



Ingeniería de Software
Arquitecturas De Software Emergentes
Profesor: Richard Leonardo Berrocal Navarro

INFORME

Startup: AlquilaEZ

Product: AlquilaFácil

Team Members:

Member	Code
Ames Oviedo, Mariano José	U202211371
Cachis Gonzales, Sebastian Nicolas	U202210846
Lecca Choccare, Christopher Bryan	U202211399
Llamo Sánchez, Amner Levi	U20221C376
Morales Montalvo, Omar Andrew	U202212383

Registro de Versiones del Informe

Version	Fecha	Autor	Descripción de modificación
1era	18/09/2025	- Ames Oviedo, Mariano José - Cachis Gonzales, Sebastian Nicolas - Lecca Choccare, Christopher Bryan - Llamo Sánchez, Amner Levi - Morales Montalvo, Omar Andrew	• Capítulo I • Capítulo II • Capítulo III • Capítulo IV
2da	07/10/2025	- Ames Oviedo, Mariano José - Cachis Gonzales, Sebastian Nicolas - Lecca Choccare, Christopher Bryan - Llamo Sánchez, Amner Levi - Morales Montalvo, Omar Andrew	• Capítulo V • Capítulo VI

Project Report Collaboration Insights

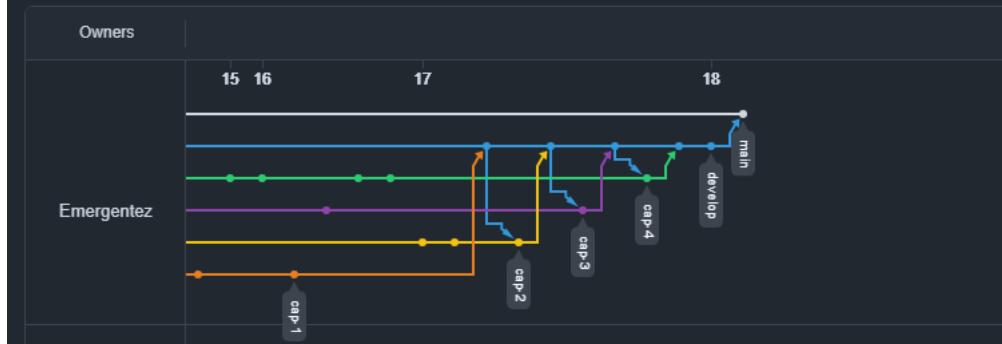
Repositorio del proyecto: <https://github.com/Emergentez/alquilafacil-document-report>

TB1:

Para este entregable, se mantuvo un orden trabajando los capítulos por ramas:

Network graph

Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



TP1:

Para este entregable, se continuó trabajando los capítulos por ramas, añadiendo el capítulo 5 y 6:





Student Outcome

ABET – EAC – Student Outcome 3

Criterio: Capacidad de comunicarse efectivamente con un rango de audiencias.

En el siguiente cuadro se describe las acciones realizadas y enunciados de conclusiones por parte del grupo, que permiten sustentar el haber alcanzado el logro del ABET – EAC - Student Outcome 3.

Criterio específico	Acciones realizadas	Conclusiones
Comunica oralmente sus ideas y/o resultados con objetividad a público de diferentes especialidades y niveles jerárquicos, en el marco del desarrollo de un proyecto en ingeniería.	<p>Ames Oviedo, Mariano José (TB1): Expuso los avances de la fase de análisis de requerimientos, aclarando dudas del equipo y utilizando ejemplos prácticos para facilitar la comprensión de las funcionalidades.</p> <p>Cachis Gonzales, Sebastián Nicolas (TB1): Presentó la propuesta de arquitectura del sistema, explicando de manera clara el flujo de datos y la interacción de módulos ante un público con distintos niveles de experiencia.</p> <p>Lecca Choccare, Christopher Bryan (TB1): Lideró sesiones de Event Storming y Context Mapping, guiando al equipo en la identificación de bounded contexts y adaptando el lenguaje técnico según la audiencia.</p> <p>Llamo Sánchez, Amner Levi (TB1): Expliqué los diagramas UML al grupo, destacando las relaciones entre clases y casos de uso, simplificando conceptos técnicos para que todos pudieran entenderlos.</p>	<p>TB1: Los integrantes del equipo demostraron capacidad de comunicación oral al exponer ideas y resultados con objetividad, adaptando el lenguaje a públicos con diferentes especialidades y niveles jerárquicos.</p> <p>TP1: El equipo fortaleció sus habilidades para presentar elementos de diseño de interfaz y experiencia de usuario, logrando transmitir conceptos visuales y de interacción de manera clara y efectiva. Los integrantes mejoraron su capacidad de articular decisiones de diseño justificando cada elección desde una perspectiva técnica y de usabilidad, lo que facilitó la comprensión y alineación del equipo en la fase de prototipado.</p>

Criterio específico	Acciones realizadas	Conclusiones
	<p>Morales Montalvo, Omar Andrew (TB1): Expuse la justificación del uso de metodologías ágiles, mostrando cómo Scrum permite adaptarse a cambios y ejemplificando con la organización de sprints.</p> <p>Ames Oviedo, Mariano José (TP1): Presenté los Wireflows y User Flows del sistema, explicando el recorrido completo de los usuarios y cómo cada flujo responde a las necesidades identificadas, facilitando la comprensión del equipo.</p> <p>Cachis Gonzales, Sebastian Nicolas (TP1): Expuse el trabajo realizado en Profiles (SC) y el agregado de enlaces de herramientas usadas (SC), explicando la estructura de perfiles de usuario y justificando las decisiones de diseño.</p> <p>Lecca Choccare, Christopher Bryan (TP1): Presenté los desarrollos de Booking (CL), Subscriptions (CL), Monitoring (CL), Mockups (CL) y Wireframes (CL), articulando la funcionalidad de cada módulo y su integración en el sistema completo.</p> <p>Llamo Sánchez, Amner Levi (TP1): Expuse el trabajo en Notifications (AS), Information Architecture (AS) y Landing Page UI Design (AS), explicando las decisiones de arquitectura de información y diseño de interfaz de manera clara.</p> <p>Morales Montalvo, Omar Andrew (TP1): Presenté los avances en Locales (OM), Style Guidelines (OM) y Prototypes (OM), mostrando la coherencia visual del sistema y las guías de estilo establecidas para el proyecto.</p> <p>Cachis Gonzales, Sebastian Nicolas (TB2): Conduje entrevistas de validación con usuarios finales. En ellas, traduje el funcionamiento técnico del backend en tareas simples, guiando a una audiencia no técnica para obtener retroalimentación objetiva sobre el producto. Además, Presenté la arquitectura e implementación del backend en el Sprint Review. Expliqué los endpoints, la lógica de negocio y las decisiones técnicas que tomé, adaptando la explicación para mi equipo.</p> <p>Llamo Sánchez, Amner Levi (TB2): Participe en las entrevistas de validación con usuarios finales, traduciendo las funcionalidades técnicas de la aplicación móvil en tareas sencillas y comprensibles para obtener retroalimentación objetiva. Además, expuse la implementación y funcionalidad de la aplicación móvil durante el Sprint Review, demostrando el producto y explicando las decisiones técnicas tomadas al equipo</p>	
Comunica en forma escrita ideas y/o resultados con objetividad a público de diferentes especialidades y niveles jerárquicos, en el marco del desarrollo de un proyecto en ingeniería	<p>Ames Oviedo, Mariano José (TB1): Redacté la documentación del análisis de requerimientos y los criterios de aceptación, asegurando claridad en el uso de términos técnicos y su comprensión general.</p> <p>Cachis Gonzales, Sebastian Nicolas (TB1): Elaboré la sección del informe sobre diseño arquitectónico, integrando diagramas y resúmenes técnicos para hacer el contenido accesible a distintos públicos.</p>	<p>TB1: El equipo evidenció solidez en la comunicación escrita al elaborar documentación técnica clara y estructurada, comprensible para distintos públicos y útil para la gestión del proyecto.</p> <p>TP1: Los integrantes consolidaron su capacidad de documentar elementos de diseño, arquitectura de información y prototipos con un nivel profesional de detalle y claridad. El equipo desarrolló habilidades para integrar elementos visuales (mockups, wireframes) con descripciones textuales coherentes, creando documentación técnica que sirve tanto como referencia para desarrolladores como guía comprensible para stakeholders no técnicos. Esta competencia resultó fundamental para</p>

Criterio específico	Acciones realizadas	Conclusiones
	<p>Lecca Choccare, Christopher Bryan (TB1): Participé en la redacción de las descripciones del modelo C4 y de las User Stories, estructurando el Product Backlog con un lenguaje claro y comprensible.</p> <p>Llamo Sánchez, Amner Levi (TB1): Documenté los diagramas UML y su interpretación funcional, conectando lo visual con lo técnico y garantizando la coherencia del informe final.</p> <p>Morales Montalvo, Omar Andrew (TB1): Redacté la sección metodológica del informe, justificando el uso de Scrum y explicando la organización de los sprints con un estilo formal y accesible.</p> <p>Ames Oviedo, Mariano José (TP1): Documenté los Wireflows y User Flows con descripciones detalladas, explicando cada paso del recorrido del usuario y su relación con los requisitos funcionales del sistema.</p> <p>Cachis Gonzales, Sebastian Nicolas (TP1): Redacté la documentación de Profiles (SC) y elaboré el enlace de herramientas usadas, estructurando la información de manera clara y accesible para diferentes audiencias.</p> <p>Lecca Choccare, Christopher Bryan (TP1): Documenté los módulos de Booking, Subscriptions, Monitoring (CL), además de Mockups y Wireframes, integrando descripciones funcionales con elementos visuales de diseño.</p> <p>Llamo Sánchez, Amner Levi (TP1): Elaboré la documentación de Notifications (AS), Information Architecture (AS) y Landing Page UI Design (AS), estructurando la arquitectura de información con claridad y coherencia.</p> <p>Morales Montalvo, Omar Andrew (TP1): Redacté las secciones de Locales (OM), Style Guidelines (OM) y Prototypes (OM), estableciendo estándares visuales y de diseño con documentación clara y profesional.</p> <p>Cachis Gonzales, Sebastian Nicolas (TB2): Documenté la evidencia de desarrollo (Development Evidence) y la documentación de servicios (Services Documentation Evidence) del backend, creando un registro técnico preciso para la trazabilidad y consulta del equipo. Además, Sintetizé los hallazgos de las entrevistas de validación en el "Registro de Entrevistas". Transformé las opiniones orales de la audiencia no técnica en un reporte analítico y objetivo que el equipo pudo usar.</p> <p>Llamo Sánchez, Amner Levi (TB2): Elaboré el "Registro de Entrevistas" de validación, sintetizando la retroalimentación oral de una audiencia no técnica (usuarios) en un informe analítico estructurado para el equipo. Además, documenté la evidencia de desarrollo, pruebas y despliegue (Development, Testing & Deployment Evidence) de la aplicación móvil, asegurando la trazabilidad del trabajo realizado en el Sprint.</p>	<p>Mantener la trazabilidad entre los requisitos iniciales y las soluciones de diseño propuestas.</p>

Contenido

- Capítulo I: Introducción
 - 1.1. Startup Profile
 - 1.1.1. Descripción de la Startup
 - 1.1.2. Perfiles de integrantes del equipo
 - 1.2. Solution Profile
 - 1.2.1. Antecedentes y problemática
 - 1.2.2. Lean UX Process
 - 1.2.2.1. Lean UX Problem Statements
 - 1.2.2.2. Lean UX Assumptions
 - 1.2.2.3. Lean UX Hypothesis Statements
 - 1.2.2.4. Lean UX Canvas
 - 1.3. Segmentos objetivo
- Capítulo II: Requirements Elicitation & Analysis
 - 2.1. Competidores
 - 2.1.1. Análisis competitivo
 - 2.1.2. Estrategias y tácticas frente a competidores
 - 2.2. Entrevistas
 - 2.2.1. Diseño de entrevistas
 - 2.2.2. Registro de entrevistas
 - 2.2.3. Análisis de entrevistas
 - 2.3. Needfinding
 - 2.3.1. User Personas
 - 2.3.2. User Task Matrix
 - 2.3.3. Empathy Mapping
 - 2.3.4. As-is Scenario Mapping
 - 2.4. Ubiquitous Language
- Capítulo III: Requirements Specification
 - 3.1. To-Be Scenario Mapping
 - 3.2. User Stories
 - 3.3. Impact Mapping
 - 3.4. Product Backlog
- Capítulo IV: Strategic-Level Software Design
 - 4.1. Strategic-Level Attribute-Driven Design
 - 4.1.1. Design Purpose
 - 4.1.2. Attribute-Driven Design Inputs
 - 4.1.2.1. Primary Functionality (Primary User Stories)
 - 4.1.2.2. Quality attribute Scenarios
 - 4.1.2.3. Constraints
 - 4.1.3. Architectural Drivers Backlog
 - 4.1.4. Architectural Design Decisions
 - 4.1.5. Quality Attribute Scenario Refinements
 - 4.2. Strategic-Level Domain-Driven Design
 - 4.2.1. EventStorming
 - 4.2.2. Candidate Context Discovery
 - 4.2.3. Domain Message Flows Modeling
 - 4.2.4. Bounded Context Canvases
 - 4.2.5. Context Mapping
 - 4.3. Software Architecture
 - 4.3.1. Software Architecture System Landscape Diagram
 - 4.3.2. Software Architecture Context Level Diagrams
 - 4.3.3. Software Architecture Container Level Diagrams
 - 4.3.4. Software Architecture Deployment Diagrams
- Capítulo V: Tactical-Level Software Design
 - Capítulo V: Tactical-Level Software Design
 - 5.1. Bounded Context:
 - 5.1.1. Domain Layer
 - 5.1.2. Interface Layer
 - 5.1.3. Application Layer
 - 5.1.4. Infrastructure Layer

- 5.1.6. Bounded Context Software Architecture Component Level Diagrams
- 5.1.7. Bounded Context Software Architecture Code Level Diagrams
 - 5.1.7.1. Bounded Context Domain Layer Class Diagrams
 - 5.1.7.2. Bounded Context Database Design Diagram
- 5.2. Bounded Context: Profile Context
 - 5.2.1. Domain Layer
 - 5.2.2. Interface Layer
 - 5.2.3. Application Layer
 - 5.2.4. Infrastructure Layer
 - 5.2.6. Bounded Context Software Architecture Component Level Diagrams
 - 5.2.7. Bounded Context Software Architecture Code Level Diagrams
 - 5.2.7.1. Bounded Context Domain Layer Class Diagrams
 - 5.2.7.2. Bounded Context Database Design Diagram
- 5.3. Bounded Context: Locals Context
 - 5.3.1. Domain Layer
 - 5.3.2. Interface Layer
 - 5.3.3. Application Layer
 - 5.3.4. Infrastructure Layer
 - 5.3.6. Bounded Context Software Architecture Component Level Diagrams
 - 5.3.7. Bounded Context Software Architecture Code Level Diagrams
 - 5.3.7.1. Bounded Context Domain Layer Class Diagrams
 - 5.3.7.2. Bounded Context Database Design Diagram
- 5.4. Bounded Context: Booking Context
 - 5.4.1. Domain Layer
 - 5.4.2. Interface Layer
 - 5.4.3. Application Layer
 - 5.4.4. Infrastructure Layer
 - 5.4.6. Bounded Context Software Architecture Component Level Diagrams
 - 5.4.7. Bounded Context Software Architecture Code Level Diagrams
 - 5.4.7.1. Bounded Context Domain Layer Class Diagrams
 - 5.4.7.2. Bounded Context Database Design Diagram
- 5.5. Bounded Context: Notifications Context
 - 5.5.1. Domain Layer
 - 5.5.2. Interface Layer
 - 5.5.3. Application Layer
 - 5.5.4. Infrastructure Layer
 - 5.5.6. Bounded Context Software Architecture Component Level Diagrams
 - 5.5.7. Bounded Context Software Architecture Code Level Diagrams
 - 5.5.7.1. Bounded Context Domain Layer Class Diagrams
 - 5.5.7.2. Bounded Context Database Design Diagram
- 5.6. Bounded Context: Subscription Context
 - 5.6.1. Domain Layer
 - 5.6.2. Interface Layer
 - 5.6.3. Application Layer
 - 5.6.4. Infrastructure Layer
 - 5.6.6. Bounded Context Software Architecture Component Level Diagrams
 - 5.6.7. Bounded Context Software Architecture Code Level Diagrams
 - 5.6.7.1. Bounded Context Domain Layer Class Diagrams
 - 5.6.7.2. Bounded Context Database Design Diagram
- 5.7. Bounded Context: Monitoring Context
 - 5.7.1. Domain Layer
 - 5.7.2. Interface Layer
 - 5.7.3. Application Layer
 - 5.7.4. Infrastructure Layer
 - 5.7.6. Bounded Context Software Architecture Component Level Diagrams
 - 5.7.7. Bounded Context Software Architecture Code Level Diagrams
 - 5.7.7.1. Bounded Context Domain Layer Class Diagrams
 - 5.7.7.2. Bounded Context Database Design Diagram
- Capítulo VI: Solution UX Design

- 6.1. Style Guidelines
 - 6.1.1. General Style Guidelines
 - 6.1.2. Web, Mobile & Devices Style Guidelines
 - 6.2. Information Architecture
 - 6.2.2. Labeling Systems
 - 6.2.3. Searching Systems
 - 6.2.4. SEO Tags and Meta Tags
 - 6.2.5. Navigation Systems
 - 6.3. Landing Page UI Design
 - 6.3.1. Landing Page Wireframe
 - 6.3.2. Landing Page Mock-up
 - 6.4. Applications UX/UI Design
 - 6.4.1. Applications Wireframes
 - 6.4.2. Applications Wireflow Diagrams
 - 6.4.3. Applications Mock-ups
 - 6.4.4. Applications User Flow Diagrams
 - 6.5. Applications Prototyping
- Capítulo VII: Product Implementation, Validation & Deployment
 - 7.1. Software Configuration Management
 - 7.1.1. Software Development Environment Configuration
 - 7.1.2. Source Code Management
 - 7.1.3. Source Code Style Guide & Conventions
 - 7.1.4. Software Deployment Configuration
 - 7.2. Solution Implementation
 - 7.2.1. Sprint 1
 - 7.2.1.1. Sprint Planning 1
 - 7.2.1.2. Sprint Backlog 1
 - 7.2.1.3. Development Evidence for Sprint Review
 - 7.2.1.4. Testing Suite Evidence for Sprint Review
 - 7.2.1.5. Execution Evidence for Sprint Review
 - 7.2.1.6. Services Documentation Evidence for Sprint Review
 - 7.2.1.7. Software Deployment Evidence for Sprint Review
 - 7.2.1.8. Team Collaboration Insights during Sprint
 - 7.3. Validation Interviews
 - 7.3.1. Diseño de Entrevistas
 - 7.3.2. Registro de Entrevistas
 - 7.3.3. Evaluaciones según heurísticas
 - 7.4. Video About-the-Product
- Conclusiones y recomendaciones
 - Bibliografía
 - Anexos

Capítulo I: Introducción

1.1. Startup Profile

1.1.1. Descripción de la Startup

AlquilaFácil es un startup fundado en el 2025, por un grupo de estudiantes de la carrera de Ingeniería de Software de la Universidad Peruana de Ciencias Aplicadas (UPC). Nuestra empresa emergente se enfoca en la simplificación del proceso de búsqueda de alquiler de espacios para eventos, así como la administración de los mismos.

