

### 6.17.1 Scamper:

Hecho en Canva, Esta herramienta permitió reinventar y optimizar funcionalidades clave de la plataforma mediante un enfoque estructurado de siete técnicas (Sustituir, Combinar, Adaptar, Modificar, Poner en otros usos, Eliminar, Reorganizar). Cada una de estas técnicas ayudó a pensar de manera diferente sobre cómo mejorar aspectos de la plataforma, desde la personalización de eventos hasta la integración de tecnologías innovadoras. (**Anexo 6.** Detalles adicionales del scamper)

Link Canva: <https://www.canva.com/design/DAGlGmqRVPg/UDrWCeq59gKWGCUh3xQjCQ/edit?utm_content=DAGlGmqRVPg&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

### 6.17.2 Brainstorming:

Hecho en Canva, Durante las sesiones de brainstorming, se generaron ideas como un mapa interactivo en tiempo real, un sistema de calificación de eventos, y recomendaciones personalizadas, entre otras. Estas propuestas fueron diseñadas para mejorar la accesibilidad y la interacción del usuario con los eventos locales. (**Anexo 7.** Detalles adicionales del brainstorming)

Link Canva: <https://www.canva.com/design/DAGlUi1LV0k/Fs9IS-hdItvBv_-o3z6U6g/edit?utm_content=DAGlUi1LV0k&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

### 6.17.3 Crazy 8’s:

Hecho en Canva, Esta técnica permitió generar rápidamente ocho ideas visuales en solo ocho minutos. Entre las propuestas surgieron conceptos como la integración de un mapa interactivo, sistemas de recomendación de eventos y un modo sin conexión para facilitar el acceso a los eventos guardados. (**Anexo 8.** Detalles adicionales del Crazy 8’s)

Link Canva: <https://www.canva.com/design/DAGlUr5xi04/P0Cmeq0ATzEYnOwVHS8oWw/edit?utm_content=DAGlUr5xi04&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

## 6.18 Prototipar

En la etapa de **Prototipar** del proceso de Design Thinking para el proyecto Desparches, el equipo se enfocó en transformar las ideas generadas en la fase anterior en representaciones tangibles. Se crearon prototipos que permitieron visualizar de manera sencilla y rápida cómo sería la interacción del usuario con la plataforma, ayudando a validar y mejorar las soluciones antes de llegar al diseño final.

Herramientas utilizadas:

### 6.18.1 Wireframes

Hecho en Canva, los wireframes son representaciones básicas de las pantallas de la plataforma, sin colores ni detalles finales, que muestran la disposición de los elementos y cómo se conectan entre sí. En este caso, se dibujaron las pantallas principales a mano y luego se digitalizaron en Figma. Esta herramienta permitió organizar las ideas, identificar posibles problemas de ubicación en la interfaz y mejorar la navegación antes de avanzar al diseño final. (**Anexo 9.** Detalles adicionales del Wireframes)

Link Canva: <https://www.canva.com/design/DAGlUhK9svU/5gqm-pypdWMEGwXfVzZcDQ/edit?utm_content=DAGlUhK9svU&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

### 6.18.2 Prototipos Rápidos

Hecho en figma, Los prototipos rápidos son maquetas funcionales de la plataforma que permiten ver cómo se comportarán las pantallas principales y probar la interacción de los botones y menús. Se crearon pantallas clicables en Figma, enfocándose en las secciones más importantes como inicio, búsqueda y reserva. Esta herramienta fue clave para detectar errores temprano y garantizar que el equipo comprendiera cómo funcionaría la plataforma, permitiendo realizar correcciones de forma rápida antes de realizar el diseño final. (**Anexo 10.** Detalles adicionales de prototipos rápidos)

### Link del prototipo en figma en computador

* Usuario:
* <https://www.figma.com/proto/kcHQJEOxhfJquLauWHwJ4B/prototipo1-dise%C3%B1o?node-id=1-2&t=TbOl2tm4mxEsjruD-1&scaling=scale-down&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=1%3A4478>
* Administrador:
* <https://www.figma.com/proto/jvxvgJwycH6YeTLQ32XxFD/Prototipo-empresas?node-id=38-3&p=f&t=e6vCAjWxamcjhcy0-0&scaling=scale-down&content-scaling=fixed&page-id=17%3A14&starting-point-node-id=38%3A3>

### Link del prototipo en figma en celular

* + Usuario:
* <https://www.figma.com/proto/iuue4diUQVcQsBMV66za1z/Prototipo-celular-Usuario?node-id=1-3&p=f&t=AaXK5ezhBCMXc7lK-0&scaling=scale-down&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=1%3A3>
  + Empresa:
* <https://www.figma.com/proto/jvxvgJwycH6YeTLQ32XxFD/Prototipo-empresas?node-id=83-67&p=f&t=e6vCAjWxamcjhcy0-0&scaling=scale-down&content-scaling=fixed&page-id=0%3A1>

## 6.19 Testear

En esta etapa del proyecto *Desaparches*, buscamos evaluar la efectividad del prototipo con personas reales del entorno escolar que enfrentan o conocen situaciones de discriminación. Se aplicaron pruebas directas donde los usuarios interactuaron con el material desarrollado y compartieron sus percepciones. Esta fase es clave porque permite identificar si el mensaje se entiende, si la propuesta es útil y cómo puede mejorarse para generar un mayor impacto en la comunidad.

Herramientas utilizadas:

### 6.19.1 Matriz de retroalimentación:

Hecho en Canva, es una herramienta que usamos para organizar lo que los participantes dijeron sobre el prototipo. Clasificamos sus respuestas en cuatro categorías: lo que les gustó, lo que no les gustó, lo que les generó dudas y las sugerencias que propusieron. Esto nos permitió visualizar rápidamente los puntos fuertes y débiles del diseño de *Desaparches*, y nos ayudó a definir los ajustes necesarios. (**Anexo 11.** Detalles adicionales de la matriz de retroalimentación)

Link Canva: <https://www.canva.com/design/DAGjBxa7gnI/RmrihC7mhKOr52AyTVi4gw/edit?utm_content=DAGjBxa7gnI&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

### **6.19.2 User Journey Map**

Hecho en Canva, con esta herramienta se visualizó el proceso completo del usuario desde su primer contacto con el sistema hasta la experiencia final. Permite identificar las emociones, frustraciones y puntos críticos que el usuario puede experimentar durante su interacción con el proyecto. Se enfoca en entender mejor las necesidades y expectativas en cada etapa del recorrido, facilitando así las mejoras en la experiencia general. (**Anexo 12.** Detalles adicionales del user journey map)

Link Canva: <https://www.canva.com/design/DAGjB03P9ec/xg0FpTerCR0C8acv-LifxQ/edit?utm_content=DAGjB03P9ec&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

## 6.20 UX:

## 6.21 Usabilidad

La usabilidad se refiere a la facilidad con la que los usuarios interactúan con un sistema para lograr sus objetivos de manera eficiente ( International Organization for Standardization, 2018). En "Desparches", esto se evidencia en:

* Navegación intuitiva: Menú con categorías claras ("Mis Eventos", "Eventos Recomendados").

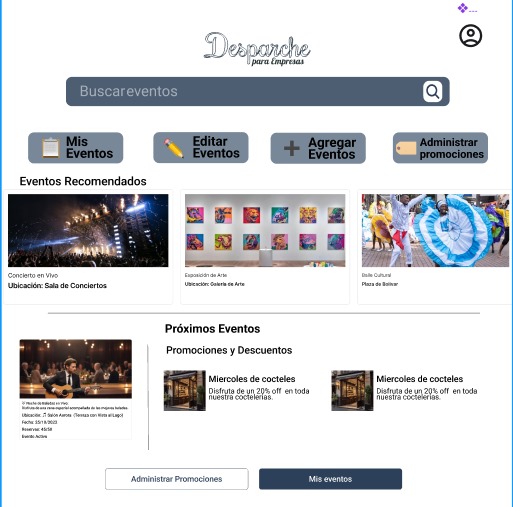


Ilustración 6 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

* Reducción de errores: Campos obligatorios marcados en formularios (ej. "Agregar Promociones").