Misión: Revolucionar el alquiler de espacios para eventos con una plataforma intuitiva que une a arrendadores y arrendatarios de manera rápida y sencilla.

Visión: Consolidarnos como la referencia global en alquiler de espacios para eventos, redefiniendo el sector con soluciones inteligentes y una experiencia excepcional.

1.1.2. Perfiles de integrantes del equipo

Nombre

Ames Oviedo,
Mariano José
(U202211371)

Mi nombre es Mariano Ames, tengo 20 años y me encuentro cursando el 8vo ciclo de la carrera de Ingeniería de Software en la Universidad Peruana de Ciencias aplicadas. Soy una persona e intento colaborar en lo más posible dentro del equipo de trabajo, también me llama la atención el aprendizaje de nuevas tecnologías que puedan ayudarme a mejorar mi ayuda en el equipo.



Nombre

Cachis Gonzales,
Sebastian Nicolas
(U202210846)

Mi nombre es Sebastian Nicolas Cachis Gonzales, soy estudiante de octavo ciclo de ingeniería de software en la UPC. Me considero una persona proactiva, organizada, meticulosa y muy enfocada en mis estudios, tanto grupales como individuales. Tengo facilidad para entender y ejemplificar los distintos temas que vemos, teniendo soltura para explicar.



Nombre

Lecca Choccare,
Christopher Bryan
(U202211399)

Mi nombre es Christopher, tengo 20 años y actualmente me encuentro cursando el 8vo ciclo de la carrera de Ingeniería de Software en la Universidad Peruana de Ciencias Aplicadas (UPC). Considero que mi compromiso con el equipo de trabajo designado y mi capacidad de aprendizaje rápida aportarán de buena manera al desarrollo del producto final.



Nombre

Llamo Sánchez,
Amner Levi
(U20221C376)

Mi nombre es Amner Levi Llamo Sánchez, soy estudiante del octavo ciclo de ingeniería de software en la UPC. Me gusta jugar fútbol y videojuegos, por eso estoy constantemente investigando sobre nuevas tecnologías. Soy responsable con los trabajos que se me asignan; además soy tolerante y me adapto a las circunstancias del equipo.



Nombre

Morales
Montalvo, Omar
Andrew
(U202212383)

Mi nombre es Omar, tengo 19 años y actualmente estoy cursando el séptimo ciclo de la carrera de Ingeniería de Software en la Universidad Peruana de Ciencias Aplicadas (UPC). Me considero una persona comprometida con lo que hace, además de ser responsable y dedicada. Me gusta aprender constantemente y trabajar en equipo, siempre buscando mejorar y aportar en cada proyecto.



1.2. Solution Profile

1.2.1. Antecedentes y problemática

Hoy en día, las personas cuentan con múltiples herramientas tecnológicas que facilitan el proceso de reservar y alquilar locales para eventos, ya sea a través de páginas web o aplicaciones móviles. Esto ha impulsado un aumento de la demanda en el mercado peruano de reservas en línea. Este crecimiento se ve reforzado por el incremento en el acceso a internet y al uso de dispositivos digitales. Según datos del INEI, durante los primeros tres meses del 2022, 73 de cada 100 personas mayores de seis años accedieron a internet en el país, lo que representa un crecimiento de 5,1 y 17,7 puntos porcentuales en comparación con el mismo periodo en 2021 (67,4%) y en 2019 (54,8%) respectivamente (Jasin, 2023).

De igual manera, las marcas apuestan cada vez más por la organización de eventos como estrategia para acercarse a su público y transmitir sus valores. Como indica Mañas (2024), estas experiencias son vistas como una vía efectiva para fortalecer el vínculo con los clientes y construir relaciones más duraderas.

Por otro lado, las empresas que ofrecen espacios para eventos enfrentan desafíos al momento de gestionar reservas, promocionar sus locales o coordinar con los arrendatarios. Muchas veces recurren a procesos manuales, canales informales o herramientas dispersas que generan fricciones en la experiencia de usuario.

En este contexto, se evidencia que en el Perú el número de usuarios con acceso a internet ha aumentado de manera significativa en los últimos años, lo que respalda la necesidad de contar con una plataforma digital centralizada y confiable que agilice la búsqueda, comparación, reserva y monitoreo de espacios para

eventos. Este tipo de solución se vuelve especialmente relevante frente a la creciente demanda, especialmente por parte de empresas, emprendimientos y usuarios que organizan eventos corporativos, sociales o culturales.

5W's y 2H's:

Elemento	Respuesta
Who	Usuarios con acceso a internet en el Perú interesados en alquilar o ofrecer espacios para eventos.
What	Desarrollo de una plataforma digital (web y app móvil) para facilitar la búsqueda, publicación, reserva y monitoreo de espacios para eventos.
When	En un contexto actual (2025), donde el acceso a internet y la demanda de servicios digitales siguen en crecimiento.
Where	Perú, especialmente en zonas urbanas con mayor acceso a internet y actividad empresarial.
Why	Porque existe una creciente necesidad de plataformas que conecten a arrendadores y arrendatarios de espacios para eventos, facilitando la gestión, seguridad y visibilidad de los locales en un mercado cada vez más competitivo.
How	A través del desarrollo de una aplicación móvil y una landing page web responsive, con funciones como publicación de espacios, filtros de búsqueda, calendario de reservas, monitoreo mediante sensores, notificaciones y sección de ayuda.
How much	El costo estimado para la publicación en Google Play es de 25 dólares más 30% de ingresos, mientras que en App Store la tasa se encuentra entre 99 y 299 dólares anuales. Este rango está basado en fuentes como: - DoApps - GoApps

1.2.2. Lean UX Process

En esta parte, hacemos el proceso Lean UX que cubre la visión de un modelo de negocio que respalda nuestro producto de software. Este enfoque metodológico nos permite centrarnos en desarrollar nuestra solución y la solución al problema identificado al pensar en el diseño.

1.2.2.1. Lean UX Problem Statements

Nuestro entorno se sitúa en el mercado peruano de alquiler de espacios para eventos, un rubro que ha mostrado un crecimiento constante en los últimos años. El incremento de la demanda por ambientes versátiles y ajustables a distintas necesidades también ha traído consigo retos en la forma en que interactúan arrendadores y arrendatarios dentro de este ecosistema.

Detectamos que, dentro de este mercado, los usuarios enfrentan complicaciones debido a la falta de integración en las distintas etapas del proceso. La dispersión en la búsqueda, la gestión de reservas y la organización genera ineficiencias y una experiencia poco satisfactoria.

Surge entonces la pregunta: ¿cómo lograr una mejor conexión entre arrendadores y arrendatarios, optimizando el proceso de alquiler de espacios para eventos y garantizando una experiencia más ágil y eficiente?

1.2.2.2. Lean UX Assumptions

Business Outcomes

- Facilitar la gestión de mínimo 1,000 reservaciones durante el primer año operativo, mejorando la comunicación y administración entre ambas partes.
- Alcanzar un índice de satisfacción superior al 90% mediante herramientas que permitan verificar ratings e historial de propiedades, facilitando decisiones informadas.
- Conseguir un 80% de renovación en suscripciones al finalizar el primer año, demostrando satisfacción y fidelización efectiva.

User Assumptions

¿Quién es el usuario?

- Usuarios que buscan espacios (arrendatarios)
- Propietarios de espacios (arrendadores)

¿Dónde entra nuestro producto en su trabajo o su vida?

- **Para arrendatarios:** La plataforma AlquilaFácil optimiza la localización y reserva de espacios, facilitando la organización de eventos y reduciendo tiempo y recursos invertidos.
- **Para arrendadores:** AlquilaFácil potencia la promoción de propiedades y mejora la administración, incrementando exposición y volumen de reservas.

¿Cuál es el problema que nuestro producto soluciona?

- Comunicación deficiente entre las partes involucradas
- Complejidad en la localización y reserva de espacios apropiados
- Ausencia de transparencia y credibilidad respecto a calidad y disponibilidad
- Carencia de una herramienta unificada para administrar reservas y promocionar propiedades

¿Cómo y cuándo nuestro producto es usado?

- **Arrendatarios:** Utilizan la plataforma para localizar, filtrar y reservar espacios durante las etapas de planificación y ejecución de eventos.
- **Arrendadores:** Emplean la plataforma para dar visibilidad a sus espacios y administrar las reservaciones.

¿Qué características son importantes?

- Interface intuitiva para búsqueda y reserva eficiente
- Sistema robusto de promoción y visibilidad de espacios
- Herramientas integradas para gestión de reservas y comunicación efectiva
- Sistema de calificaciones y reseñas para decisiones fundamentadas

User Outcomes

- **Eficiencia para arrendatarios:** Reducción del 20% en tiempo promedio de búsqueda y reserva durante los primeros seis meses de uso.
- **Crecimiento para arrendadores:** Incremento del 25% en reservas y consultas recibidas durante el primer año de implementación.
- **Satisfacción mejorada:** Aumento del 30% en satisfacción con calidad y adecuación de espacios reservados, medido através del sistema de reseñas y calificaciones durante el primer año.

1.2.2.3. Lean UX Hypothesis Statements

Para la elaboración de los Hypothesis Statements se utilizó el formato Lean UX: [We believe that], [This will achieve] y [We will have demonstrated this when]

Hipótesis 1

Creemos que, al simplificar el proceso de búsqueda y comparación de espacios para eventos en nuestra plataforma, aumentaremos la fidelización de clientes y la recurrencia de uso.

Esto permitirá que los usuarios encuentren más fácilmente opciones adecuadas a sus necesidades, mejorando su experiencia.

Sabremos que estamos en lo correcto cuando observemos un incremento del 25% en la cantidad de reservas realizadas por clientes recurrentes en los primeros seis meses tras la implementación.

Hipótesis 2

Creemos que, al mejorar la función de búsqueda avanzada, incorporando filtros detallados como capacidad, ubicación y servicios incluidos, optimizaremos la experiencia de exploración y selección de espacios.

Esto facilitará que los usuarios encuentren rápidamente opciones relevantes y reduzcan la necesidad de soporte.

Sabremos que estamos en lo correcto cuando se registre una disminución del 25% en las consultas al área de soporte relacionadas con la búsqueda, y un aumento en el tiempo promedio de navegación en la plataforma.

Hipótesis 3

Creemos que, al implementar un dashboard personalizado para los arrendadores, donde puedan monitorear en tiempo real el estado de sus espacios (reservas activas, ingresos generados, evaluaciones, disponibilidad, etc.), mejoraremos la gestión de los locales y el compromiso de los arrendadores con la plataforma.

Esto permitirá a los arrendadores tomar decisiones informadas y ofrecer un mejor servicio.

Sabremos que estamos en lo correcto cuando observemos un aumento del 30% en la actividad de los arrendadores dentro del panel, así como una mejora en la tasa de respuesta y actualización de disponibilidad en sus publicaciones, dentro de los primeros seis meses.

Hipótesis 4

Creemos que, al habilitar un sistema de calificación y comentarios para los espacios, incrementaremos la confianza de los usuarios y la percepción de calidad de los servicios ofrecidos.

Esto ayudará a generar transparencia y a tomar decisiones basadas en la experiencia de otros clientes.

Sabremos que estamos en lo correcto cuando se evidencie un aumento del 20% en la cantidad de reservas completadas y una mejora progresiva en la calidad de los comentarios dentro de los primeros tres meses tras la implementación.

Hipótesis 5

Creemos que, al implementar un sistema de notificaciones en tiempo real que alerte a los arrendadores y al equipo de soporte sobre posibles infracciones o comportamientos indebidos en los locales (como exceder la capacidad, ruido excesivo o uso indebido del espacio), reforzaremos la seguridad y el control sobre los espacios alquilados.

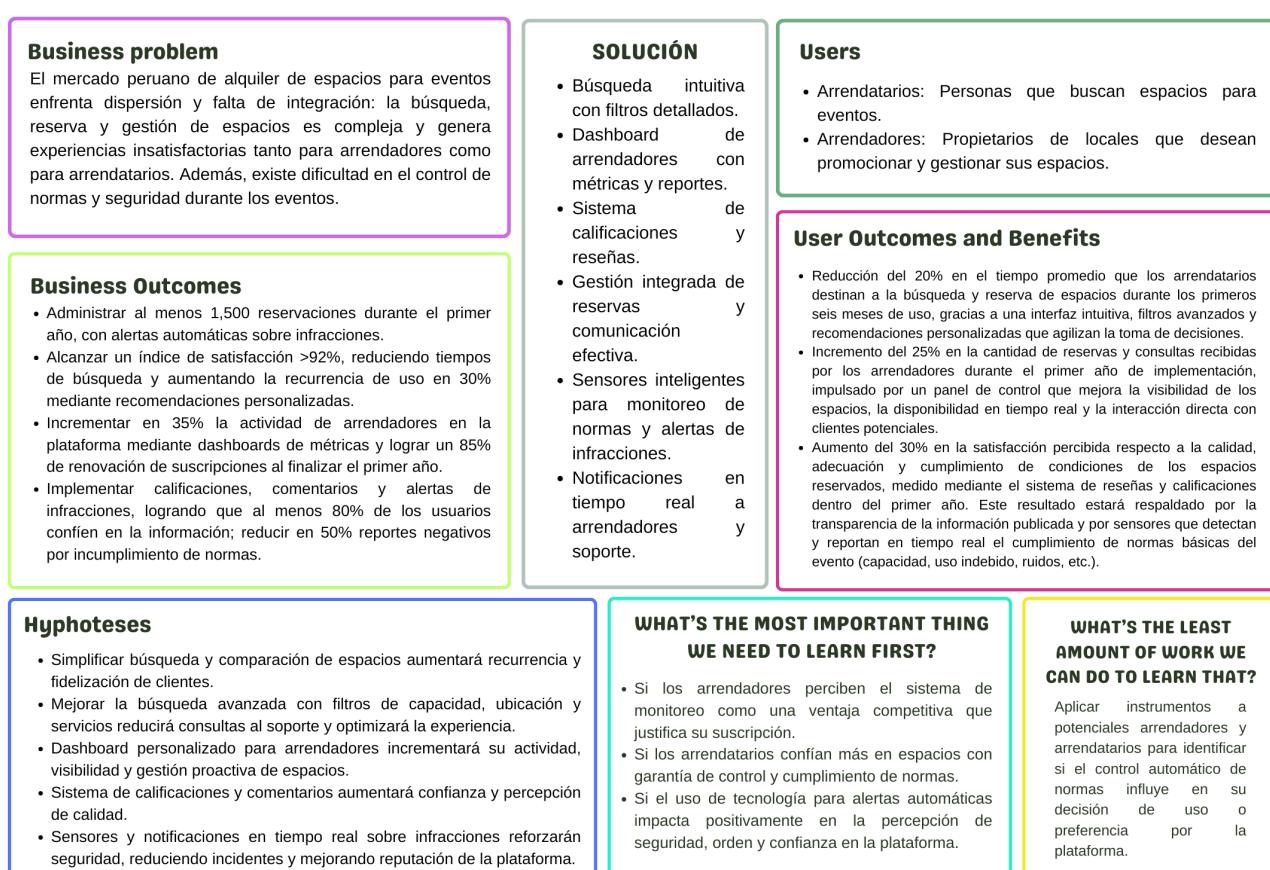
Esto contribuirá a mejorar la reputación de la plataforma y a reducir conflictos entre usuarios.

Sabremos que estamos en lo correcto cuando se registre una reducción del 40% en reportes de incidentes posteriores a las reservas, así como una mejora en las calificaciones promedio de los espacios afectados, dentro de los primeros tres meses de implementación.

1.2.2.4. Lean UX Canvas

Este Lean UX Canvas representa la propuesta integral de valor de AlquilaFácil, sintetiza los problemas detectados, los objetivos de negocio, los supuestos sobre los usuarios y las hipótesis de diseño que guían el desarrollo de la solución. Cada sección del canvas refleja cómo la solución aborda los retos de eficiencia, comunicación y confianza para arrendadores y arrendatarios, y cómo mediremos el éxito de nuestras acciones a través de métricas concretas.

Lean UX Canvas



1.3. Segmentos objetivo

En esta sección, identificamos los segmentos de clientes específicos a los que se dirige AlquilaFácil, basándonos en características demográficas, comportamientos y necesidades compartidas.

Arrendatarios:

- Descripción: Individuos que planifican eventos sociales como bodas, fiestas de cumpleaños, reuniones familiares, entre otros, y aquellos que organizan eventos regularmente, ya sea social o corporativo.
- Características: Buscan espacios que se adapten a sus necesidades específicas, como capacidad, ubicación y servicios disponibles. Realizan múltiples reservas de espacios a lo largo del año.
- Necesidades: Facilidad de búsqueda y reserva de espacios, acceso a información detallada sobre las instalaciones y servicios ofrecidos, programas de fidelización y alertas de acontecimientos que puedan afectar el desarrollo de su evento.

Arrendadores:

- Descripción: Individuos o empresas que poseen espacios aptos para eventos, como salones de banquetes, jardines, locales comerciales, etc.
- Características: Ofrecen una variedad de espacios con diferentes capacidades y servicios.
- Necesidades: Promoción efectiva de sus espacios, gestión eficiente de reservas, acceso a herramientas para administrar sus listados y proyectos de eventos.