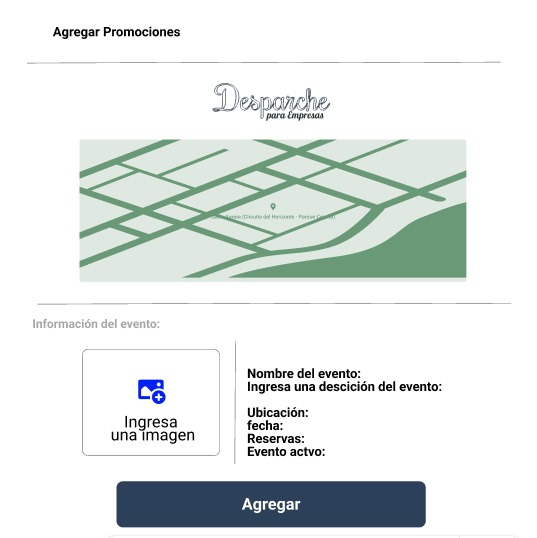


Ilustración 7 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

* Feedback inmediato: Confirmaciones visuales al agregar eventos ("Label 'Agregado'").

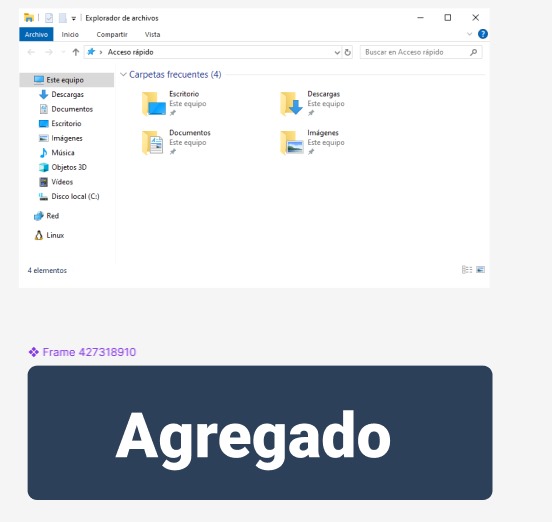


Ilustración 8 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

## 6.22 Accesibilidad

La accesibilidad garantiza que el sistema sea usable por personas con discapacidades (World Wide Web Consortium, 2018). Se observa:

* Contraste de colores: Texto legible sobre fondos oscuros/claros (ej. botones "Agregar").



Ilustración 9 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

* Estructura semántica: Uso de encabezados para jerarquía en "Perfil" y "Calendario".



Ilustración 10 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

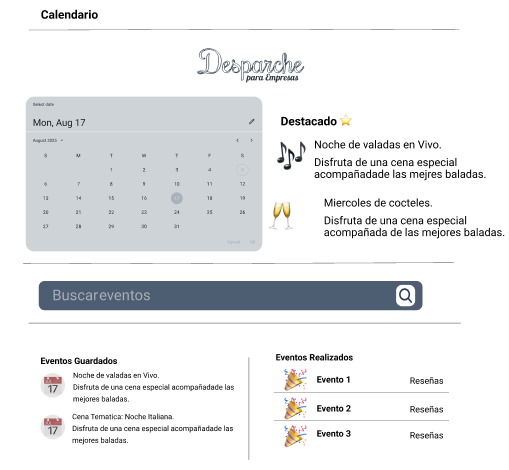


Ilustración 11 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

* Alternativas texto alternativo para íconos, garantizando que usuarios con discapacidades visuales puedan entender la función de cada elemento mediante lectores de pantalla.



Ilustración 12 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.



Ilustración 13 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

## 6.23 Interacción Humano-Computadora (HCI)

La HCI estudia cómo las personas usan sistemas digitales. En el proyecto:

* Consistencia: Diseño uniforme en tarjetas de eventos (ej. "Microsales de cocteles").

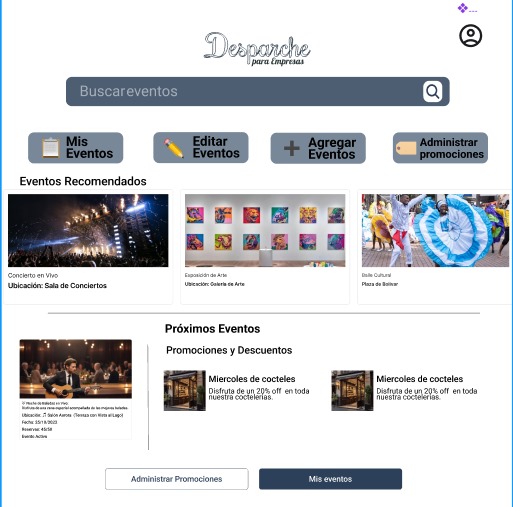


Ilustración 14 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

* Lenguaje natural: Frases como "Disfruta de una cena especial..." alineadas al usuario.



Ilustración 15 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

* Control y libertad: Opciones para editar o eliminar eventos ("Administrar Promociones").



Ilustración 16 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

## 6.24 Principios de UX Aplicados

### Diseño centrado en el usuario:

Ejemplo: Sección "Perfil" con eventos guardados/realizados para personalización.



Ilustración 17 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

### Jerarquía visual:

Uso de negritas y títulos en "Buscar eventos" para priorizar información.

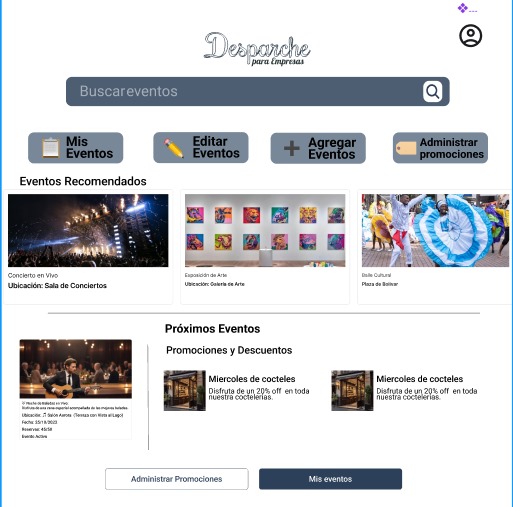


Ilustración 18 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

### Consistencia:

Botones similares en todas las pantallas (ej. "Agregar" en formularios).

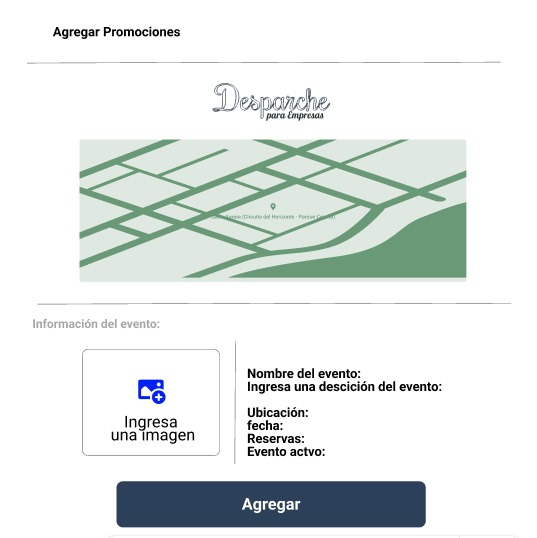


Ilustración 19 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

### Retroalimentación (Feedback):

Mensajes como "Agregado" confirman acciones (Nielsen, Usability 101: Introduction to Usability, 2021).