Capítulo II: Requirements Elicitation & Analysis

2.1. Competidores

2.1.1. Análisis competitivo

Competitive Analysis Landscape

¿Por qué llevar a cabo este análisis?		Este análisis se realizó con la finalidad de poder identificar a nuestros potenciales competidores e idear estrategias y tácticas para diferenciarnos de estos.			
¿Por qué llevar a cabo este análisis?		Este análisis se realizó con la finalidad de poder identificar a nuestros potenciales competidores e idear estrategias y tácticas para diferenciarnos de estos.			
Empresas		AlquilaFácil	Airbnb	Vrbo	Booking.com
Perfil	Overview	Es una plataforma en línea que facilita el alquiler de una amplia gama de espacios para eventos, desde salones de eventos hasta casas y alojamientos temporales. Con una interfaz fácil de usar, conecta a arrendadores con arrendatarios, ofreciendo una solución conveniente y eficiente para satisfacer las necesidades de ambos. Asimismo, presenta una interfaz para poder recibir reportes de eventos ocurridos en el local.	Plataforma en línea que revoluciona el alquiler de alojamientos a corto plazo, conectando anfitriones y huéspedes en todo el mundo. Ofrece una amplia variedad de opciones de alojamiento, desde habitaciones individuales hasta casas completas.	Plataforma en línea que permite a los usuarios buscar y reservar alojamientos vacacionales directamente a través de arrendadores. Ofrece una variedad de opciones de alojamiento, como casas, apartamentos, cabañas y villas, en diferentes destinos, adaptándose a las necesidades y preferencias de los viajeros.	Booking.com es una plataforma líder para reservar alojamiento y actividades turísticas en todo el mundo. Ofrece una amplia variedad de opciones, desde hoteles hasta apartamentos, y facilita la búsqueda y reserva según las preferencias y presupuesto de los usuarios.
	¿Qué valor ofrece a los clientes?	Proporciona una solución completa para la planificación de eventos, ofreciendo una amplia gama de espacios para eventos y una plataforma intuitiva para gestionar reservas y espacios. Con características como registro gratuito de espacios, búsqueda avanzada y servicio al cliente dedicado, ArrendaFácil simplifica el proceso de planificación de eventos para arrendatarios y arrendadores.	Ofrece una amplia variedad de alojamientos en todo el mundo, desde habitaciones individuales hasta casas completas, junto con experiencias locales únicas organizadas por anfitriones. Esto permite a los viajeros personalizar su experiencia y sumergirse en la cultura local.	Se especializa en alquileres vacacionales directamente a través de arrendadores, brindando a los viajeros la oportunidad de disfrutar de una experiencia más auténtica y personalizada. Con una amplia selección de alojamientos vacacionales, Vrbo ofrece opciones para todos los gustos y presupuestos.	Destaca por su amplia oferta de alojamiento y servicios, que incluyen hoteles, vuelos, alquiler de coches y actividades turísticas. La plataforma fácil de usar permite a los usuarios encontrar y reservar alojamiento de manera rápida y sencilla, ofreciendo una solución integral para los viajeros.
Perfil de Marketing	Mercado objetivo	Dirigido a arrendatarios adultos de 18 años en adelante que buscan espacios para eventos sociales, corporativos o especiales, como bodas, conferencias, fiestas de empresa, entre otros. Además, atrae a empresas y organizaciones que buscan espacios para eventos corporativos y reuniones profesionales dirigidas a un público adulto.	Se centra en viajeros adultos de 18 años en adelante, ofreciendo una amplia variedad de alojamientos en todo el mundo, desde habitaciones individuales hasta casas completas, para adaptarse a diferentes necesidades y preferencias de viaje. Además, apunta a grupos de amigos, parejas y familias adultas que buscan opciones de alojamiento que se ajusten a sus requerimientos y presupuestos durante sus vacaciones o escapadas.	Está dirigido a familias y grupos de amigos adultos de 18 años en adelante que buscan alquilar casas, villas o cabañas para vacaciones o escapadas grupales. Además, atrae a parejas y grupos de amigos adultos que buscan alquilar alojamientos vacacionales para eventos especiales, como bodas, reuniones familiares o celebraciones de cumpleaños.	Orientado a viajeros adultos de 18 años en adelante, busca ofrecer una amplia gama de opciones de alojamiento para adaptarse a diversas necesidades y preferencias durante sus viajes. También se dirige a personas que viajan por negocios, parejas en escapadas románticas, grupos de amigos en vacaciones y familias en busca de opciones de alojamiento cómodas y convenientes.
	Estrategias de marketing	ArrendaFácil destaca con marketing de contenidos, participación en eventos de la industria y alianzas con proveedores de servicios de eventos para ofrecer soluciones completas a arrendatarios.	Airbnb emplea campañas publicitarias en redes sociales y colaboraciones con influencers para promocionar destinos y experiencias únicas. También utiliza programas de referidos para aumentar su base de usuarios.	Vrbo se centra en contenido educativo en su sitio web, colaboraciones con agencias de viajes y ofertas exclusivas para atraer a familias y grupos de amigos a reservar a través de su plataforma.	Booking.com utiliza estrategias de SEO y SEM, junto con programas de fidelización, para atraer tráfico y fomentar la lealtad del cliente a través de una experiencia personalizada.

Competitive Analysis Landscape					
¿Por qué llevar a cabo este análisis?		Este análisis se realizó con la finalidad de poder identificar a nuestros potenciales competidores e idear estrategias y tácticas para diferenciarnos de estos.			
Perfil de Producto	Productos & Servicios	Aplicación web que conecta a arrendadores de diversos espacios, como salones de eventos, jardines y locales comerciales, con arrendatarios que buscan alquilar estos espacios para celebrar eventos sociales, corporativos o especiales. Proporciona una variedad de herramientas y servicios para facilitar la búsqueda, reserva y gestión de espacios para eventos.	Plataforma en línea donde los usuarios pueden alquilar una variedad de alojamientos a corto plazo, desde habitaciones individuales hasta casas completas, en todo el mundo. Además de alojamiento, Airbnb también proporciona experiencias locales únicas organizadas por anfitriones, como recorridos gastronómicos, clases de cocina y excursiones.	Plataforma para alquilar alojamientos vacacionales directamente a través de los arrendadores. Los usuarios pueden encontrar y reservar una amplia gama de propiedades, que van desde casas y apartamentos hasta cabañas y villas, para sus vacaciones o escapadas en todo el mundo.	Plataforma en línea que permite a los usuarios reservar una variedad de alojamientos, incluyendo hoteles, hostales, apartamentos y otros tipos de hospedaje en todo el mundo. Además de alojamiento, Booking.com ofrece la posibilidad de reservar vuelos, alquilar coches y reservar actividades turísticas para completar la experiencia de viaje del usuario.
	Precios & Costos	Los precios de ArrendaFácil varían dependiendo del lugar, el tamaño de la propiedad, servicios, y el tiempo de uso.	El costo promedio puede variar significativamente dependiendo de varios factores, como la ubicación, el tipo de alojamiento, la época del año y la demanda local.	Los precios en VRBO pueden variar significativamente dependiendo de la ubicación, el tamaño de la propiedad, las comodidades ofrecidas y la temporada del año.	Los precios pueden variar significativamente según la ubicación, la temporada, la demanda y el tipo de alojamiento.
	Canales de distribución (Web y/o móvil)	Redes sociales y aplicación web donde los usuarios pueden poner en renta su espacio o alquilar un espacio para eventos.	Sitio web de Airbnb, aplicación móvil de Airbnb, socios afiliados y asociaciones, redes sociales y marketing digital.	Principalmente su sitio web y su aplicación móvil, así como acuerdos de distribución con otros sitios web de viajes o agencias de viajes en línea.	Sitio web de Booking.com, aplicación móvil de Booking, agencias de viajes en línea, alianzas con compañías de viajes, afiliados y asociados.
Análisis SWOT	Fortalezas	Solución completa para la planificación de eventos.	Gran comunidad de anfitriones y usuarios.	Variedad de alojamientos en todo el mundo.	Interfaz fácil de usar y experiencia intuitiva del usuario.
	Debilidades	Dependencia de la disponibilidad de espacios para eventos.	Dependencia de la reputación y opiniones de los usuarios.	Possible saturación del mercado de alquiler vacacional.	Competencia intensa con otras plataformas de alquiler de alojamiento.
	Oportunidades	Expansión mediante aplicaciones integradas para mejorar la comunicación entre sus usuarios.	Desarrollo de nuevas características y servicios para mejorar la experiencia del usuario.	Alianzas estratégicas con proveedores de servicios de eventos.	Aprovechamiento de la tendencia creciente del turismo y los viajes.
	Amenazas	Cambios en la regulación de alquileres vacacionales y eventos.	Possible disminución de la demanda de viajes debido a crisis económicas o sanitarias.	Innovaciones tecnológicas que podrían ser adoptadas por competidores.	Possible pérdida de confianza del usuario debido a problemas de seguridad o calidad del servicio.

2.1.2. Estrategias y tácticas frente a competidores

2.1.2. Estrategias y tácticas frente a competidores

En esta sección se analizarán las estrategias y tácticas que se implementarán para aprovechar las debilidades de la competencia y enfrentar sus fortalezas, así como para abordar las amenazas y oportunidades del mercado. Para ello, hemos empleado un análisis FODA que nos ha permitido identificar nuestras fortalezas y debilidades internas, así como las oportunidades y amenazas externas, todo en función de nuestros dos segmentos objetivos: arrendadores y arrendatarios.

Estrategia de diferenciación:

Para los arrendadores: A diferencia de otras plataformas de alquiler de espacios para eventos, AlquilaFácil ofrecerá dispositivos IoT para la gestión de sus propiedades durante la realización de algún evento, de modo que el arrendador esté al tanto de lo que está sucediendo en todo momento.

Para los arrendatarios: Nuestra interfaz será fácil de usar y estará diseñada para simplificar todo el proceso de búsqueda, comparación y reserva de espacios. Además, gracias a los dispositivos IoT, no tendrán que preocuparse por romper alguna regla de seguridad, debido a que serán notificados en el momento de cometer la infracción.

Estrategia de liderazgo en Costos:

AlquilaFácil buscará ofrecer un modelo flexible y competitivo en costos. Si bien se mantendrán tarifas estándar por el uso de la plataforma, se explorarán modelos de monetización creativa que permitan diversificar los ingresos sin afectar la accesibilidad. Se ofrecerán servicios premium opcionales para arrendadores que deseen destacar sus listados o acceder a herramientas avanzadas de gestión. También se considerará la inclusión de servicios adicionales bajo demanda, así como publicidad no intrusiva y patrocinios de eventos como fuentes complementarias de ingreso.

Estrategia de marketing:

El marketing de AlquilaFácil estará enfocado en construir una comunidad activa y comprometida de arrendadores y arrendatarios. Se buscará destacar los beneficios de la plataforma no solo desde el punto de vista funcional, sino también como un espacio para compartir experiencias, generar confianza y fidelizar usuarios.

Tácticas:

- **Campañas segmentadas:** Se lanzarán campañas digitales específicas para arrendadores y otras para arrendatarios, empleando publicidad en redes sociales, colaboraciones con influencers del sector de eventos y alianzas con proveedores de servicios relacionados.
- **Fomento de comunidad:** Se habilitarán espacios dentro de la plataforma para que los usuarios comparten sus experiencias, recomiendan espacios o servicios, e interactúen entre sí, fortaleciendo así el sentido de comunidad.
- **Eventos y participación activa:** AlquilaFácil participará en ferias comerciales, eventos del sector y actividades colaborativas que aumenten la visibilidad de la marca y fomenten la interacción directa con los usuarios.

2.2. Entrevistas

2.2.1. Diseño de entrevistas

Se definieron preguntas por segmento objetivo para obtener información cualitativa que apoye el desarrollo de nuestra solución.

Preguntas generales:

1. ¿Cuál es tu nombre?
2. ¿Qué edad tienes?
3. ¿Dónde vives actualmente?
4. ¿A qué te dedicas?

Preguntas segmento de arrendadores de espacios para eventos:

El objetivo de entrevistar a los arrendadores es comprender sus necesidades, expectativas y preocupaciones en la gestión de propiedades para eventos. Buscamos conocer su disposición a usar una plataforma como AlquilaFácil que centralice reservas, pagos, comunicación y solución de conflictos, así como su interés en herramientas que les den mayor control y visibilidad de sus espacios.

Con esta información podremos diseñar funciones que respondan a sus prioridades, como aprobación de usuarios, monitoreo, pagos integrados y soporte ante imprevistos. De esta manera, buscamos generar confianza y ofrecer un valor añadido que mejore su experiencia como anfitriones.

1. ¿Qué tan útil te parecería tener una sola plataforma como AlquilaFácil para gestionar reservas, pagos y comunicación con los usuarios?
2. ¿Te gustaría que los usuarios puedan dejar calificaciones y comentarios sobre tu espacio? ¿Por qué?
3. ¿Qué tan importante es para ti poder aprobar o rechazar reservas según el perfil o reputación del usuario?
4. ¿Te serviría ver reportes o estadísticas sobre las reservas y uso de tu espacio?
5. ¿Cómo te gustaría recibir los pagos de tus alquileres? ¿Todo desde la misma plataforma?
6. ¿Cuánto valorás tener una app que te ayude a resolver reclamos o problemas con arrendatarios de forma rápida?
7. ¿Qué tan importante es para vos poder tener visibilidad de lo que ocurre en tu espacio durante un evento?
8. ¿Te gustaría recibir alertas si algo inusual sucede mientras alguien está usando tu espacio?
9. ¿Qué tipo de herramientas o soluciones tecnológicas creés que te ayudarían a gestionar mejor tu propiedad?
10. ¿Te interesaría usar tecnología que te permita tener más control sobre el uso de tu espacio sin necesidad de estar presente?

Preguntas segmento de arrendatarios sociales / usuarios frecuentes de espacios para eventos:

El objetivo de entrevistar a este segmento es identificar sus hábitos de búsqueda, reserva y uso de espacios para eventos, además de sus expectativas en relación con la seguridad, la transparencia y la facilidad de los procesos. También nos interesa conocer la importancia que le dan a las calificaciones, su nivel de comodidad con el sistema de alquiler actual y el valor que otorgan a funciones como reclamos o notificaciones durante el evento.

Con estos hallazgos podremos diseñar una experiencia enfocada en el usuario, que simplifique la búsqueda de espacios confiables e integre herramientas para reforzar la seguridad, la reputación compartida y la comunicación efectiva, logrando así una plataforma clara, accesible y funcional.

1. ¿Qué tan útil te parecería una app como AlquilaFácil que te permita ver espacios disponibles, comparar precios y reservar en el mismo lugar?
2. ¿Qué tanto valoras poder leer opiniones y calificaciones de otros usuarios antes de alquilar un espacio?
3. ¿Cómo te gustaría que sea el proceso de reserva: rápido, seguro, desde la misma plataforma?

4. ¿Qué tan importante sería para vos poder reportar problemas o hacer reclamos directamente desde la app?
5. ¿Qué tan fácil te resulta hoy en día encontrar espacios adecuados para tus eventos?
6. ¿Te sentirías más cómodo/a alquilando si la app tuviera un sistema de reputación tanto para arrendadores como para arrendatarios?
7. ¿Te interesaría recibir notificaciones sobre el estado del lugar durante tu evento? (por ejemplo, si algo se sale de control)
8. ¿Qué tan importante es para ti saber que el espacio cuenta con herramientas que ayuden a garantizar tu seguridad durante el evento?
9. ¿Prefieres lugares que cuenten con tecnología que permita controlar ciertas situaciones mientras se alquila?
10. ¿Qué tipo de herramientas tecnológicas te gustaría que tuvieran los espacios que alquilás para sentirte más tranquilo/a?

2.2.2. Registro de entrevistas

Se llevarán a cabo tres entrevistas por segmento, sumando un total de seis. El registro será en formato mp4, y cada entrevista será independiente, ya que las preguntas y respuestas varían según los entrevistados de cada grupo.

Segmento 1: Arrendadores de espacios para eventos

Entrevista #1

Nombre	Luis David
Apellidos	García Díaz
Edad	21 años
Distrito	Magdalena
Aplicaciones usadas	Instagram, Excel.
Motivación	Ver crecer su negocio y convertir su local en un lugar de referencia para eventos en su comunidad.
Frustración	Posibles accidentes dentro de su local por falta de herramientas para gestionarlo.
Tecnologías	Android, Windows.
Browsers	Opera GX
Entrevistador	Mariano Ames



Evidencia	Microsoft Stream
Duración	00:00 min - 03:42 min
Resumen	El entrevistado señaló que estaría interesado en la aplicación, ya que le permitiría gestionar de manera más sencilla tanto sus locales como a los arrendatarios, mejorando así su negocio. Comentó que en el pasado enfrentó dificultades al organizar reservas por no contar con una herramienta especializada para llevar un control adecuado. También indicó que tuvo inconvenientes con ciertos clientes, los cuales podrían haberse evitado si hubiera tenido acceso a un perfil previo de ellos. Finalmente, expresó su interés en contar con una solución que integre todo el proceso en una sola aplicación, desde la promoción del local y la gestión de reservas y pagos, hasta el control y administración del espacio.

Entrevista #2

Nombre	Manuel Jesús
--------	--------------

Entrevista #2

Apellidos	Chávez Cuba
Edad	21
Distrito	San Martín de Porres
Aplicaciones usadas	Facebook Marketplace
Motivación	Ofrecer experiencias únicas y memorables para sus clientes.
Frustración	Limitaciones presupuestarias que puedan afectar la calidad de los eventos.
Tecnologías	iOS, Android, Windows.
Browsers	Google Chrome, Brave
Entrevistador	Christopher Lecca



Enlace [Microsoft Stream](#)

Duración 00:00 min - 06:31 min

Resumen El entrevistado se presenta como Manuel Chavez, un estudiante de Ingeniería de 21 años. Manuel nos comentó que se encuentra viendo la manera de generar ingresos, por lo que se lo ocurrió adentrarse al mercado de alquiler de espacios para organizar eventos. Durante la entrevista, resaltó la necesidad de una aplicación que facilite la reserva de este tipo de espacios, enfocándose en la gestión del local por parte del arrendador, de modo que pueda estar al tanto de aquellas ocurrencias que puedan perjudicar el desarrollo de su evento. Para finalizar, nos comentó su experiencia previa con otras herramientas similares y cómo es que estas no cuentan con herramientas de gestión inteligentes que faciliten el control del aforo o detección de riesgos.

Entrevista #3

Nombre	Angello Rodolfo
Apellidos	Sosa Colca
Edad	20
Distrito	San Borja
Aplicaciones usadas	Facebook Marketplace
Motivación	Maximizar el uso de sus espacios y generar ingresos adicionales mediante alquileres frecuentes.
Frustración	La falta de control y visibilidad sobre el comportamiento de los inquilinos y posibles daños en los espacios alquilados.
Tecnologías	Android, Windows
Browsers	Brave
Entrevistador	Omar Morales

Evidencia

Enlace [Microsoft Stream](#)

Duración 00:00 min - 04:58 min

Resumen El entrevistado, Angello Sosa, comentó que ve con entusiasmo la propuesta de una aplicación enfocada en la gestión de alquileres de espacios. Considera que esta herramienta es innovadora porque facilitaría el control y la administración de sus propiedades, reduciendo riesgos y optimizando la experiencia tanto para arrendadores como para arrendatarios. Mencionó que la idea le parece acertada, ya que podría marcar la diferencia frente a otras alternativas poco prácticas del mercado actual.