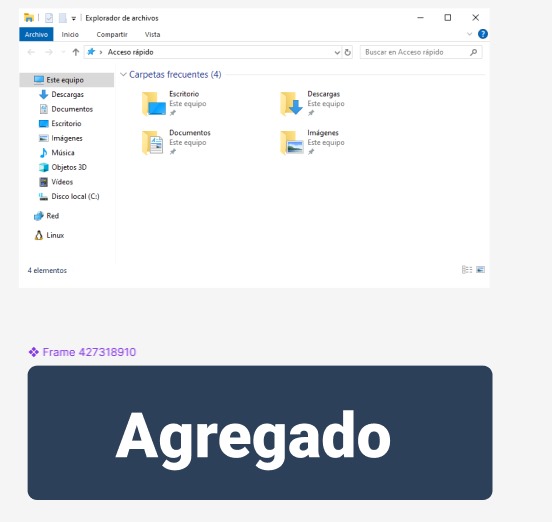


Ilustración 20Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

### Eficiencia:

Filtros rápidos ("distancia, precio") en "Administrar Promociones".



Ilustración 21 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

## 6.25 Diferencias Relevantes

### UX vs UI:

* UX: Flujo intuitivo para reservar eventos.
* UI: Diseño visual de botones y tarjetas.

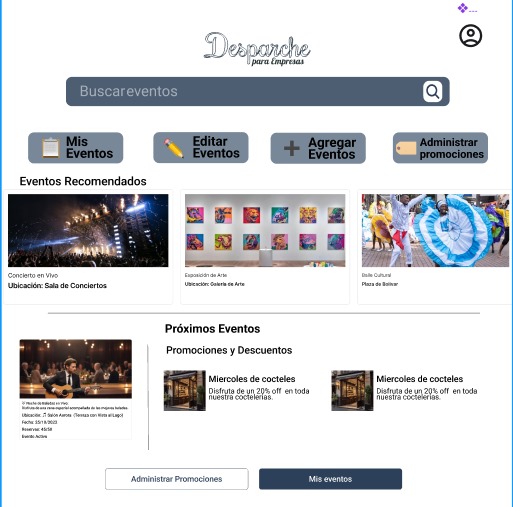


Ilustración 22 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

### Usabilidad vs Accesibilidad:

* Usabilidad: Formularios fáciles de completar.
* Accesibilidad: Contraste de colores para usuarios con baja visión.



Ilustración 23 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

### Interacción vs Experiencia:

* Interacción: Clic en "Agregar Promoción".
* Experiencia: Satisfacción al encontrar eventos rápidamente.



Ilustración 24 Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

## 6.26 Arquitectura de la información

9.1 ¿Qué es y por qué la usamos?

En Desparches, se aplicaron principios de arquitectura de la información para facilitar la navegación y la búsqueda de eventos. El contenido se organizó en secciones claras como "Eventos destacados", "Promociones cercanas", "Mis eventos" y "Perfil". Cada sección cuenta con una jerarquía visual que permite al usuario identificar rápidamente lo más importante.

Además, se diseñó un menú principal fijo que permite acceder a las funciones principales desde cualquier pantalla. Los eventos se presentan en tarjetas visuales con información clave (nombre, fecha, lugar), y los filtros ayudan a refinar la búsqueda. Esta estructura organizada mejora la usabilidad, reduce la carga cognitiva y asegura una experiencia más intuitiva para todos los usuarios.

Esta arquitectura fue diseñada pensando en mejorar la usabilidad y la experiencia del usuario, permitiendo que cada visitante encuentre lo que busca de forma rápida, sin esfuerzo y desde cualquier dispositivo.

### 6.26.1 Evidencia visual: Capturas del prototipo en Figma

- **Menú principal fijo**

Esta captura muestra la estructura superior de navegación de la plataforma. Se incluyen las secciones más usadas por los usuarios como “Inicio”, “Eventos”, “Filtros” y “Iniciar sesión”. Este menú está disponible en todas las pantallas para mejorar la orientación del usuario.

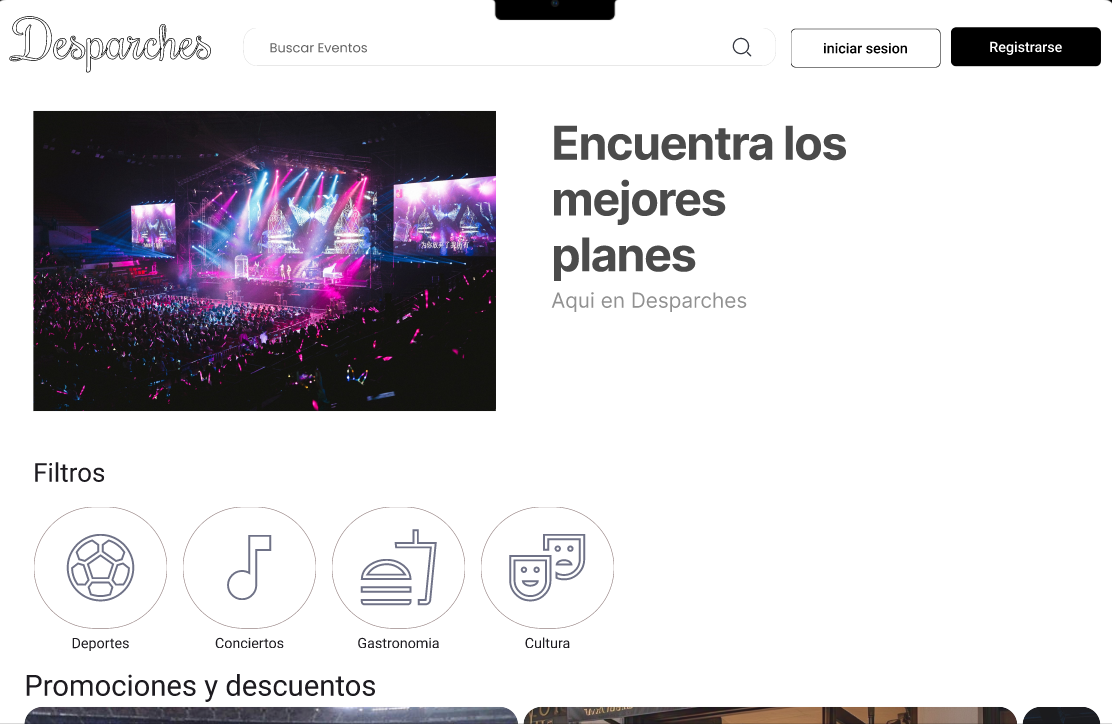


Ilustración  Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

**- Organización de tarjetas de eventos**

Se presentan los eventos mediante tarjetas visuales con nombre, imagen, lugar y tipo de evento. Esta estructura ayuda al usuario a identificar rápidamente las actividades relevantes y facilita la exploración visual.

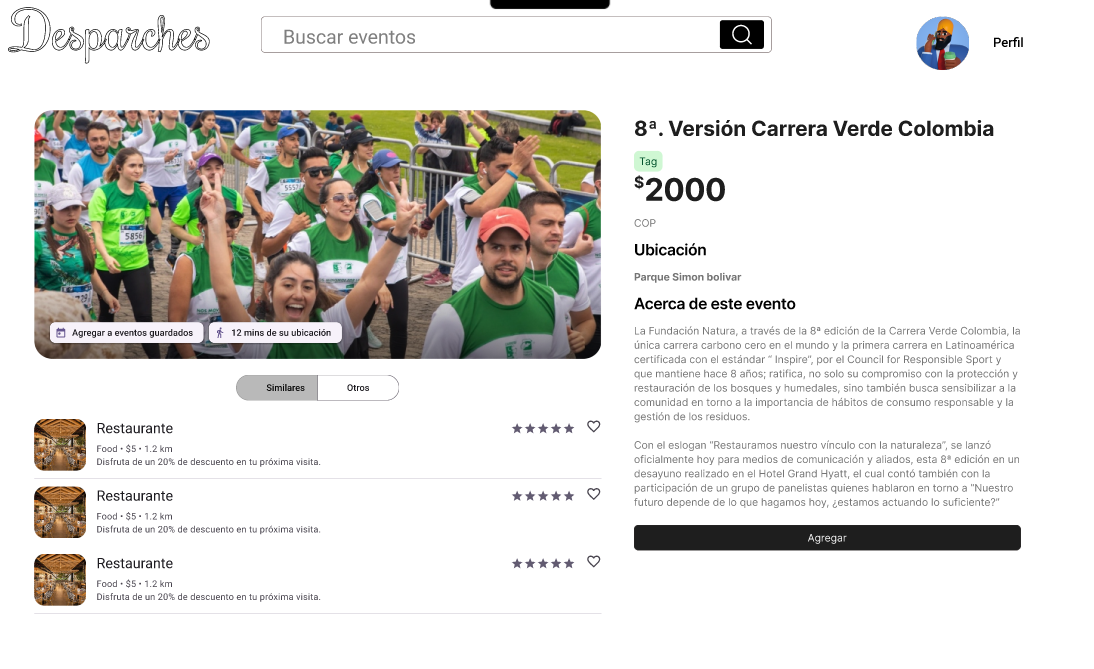


Ilustración  Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

**-** Sistema de filtros personalizados

Los filtros permiten al usuario refinar su búsqueda según categoría, ubicación, precio o tipo de actividad. Esto mejora la eficiencia en la navegación y la precisión de los resultados.