Segmento 2: Arrendatarios sociales frecuentes de espacios para eventos

Nombre	Ricardo Rafael
Apellidos	Rivas Carrillo
Edad	20
Distrito	Callao
Aplicaciones usadas	Facebook Marketplace
Motivación	Realizar practicar en espacio adecuados para su grupo de baile
Frustración	Limitaciones para encontrar espacios que cumplan las necesidades de su grupo de baile.
Tecnologías	Android, Windows.
Browsers	Opera, Edge
Entrevistador	Sebastian Cachis

Evidencia

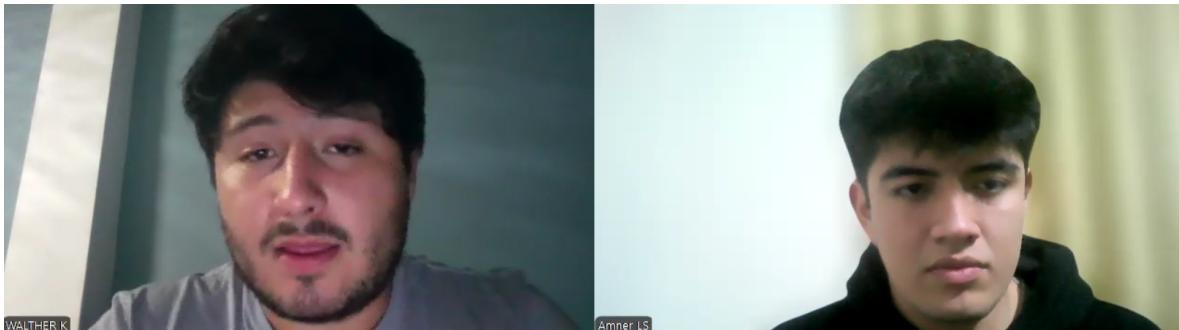
Enlace [Microsoft Stream](#)

Entrevista #1

Duración	00:00 min - 07:44 min
Resumen	<p>En la entrevista, Ricardo Rivas, estudiante de 20 años de ciencias de la computación, expresó que una aplicación como Alquila Fácil le resultaría muy útil para organizar eventos de baile o actuación, ya que actualmente es complicado encontrar y reservar espacios adecuados, con información clara sobre costos, capacidad y disponibilidad. Destacó la importancia de poder comparar precios, ver fotos, leer opiniones y calificaciones, ya que le permitiría ahorrar tiempo y confiar en la calidad del lugar, incluso al alquilar con poca anticipación. Considera fundamental que el proceso de reserva sea rápido, seguro y con contacto directo con el arrendador, y que la app funcione como intermediario en caso de reclamos, brindando respaldo y confianza. También valoró un sistema de reputación tanto para arrendadores como arrendatarios, así como recibir notificaciones durante el evento ante imprevistos. Para él, la seguridad es prioritaria, por lo que aprecia que los espacios cuenten con herramientas como cámaras de vigilancia, sistemas de control de ingreso y tecnología que garantice protección. En resumen, su visión resalta la necesidad de una plataforma confiable, segura y tecnológica que simplifique el acceso a espacios para eventos, brinde transparencia y fortalezca la experiencia de los usuarios.</p>

Entrevista #2

Nombre	Walther
Apellidos	Cachay Villar
Edad	27
Distrito	Surco
Aplicaciones usadas	Facebook Marketplace
Motivación	Ahorrar tiempo y esfuerzo al buscar espacios adecuados, con precios claros y disponibilidad inmediata.
Frustración	Tener que revisar múltiples páginas o redes sociales sin información confiable ni opiniones verificadas.
Tecnologías	Android, Windows.
Browsers	Chrome, Brave
Entrevistador	Amner Llamo



Evidencia	A screenshot of a Microsoft Stream video player. It shows two video feeds side-by-side. The left feed is labeled "WALTHER K" and the right feed is labeled "Amner LS". The video player has a standard interface with play/pause controls and a progress bar.
Enlace	Microsoft Stream
Duración	00:00 min - 08:25 min

Resumen	Walther Kevin, de 27 años y organizador de eventos en Santiago de Surco, considera que una aplicación como AlquilaFácil sería muy útil porque centralizaría la búsqueda, comparación y reserva de espacios, ahorrando tiempo y brindando confianza. Para él, las opiniones de otros usuarios son fundamentales, y valora que el proceso de reserva sea rápido, seguro y dentro de la misma plataforma, con pagos en línea y confirmación inmediata. Actualmente, encuentra difícil localizar espacios adecuados debido a la fragmentación de la información en redes sociales y páginas web. Destaca la importancia de contar con un sistema de reputación, canales de reclamo directos y notificaciones en tiempo real sobre seguridad o fallas técnicas. Además, prefiere lugares con tecnología de control (acceso digital, cámaras, sensores, monitoreo de aforo, internet estable y botón de emergencia), ya que considera la seguridad y la transparencia como factores clave en la experiencia de alquiler.
---------	--

Entrevista #3

Nombre	Alonso Rafael
Apellidos	Castro
Edad	25
Distrito	Miraflores
Aplicaciones usadas	Airbnb, Facebook Marketplace

Entrevista #3

Motivación	Acceder de manera rápida a espacios adecuados para eventos sociales y familiares.
Frustración	La poca transparencia en precios y condiciones de alquiler, así como la dificultad para comparar opciones confiables.
Tecnologías	Android, Windows
Browsers	Google Chrome
Entrevistador	Mariano Ames



Enlace	Microsoft Stream
Duración	00:00 min - 03:53 min

Resumen El entrevistado, Alonso Castro, pertenece al segmento de arrendatarios y compartió su experiencia buscando espacios para eventos. Comentó que la aplicación le resulta una propuesta útil y novedosa, ya que facilitaría encontrar lugares disponibles con mayor rapidez y transparencia. Resaltó que el principal valor estaría en centralizar la información de precios y condiciones, evitando la pérdida de tiempo en negociaciones poco claras. Finalmente, expresó que le parece una idea con gran potencial para mejorar la experiencia de los usuarios que alquilan espacios.

2.2.3. Análisis de entrevistas

Las entrevistas realizadas ofrecen una perspectiva clara y detallada sobre las expectativas y necesidades de los distintos actores que participan en el proceso de alquiler de espacios para eventos. En términos generales, tanto arrendadores como arrendatarios coinciden en la importancia de contar con plataformas que garanticen la seguridad en las transacciones financieras, la transparencia en los acuerdos contractuales y una comunicación directa, clara y eficiente.

Arrendadores de espacios para eventos

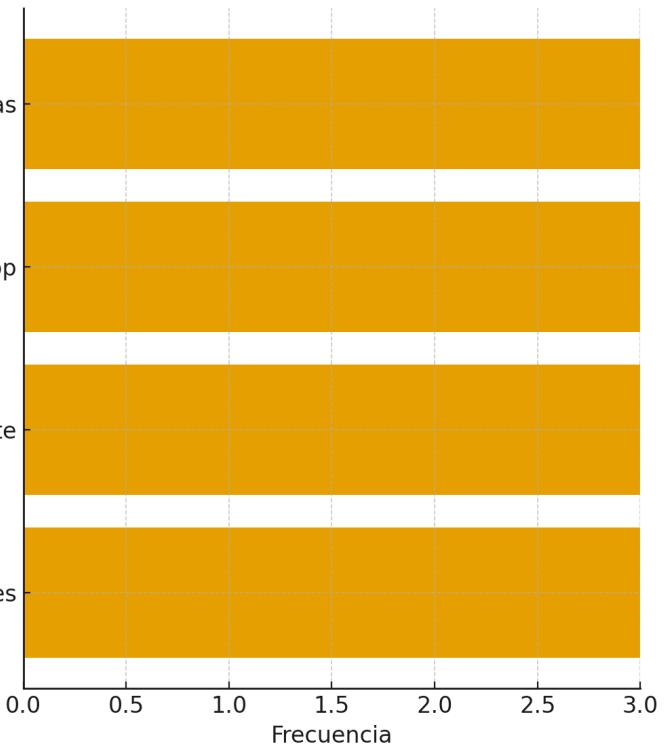
Los arrendadores priorizan la seguridad en los pagos y la validez de los contratos como factores críticos. Además, valoran la posibilidad de promocionar sus locales de manera efectiva, llegando a un público más amplio y relevante. También consideran indispensable una comunicación fluida con los arrendatarios para evitar conflictos o malentendidos. De hecho, algunos están dispuestos a invertir en planes premium que les brinden beneficios adicionales, como una mayor visibilidad en la plataforma.

En conclusión, los arrendadores esperan que la plataforma les proporcione seguridad, visibilidad, canales de comunicación confiables y herramientas que potencien el rendimiento de sus espacios.

Característica objetiva	Frecuencia	Porcentaje	Sustento desde entrevistas
Utiliza redes sociales para captar clientes	3 de 3	100%	Se mencionan principalmente Instagram y Facebook
Gestiona reservas manualmente	3 de 3	100%	Utilizan agenda física o Excel
Recibe consultas por WhatsApp	3 de 3	100%	Todo el proceso se coordina vía WhatsApp
Necesita un sistema de agenda para evitar dobles reservas	3 de 3	100%	Reportaron problemas por doble reserva

Análisis cuantitativo - Arrendadores

Necesita un sistema de agenda para evitar dobles reservas



Arrendatarios para eventos

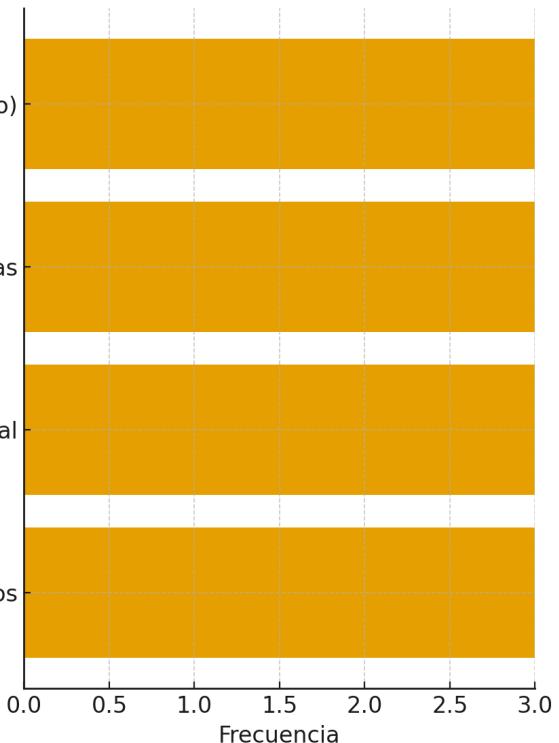
Por su parte, los arrendatarios valoran sobre todo la facilidad para encontrar y reservar espacios adecuados, especialmente en situaciones imprevistas o de última hora. La claridad en los contratos y una comunicación transparente con los arrendadores también son aspectos determinantes para garantizar experiencias satisfactorias. Además, esperan encontrar una amplia variedad de locales que se adapten a distintos tipos de actividades y horarios. La disponibilidad de filtros y herramientas de búsqueda avanzada resulta clave para optimizar la elección del espacio ideal.

En resumen, los arrendatarios demandan plataformas fáciles de usar, con procesos transparentes, comunicación eficiente y una oferta amplia y diversificada de espacios.

Característica objetiva	Frecuencia	Porcentaje	Sustento desde entrevistas
Usa redes sociales o sitios web para buscar espacios	3 de 3	100%	Todos señalaron estas herramientas como su primera opción
Realiza reservas de forma habitual	3 de 3	100%	Mencionan alquilar espacios regularmente
Busca espacios para actividades familiares o recreativas	3 de 3	100%	Los eventos más frecuentes son de carácter recreativo o familiar
Valora herramientas de organización del tiempo (agenda/calendario)	3 de 3	100%	Reconocen la utilidad de agendas digitales para evitar desorden

Análisis cuantitativo - Arrendatarios

Valora herramientas de organización del tiempo (agenda/calendario)



Determinación de Personas

- **Persona Principal: Arrendatarios**

Justificación: Son quienes interactúan de manera más constante con la plataforma y dependen de ella para encontrar y reservar espacios. Sus necesidades (usabilidad, transparencia, comunicación y variedad) son determinantes para el éxito del producto, por lo que constituyen el foco principal del diseño y desarrollo.

- **Persona Secundaria: Arrendadores**

Justificación: Aunque fundamentales para el funcionamiento de la plataforma, los arrendadores cumplen un rol complementario en comparación con los arrendatarios. Sus prioridades giran en torno a la seguridad en transacciones, la promoción de sus espacios y la eficiencia en la gestión, pero su nivel de interacción es menos frecuente que el de los usuarios que reservan.

2.3. Needfinding

2.3.1. User Personas

Presentaremos los User Persona por cada segmento objetivo, en los cuales nos basamos en los usuarios ideales de cada segmento: Sergio Aguirre (Arrendatario), Claudia Camareño (Arrendador).

Segmento 1: Arrendatario

Este perfil representa a Sergio, un joven emprendedor de 19 años estudiante de Ingeniería de Software, ubicado en San Isidro, Lima. Como arrendatario que representa el 60% del mercado objetivo, Sergio se especializa en la organización de eventos, habiendo expandido su negocio desde pequeñas reuniones sociales hasta eventos corporativos y lanzamientos de productos. Sus principales fortalezas incluyen la planificación de eventos y el diseño creativo de experiencias, aunque enfrenta desafíos relacionados con limitaciones presupuestarias y la gestión eficiente de recursos durante el desarrollo de sus eventos.

NAME

Sergio Aguirre

MARKET SIZE



60 %

TYPE

Arrendatario**Demographic**

Male 19 years

San Isidro, Lima

Estudiante de Ingeniería de Software

Skills

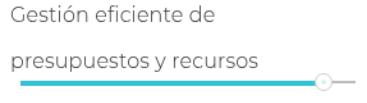
Planificación de eventos



Creatividad en diseño de experiencias



Gestión eficiente de presupuestos y recursos

**Background**

Sergio es un joven emprendedor que ha estado organizando eventos durante los últimos 3 años. Comenzó organizando pequeñas reuniones sociales y fiestas para amigos y familiares, pero con el tiempo ha expandido su negocio para incluir eventos corporativos, lanzamientos de productos y bodas. Tiene un enfoque creativo y le apasiona crear experiencias memorables para sus clientes.

Goals

Coordinar todos los aspectos logísticos de los eventos, desde la selección del lugar hasta la gestión del personal y aforo. Diseñar conceptos únicos para cada evento, incluyendo la decoración, la ambientación y las actividades especiales. Asegurar que los eventos se ejecuten sin problemas y cumplan con las expectativas de los clientes en términos de calidad y experiencia.

Motivators

- Ofrecer experiencias únicas y memorables para sus clientes.
- Expandir su negocio y establecerse como un organizador de eventos reconocido en su área.

Challenges

- Limitaciones presupuestarias que pueden afectar la calidad de los eventos.
- Dificultades para encontrar espacios que cuenten con características específicas.
- Falta de herramientas de gestión durante el desarrollo de sus eventos.

Preferred channels

PC



Web app



Smartphone

Technology**UXPRESSIA**

This persona was built in upressoia.com

Segmento 2: Arrendador

Claudia es una podóloga de 46 años de Surco, Lima, que representa el 40% del mercado como arrendador. Después de más de 10 años en su profesión, ha decidido aventurarse en el negocio del alquiler de espacios para generar ingresos adicionales. Su perfil destaca por sus habilidades en organización, comunicación

y adaptabilidad. Su principal motivación es convertir su local en un lugar de referencia para eventos comunitarios, aunque enfrenta desafíos en la promoción efectiva de su espacio y la gestión de reservas durante períodos de alta demanda.

PERSONA: Claudia Camareño

NAME

MARKET SIZE

TYPE

Claudia Camareño

40 %

Arrendador

Background

Claudia ha trabajado como podóloga durante más de 10 años. Con el fin de obtener un ingreso adicional, es que recientemente se aventuró en el negocio del alquiler de espacios. Para ello, ha decidido arrendar un local que ella posee, con la esperanza de que esta idea funcione y pueda expandirse.

Demographic

Female 46 years
 Surco, Lima
 Podóloga

Goals

Aumentar la popularidad de su local, de forma que pueda ser alquilado en distintos eventos, promocionándolo activamente en redes sociales y otros canales de marketing para incrementar su visibilidad.

Mantener un estándar alto de organización de su propiedad, de manera que se encuentre en excelentes condiciones para cualquier evento que se pueda presentar.

Skills

Organización

Comunicación y negociación

Adaptabilidad

Motivators

- Ver crecer su negocio y convertir su local en un lugar de referencia para eventos en su comunidad.
- Proporcionar experiencias memorables y satisfactorias a sus clientes, contribuyendo a la felicidad y la celebración de momentos importantes.
- Explorar nuevas oportunidades de negocio y expandir su presencia en el mercado de eventos sociales.

Challenges

- Dificultades para promocionar eficazmente su local y atraer clientes nuevos y recurrentes.
- Desafíos en la gestión de reservas y la coordinación de eventos, especialmente en períodos de alta demanda.
- Preocupación la competencia de otros locales para eventos en la zona y la posibilidad de perder clientes ante ofertas más atractivas.
- Posibles accidentes dentro de su local por falta de herramientas para gestionarlo.

Preferred channels

PC

Smartphone

Technology

UXPRESSIA

This persona was built in uxpressia.com

2.3.2. User Task Matrix

PROFESSEUR : M.DA ROS

♦ 24 / 194 ♦

BTS SIO BORDEAUX - LYCÉE GUSTAVE EIFFEL

User Persona Principal

Nombre: Sergio Aguirre

Rol: Arrendatario

Task	Frequency	Importance
Buscar y reservar espacios para eventos	High	High
Comunicarse con los arrendadores de los espacios	High	High
Buscar variedad en los tipos de espacios disponibles	High	High
Reservar espacios para diferentes tipos de eventos	High	High
Estar dispuesto a pagar por una mejor experiencia	Medium	Medium
Utilizar una plataforma fácil de usar e intuitiva	High	High

User Persona Secundaria

Nombre: Claudia Cañamero

Rol: Arrendadora

Task	Frequency	Importance
Gestionar reservas y coordinar eventos	High	High
Promocionar el local en redes sociales y otros canales de marketing	High	High
Mantener el local en óptimas condiciones de limpieza y mantenimiento	High	High
Establecer comunicación efectiva con clientes y proveedores	High	High
Explorar nuevas oportunidades de negocio y crecimiento	Medium	Medium

2.3.3. Empathy Mapping

Persona Principal: Arrendatarios de eventos sociales

Este mapa de empatía profundiza en la experiencia del usuario arrendatario, mostrando cómo Sergio busca espacios para sus eventos a través de recomendaciones y plataformas online. Sus principales dolores incluyen el tiempo dedicado a la búsqueda y evaluación de espacios, así como los esfuerzos de comunicación con propietarios. Por otro lado, valora el acceso a una variedad de opciones y la conveniencia de las plataformas digitales para reservas rápidas y seguras, expresando satisfacción cuando encuentra espacios que cumplen con todas sus expectativas y necesidades específicas.

1.WHO are we empathizing with?

Organizadores de eventos como Sergio Aguirre, que buscan espacios para sus reuniones familiares, fines de semana con amigos u otros eventos recreativos.

7.What do they THINK and FEEL?

“
Piensan en la importancia de encontrar un espacio que se ajuste a sus necesidades y preferencias.

Sienten curiosidad por descubrir nuevas opciones de espacios para eventos que satisfagan sus expectativas.

Pueden sentirse frustrados si encuentran dificultades para encontrar un espacio adecuado o si experimentan malentendidos en el proceso de reserva.

2.What do they need to DO?

Necesitan buscar comprobadamente opciones de espacios para eventos que se adapten a sus requisitos.

Deben evaluar y comparar diferentes alternativas para tomar una decisión informada sobre qué espacio reservar.

Necesitan realizar la reserva del espacio de manera fácil y segura.

6.What do they HEAR?

Oyen sobre diferentes opciones de espacios para eventos a través de amigos y redes.