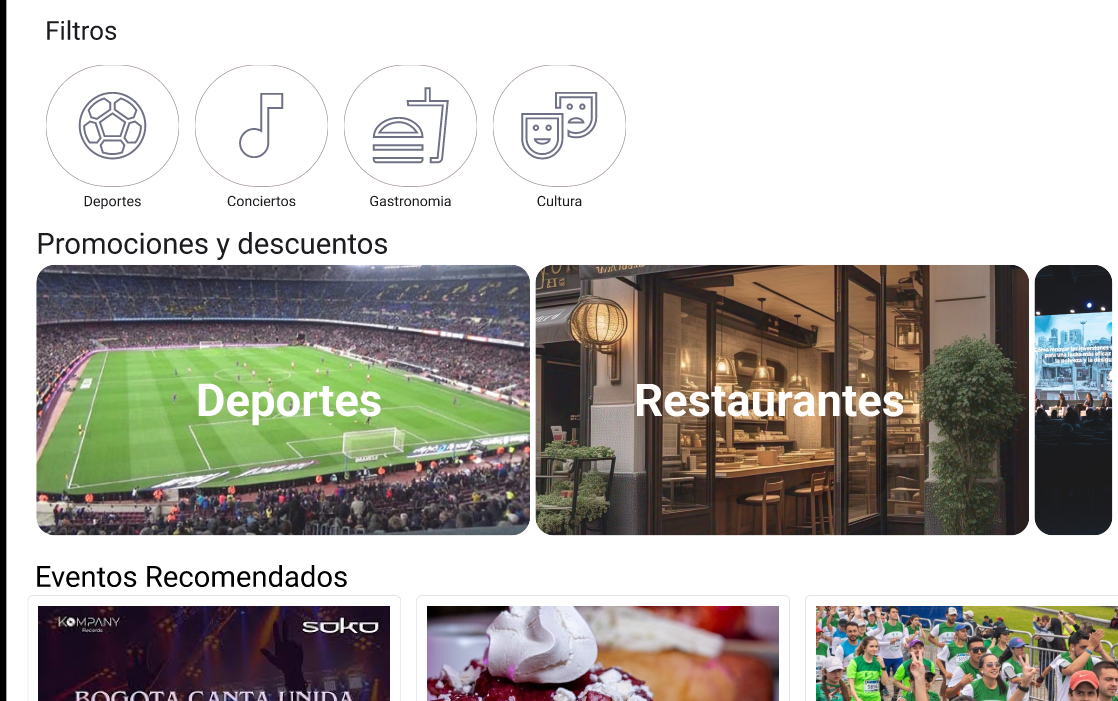


Ilustración  Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

**-** Jerarquía de información en la pantalla de detalles del evento

En esta vista se aplicó una jerarquía de contenidos, donde el título, la fecha y el botón de acción están en la parte superior. Luego, se encuentra la descripción detallada y las recomendaciones.

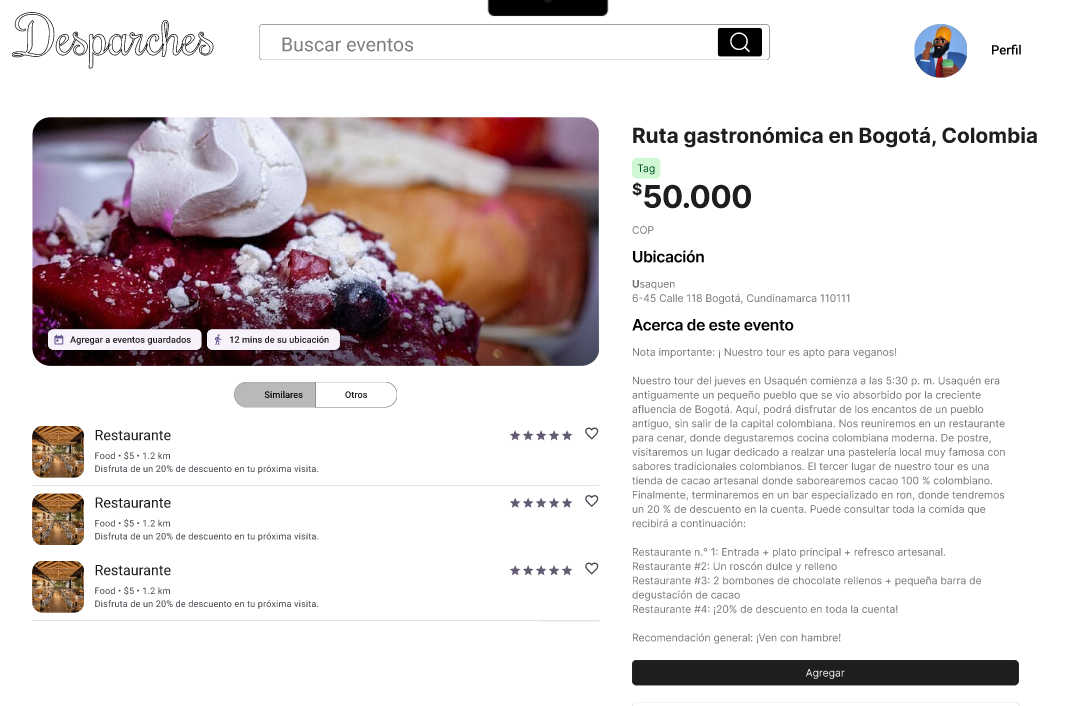


Ilustración  Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

Las decisiones tomadas en la arquitectura de la información del prototipo buscan que los usuarios encuentren contenido relevante de manera clara y rápida. Gracias a una estructura lógica y visualmente organizada, la plataforma Desparches mejora la experiencia de quienes buscan actividades y promociones en su entorno.

## 6.27 Diseño de interacción

El diseño de interacción se encarga de definir cómo las personas interactúan con una plataforma digital. Su objetivo es lograr que cada acción del usuario (como hacer clic, tocar un botón, desplazarse o recibir una respuesta) sea clara, eficiente y satisfactoria.

En el prototipo de Desparches se aplicaron varios principios del diseño de interacción para garantizar una experiencia fluida e intuitiva:

### - Botones interactivos con retroalimentación

En la plataforma, se diseñaron botones como "Agregar evento", "Guardar" y "Reservar", que simulan acciones reales mediante hipervínculos en Figma. Estos botones responden visualmente al hacer clic y confirman la acción con mensajes como “Agregado”, lo que mejora la experiencia del usuario al brindar retroalimentación inmediata.