Recibe recomendaciones de sitios web y plataformas para encontrar locales para eventos.



3.What do they SEE?

Una variedad de opciones de espacios para eventos disponibles en plataformas en línea y redes sociales.

Observen las características y fotos de los espacios para evaluar si cumplen con sus necesidades y expectativas.

5.What do they DO?

Utilizan plataformas en línea como Airbnb y otras para buscar, evaluar y reservar espacios para eventos.

Interactúan con los dueños de los espacios a través de mensajes para aclarar dudas y coordinar detalles adicionales.

PAINS

Dedica tiempo y esfuerzo en la búsqueda y evaluación de diferentes opciones de espacios para eventos.

Se esfuerzan por comunicarse efectivamente con los propietarios de los espacios para garantizar una reserva exitosa.

GAINS

Obtenga acceso a una variedad de opciones de espacios para eventos que de otra manera no podrían haber descubierto fácilmente.

Disfrutan de la conveniencia de reservar espacios para eventos de manera rápida y segura a través de plataformas en línea.

4.What do they SAY?

“
Pueden expresar su satisfacción si encuentran un espacio que cumple con todas sus expectativas y necesidades.

Comenta sobre su experiencia positiva al utilizar plataformas en línea para encontrar y reservar espacios para eventos.

UXPRESSIA

This persona was built in upressoia.com

Persona Secundaria: Arrendadores de espacios para eventos

El mapa de empatía de Claudia ilustra la perspectiva del propietario que busca generar ingresos adicionales alquilando su espacio. Escucha sobre oportunidades a través de redes sociales y recibe sugerencias de plataformas online para promocionar su local. Sus principales desafíos incluyen la dedicación de tiempo y recursos para mantener y promocionar su espacio, así como el esfuerzo por ofrecer experiencias excepcionales. Sus beneficios se centran en obtener ingresos extra y la satisfacción de proporcionar un espacio funcional para organizadores de eventos, lo que puede generar recomendaciones y reservas futuras.

1.WHO are we empathizing with?

Propietarios de casas, locales u otros espacios que ofrecen sus propiedades para eventos, como Claudia Cañamero y otros propietarios en situaciones similares.

7.What do they THINK and FEEL?

“
Piensan en la importancia de encontrar un espacio que se ajuste a sus necesidades y preferencias.

Sienten curiosidad por descubrir nuevas opciones de espacios para eventos que satisfagan sus expectativas.

Pueden sentirse frustrados si encuentran dificultades para encontrar un espacio adecuado o si experimentan malentendidos en el proceso de reserva.

2.What do they need to DO?

Necesitan promocionar activamente sus espacios para eventos en plataformas en línea y redes sociales.

Deben mantener sus propiedades limpias, seguras y bien mantenidas para atraer y retener a los clientes.

Necesitan establecer una comunicación clara y efectiva con los organizadores de eventos para garantizar una experiencia sin problemas.

6.What do they HEAR?

Escuchan sobre la demanda de espacios para eventos a través de amigos, familiares y redes sociales.

Reciben sugerencias de plataformas en línea y sitios web para promocionar sus espacios y llegar a posibles organizadores de eventos.



3.What do they SEE?

Ven la oportunidad de generar ingresos adicionales al alquilar sus propiedades para eventos.

Observan las características similares y fotos de otros espacios en plataformas en línea para evaluar la competencia y mejorar su oferta.

5.What do they DO?

Utilizan plataformas en línea como Airbnb y otras para buscar, evaluar y reservar espacios para eventos.

Interactúan con los organizadores de eventos a través de mensajes para aclarar dudas y coordinar detalles adicionales.

PAINS

Dedican tiempo y recursos en la promoción y mantenimiento de sus espacios para eventos.

Se esfuerzan por ofrecer una experiencia excepcional a los organizadores de eventos para generar comentarios positivos y retener clientes.

GAINS

Obtienen ingresos adicionales al alquilar sus propiedades para eventos.

Disfrutan de la satisfacción de proporcionar un espacio agradable y funcional para los organizadores de eventos, lo que puede generar recomendaciones y reservas futuras.

4.What do they SAY?

“
Pueden expresar su satisfacción si logran alquilar sus espacios con éxito y recibir comentarios positivos de los organizadores de eventos.

Comparten su experiencia al trabajar con plataformas en línea para gestionar reservas de sus espacios para eventos.

2.3.4. As-is Scenario Mapping

Persona Principal: Arrendatarios

Este mapa de experiencia ilustra el proceso actual que atraviesan los arrendatarios al buscar y reservar espacios para eventos. El journey comienza con la búsqueda y selección de espacios a través de plataformas online, donde los usuarios evalúan diferentes opciones según el tipo de evento que planean organizar. Durante las fases de lectura de opiniones y negociación, experimentan una mezcla de expectativas positivas y ansiedad por encontrar el espacio perfecto. El proceso se vuelve

más complejo durante la confirmación y programación de la reserva, donde deben coordinar múltiples detalles logísticos. A lo largo del journey, los arrendatarios enfrentan incertidumbre sobre estándares mínimos, preocupaciones por malentendidos en términos de alquiler, y la necesidad constante de planificación logística para garantizar una experiencia exitosa.

As-is Scenario: Arrendatarios		Búsqueda y selección de espacios	Lectura de opiniones sobre el espacio seleccionado	Negociación de la reserva	Confirmar la reserva	Agendar la reserva			
Steps	Doing	Navegación en plataformas en línea	Búsqueda de espacios que cumplen con los requisitos del evento	Lectura en foros sobre las opiniones de otros clientes sobre el espacio	Intercambio de consultas, condiciones y negociación de precios	Coordinación de fecha y hora de la reserva	Coordinación de detalles del espacio para la fecha	Apuntar los detalles del espacio, como ubicación, salones, etc	Guardar en una agenda externa el día y hora acordados para no olvidarse
Thinking	Evaluación de la idoneidad de los espacios para diferentes tipos de eventos	Incertidumbre por encontrar un espacio que cumpla con los estándares mínimos	Evaluación de la confiabilidad y seguridad del espacio para ser alquilado	Evaluación del trato brindado por el arrendador	Evaluación de la confiabilidad y conveniencia	Planificación logística para garantizar una experiencia sin problemas durante el evento	Elección del lugar adecuado para guardar el recordatorio de la reserva		
Feeling	Emoción por la planificación de futuros eventos	Ansiedad por encontrar el espacio perfecto que se ajuste a sus necesidades	Expectativas de valoraciones positivas para animarse a alquilar	Expectativas de respuestas rápidas y claras por parte del arrendador	Preocupación por posibles malentendidos o conflictos en los términos de alquiler	Alivio por la confirmación de la reserva	Cansancio por el extenso proceso de búsqueda realizado	Preocupación por que el evento se realice de la forma planeada	

Persona Secundaria: Arrendadores

Arrendadores Este mapa muestra la experiencia actual de los propietarios de espacios que buscan alquilar sus locales para eventos. El proceso inicia con la promoción del espacio en diferentes medios y plataformas, donde los arrendadores enfrentan incertidumbre sobre cómo encontrar clientes adecuados para su espacio. Durante el contacto inicial y la negociación, deben evaluar cuidadosamente las solicitudes de los posibles clientes mientras manejan la tensión de las negociaciones. Las fases de confirmación y programación generan alivio, pero también requieren una coordinación detallada y la preocupación por no olvidar aspectos importantes del acuerdo. Finalmente, durante la gestión del evento, los arrendadores deben mantener comunicación constante con los arrendatarios y enfrentar la preocupación de que el local pueda resultar dañado, mientras buscan asegurar opiniones positivas para futuras reservas.

As-is Scenario: Arrendadores		Promoción para alquilar su espacio	Contacto con posible arrendatario	Negociación de la reserva	Confirmar la reserva	Agendar la reserva	Gestión del espacio durante el evento			
Steps	Doing	Publicación del alquiler del espacio en diferentes medios	Comenta con conocidos acerca de la disponibilidad de su local	Se comunica por diferentes medios con el posible cliente para brindar detalles del local	Intercambio de consultas, condiciones y negociación de precios	Coordinación de fecha y hora de la reserva	Coordinación de detalles del espacio para la fecha	Apuntar los detalles del arrendatario	Guardar en una agenda externa el día y hora acordados para no olvidarse	Comunicación constante con el arrendatario para asegurarse que el local esté bien
Thinking	Incertidumbre por encontrar un cliente para su espacio	Piensa en cómo persuadir al posible cliente	Evaluación de los requisitos pedidos por el posible cliente	Evaluación de la confiabilidad y conveniencia	Planificación de beneficios que brindará al arrendatario	Elección del lugar adecuado para guardar el recordatorio de la reserva	Considerar en posibles circunstancias donde el local resulte dañado			
Feeling	Ansiedad por encontrar un arrendatario que deseé su local	Tensión por la negociación que se aproxima	Expectativas de respuestas rápidas y favorables	Preocupación por posibles malentendidos durante la negociación	Alivio por la confirmación de la reserva	Considerar por el extenso proceso de búsqueda de cliente realizado	Preocupación por no olvidarse del acuerdo realizado	Preocupación por resultados de la local en óptimas condiciones		

2.4. Ubiquitous Language

Término Definición

Tenant Arrendatario, quien realiza la reserva de un espacio.

Landlord Arrendador, quien publica sus espacios para ser reservados.

Local Espacio para eventos que contiene diferentes características de acuerdo con el arrendador.

Booking Reserva de un espacio, con identificación de arrendatario, fecha y hora de inicio y fin.

Comment Comentario acerca de un local anexado a su reseña con calificación.

Calendar Agenda de reservas presentadas tanto para arrendador como arrendatario.

Report Denuncia hacia algún espacio publicada por un arrendatario de este mismo.

District Distrito de ubicación del espacio publicado.

Features Características adicionales del espacio publicado.

Capítulo III: Requirements Specification

3.1. To-Be Scenario Mapping

Persona Principal: Arrendatarios

Este mapa representa la experiencia optimizada para los arrendatarios utilizando la plataforma AlquilaFácil. El journey mejorado comienza con una navegación intuitiva y filtros personalizados que permiten a los usuarios encontrar espacios que se adapten específicamente a sus necesidades. Durante la fase de evaluación, los arrendatarios experimentan mayor seguridad al contar con información detallada sobre cada espacio y una comunidad activa de reseñas de otros organizadores. El proceso de reserva se simplifica significativamente con la selección automática de fechas y horarios, precios transparentes basados en horas reservadas, y un sistema de pago establecido. La experiencia emocional evoluciona desde el asombro inicial por la variedad de opciones disponibles hasta la tranquilidad final de poder ahorrar tiempo en la planificación del evento, eliminando la incertidumbre y ansiedad del proceso tradicional.



Persona Secundaria: Arrendadores

Este mapa muestra cómo la plataforma AlquilaFácil transforma la experiencia de los propietarios de espacios. El proceso optimizado permite a los arrendadores publicar información detallada de sus locales directamente en la aplicación, eliminando la necesidad de promoción en múltiples canales. La plataforma facilita el contacto directo con arrendatarios interesados y automatiza gran parte del proceso de negociación mediante precios preestablecidos por hora de reserva. Las confirmaciones se manejan a través de notificaciones automáticas, y las reservas se integran seamless en un sistema de calendario. La experiencia emocional mejora considerablemente, pasando de la ansiedad inicial por encontrar clientes a la tranquilidad de saber que el proceso está bien estructurado. Los arrendadores sienten alivio al no perder tiempo en búsquedas activas de clientes y experimentan alegría por la simplicidad del sistema de reservas, culminando con la tranquilidad de recibir notificaciones automáticas sobre cualquier novedad en sus espacios.



3.2. User Stories

Epics

Epic / Story ID	Título	Descripción	Historias de Usuario Relacionadas
EP01	Gestión de usuarios y autenticación	Como usuario de AlquilaFácil, quiero poder registrarme fácilmente en la aplicación y acceder a mi cuenta, para poder utilizar las funcionalidades principales sin complicaciones.	US01, US02, US14, US17

Epic / Story ID	Título	Descripción	Historias de Usuario Relacionadas
EP02	Registro y gestión de espacios	Como arrendador de AlquilaFácil, quiero poder registrar, modificar y promocionar mis espacios en la plataforma para asegurarme de que los arrendatarios puedan ver y reservar mis propiedades de manera fácil y rápida.	US03, US12, US13
EP03	Búsqueda y reserva de espacios	Como arrendatario de AlquilaFácil, quiero poder buscar y reservar espacios de acuerdo con mis necesidades (por capacidad, categoría, y ubicación), para encontrar el lugar perfecto para mi evento y asegurar su disponibilidad.	US04, US05, US06, US07, US11, US15, US16
EP04	Gestión de agenda de usuario	Como usuario de AlquilaFácil, quiero poder gestionar mis reservas y recibir notificaciones sobre cambios y actividades relacionadas con mis espacios, para mantenerme organizado y al tanto de todo en tiempo real.	US08, US09, US10
EP05	Seguridad y monitoreo de eventos	Como arrendador, quiero contar con sensores que me alerten sobre aforo, ruido, humo o movimiento no autorizado, para garantizar la seguridad durante los eventos.	US18, US19, US20, US21, US22
EP06	Experiencia de visitante y landing page	Como visitante de AlquilaFácil, quiero acceder a una landing clara, informativa y navegable, para comprender rápidamente los beneficios de la plataforma y cómo interactuar con ella sin necesidad de registrarme.	US23, US24, US25
User Stories			
ID de la user story	Título	Descripción	Criterios de aceptación
US01	Registrar usuario	Como usuario de la aplicación móvil de AlquilaFácil de un espacio para eventos, quiero poder registrarme fácilmente en AlquilaFácil para ofrecer mi espacio en alquiler y llegar a más clientes potenciales.	Escenario 01: Registro exitoso Dado que un usuario desea registrar su espacio en AlquilaFácil. Cuando el usuario completa el formulario de registro con la información requerida. Entonces el usuario recibe una confirmación de registro y puede acceder a su cuenta.
			EP01 Escenario 02: Validación de datos Dado que un usuario completa el formulario de registro en AlquilaFácil. Cuando el usuario envía el formulario. Entonces los datos proporcionados se validan para garantizar la precisión y la autenticidad.
US02	Iniciar sesión	Como usuario registrado de la aplicación móvil de AlquilaFácil, quiero poder iniciar sesión fácilmente en mi cuenta, para gestionar mi espacio en alquiler y acceder a mis mensajes y notificaciones.	Escenario 01: Inicio de sesión exitoso Dado que un usuario registrado desea acceder a su cuenta en AlquilaFácil. Cuando el usuario ingresa su correo electrónico y contraseña correctos en el formulario de inicio de sesión. Entonces el usuario es autenticado exitosamente y se le otorga acceso a su cuenta.
			EP01 Escenario 02: Error en el inicio de sesión por credenciales incorrectas Dado que un usuario registrado intenta acceder a su cuenta en AlquilaFácil. Cuando el usuario ingresa una combinación incorrecta de correo electrónico o contraseña. Entonces se le muestra un mensaje de error indicando que las credenciales son incorrectas y se le solicita que intente nuevamente.
US03	Registrar espacios	Como arrendador de un espacio para eventos en AlquilaFácil, quiero poder registrar mi espacio para comenzar a recibir solicitudes de reserva lo antes posible.	Escenario 01: Registro de datos básicos Dado que un arrendador desea registrar su espacio en AlquilaFácil. Cuando completa distintos campos organizados en varios pasos detallados que incluyen información detallada sobre la propiedad. Entonces puede enviar el registro con éxito y recibir confirmación de su inclusión en la plataforma.
			EP02 Escenario 02: Validación de datos requeridos en cada paso del registro Dado que un arrendador está completando el registro de su espacio en AlquilaFácil. Cuando rellena todos los campos correspondientes a un paso en específico. Entonces se activa el botón que le permite continuar al siguiente paso.

ID de la user story	Título	Descripción	Criterios de aceptación	Relacionado con (Epic ID)
US04	Buscar espacios disponibles	Como arrendatario, quiero poder buscar fácilmente espacios disponibles en AlquilaFácil para encontrar el lugar perfecto para mi evento.	<p>Escenario 01: Búsqueda principal por ubicación Dado que un arrendatario busca un espacio para eventos en una ubicación específica. Cuando el arrendatario ingresa la ubicación deseada en el campo de búsqueda. Entonces se muestran los espacios disponibles en esa ubicación.</p>	EP03
US05	Filtrar espacios disponibles	Como arrendatario, quiero poder filtrar los espacios disponibles por capacidad y categoría, para encontrar uno que cumpla con mis criterios específicos.	<p>Escenario 01: Filtrado por capacidad Dado que un arrendatario desea un espacio con capacidad para un número específico de personas. Cuando el arrendatario aplica un filtro de capacidad en la búsqueda. Entonces se muestran solo los espacios que cumplen con ese criterio.</p> <p>Escenario 02: Filtrado por categoría Dado que un arrendatario desea un espacio de una categoría en específico. Cuando el arrendatario aplica un filtro de categoría en la búsqueda. Entonces se muestran solo los espacios que cumplen con ese criterio.</p>	EP03
US06	Visualizar información del espacio	Como arrendatario, quiero poder visualizar información detallada del espacio, para evaluar si cumple con los requisitos de mi evento antes de hacer una reserva.	<p>Escenario 01: Visualización de información Dado que el arrendatario selecciona un espacio en AlquilaFácil. Cuando accede a la página de detalles del espacio. Entonces puede visualizar información detallada como el aforo máximo, descripción del espacio y servicios disponibles.</p> <p>Escenario 02: Visualización de reseñas del espacio Dado que el arrendatario está revisando un espacio. Cuando accede a la página de comentarios. Entonces puede visualizar las reseñas y calificaciones dejadas por otros usuarios sobre ese espacio.</p>	EP03
US07	Reservar espacios	Como arrendatario, quiero poder reservar un espacio para mi evento en AlquilaFácil para garantizar su disponibilidad en la fecha deseada.	<p>Escenario 01: Proceso de Reserva Dado que un arrendatario ha encontrado el espacio ideal en AlquilaFácil. Cuando selecciona un espacio y una fecha. Entonces se muestra un formulario para completar los detalles de reserva.</p> <p>Escenario 02: Pago de la reserva con comprobantes Dado que el usuario está a punto de culminar el proceso de reserva de un espacio. Cuando el usuario presione el botón de realizar reserva. Entonces el usuario realizará el pago de la reserva a través de comprobantes.</p> <p>Escenario 03: Confirmación de Reserva Dado que un arrendatario ha realizado el pago de la reserva en AlquilaFácil. Cuando el arrendatario es redirigido a la aplicación. Entonces recibe una confirmación de reserva y los detalles se actualizan en su cuenta.</p>	EP03