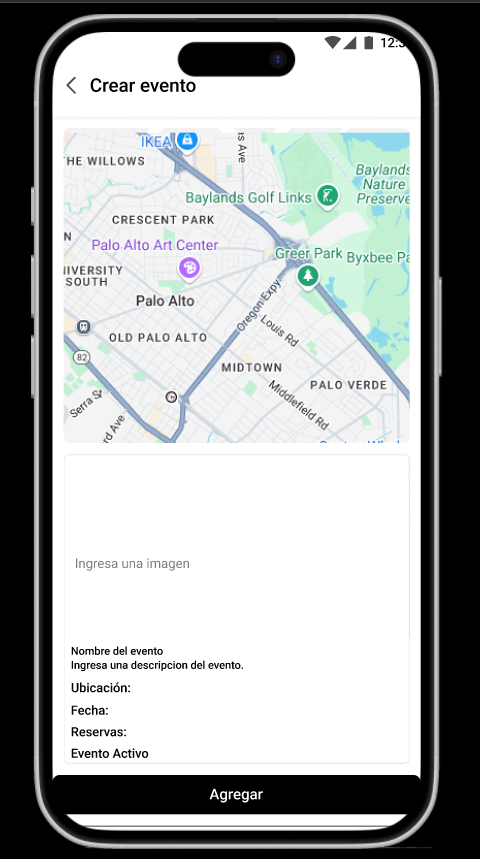


Ilustración  Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

- Navegación consistente entre pantallas

En todas las pantallas se mantuvieron elementos constantes, como el menú inferior y los íconos de acción, lo que permite que el usuario entienda rápidamente cómo moverse por la plataforma sin confusión. Este patrón coherente es clave para que la interacción sea fluida y predecible.



Ilustración  Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

### -Flujo de interacción: buscar - ver - reservar

El flujo para explorar eventos fue diseñado con pasos claros y ordenados: el usuario puede buscar eventos, ver detalles, aplicar filtros y finalmente reservar. Este diseño guiado facilita la navegación, especialmente para personas que no están familiarizadas con plataformas similares.



Ilustración  Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

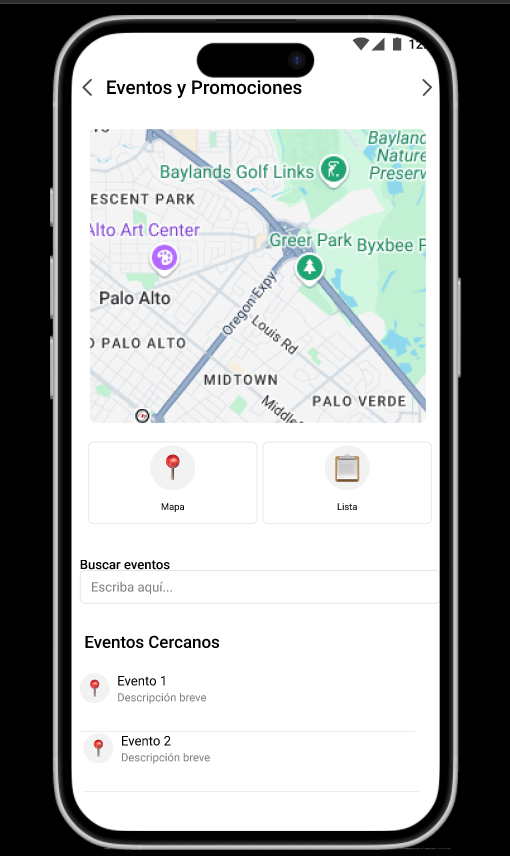


Ilustración  Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

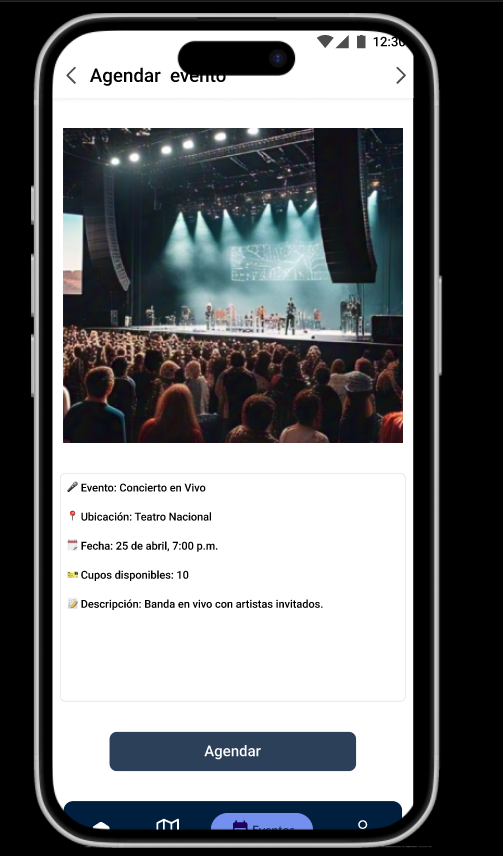


Ilustración  Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

### -Control y libertad del usuario

La plataforma permite que el usuario explore libremente, filtre por preferencias, y edite o elimine eventos desde su perfil. Esta libertad le da más control sobre su experiencia y evita frustraciones.



Ilustración  Captura del prototipo de la plataforma Desparches en Figma. Fuente: Elaboración propia.

Gracias al diseño de interacción aplicado en Desparches, se logró una experiencia centrada en el usuario. Las acciones son claras, las respuestas del sistema son visibles y predecibles, y la navegación se adapta a distintos dispositivos. Esto garantiza que cada interacción aporte valor y comodidad al usuario.

## 6.14 Metodologías de Desarrollo de Software

6.14.1 Rational Unified Process (RUP)  
El Proceso Unificado de Rational, propuesto por Jacobson, Booch y Rumbaugh (1999), es una metodología iterativa e incremental para el desarrollo de software que facilita la gestión de proyectos complejos. Según Kruchten (2003), RUP se basa en tres pilares: desarrollo dirigido por casos de uso, arquitectura centrada en el sistema y desarrollo iterativo, lo que permite controlar riesgos y garantizar la calidad. Estudios de Rational Software Corporation (2001) han evidenciado que su uso reduce defectos tardíos y mejora la documentación arquitectónica, aspectos clave en plataformas digitales como Desparches.

6.14.2 Metodologías Ágiles - SCRUM  
SCRUM, descrito por Schwaber y Sutherland (2020), es una metodología ágil especialmente útil para entornos con requisitos cambiantes y equipos multidisciplinarios. La revisión de Dybå y Dingsøyr (2008) muestra que SCRUM mejora la satisfacción del cliente y la productividad del equipo, favoreciendo la adaptabilidad en el desarrollo de plataformas interactivas. Cockburn (2006) propone combinar RUP con SCRUM para obtener tanto estructura inicial como flexibilidad continua, recomendación valiosa para proyectos como Desparches.

## 6.15 Arquitecturas de Software para Plataformas Digitales

6.15.1 Arquitecturas Multicapa  
Según Fowler (2002), las arquitecturas multicapa facilitan la mantenibilidad y escalabilidad al separar responsabilidades en capas lógicas. Bass, Clements y Kazman (2012) destacan su efectividad para plataformas web, permitiendo integración con servicios externos, crucial para Desparches al manejar eventos y usuarios diversos.

6.15.2 Microservicios y Arquitecturas Distribuidas  
Newman (2015) plantea que los microservicios permiten escalabilidad horizontal y despliegues independientes. Dragoni et al. (2017) confirman que estas arquitecturas aumentan la resiliencia y facilitan equipos distribuidos, ventajas importantes para plataformas que requieren crecimiento y actualización constante, como Desparches.

## 6.16 Sistemas de Geolocalización y Computación Espacial

6.16.1 Tecnologías de Geolocalización y Sistemas de Información Geográfica (GIS)  
Longley et al. (2015) definen los fundamentos de GIS para servicios basados en ubicación, que combinan GPS, triangulación y WiFi para mejorar precisión (Schiller y Voisard, 2004). En Desparches, estos sistemas permiten ofrecer eventos y actividades cercanas y personalizadas.

6.16.2 Location-Based Services (LBS)  
Küpper (2005) identifica los componentes críticos para aplicaciones LBS: precisión espacial, relevancia contextual y personalización, mientras Junglas y Watson (2008) relacionan la adopción con la facilidad de uso y utilidad percibida, factores esenciales para la experiencia en Desparches.