ID de la user story	Título	Descripción	Criterios de aceptación	Relacionado con (Epic ID)
US08	Gestionar calendario de reservas	Como usuario de AlquilaFácil, quiero poder controlar una agenda de reservas para tener un horario organizado y evitar conflictos futuros.	<p>Escenario 01: Existencia de reserva de usuario normal Dado que un arrendatario ha realizado una reserva de uno de mis espacios. Cuando el propietario accede al calendario. Entonces puede ver el día de la reserva resaltada en color rojo.</p>	
			<p>Escenario 02: Existencia de reserva de usuario premium Dado que un arrendatario con suscripción premium ha realizado una reserva de uno de mis espacios. Cuando el propietario accede al calendario. Entonces puede ver el día de la reserva resaltada en color amarillo.</p>	EP04
			<p>Escenario 03: Existencia de reserva de espacio ajeno Dado que un arrendatario ha realizado una reserva de espacio. Cuando este accede al calendario. Entonces puede ver el día de su reserva resaltada en color azul.</p>	
US09	Calificar y comentar sobre espacios	Como arrendatario, quiero poder publicar mi reseña sobre un espacio que he reservado para que otros usuarios puedan conocer mi experiencia sobre este.	<p>Escenario 01: Permiso para opinar sobre un espacio Dado que un arrendatario tiene una reserva culminada de un espacio. Cuando accede a su información a través del calendario. Entonces se presenta la opción habilitada para que pueda publicar su reseña sobre este.</p>	EP04
			<p>Escenario 02: Aporte de reseña Dado que un arrendatario ha llenado todos los campos de reseña. Cuando presiona el campo de realizar reseña. Entonces esta se publica para que todos los usuarios la puedan ver.</p>	
US10	Notificar sobre actividades de arrendatarios	Como arrendador de un espacio para eventos en AlquilaFácil, quiero recibir notificaciones Cuando un arrendatario ha realizado cierta acción sobre mi espacio para estar al tanto de su estado y tomar las acciones pertinentes.	<p>Escenario 01: Notificación de reserva Dado que el arrendador ha publicado un espacio en AlquilaFácil. Cuando un arrendatario realiza una reserva para dicho espacio. Entonces el arrendador recibe una notificación en la aplicación indicando que el espacio ha sido reservado, con detalles de la fecha y el nombre del arrendatario.</p>	EP04
			<p>Escenario 02: Notificación de reseña Dado que el arrendador ha publicado un espacio en AlquilaFácil. Cuando un arrendatario realiza una reseña sobre dicho espacio. Entonces el arrendador recibe una notificación en la aplicación que le permite ver el comentario y la calificación otorgada, ayudándole a monitorear la satisfacción del cliente.</p>	
US11	Controlar espacios favoritos	Como arrendatario en AlquilaFácil, quiero poder agregar un espacio a mis favoritos para guardar y acceder rápidamente a los lugares de interés para futuras reservas.	<p>Escenario 01: Agregar a favoritos Dado que el arrendatario está visualizando la página de detalles de un espacio. Cuando selecciona la opción de "Agregar a favoritos". Entonces el espacio se guarda en la lista de favoritos de forma local.</p> <p>Escenario 02: Eliminar de favoritos Dado que el arrendatario ha guardado un espacio como favorito. Cuando selecciona la opción de "Eliminar de favoritos" en dicho espacio. Entonces el espacio se elimina de la lista de favoritos, y ya no se mostrará al realizar el filtrado.</p>	EP03

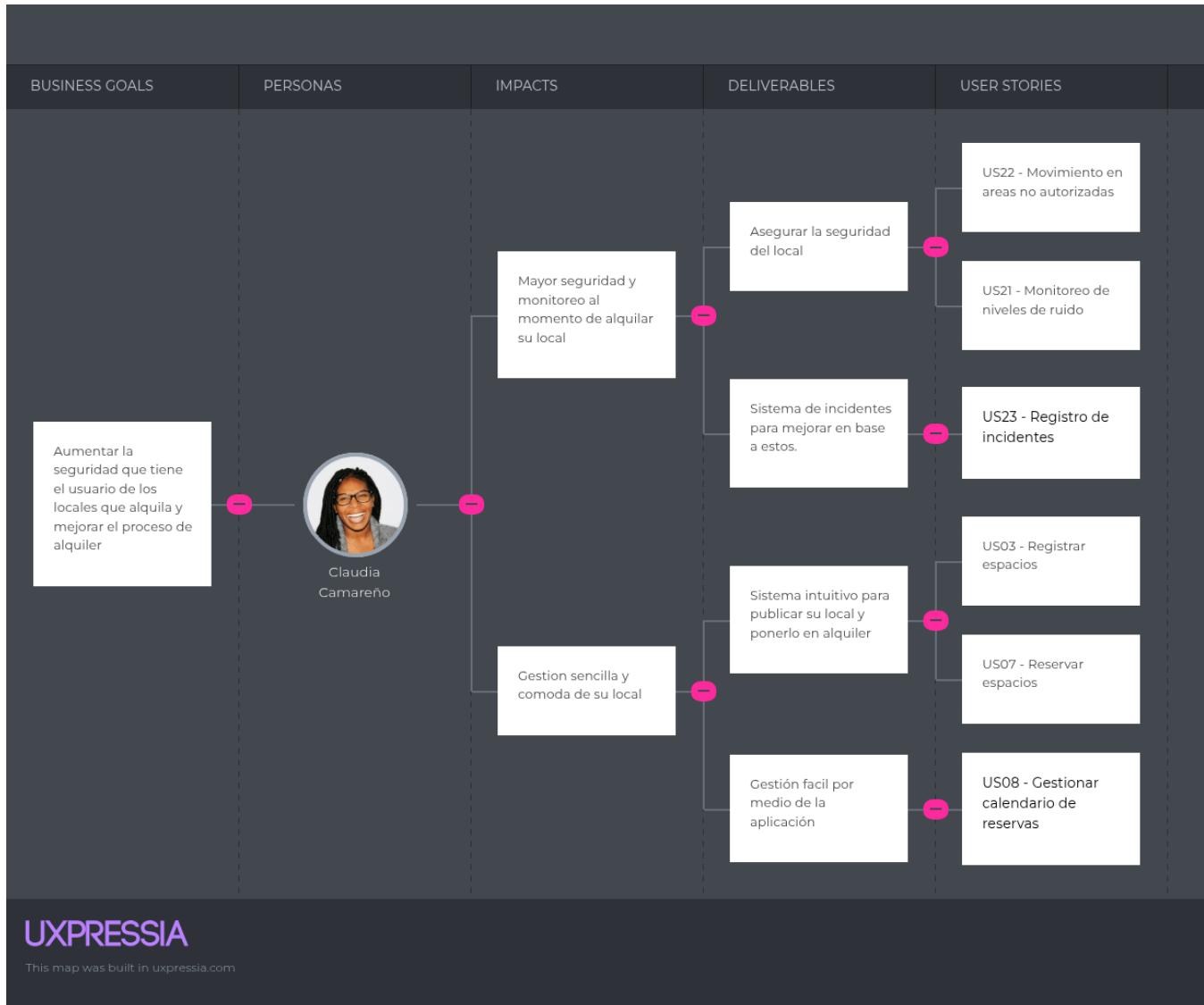
ID de la user story	Título	Descripción	Criterios de aceptación	Relacionado con (Epic ID)
US12	Visualizar espacios propios publicados	Como arrendador en AlquilaFácil, quiero poder visualizar todos los espacios que he publicado en la plataforma, para gestionar fácilmente la información de mis propiedades.	<p>Escenario 01: Listado de espacios publicados</p> <p>Dado que el arrendador ha iniciado sesión en su cuenta de AlquilaFácil. Cuando accede a la sección "Mis espacios". Entonces puede ver una lista de todos los espacios que ha publicado, incluyendo información básica como nombre del espacio, ubicación.</p>	EP02
US13	Modificar espacios publicados	Como arrendador de un espacio en AlquilaFácil, quiero poder modificar la información de los espacios que he publicado, incluyendo detalles como la descripción, aforo, servicios disponibles, tarifas y visibilidad temporal, para asegurar que los datos sean precisos y estén actualizados.	<p>Escenario 01: Edición de detalles del espacio</p> <p>Dado que el arrendador desea actualizar la información de su espacio. Cuando accede a la sección de edición de su espacio y realiza modificaciones en campos como descripción, aforo, servicios disponibles y tarifas. Entonces puede guardar los cambios, y estos se reflejan en la página de detalles del espacio para todos los usuarios una vez finalizada la edición.</p> <p>Escenario 02: Modificación de visibilidad Temporal del espacio</p> <p>Dado que el arrendador está realizando modificaciones en su espacio. Cuando activa la opción de "No disponible temporalmente" para realizar cambios. Entonces el espacio se muestra como no disponible para reservas.</p>	EP02
US14	Actualizar perfil y sesión de usuario	Como usuario de AlquilaFácil, quiero poder modificar mi perfil, incluyendo la opción de cambiar mi nombre de usuario y cerrar sesión, para mantener actualizada mi información personal y gestionar mi acceso a la aplicación de manera conveniente.	<p>Escenario 01: Modificación del nombre de usuario</p> <p>Dado que el usuario ha iniciado sesión en su cuenta de AlquilaFácil. Cuando accede a la sección de configuración de perfil y edita el campo "Nombre de usuario". Entonces puede guardar los cambios, y el nuevo nombre de usuario se refleja en su perfil y en todas las interacciones futuras en la plataforma.</p> <p>Escenario 02: Cierre de sesión</p> <p>Dado que el usuario desea salir de su cuenta en AlquilaFácil. Cuando selecciona la opción "Cerrar sesión" en la configuración de perfil. Entonces es desconectado de su cuenta y redirigido a la pantalla de inicio de sesión, asegurando que su sesión se haya cerrado de manera segura.</p>	EP01
US15	Reportar espacios inseguros	Como usuario de AlquilaFácil, quiero poder reportar espacios inseguros, para contribuir a la seguridad de la comunidad y alertar sobre situaciones que puedan poner en riesgo a otros usuarios.	<p>Escenario 01: Selección del espacio inseguro</p> <p>Dado que el usuario ha iniciado sesión en su cuenta de AlquilaFácil. Cuando accede a la sección de "Reportar espacio" y selecciona el espacio desde la lista de reservas activas o buscando un espacio específico. Entonces puede ingresar un asunto y descripción del problema.</p> <p>Escenario 02: Confirmación del reporte</p> <p>Dado que el usuario ha completado el formulario de reporte con todos los detalles requeridos. Cuando envía el reporte mediante el botón "Enviar". Entonces el reporte será registrado y revisado por el equipo de soporte de AlquilaFácil.</p>	EP03

ID de la user story	Título	Descripción	Criterios de aceptación	Relacionado con (Epic ID)
US16	Visualizar espacios reportados por el usuario	Como usuario de AlquilaFácil, quiero poder visualizar los espacios que he reportado, para realizar un seguimiento de mis reportes y gestionarlos.	<p>Escenario 01: Navegar a la sección de "Ver Espacios Reportados"</p> <p>Dado que el usuario ha iniciado sesión en su cuenta de AlquilaFácil. Cuando accede a la sección de "Ver Espacios Reportados" desde el panel de control. Entonces puede ver una lista de los espacios que ha reportado, incluyendo información como el nombre del espacio, la fecha del reporte, y el motivo del reporte.</p>	EP03
US17	Acceder a información de ayuda	Como usuario de AlquilaFácil, quiero poder acceder fácilmente a una sección de preguntas frecuentes e información de contacto para resolver mis dudas o comunicarme con el soporte. Cuando lo necesite.	<p>Escenario 01: Consultar preguntas frecuentes</p> <p>Dado que un usuario tiene una duda común sobre el funcionamiento de la plataforma. Cuando selecciona la opción de "Preguntas frecuentes" en el menú de ayuda. Entonces se muestra una lista de preguntas con sus respuestas para ayudar al usuario.</p> <p>Escenario 02: Acceder a información de contacto</p> <p>Dado que un usuario necesita comunicarse con el soporte. Cuando selecciona la opción de "Información de contacto" en el menú de ayuda. Entonces se muestra la información de contacto, incluyendo el correo electrónico y número de teléfono del soporte.</p>	EP01
US18	Control de Aforo	Como arrendador, quiero recibir información en tiempo real sobre el número de personas presentes en mi local durante un evento, para asegurarme de que no se exceda el aforo permitido.	<p>Escenario 01: Alerta de aforo</p> <p>Dado que hay un evento activo, cuando se detecta que el número de personas se acerca al límite permitido, entonces se muestra una alerta en la app del arrendador.</p> <p>Escenario 02: Aforo excedido</p> <p>Dado que se ha superado el aforo máximo permitido, cuando esto ocurre, entonces el sistema debe notificar al arrendador inmediatamente por notificación push y correo electrónico.</p>	EP05
US19	Monitoreo de Niveles de Ruido	Como arrendador, quiero que el sistema me notifique si el nivel de ruido supera un umbral establecido, para poder tomar acciones preventivas.	<p>Escenario 01: Umbral de ruido excedido</p> <p>Dado que se ha configurado un umbral de ruido, cuando los niveles exceden ese límite, entonces el sistema envía una notificación al arrendador.</p> <p>Escenario 02: Actualización de umbral</p> <p>Dado que el evento sigue en curso, cuando los niveles de ruido vuelven a la normalidad, entonces el sistema registra automáticamente el tiempo total del exceso.</p>	
US20	Detección de Humo	Como arrendador, quiero que se active una alerta automática si se detecta humo en el local durante un evento, para verificar si hay una situación de riesgo.	<p>Escenario 01: Presencia de humo detectada</p> <p>Dado que hay sensores de humo activos, cuando se detecta presencia de humo, entonces se envía una notificación crítica al arrendador.</p> <p>Escenario 02: Visualización de incidente</p> <p>Dado que se ha enviado una alerta de humo, cuando el usuario accede a la app, entonces debe visualizar el evento en un panel de incidentes con la hora exacta de detección.</p>	EP05

ID de la user story	Título	Descripción	Criterios de aceptación	Relacionado con (Epic ID)
US21	Movimiento en Áreas No Autorizadas	Como arrendador, quiero recibir notificaciones si se detecta movimiento en zonas restringidas, para asegurar que los arrendatarios respeten los espacios delimitados.	<p>Escenario 01: Movimiento detectado en zona restringida Dado que se ha definido una zona como restringida, cuando se detecta movimiento en esa área, entonces el arrendador recibe una alerta inmediata con la ubicación del incidente.</p> <p>Escenario 02: Registro de movimientos detectados Dado que ocurre una intrusión, cuando el evento es registrado, entonces debe quedar un historial con fecha, hora y duración del movimiento.</p>	EP05
US22	Registro de Incidentes	Como arrendador, quiero acceder a un historial con todos los eventos registrados por sensores, para tener evidencia y mejorar la gestión.	<p>Escenario 01: Historial de incidentes Dado que han ocurrido incidentes durante un evento, cuando el arrendador accede al panel de historial, entonces puede ver un listado con fecha, tipo de incidente y duración.</p> <p>Escenario 02: Detalle de incidente registrado Dado que se selecciona un evento del historial, cuando el usuario lo abre, entonces debe poder ver detalles como gráfico de actividad, sensores involucrados y ubicación dentro del local.</p>	EP05
US23	Explorar landing informativa	Como visitante, quiero acceder a una landing clara y atractiva para entender rápidamente los beneficios de usar AlquilaFácil como arrendador o arrendatario.	<p>Escenario 01: Visualización de beneficios Dado que el visitante accede a la landing page, cuando hace scroll hasta la sección de beneficios, entonces ve tarjetas explicativas con iconos, título corto y descripción concisa.</p> <p>Escenario 02: Comprensión rápida de roles Dado que el visitante no conoce la plataforma, cuando observa las secciones específicas por segmento, entonces entiende qué ofrece la plataforma para arrendadores y arrendatarios.</p>	EP06
US24	Navegar por secciones clave desde el header	Como visitante, quiero usar el menú superior para desplazarme fácilmente a secciones como "Espacios", "Publicar", "Monitoreo" y "Beneficios", para encontrar rápidamente lo que me interesa sin tener que hacer scroll interminable.	<p>Escenario 01: Menú funcional Dado que el visitante está en la landing, cuando hace clic en un ítem del header (por ejemplo "Beneficios"), entonces la página se desplaza de forma suave hasta la sección correspondiente.</p> <p>Escenario 02: Responsive en móviles Dado que el visitante usa un dispositivo móvil, cuando abre el menú del header, entonces ve un menú desplegable con los mismos ítems y todos son funcionales.</p>	EP06
US25	Conocer los beneficios diferenciados	Como visitante, quiero ver tarjetas explicativas con los beneficios que ofrece AlquilaFácil tanto para arrendadores como arrendatarios, para valorar mejor si la plataforma se adapta a lo que busco.	<p>Escenario 01: Visualización de tarjetas Dado que el visitante llega a la sección de beneficios, cuando la sección carga, entonces se muestran tarjetas separadas para arrendadores y arrendatarios con título, ícono y breve descripción.</p> <p>Escenario 02: Diseño atractivo Dado que el visitante revisa los beneficios, cuando pasa el cursor o toca una tarjeta, entonces esta responde con un efecto visual como sombreado o cambio de color.</p>	EP06

3.3. Impact Mapping

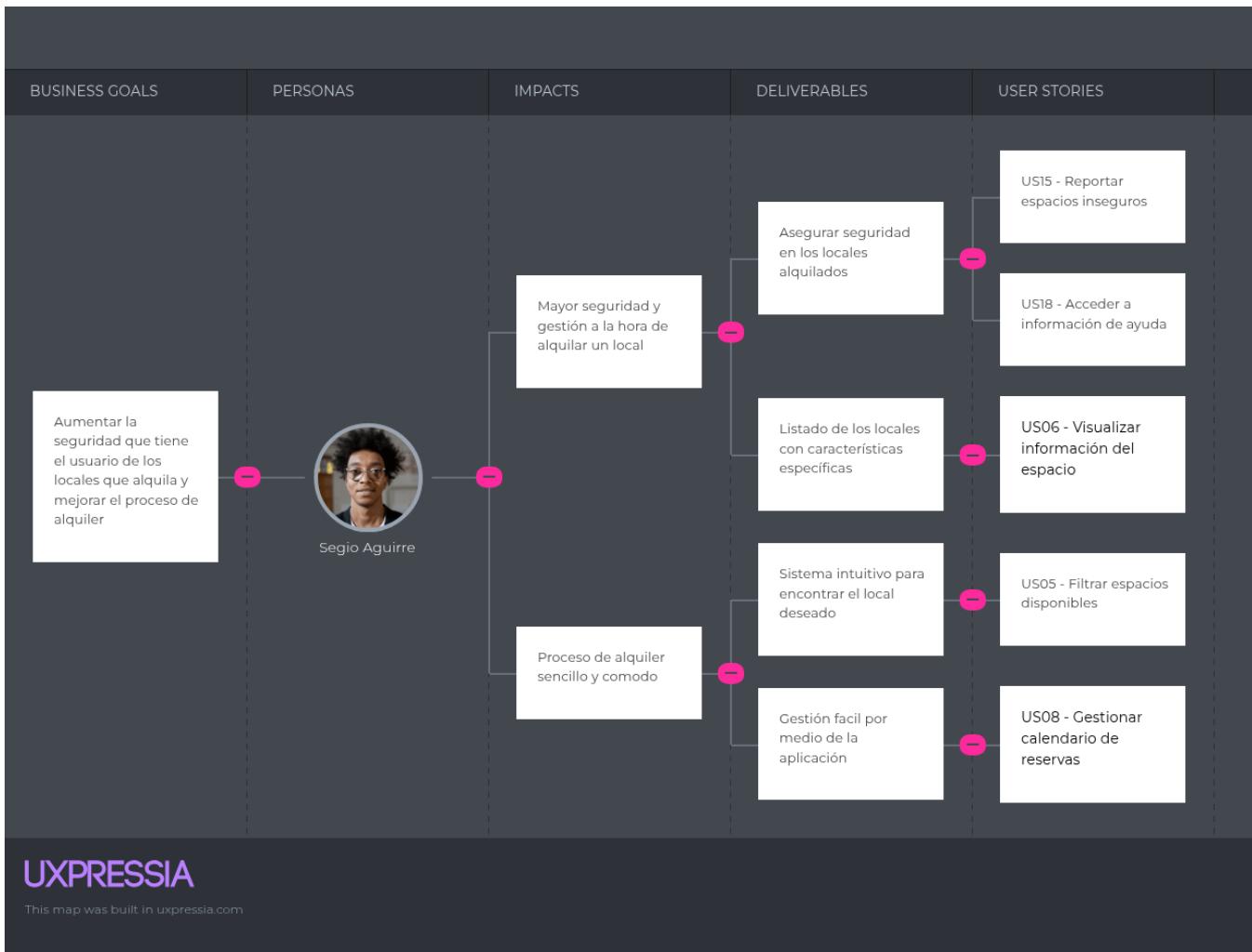
Segmento Arrendador



UXPRESSIA

This map was built in upressoia.com

Segmento Arrendatario



3.4. Product Backlog

User Stories ID	Título	Descripción	Story Points (1/2/3/5/8)
US03	Registrar espacios	Como arrendador de un espacio para eventos en AlquilaFácil, quiero poder registrar mi espacio para comenzar a recibir solicitudes de reserva lo antes posible.	8
US04	Buscar espacios disponibles	Como arrendatario, quiero poder buscar fácilmente espacios disponibles en AlquilaFácil para encontrar el lugar perfecto para mi evento.	5
US05	Filtrar espacios disponibles	Como arrendatario, quiero poder filtrar los espacios disponibles por capacidad y categoría, para encontrar uno que cumpla con mis criterios específicos.	5
US06	Visualizar información del espacio	Como arrendatario, quiero poder visualizar información detallada del espacio, para evaluar si cumple con los requisitos de mi evento antes de hacer una reserva.	5
US07	Reservar espacios	Como arrendatario, quiero poder reservar un espacio para mi evento en AlquilaFácil para garantizar su disponibilidad en la fecha deseada.	8
US12	Visualizar espacios propios publicados	Como arrendador en AlquilaFácil, quiero poder visualizar todos los espacios que he publicado en la plataforma, para gestionar fácilmente la información de mis propiedades.	3
US13	Modificar espacios publicados	Como arrendador de un espacio en AlquilaFácil, quiero poder modificar la información de los espacios que he publicado, incluyendo detalles como la descripción, aforo, servicios disponibles, tarifas y visibilidad temporal, para asegurar que los datos sean precisos y estén actualizados.	3
US08	Gestionar calendario de reservas	Como usuario de AlquilaFácil, quiero poder controlar una agenda de reservas para tener un horario organizado y evitar conflictos futuros.	5
US18	Control de aforo	Como arrendador, quiero recibir información en tiempo real sobre el número de personas presentes en mi local durante un evento, para asegurarme de que no se exceda el aforo permitido.	3

User Stories ID	Título	Descripción	Story Points (1/2/3/5/8)
US19	Monitoreo de niveles de ruido	Como arrendador, quiero que el sistema me notifique si el nivel de ruido supera un umbral establecido, para poder tomar acciones preventivas.	3
US20	Detección de humo	Como arrendador, quiero que se active una alerta automática si se detecta humo en el local durante un evento, para verificar si hay una situación de riesgo.	5
US21	Movimiento en áreas No autorizadas	Como arrendador, quiero recibir notificaciones si se detecta movimiento en zonas restringidas, para asegurar que los arrendatarios respeten los espacios delimitados.	5
US22	Registro de incidentes	Como arrendador, quiero acceder a un historial con todos los eventos registrados por sensores, para tener evidencia y mejorar la gestión.	5
US09	Calificar y comentar sobre espacios	Como arrendatario, quiero poder publicar mi reseña sobre un espacio que he reservado para que otros usuarios puedan conocer mi experiencia sobre este.	3
US11	Controlar espacios favoritos	Como arrendatario en AlquilaFácil, quiero poder agregar un espacio a mis favoritos para guardar y acceder rápidamente a los lugares de interés para futuras reservas.	3
US15	Reportar espacios inseguros	Como usuario de AlquilaFácil, quiero poder reportar espacios inseguros, para contribuir a la seguridad de la comunidad y alertar sobre situaciones que puedan poner en riesgo a otros usuarios.	3
US16	Visualizar espacios reportados	Como usuario de AlquilaFácil, quiero poder visualizar los espacios que he reportado, para realizar un seguimiento de mis reportes y gestionarlos.	3
US01	Registrar usuario	Como usuario de la aplicación móvil de AlquilaFácil de un espacio para eventos, quiero poder registrarme fácilmente en AlquilaFácil para ofrecer mi espacio en alquiler y llegar a más clientes potenciales.	3
US02	Iniciar sesión	Como usuario registrado de la aplicación móvil de AlquilaFácil, quiero poder iniciar sesión fácilmente en mi cuenta, para gestionar mi espacio en alquiler y acceder a mis mensajes y notificaciones.	3
US10	Notificar sobre actividades de arrendatarios	Como arrendador de un espacio para eventos en AlquilaFácil, quiero recibir notificaciones Cuando un arrendatario ha realizado cierta acción sobre mi espacio para estar al tanto de su estado y tomar las acciones pertinentes.	3
US14	Actualizar perfil y sesión de usuario	Como usuario de AlquilaFácil, quiero poder modificar mi perfil, incluyendo la opción de cambiar mi nombre de usuario y cerrar sesión, para mantener actualizada mi información personal y gestionar mi acceso a la aplicación de manera conveniente.	2
US17	Acceder a información de ayuda	Como usuario de AlquilaFácil, quiero poder acceder fácilmente a una sección de preguntas frecuentes e información de contacto para resolver mis dudas o comunicarme con el soporte Cuando lo necesite.	5
US23	Explorar landing informativa	Como visitante, quiero acceder a una landing clara y atractiva para entender rápidamente los beneficios de usar AlquilaFácil como arrendador o arrendatario.	5
US24	Navegar por secciones clave desde el header	Como visitante, quiero usar el menú superior para desplazarme fácilmente a secciones como "Espacios", "Publicar", "Monitoreo" y "Beneficios".	5
US25	Conocer los beneficios diferenciados	Como visitante, quiero ver tarjetas explicativas con los beneficios que ofrece AlquilaFácil tanto para arrendadores como arrendatarios.	5

Capítulo IV: Strategic-Level Software Design

4.1. Strategic-Level Attribute-Driven Design

4.1.1. Design Purpose

El propósito del proceso de diseño de la solución **Alquila Facil** es diseñar una arquitectura de software que resuelva la problemática de dispersión de la oferta y demanda de alquileres en múltiples canales desconectados, así como la complejidad de los procesos de búsqueda y gestión identificada previamente.

- **Arrendadores:** Requieren una plataforma que maximice la visibilidad de sus propiedades y simplifique la gestión de arrendatarios potenciales.
- **Arrendatarios:** Necesitan un punto centralizado para búsqueda eficiente de propiedades y comunicación directa con arrendadores.

El diseño busca generar valor para el negocio mediante la creación de un ecosistema digital que conecte eficientemente a ambos segmentos, reduciendo fricciones en el proceso de alquiler y posibilitando la monetización a través de transacciones exitosas facilitadas por la plataforma.

4.1.2 Attribute-Driven Design Inputs

El diseño arquitectónico de AlquilaFácil se basa en la metodología Attribute-Driven Design (ADD), que prioriza la definición sistemática de inputs arquitectónicos críticos. Esta sección presenta los elementos fundamentales que orientarán las decisiones de diseño: Primary Functionality (historias de usuario de mayor impacto), Quality Attribute Scenarios (escenarios cuantificables de rendimiento, seguridad y escalabilidad), y Constraints (restricciones técnicas y regulatorias no negociables). Esta aproximación asegura que la arquitectura resultante satisfaga los requisitos funcionales, los atributos de calidad esperados y las limitaciones del contexto operativo.

4.1.2.1. Primary Functionality (Primary User Stories)

En esta sección se presentan las historias de usuario que representan la **funcionalidad primaria** de AlquilaFácil. Estas historias son consideradas de mayor relevancia, ya que reflejan los **requisitos funcionales esenciales** del sistema y tienen un **impacto directo sobre la arquitectura de la solución**. La selección se centra en aquellas funcionalidades que permiten a los usuarios interactuar con la plataforma de manera clave: búsqueda y reserva de espacios, gestión de espacios por parte de arrendadores y monitoreo de eventos para garantizar la seguridad.

A continuación, se detallan las historias de usuario seleccionadas, con su descripción, criterios de aceptación y la Epic a la que están relacionadas:

ID de la user story	Título	Descripción	Criterios de aceptación	Relacionado con (Epic ID)
US04	Buscar espacios disponibles	Como arrendatario, quiero poder buscar fácilmente espacios disponibles en AlquilaFácil para encontrar el lugar perfecto para mi evento.	<p>Escenario 01: Búsqueda principal por ubicación Dado que un arrendatario busca un espacio para eventos en una ubicación específica. Cuando el arrendatario ingresa la ubicación deseada en el campo de búsqueda. Entonces se muestran los espacios disponibles en esa ubicación.</p> <p>Escenario 02: Búsqueda general de espacios Dado que un arrendatario no ingresa algún texto en la barra de búsqueda. Cuando presiona sobre el ícono para buscar espacios. Entonces se muestran todos los espacios disponibles.</p>	EP03
US05	Filtrar espacios disponibles	Como arrendatario, quiero poder filtrar los espacios disponibles por capacidad y categoría, para encontrar uno que cumpla con mis criterios específicos.	<p>Escenario 01: Filtrado por capacidad Dado que un arrendatario desea un espacio con capacidad para un número específico de personas. Cuando el arrendatario aplica un filtro de capacidad en la búsqueda. Entonces se muestran solo los espacios que cumplen con ese criterio.</p> <p>Escenario 02: Filtrado por categoría Dado que un arrendatario desea un espacio de una categoría en específico. Cuando el arrendatario aplica un filtro de categoría en la búsqueda. Entonces se muestran solo los espacios que cumplen con ese criterio.</p>	EP03
US07	Reservar espacios	Como arrendatario, quiero poder reservar un espacio para mi evento en AlquilaFácil para garantizar su disponibilidad en la fecha deseada.	<p>Escenario 01: Proceso de Reserva Dado que un arrendatario ha encontrado el espacio ideal en AlquilaFácil. Cuando selecciona un espacio y una fecha. Entonces se muestra un formulario para completar los detalles de reserva.</p> <p>Escenario 02: Pago de la reserva con PayPal Dado que el usuario está a punto de culminar el proceso de reserva de un espacio. Cuando el usuario presione el botón de realizar reserva. Entonces el usuario realizará el pago de la reserva a través de PayPal.</p> <p>Escenario 03: Confirmación de Reserva Dado que un arrendatario ha realizado el pago de la reserva en AlquilaFácil. Cuando el arrendatario es redirigido a la aplicación. Entonces recibe una confirmación de reserva y los detalles se actualizan en su cuenta.</p>	EP03

ID de la user story	Título	Descripción	Criterios de aceptación	Relacionado con (Epic ID)
US08	Gestionar calendario de reservas	Como usuario de AlquilaFácil, quiero poder controlar una agenda de reservas para tener un horario organizado y evitar conflictos futuros.	Escenario 01: Existencia de reserva de usuario normal Dado que un arrendatario ha realizado una reserva de uno de mis espacios. Cuando el propietario accede al calendario. Entonces puede ver el día de la reserva resaltada en color rojo.	
US18	Control de Aforo	Como arrendador, quiero recibir información en tiempo real sobre el número de personas presentes en mi local durante un evento, para asegurarme de que no se exceda el aforo permitido.	Escenario 02: Existencia de reserva de usuario premium Dado que un arrendatario con suscripción premium ha realizado una reserva de uno de mis espacios. Cuando el propietario accede al calendario. Entonces puede ver el día de la reserva resaltada en color amarillo.	EP04
US19	Monitoreo de Niveles de Ruido	Como arrendador, quiero que el sistema me notifique si el nivel de ruido supera un umbral establecido, para poder tomar acciones preventivas.	Escenario 03: Existencia de reserva de espacio ajeno Dado que un arrendatario ha realizado una reserva de espacio. Cuando este accede al calendario. Entonces puede ver el día de su reserva resaltada en color azul.	
US20	Detección de Humo	Como arrendador, quiero que se active una alerta automática si se detecta humo en el local durante un evento, para verificar si hay una situación de riesgo.	Escenario 01: Alerta de aforo Dado que hay un evento activo, cuando se detecta que el número de personas se acerca al límite permitido, entonces se muestra una alerta en la app del arrendador. Escenario 02: Aforo excedido Dado que se ha superado el aforo máximo permitido, cuando esto ocurre, entonces el sistema debe notificar al arrendador inmediatamente por notificación push y correo electrónico.	EP05
US21	Movimiento en Áreas No Autorizadas	Como arrendador, quiero recibir notificaciones si se detecta movimiento en zonas restringidas, para asegurar que los arrendatarios respeten los espacios delimitados.	Escenario 01: Umbral de ruido excedido Dado que se ha configurado un umbral de ruido, cuando los decibeles exceden ese límite, entonces el sistema envía una notificación al arrendador. Escenario 02: Actualización de umbral Dado que el evento sigue en curso, cuando los niveles de ruido vuelven a la normalidad, entonces el sistema registra automáticamente el tiempo total del exceso.	EP05
US22	Registro de Incidentes	Como arrendador, quiero acceder a un historial con todos los eventos registrados por sensores, para tener evidencia y mejorar la gestión.	Escenario 01: Presencia de humo detectada Dado que hay sensores de humo activos, cuando se detecta presencia de humo, entonces se envía una notificación crítica al arrendador. Escenario 02: Visualización de incidente Dado que se ha enviado una alerta de humo, cuando el usuario accede a la app, entonces debe visualizar el evento en un panel de incidentes con la hora exacta de detección.	EP05
			Escenario 01: Movimiento detectado en zona restringida Dado que se ha definido una zona como restringida, cuando se detecta movimiento en esa área, entonces el arrendador recibe una alerta inmediata con la ubicación del incidente. Escenario 02: Registro de movimientos detectados Dado que ocurre una intrusión, cuando el evento es registrado, entonces debe quedar un historial con fecha, hora y duración del movimiento.	
			Escenario 01: Historial de incidentes Dado que han ocurrido incidentes durante un evento, cuando el arrendador accede al panel de historial, entonces puede ver un listado con fecha, tipo de incidente y duración. Escenario 02: Detalle de incidente registrado Dado que se selecciona un evento del historial, cuando el usuario lo abre, entonces debe poder ver detalles como gráfico de actividad, sensores involucrados y ubicación dentro del local.	EP05

4.1.2.2. Quality attribute Scenarios

En esta sección se incluye la especificación de la primera versión de los escenarios de atributos de calidad que tienen mayor impacto en la arquitectura de la solución, los cuales sirven de input para el proceso de diseño. La sección inicia con un texto de introducción en el cual se resume los escenarios de atributos de calidad identificados en primera instancia y se incluye el siguiente cuadro para especificar los Quality Attribute Scenarios.

Los escenarios de atributos de calidad identificados para **Alquila Fácil** se centran en garantizar una experiencia de usuario óptima, seguridad de datos, disponibilidad del sistema y escalabilidad para soportar el crecimiento esperado de la plataforma.

Atributo	Fuente	Estímulo	Artefacto	Entorno	Respuesta	Medida
Performance	Usuario (arrendador/arrendatario)	Búsqueda de propiedades con filtros múltiples	Sistema de búsqueda	Operación normal con 1000 usuarios concurrentes	El sistema retorna resultados de búsqueda	Tiempo de respuesta ≤ 2 segundos
Availability	Usuario	Acceso a la plataforma	Sistema completo	Operación normal	El sistema permanece disponible y operativo	Disponibilidad ≥ 99.5% del tiempo
Security	Usuario malicioso	Intento de acceso no autorizado a datos personales	Módulo de autenticación	Operación normal	El sistema bloquea el acceso y registra el intento	100% de intentos no autorizados bloqueados
Scalability	Carga del sistema	Incremento súbito de 500 usuarios simultáneos	Sistema completo	Pico de demanda	El sistema mantiene el rendimiento sin degradación	Tiempo de respuesta se mantiene ≤ 3 segundos
Usability	Arrendatario nuevo	Registro y primera búsqueda en la plataforma	Interfaz de usuario	Operación normal	El usuario completa el proceso sin asistencia	≥ 90% de usuarios completan el proceso en ≤ 5 minutos

4.1.2.3. Constraints

En esta sección se incluye la especificación de restricciones, es decir características que no pueden ser negociadas y son impuestas por el cliente o el propio negocio como guía para la elaboración de la solución. La sección inicia con una introducción donde se resume los principales constraints a considerar. Luego de ello se incluye el cuadro de Constraints representados como Technical Stories, donde cada constraint considerado ocupa una fila.

Los constraints identificados para **Alquila Fácil** incluyen restricciones técnicas relacionadas con la seguridad de datos personales, compliance regulatorio, limitaciones presupuestarias y requerimientos de compatibilidad que son fundamentales para el éxito del proyecto.

Technical Story ID	Título	Descripción	Criterios de Aceptación	Relacionado con (Epic ID)
C-001	Cumplimiento de Ley de Protección de Datos	El sistema debe cumplir con la normativa peruana de protección de datos personales	- Implementar consentimiento explícito para recolección de datos - Permitir eliminación de datos bajo solicitud - Cifrar datos personales sensibles	Epic EP01
C-002	Compatibilidad con Navegadores Web	La aplicación web debe ser compatible con navegadores principales	- Funcionar correctamente en Chrome, Firefox, Safari y Edge - Versiones de los últimos 2 años - Responsive design para dispositivos móviles	Epic EP02
C-003	Presupuesto de Infraestructura Cloud	Los costos de infraestructura no deben exceder el presupuesto inicial	- Utilizar servicios cloud con modelo de pricing escalable - Implementar auto-scaling para optimizar costos - Monitorear gastos mensuales	Epic EP03
C-004	Tiempo de Desarrollo Limitado	El MVP debe estar listo en 6 meses	- Priorizar funcionalidades core - Utilizar frameworks y librerías existentes - Implementar metodología ágil	Epic EP03

4.1.3. Architectural Drivers Backlog

La arquitectura de la solución AlquilaFácil está guiada por un conjunto de drivers que garantizan el cumplimiento de los objetivos funcionales del sistema, los atributos de calidad priorizados y las restricciones tecnológicas definidas. Estos drivers representan las decisiones críticas que orientan el diseño de los distintos componentes, desde la autenticación segura y la gestión de espacios, hasta la integración de sensores IoT y la escalabilidad en la nube, asegurando que la plataforma sea confiable, segura y consistente para todos los usuarios.