6.16.3 Algoritmos de Recomendación Espacial  
Bao et al. (2015) y Lian et al. (2014) demuestran que integrar filtrado colaborativo con datos geoespaciales mejora la precisión y satisfacción de los usuarios, permitiendo que Desparches sugiera eventos relevantes basados en preferencias y proximidad.

## 6.17 Experiencia de Usuario y Diseño de Interfaces

6.17.1 Principios de Usabilidad Móvil y Diseño Responsivo  
Marcotte (2010) estableció que el diseño responsivo es clave para que interfaces se adapten a diferentes dispositivos. Luke (2011) enfatiza el diseño “mobile-first” para mejorar la experiencia móvil, fundamental para Desparches, cuyo público usa principalmente smartphones.

6.17.2 Interacción Táctil  
Wigdor y Wixon (2011) señalan que la optimización de interfaces táctiles aumenta la eficiencia y satisfacción en dispositivos móviles. Hoober y Berkman (2011) confirman que una buena jerarquía visual y navegación gestual incrementan la tasa de tareas completadas, mejorando la interacción en plataformas como Desparches.

6.17.3 Comportamiento del Usuario en Plataformas de Eventos  
Liu et al. (2016) encontraron que la relevancia geográfica, la afinidad por categorías y las recomendaciones sociales influyen en la participación, mientras Brown y Chalmers (2003) destacan que la confianza depende de la calidad de información y facilidad de uso, aspectos que Desparches debe priorizar.

## 6.18 Plataformas Digitales y Modelos de Negocio

6.18.1 Economías de Plataforma y Efectos de Red  
Parker, Van Alstyne y Choudary (2016) explican que el éxito de plataformas depende de efectos de red positivos entre usuarios y proveedores, mientras Eisenmann et al. (2011) destacan que alcanzar masa crítica rápido es vital. Desparches, al conectar usuarios con eventos locales, debe fomentar esta interacción para crecer.

6.18.2 Monetización en Plataformas Locales  
Kumar (2014) sugiere que modelos híbridos (freemium, comisiones y publicidad) son sostenibles, y Rochet y Tirole (2003) establecen estrategias de precios óptimas para mercados bilaterales, claves para la viabilidad económica de Desparches.

## 6.19 Tecnologías de Desarrollo Web y Móvil

6.19.1 Frameworks de Desarrollo Multiplataforma  
Malavolta et al. (2015) y Biørn-Hansen et al. (2019) demuestran que React Native ofrece un buen balance entre rendimiento y eficiencia, ideal para aplicaciones móviles como Desparches.

6.19.2 Arquitecturas Backend Modernas  
Newman (2015) y Chaniotis et al. (2015) destacan Node.js para aplicaciones con alta concurrencia, importante para manejar múltiples usuarios simultáneos en Desparches.

## 6.20 Seguridad y Privacidad en Aplicaciones Geolocalizadas

6.20.1 Protección de Datos de Ubicación y Cumplimiento Regulatorio  
Dwork (2008) y Shokri et al. (2011) desarrollaron técnicas para proteger la privacidad del usuario en servicios de ubicación, mientras Hoofnagle et al. (2019) y Martins et al. (2020) destacan la importancia del cumplimiento de normativas como GDPR y LGPD para ganar confianza. (**Anexo 13.** Detalles adicionales del marco teórico)

# Metodología

El proyecto "Desparches" adopta un enfoque metodológico híbrido que combina las fortalezas del Proceso Unificado de Rational (RUP) y Scrum, integrando la estructuración por fases de RUP con la flexibilidad y adaptabilidad de Scrum. Esta combinación permite gestionar eficientemente la complejidad del desarrollo, asegurando tanto el cumplimiento de los requisitos como la capacidad de adaptación a cambios durante el ciclo de vida del proyecto.

## 7.1 Uso de la metodología RUP

El proceso de desarrollo del proyecto *Desparches* se basa en el uso de la metodología RUP (Proceso Unificado Rational), que organiza el ciclo de vida del software en cuatro fases: Inicio, Elaboración, Construcción y Transición. Esta estructura permite una planificación detallada, gestión de riesgos y validación progresiva del producto.

* **Inicio:** Se define el alcance general del sistema, se identifican los casos de uso críticos y se realiza una evaluación inicial de riesgos.
* **Elaboración:** Se construyen prototipos y se valida la arquitectura del sistema. Además, se detallan los requisitos funcionales y no funcionales.
* **Construcción:** Se lleva a cabo el desarrollo de los componentes del sistema y su integración, siguiendo los casos de uso definidos.
* **Transición:** Se realizan pruebas finales, documentación, capacitación y entrega al usuario final.

Esta metodología estructurada facilita una visión clara del progreso del proyecto y permite tomar decisiones fundamentadas durante cada etapa del desarrollo.

(**Anexo 14.** Detalles adicionales del la metodología RUP)

(**Anexo 15.** Detalles adicionales de la plantilla de vision)

## 7.2 Ingeniería de Requerimientos en la etapa de Inicio

Durante la fase de Inicio del proyecto, se aplicó la Ingeniería de Requerimientos como un proceso sistemático orientado a identificar, documentar y validar las necesidades del sistema Desparches. Esta fase es clave dentro de la metodología RUP, ya que permite establecer una base sólida sobre la cual se construirá el sistema, alineando las expectativas de los usuarios con las capacidades técnicas del desarrollo.

El proceso se llevó a cabo empleando diversas técnicas, actividades y herramientas, entre las cuales destacan:

**1. Técnicas de elicitación:**

* **Entrevistas** con stakeholders clave (usuarios finales, organizadores de eventos y administradores), para recopilar información directa.
* **Talleres de requisitos**, que permitieron identificar necesidades compartidas y generar consenso.
* **Análisis de competidores (benchmarking)**, con el fin de establecer funcionalidades diferenciadoras frente a otras plataformas existentes.

**2. Actividades clave:**

* **Identificación y priorización de casos de uso críticos**, como la búsqueda geolocalizada de eventos o el registro de usuarios.
* **Definición de requisitos funcionales y no funcionales**, según el estándar IEEE 830, que garantizan un desarrollo alineado con los objetivos del negocio y con criterios de calidad como rendimiento, disponibilidad y seguridad.
* **Creación de un glosario técnico**, con el propósito de unificar términos y evitar ambigüedades en el diseño e implementación.

**3. Herramientas utilizadas:**

* **Plantillas estandarizadas para la especificación de requisitos (SRS).**
* **Diagramas de casos de uso UML,** para representar de forma visual las interacciones entre actores y el sistema.
* **Matrices de trazabilidad**, que permiten mantener la relación entre cada requisito y los objetivos del sistema, asegurando que nada relevante quede fuera del desarrollo.

(**Anexo 16.** Detalles adicionales Ingeniería de Requerimientos)

(**Anexo 17.** Detalles adicionales de Casos de Uso)

(**Anexo 18.** Detalles adicionales de requerimientos)

## 7.3 Diagramas UML como apoyo para el análisis y diseño

os diagramas UML juegan un papel fundamental en el análisis y diseño del sistema, proporcionando representaciones visuales claras que facilitan la comprensión de la estructura, el comportamiento y las relaciones del sistema. En el proyecto *Desparches* se utilizan al menos seis tipos de diagramas UML para cubrir diferentes aspectos del desarrollo:

1. **Diagrama de casos de uso:**  
   Describe las interacciones entre los actores (usuarios finales, organizadores, administradores) y el sistema. Ejemplos incluyen casos como "Buscar eventos", "Registrar usuario" o "Publicar promoción". Se representan flujos básicos y alternativos para abarcar distintos escenarios.
2. **Diagrama de secuencia:**  
   Detalla el orden de mensajes intercambiados entre objetos durante operaciones críticas, como el proceso de reserva o compra de entradas para un evento, mostrando cómo se desarrolla la interacción en el tiempo.
3. **Diagrama de clases:**  
   Representa la estructura estática del sistema, mostrando clases principales como Usuario, Evento, Promoción, y sus relaciones (herencia, asociación, composición). Por ejemplo, la clase Usuario se especializa en UsuarioFinal, Organizador y Administrador, cada uno con sus atributos y métodos específicos.
4. **Diagrama entidad-relación (ER):**  
   Modela la estructura de la base de datos, definiendo las entidades, sus atributos y las relaciones entre ellas, lo cual es clave para el diseño del almacenamiento de la información del sistema.
5. **Diagrama de estados:**  
   Describe los posibles estados por los que puede pasar un objeto durante su ciclo de vida y las transiciones entre estos estados. Por ejemplo, el estado de un evento puede ser "Creado", "Publicado", "Cancelado" o "Finalizado".
6. **Diagrama de colaboración:**  
   Muestra las interacciones entre objetos y sus enlaces, enfatizando cómo colaboran para cumplir con un comportamiento específico del sistema.

Estos diagramas UML facilitan la comunicación efectiva entre el equipo de desarrollo y los stakeholders, garantizando que todos tengan una comprensión compartida y clara del sistema, desde sus requisitos hasta su diseño detallado.

(**Anexo 19.** Detalles adicionales de Diagramas UML )

## 7.4 Uso de la metodología Scrum

Scrum se aplica de manera complementaria para gestionar de forma ágil el desarrollo de Desparches, permitiendo adaptarse a cambios y entregar resultados de valor rápidamente mediante iteraciones cortas llamadas sprints, de 2 a 3 semanas.

En cada sprint se trabaja en funcionalidades priorizadas por el equipo de desarrollo y los interesados, permitiendo entregas parciales y validadas que mejoran el producto de forma continua.

* Se construye y mantiene un **Product Backlog**, con historias de usuario que representan los requisitos.
* Se realizan reuniones diarias de seguimiento (**Daily Scrum**) para monitorear avances y resolver obstáculos.
* Al finalizar cada sprint, se hace una **revisión** y una **retrospectiva**, donde se valida el incremento y se identifican mejoras al proceso.

El uso de Scrum proporciona flexibilidad, mejora la comunicación entre los miembros del equipo y garantiza que el producto evolucione de acuerdo con las necesidades reales de los usuarios y del negocio.

(**Anexo 20.** Detalles adicionales de metodología Scrum )

(**Anexo 21.** Detalles adicionales de cuadro comparativo de metodologías de desarrollo)

# 8. Resultados

El desarrollo del proyecto *Desparches* se orientó a la construcción de un prototipo funcional de alta fidelidad en Figma, como representación interactiva de la interfaz y las funcionalidades esenciales del sistema propuesto. Dado el carácter académico del proyecto, no se desarrolló código fuente ni se implementaron servicios de backend, centrándose exclusivamente en el diseño conceptual, la definición arquitectónica y la validación temprana de los requerimientos a través de prototipos visuales.

## 8.1 Modelo arquitectónico propuesto

La arquitectura propuesta para el sistema *Desparches* corresponde a un modelo multicapa bajo el paradigma cliente-servidor. Esta estructura contempla tres capas principales:

* **Capa de presentación (Frontend):** Corresponde a las interfaces móviles y web que permiten la interacción directa con los usuarios. Se contempla el uso de tecnologías como React Native para garantizar una experiencia nativa en dispositivos móviles y facilitar la escalabilidad hacia plataformas web.
* **Capa de lógica de negocio (Backend):** Diseñada como un conjunto de microservicios en Node.js que gestionan la lógica de recomendación, autenticación, búsqueda geolocalizada, reservas y control de eventos. Esta capa centraliza las reglas del sistema y permite una integración modular de servicios.
* **Capa de datos (Persistencia):** Implementada sobre PostgreSQL con extensiones geoespaciales para gestionar eventos, promociones y perfiles de usuario. Se propone también el uso de Redis como sistema de caché para optimizar el rendimiento de las consultas frecuentes y MongoDB para el almacenamiento de datos no estructurados.

Esta arquitectura fue seleccionada por su capacidad para garantizar la escalabilidad, la reutilización de componentes, la separación de responsabilidades y la flexibilidad ante futuras integraciones con APIs externas (por ejemplo, servicios de pago, mapas o redes sociales).

## 8.2 Prototipos funcionales en Figma

Como parte del proceso de validación temprana de requerimientos y experiencia de usuario, se desarrollaron dos versiones del prototipo funcional en Figma: una versión web y una versión móvil. Ambos prototipos fueron diseñados conforme a los principios de usabilidad, accesibilidad y coherencia visual.

**Versión móvil:** Enfocada en la exploración geolocalizada de eventos en tiempo real, esta interfaz incluye funcionalidades clave como:

* Mapa interactivo con marcadores dinámicos.
* Filtros por categoría, fecha y distancia.
* Detalles ampliados de eventos con opción de reserva.
* Perfil del usuario con historial, eventos guardados y reseñas.

**Versión web:** Diseñada para navegadores de escritorio, con énfasis en la administración de contenido. Esta versión facilita el acceso de organizadores y comercios para:

* Crear y gestionar eventos o promociones.
* Visualizar métricas de actividad.
* Filtrar y actualizar contenido existente.

Ambas versiones permiten simular las acciones críticas de los distintos actores del sistema (usuarios, administradores y organizadores), incluyendo procesos de autenticación, navegación, edición de perfil, búsqueda avanzada, interacción con el mapa y valoración de eventos.

Los prototipos reflejan fielmente los casos de uso definidos durante la fase de elaboración, asegurando la trazabilidad entre los requerimientos funcionales y las soluciones propuestas. Su construcción permitió verificar la coherencia del modelo de interacción y facilitó la identificación de mejoras en la arquitectura de navegación y flujo de tareas.

Resultado del desarrollo del prototipo de Desparches:

### - Estructura visual y funcional definida

Se diseñó una interfaz clara e intuitiva en Figma, asegurando una navegación sencilla para los usuarios, tanto en versión móvil como en escritorio.

### Link del prototipo en figma en computador

* Usuario:
* <https://www.figma.com/proto/kcHQJEOxhfJquLauWHwJ4B/prototipo1-dise%C3%B1o?node-id=1-2&t=TbOl2tm4mxEsjruD-1&scaling=scale-down&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=1%3A4478>
* Administrador:
* <https://www.figma.com/proto/jvxvgJwycH6YeTLQ32XxFD/Prototipo-empresas?node-id=226-911&t=qbDwPtKJOpYP2emP-0&scaling=scale-down&content-scaling=fixed&page-id=17%3A14&starting-point-node-id=226%3A911>

### Link del prototipo en figma en celular

* + Usuario:
* <https://www.figma.com/proto/iuue4diUQVcQsBMV66za1z/Prototipo-celular-Usuario?node-id=1-3&p=f&t=AaXK5ezhBCMXc7lK-0&scaling=scale-down&content-scaling=fixed&page-id=0%3A1&starting-point-node-id=1%3A3>
  + Empresa:
* <https://www.figma.com/proto/jvxvgJwycH6YeTLQ32XxFD/Prototipo-empresas?node-id=83-67&p=f&t=e6vCAjWxamcjhcy0-0&scaling=scale-down&content-scaling=fixed&page-id=0%3A1>

(**Anexo 22.** Detalles adicionales modelo en figma)

## 8.3 Validación del diseño

Aunque no se realizó una implementación técnica del sistema, los prototipos funcionaron como herramienta de validación del diseño arquitectónico y de la propuesta funcional. A través de simulaciones y pruebas de navegación con usuarios potenciales, se logró confirmar la pertinencia de las funcionalidades clave y la claridad de la interfaz. Este ejercicio sirvió también para mitigar riesgos asociados a la usabilidad y a la complejidad de interacción entre módulos.

# 9. Conclusiones

El desarrollo del proyecto Desparches permitió al equipo adquirir valiosas experiencias en la aplicación de metodologías estructuradas para el diseño de soluciones digitales. A través del enfoque iterativo propuesto por RUP y el uso de herramientas como Figma para la creación de prototipos funcionales, fue posible abordar de manera organizada los distintos retos que implica conceptualizar una plataforma centrada en la experiencia del usuario y la integración de servicios de geolocalización y recomendaciones personalizadas.

Durante el proceso, se logró una comprensión profunda de la importancia de definir de forma clara los requerimientos funcionales y no funcionales, así como de estructurar una arquitectura técnica escalable que responda a las necesidades del proyecto. Esto se reflejó en la calidad y completitud de los prototipos desarrollados, los cuales permitieron validar anticipadamente los principales flujos de interacción de los usuarios y administradores, brindando una base sólida para una futura implementación técnica.

En cuanto al cumplimiento de los objetivos propuestos, se considera que el objetivo general fue alcanzado satisfactoriamente dentro del marco académico del proyecto. Se diseñó una plataforma digital capaz de centralizar eventos y promociones con base en la ubicación del usuario, y se facilitó una experiencia personalizada para su descubrimiento. Asimismo, los objetivos específicos se cumplieron de manera progresiva y coherente:

Se identificaron y organizaron los requerimientos clave de la plataforma, asegurando su alineación con las expectativas tanto de los usuarios como de los comercios y organizadores.

Se diseñó una arquitectura técnica sólida, contemplando elementos esenciales como la integración de APIs de geolocalización, módulos de recomendación, y una gestión eficiente del contenido.

Se implementaron prototipos interactivos para dispositivos móviles y versión web, cuidando la usabilidad, la navegabilidad y la adaptabilidad visual, lo cual permitió representar fielmente las funcionalidades esperadas.

En conclusión, el proyecto Desparches no solo cumplió con sus objetivos académicos, sino que también brindó a los participantes una experiencia integral de trabajo colaborativo, análisis técnico y diseño centrado en el usuario, sentando las bases para futuros desarrollos más avanzados en el ámbito del software orientado a servicios locales y personalizados.

# 10. Bibliografía

* EDteam. (s.f.). *¿Cuáles son los roles en el desarrollo web?* Obtenido de EDteam - En Español Nadie Te Explica Mejor: https://ed.team/blog/cuales-son-los-roles-en-el-desarrollo-web
* Martins, J. (16 de agosto de 2024). *Qué es la teoría de las restricciones*. Obtenido de Asana Resources: https://asana.com/es/resources/theory-of-constraints
* Nielsen, J. (2012). *Usability 101: Introduction to Usability*. Obtenido de Nielsen Norman Group Articles: https://www.nngroup.com/articles/usability-101-introduction-to-usability/
* Pixartprinting. (s.f.). *Figma: qué es y cómo funciona*. Obtenido de El Blog de Pixartprinting: https://www.pixartprinting.es/blog/figma-que-es/
* Project Management Institute. (2021). *PMBOK Guide (7a ed.)*. Obtenido de PMI Standards: https://www.pmi.org/pmbok-guide-standards/foundational/pmbok
* Rodríguez, J. (2015). *Métodos de Diseño en Ingeniería*. Obtenido de Prezi: https://prezi.com/ej4llfzyc6d1/metodos-de-diseno-en-ingenieria/
* Visure Solutions. (1 de enero de 2023). *Ingeniería de Requerimientos: Paso a Paso*. Obtenido de Visure Solutions Blog: https://visuresolutions.com/es/blog/requirements-engineering-process/
* Bao, Y., Zheng, Y., & Li, X. (2015). Location-based and preference-aware recommendation using sparse geo-social networking data. *Proceedings of the 23rd International Conference on World Wide Web*, 337-346.
* Bass, L., Clements, P., & Kazman, R. (2012). *Software Architecture in Practice* (3rd ed.). Addison-Wesley.
* Brown, B., & Chalmers, M. (2003). Tourism and mobile technology. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1-8.
* Chaniotis, P., Trihinas, D., & Vassilakis, V. (2015). Performance evaluation of Node.js and PHP. *International Journal of Computer Applications*, 123(10), 1-6.
* Cockburn, A. (2006). Agile software development: The people factor. *Computer*, 39(5), 99-101.
* Deterding, S. (2012). Gamification: Designing for motivation. *Interactions*, 19(4), 14-17.
* Dragoni, N., et al. (2017). Microservices: Yesterday, today, and tomorrow. *Present and Ulterior Software Engineering*, 195-216.
* Dwork, C. (2008). Differential privacy: A survey of results. *Theory and Applications of Models of Computation*, 1-19.
* Eisenmann, T., Parker, G., & Van Alstyne, M. (2011). Platform envelopment. *Strategic Management Journal*, 32(12), 1270-1285.
* Fowler, M. (2002). *Patterns of Enterprise Application Architecture*. Addison-Wesley.
* Hamari, J., Koivisto, J., & Sarsa, H. (2014). Does gamification work? — A literature review of empirical studies on gamification. *47th Hawaii International Conference on System Sciences*, 3025-3034.
* Hoober, S., & Berkman, E. (2011). *Designing Mobile Interfaces*. O'Reilly Media.
* Hoofnagle, C. J., van der Sloot, B., & Zuiderveen Borgesius, F. J. (2019). The European Union General Data Protection Regulation: What it is and what it means. *Information & Communications Technology Law*, 28(1), 65-98.
* Jacobson, I., Booch, G., & Rumbaugh, J. (1999). *The Unified Software Development Process*. Addison-Wesley.
* Junglas, I. A., & Watson, R. T. (2008). Location-based services. *Communications of the ACM*, 51(3), 65-69.
* Kruchten, P. (2003). The Rational Unified Process: An introduction. Addison-Wesley.
* Kumar, V. (2014). Managing local platform business models: A strategic framework. *Journal of Business Models*, 2(2), 1-20.
* Larman, C. (2004). *Agile and Iterative Development: A Manager's Guide*. Addison-Wesley.
* Lian, J., Zhao, X., Xie, X., Sun, G., & Chen, E. (2014). GeoMF: Joint geographical modeling and matrix factorization for point-of-interest recommendation. *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 831-840.
* Liu, D., et al. (2016). User behavior in event platforms: A study on location relevance and social factors. *International Journal of Human-Computer Studies*, 92, 12-25.
* Longley, P. A., Goodchild, M. F., Maguire, D. J., & Rhind, D. W. (2015). *Geographic Information Systems and Science* (4th ed.). Wiley.
* Luke, J. (2011). Designing mobile user experiences. *Morgan Kaufmann*.

# 11. Anexos

* (**Anexo 1.** Detalles adicionales del problema)
* (**Anexo 2.** Detalles adicionales del mapa de empatía)
* (**Anexo 3.** Detalles adicionales del user persona)
* (**Anexo 4.** Detalles adicionales del diagrama de afinidad)
* (**Anexo 5.** Detalles adicionales de la declaración del problema)
* (**Anexo 6.** Detalles adicionales del scamper)
* (**Anexo 7.** Detalles adicionales del brainstorming)
* (**Anexo 8.** Detalles adicionales del Crazy 8’s)
* (**Anexo 9.** Detalles adicionales del Wireframes)
* (**Anexo 10.** Detalles adicionales de prototipos rápidos)
* (**Anexo 11.** Detalles adicionales de la matriz de retroalimentación)
* (**Anexo 12.** Detalles adicionales del user journey map)
* (**Anexo 13.** Detalles adicionales del marco teórico)
* (**Anexo 14.** Detalles adicionales de la metodología RUP)
* (**Anexo 15.** Detalles adicionales de la plantilla de visión)
* (**Anexo 16.** Detalles adicionales Ingeniería de Requerimientos)
* (**Anexo 17.** Detalles adicionales de Casos de Uso)
* (**Anexo 18.** Detalles adicionales de requerimientos)
* (**Anexo 19.** Detalles adicionales de Diagramas UML)
* (**Anexo 20.** Detalles adicionales de metodología Scrum)
* (**Anexo 21.** Detalles adicionales de cuadro comparativo de metodologías de desarrollo)
* (**Anexo 22.** Detalles adicionales modelo en figma)