Driver ID	Título de Driver	Descripción	Importancia para Stakeholders	Impacto en Architecture Technical Complexity
FD-01	Gestión de Usuarios y Autenticación Segura	Los usuarios deben registrarse e iniciar sesión de forma segura (JWT, hashing). Base para todas las demás interacciones.	High	High
FD-02	Gestión de Espacios (Locals)	Permitir a arrendadores publicar y gestionar espacios, y a arrendatarios reservarlos.	High	Medium
FD-03	Integración de Sensores IoT	Captura de datos de humo, sonido, aforo y movimiento en tiempo real mediante Edge Node y Embedded App.	High	High
FD-04	Notificaciones en Tiempo Real	Enviar alertas (seguridad, reservas, estado de locales) a usuarios vía Web y Mobile.	High	High
QAD-01	Seguridad de Datos	Proteger datos sensibles de usuarios y locales mediante cifrado, autenticación robusta y control de acceso por roles.	High	High
QAD-02	Escalabilidad en la Nube	El sistema debe soportar un incremento de usuarios, locales y sensores sin degradar el rendimiento, aprovechando Azure Cloud.	High	High
QAD-03	Disponibilidad y Resiliencia	El sistema debe garantizar alta disponibilidad y recuperación rápida ante fallos (despliegue distribuido, redundancia).	High	High
QAD-04	Experiencia de Usuario Consistente	El diseño UX/UI debe ser consistente entre Landing, Web App y Mobile App, con flujos simples e inclusivos.	High	Medium
QAD-05	Interoperabilidad con Servicios Externos	El sistema debe integrarse con APIs externas (ej. pasarela de pagos, mapas de localización).	Medium	High
C-01	Restricción de Uso de Tecnologías	Todas las capas deben implementar soluciones basadas en tecnologías establecidas (.NET, Vue y Flutter).	High	Medium
C-02	Restricción de Despliegue en Azure	Todos los componentes (Backend, Edge Node, Web) deben ser desplegados en Azure según lineamientos del curso.	High	Medium
C-03	Restricción de Evidencias de Desarrollo	Deben generarse evidencias (commits, ramas GitFlow, videos, despliegues) en cada sprint para evaluación académica.	High	Low

4.1.4. Architectural Design Decisions

Introducción

Durante el **Quality Attribute Workshop** se evaluaron los drivers arquitectónicos definidos previamente, identificando patrones candidatos y analizando ventajas y desventajas para cada caso. El objetivo fue asegurar que las decisiones arquitectónicas estuvieran alineadas con los objetivos de negocio, los atributos de calidad priorizados y las restricciones del proyecto, garantizando al mismo tiempo la viabilidad técnica de la solución **AlquilaFácil**.

El análisis se organizó en torno a los principales *drivers*, considerando tres patrones candidatos por cada uno, evaluando sus beneficios y limitaciones, y seleccionando la alternativa más adecuada para el contexto del proyecto. El resultado se presenta en la siguiente **Candidate Pattern Evaluation Matrix**, seguida de una síntesis de las decisiones finales adoptadas.

Candidate Pattern Evaluation Matrix

D1 — Gestión de Usuarios y Autenticación (FD-01)

Driver ID	Título de Driver	Pattern 1: JWT Stateless (Spring Security)	Pattern 2: OAuth2/OIDC con IdP	Pattern 3: Session-Based Auth
FD-01	Autenticación segura	Pro: Simplicidad, escalabilidad horizontal, soporte Web/Mobile. Con: Gestión de refresh tokens es propia.	Pro: Estándar enterprise, SSO, scopes. Con: Alta complejidad y sobrecarga para un MVP.	Pro: Fácil de implementar. Con: No escala bien, requiere sticky sessions.

Decisión: Se adopta **JWT stateless** con refresh tokens y hashing de contraseñas con BCrypt/Argon2, asegurando simplicidad y escalabilidad para el MVP.

D2 — Gestión de Espacios y Reservas (FD-02)

Driver ID	Título de Driver	Pattern 1: Monolito modular DDD	Pattern 2: Microservicios	Pattern 3: Microkernel
FD-02	Locals & Bookings	Pro: Organización clara, bajo costo operativo, fácil evolución. Con: Requiere disciplina modular.	Pro: Aislamiento fuerte. Con: Sobrecoste en despliegue y monitoreo.	Pro: Extensibilidad. Con: Complejidad innecesaria para MVP.

Decisión: Se mantiene un **monolito modular con bounded contexts** (IAM, Locals, Sensors, Notifications), listo para futura evolución hacia microservicios.

D3 — Integración IoT en Tiempo Real (FD-03)

Driver ID	Título de Driver	Pattern 1: MQTT + Edge Processing	Pattern 2: HTTP Polling	Pattern 3: AMQP directo
FD-03	Sensores IoT	Pro: Eficiente, bajo consumo, QoS configurable. Con: Requiere broker.	Pro: Simplicidad. Con: Ineficiente, mayor latencia.	Pro: Fiable en enterprise. Con: Pesado para microcontroladores.

Decisión: Se adopta **MQTT con Edge Processing**, asegurando eficiencia y filtrado de datos en tiempo real.

D4 — Notificaciones en Tiempo Real (FD-04)

Driver ID	Título de Driver	Pattern 1: WebSocket	Pattern 2: SSE + FCM	Pattern 3: Polling
FD-04	Notificaciones	Pro: Bidireccional y baja latencia. Con: Mayor complejidad de conexión.	Pro: Simplicidad (SSE) + fiabilidad en Mobile (FCM). Con: SSE es unidireccional.	Pro: Fácil. Con: Ineficiente, mala UX.

Decisión: Se implementa **SSE en Web y FCM en Mobile**, respaldados por un bus interno Pub/Sub para distribución de eventos.

D5 — Seguridad Transversal (QAD-01)

Driver ID	Título de Driver	Pattern 1: API Gateway	Pattern 2: Zero-Trust con JWT	Pattern 3: Vault + TLS
QAD-01	Seguridad	Pro: Rate-limiting y control centralizado. Con: Capa extra a mantener.	Pro: Control granular de accesos. Con: Requiere gestión de claves.	Pro: Cifrado integral. Con: Administración de Vault.

Decisión: Se adopta un **API Gateway** con limitación de peticiones, junto con **JWT con roles** y **TLS end-to-end**.

D6 — Escalabilidad (QAD-02)

Driver ID	Título de Driver	Pattern 1: CQRS light	Pattern 2: Cache distribuido	Pattern 3: Sharding
QAD-02	Escalabilidad	Pro: Consultas optimizadas, separa lecturas/escrituras. Con: Mayor complejidad en sincronización.	Pro: Mejora de rendimiento. Con: Riesgo en invalidación.	Pro: Escala masiva. Con: Sobrecoste para MVP.

Decisión: Se combina **CQRS light** con **Redis cache**, permitiendo escalabilidad horizontal en Azure.

D7 — Disponibilidad y Resiliencia (QAD-03)

Driver ID	Título de Driver	Pattern 1: Resilience4j	Pattern 2: Health-checks	Pattern 3: Outbox Pattern
QAD-03	Alta disponibilidad	Pro: Evita cascadas de fallos. Con: Configuración detallada.	Pro: Recuperación automática. Con: Limitado frente a fallos lógicos.	Pro: Garantiza entrega de eventos. Con: Complejidad extra.

Decisión: Se adopta **Resilience4j**, junto con **health-checks en Azure** y **Outbox Pattern** para asegurar consistencia eventual.

D8 — Experiencia de Usuario Consistente (QAD-04)

Driver ID	Título de Driver	Pattern 1: Design System	Pattern 2: BFF (Backend for Frontends)	Pattern 3: GraphQL
QAD-04	Consistencia UX	Pro: Uniformidad visual y coherencia. Con: Requiere mantenimiento.	Pro: APIs adaptadas a cada cliente. Con: Nuevos servicios por canal.	Pro: Flexibilidad de consultas. Con: Complejidad añadida.

Decisión: Se implementa un **Design System común** y un **BFF por canal (Web/Mobile)** para asegurar consistencia.

D9 — Interoperabilidad con Servicios Externos (QAD-05)

Driver ID	Título de Driver	Pattern 1: ACL/Adapter	Pattern 2: Integración directa	Pattern 3: ESB
-----------	------------------	------------------------	--------------------------------	----------------

Driver ID	Título de Driver	Pattern 1: ACL/Adapter	Pattern 2: Integración directa	Pattern 3: ESB							
QAD-05	Integraciones	Pro: Aísla dependencias externas. Con: Sobrecoste de implementación.	Pro: Simplicidad. Con: Alto acoplamiento.	Pro: Orquestación potente. Con: Excesivo para MVP.							
Decisión: Se utiliza ACL/Adapter para cada integración externa (imágenes), minimizando riesgos de acoplamiento.											
Síntesis de Decisiones											
<ul style="list-style-type: none"> Backend: Monolito modular con DDD, CQRS light y Redis. IoT: MQTT con Edge Processing para sensores. Tiempo real: SSE en Web + FCM en Mobile, con Pub/Sub interno. Seguridad: API Gateway, JWT con roles, TLS end-to-end y secretos externos. UX: Design System común y BFF por canal. Resiliencia: Resilience4j + health-checks + Outbox Pattern. Integraciones: ACL/Adapters para proveedores externos. Despliegue: Contenerización y escalado horizontal en Azure. 											
Estas decisiones permiten balancear las necesidades de negocio con la factibilidad técnica, asegurando que la solución sea segura, escalable, resiliente y fácil de evolucionar en el futuro.											
4.1.5. Quality Attribute Scenario Refinements											
Scenario Refinement 1 - Performance											
Scenario Refinement for Performance											
Scenario(s): Búsqueda eficiente de propiedades con filtros múltiples bajo carga concurrente											
Business Goals: Garantizar una experiencia de usuario fluida que mantenga a los usuarios comprometidos con la plataforma y reduzca la tasa de abandono durante las búsquedas											
Relevant Quality Attributes: Performance, Usability											
Scenario Components	Stimulus:	Arrendatario/arrendador ejecuta búsqueda con múltiples filtros (ubicación, precio, tipo de propiedad, características)									
	Stimulus Source:	Usuario final (arrendador/arrendatario)									
	Environment:	Sistema en operación normal con hasta 1000 usuarios concurrentes									
	Artifact (if known):	Sistema de búsqueda, base de datos, índices de búsqueda, cache distribuido									
	Response:	El sistema procesa la consulta, aplica filtros, consulta índices optimizados y retorna resultados paginados									
	Response Measure:	Tiempo de respuesta ≤ 2 segundos para el 95% de las consultas									
Questions:	¿Cómo se comporta el sistema cuando los filtros requieren joins complejos? ¿El cache distribuido maneja correctamente la invalidación de resultados?										
Issues:	Posible degradación con consultas geoespaciales complejas, necesidad de estrategias de cache inteligentes										
Scenario Refinement 2 - Availability											
Scenario Refinement for Availability											
Scenario(s): Acceso ininterrumpido a la plataforma durante picos de demanda y fallos parciales											
Business Goals: Mantener la confianza del usuario y evitar pérdida de ingresos por indisponibilidad del servicio											
Relevant Quality Attributes: Availability, Reliability, Recoverability											
Scenario Components	Stimulus:	Fallo de componente crítico o pico inesperado de tráfico									
	Stimulus Source:	Fallo de infraestructura, sobrecarga del sistema, mantenimiento programado									
	Environment:	Operación normal y condiciones de estrés									
	Artifact (if known):	Sistema completo, load balancers, instancias redundantes, base de datos									
	Response:	El sistema redirige tráfico a instancias saludables, activa réplicas y mantiene operatividad									
	Response Measure:	Disponibilidad ≥ 99.5% mensual, tiempo de recuperación ≤ 5 minutos									
Questions:	¿Cómo se maneja la sincronización de datos entre réplicas? ¿El sistema detecta automáticamente fallos parciales?										

Issues:	Complejidad en la gestión de estado distribuido, costos de infraestructura redundante
----------------	---

Scenario Refinement 3 - Security

Scenario Refinement for Security	
Scenario(s):	Protección contra accesos no autorizados y ataques maliciosos a datos sensibles
Business Goals:	Cumplir con regulaciones de protección de datos y mantener la confianza del usuario en la seguridad de sus datos personales
Relevant Quality Attributes:	Security, Privacy, Integrity
Scenario Components	<p>Stimulus: Intento de acceso no autorizado, ataque de fuerza bruta, inyección SQL</p> <p>Stimulus Source: Usuario malicioso, bot automatizado, atacante interno</p> <p>Environment: Operación normal, bajo ataque activo</p> <p>Artifact (if known): Módulo de autenticación, API Gateway, base de datos, logs de seguridad</p> <p>Response: Sistema bloquea acceso, registra intento, activa medidas defensivas (rate limiting, CAPTCHA)</p> <p>Response Measure: 100% de intentos no autorizados bloqueados, tiempo de detección ≤ 30 segundos</p>
Questions:	¿El sistema diferencia entre ataques automatizados y errores legítimos de usuario? ¿Cómo se balancea seguridad con usabilidad?
Issues:	Falsos positivos que bloquen usuarios legítimos, complejidad en la gestión de tokens JWT

Scenario Refinement 4 - Scalability

Scenario Refinement for Scalability	
Scenario(s):	Crecimiento súbito de usuarios simultáneos sin degradación del rendimiento
Business Goals:	Soportar el crecimiento del negocio sin requerir rediseño arquitectónico mayor, optimizando costos de infraestructura
Relevant Quality Attributes:	Scalability, Performance, Cost-effectiveness
Scenario Components	<p>Stimulus: Incremento de 500 usuarios simultáneos adicionales en corto período</p> <p>Stimulus Source: Crecimiento orgánico, campaña de marketing, evento viral</p> <p>Environment: Pico de demanda, operación bajo estrés</p> <p>Artifact (if known): Sistema completo, auto-scaling groups, load balancers, base de datos</p> <p>Response: Sistema activa auto-scaling, distribuye carga, mantiene rendimiento estable</p> <p>Response Measure: Tiempo de respuesta se mantiene ≤ 3 segundos, escalado automático en ≤ 2 minutos</p>
Questions:	¿El auto-scaling considera métricas de negocio además de técnicas? ¿Cómo se manejan las sesiones durante el escalado?
Issues:	Latencia en el aprovisionamiento de recursos, costos variables impredecibles

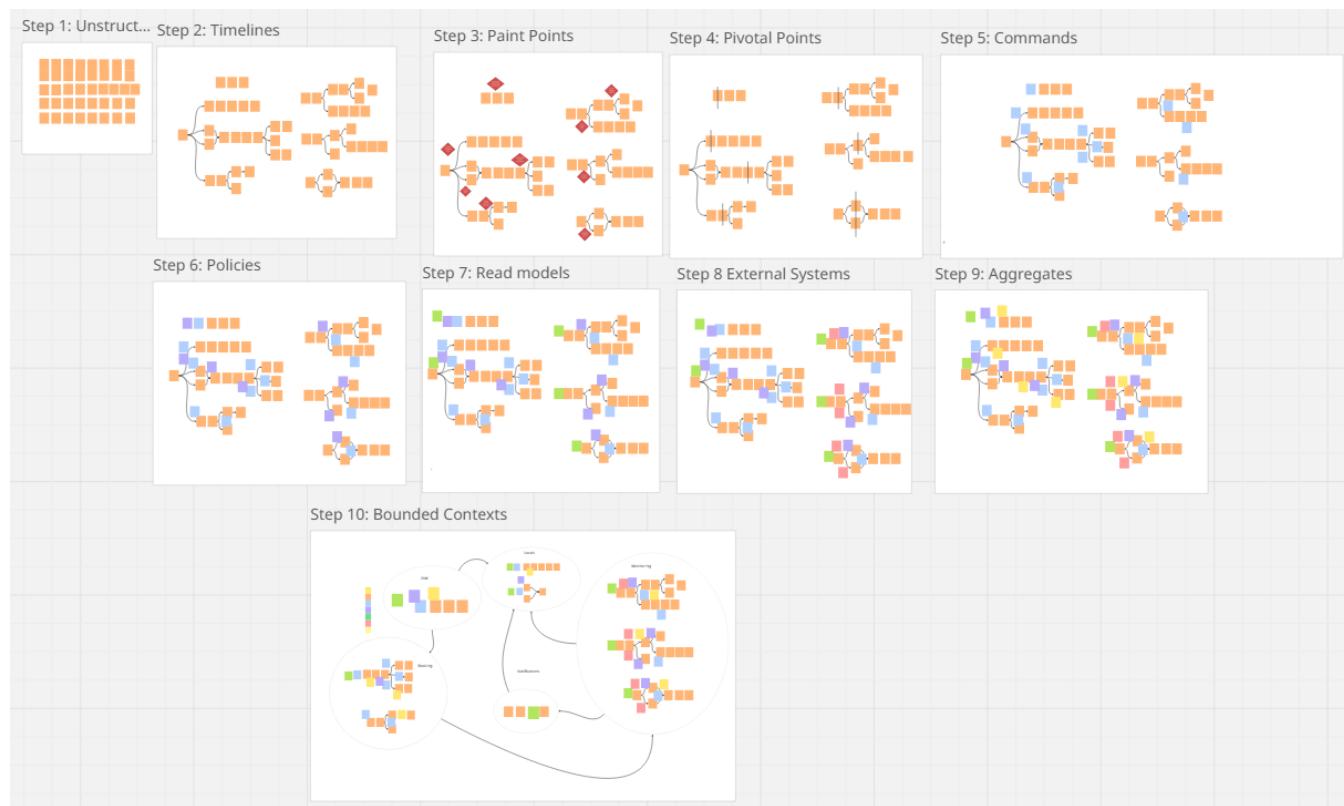
Scenario Refinement 5 - Usability

Scenario Refinement for Usability	
Scenario(s):	Onboarding intuitivo para nuevos usuarios sin experiencia previa en la plataforma
Business Goals:	Maximizar conversión de visitantes a usuarios registrados y reducir abandono durante el proceso de registro inicial
Relevant Quality Attributes:	Usability, Accessibility, Learnability
Scenario Components	<p>Stimulus: Nuevo arrendatario accede por primera vez y realiza registro completo + primera búsqueda</p> <p>Stimulus Source: Arrendatario nuevo sin experiencia previa en la plataforma</p> <p>Environment: Operación normal, interfaz web/móvil</p> <p>Artifact (if known): Interfaz de usuario, formularios de registro, sistema de onboarding, tutoriales</p> <p>Response: Sistema guía al usuario a través del proceso con indicadores claros y validación en tiempo real</p> <p>Response Measure: ≥ 90% de usuarios completan registro y primera búsqueda en ≤ 5 minutos sin asistencia</p>
Questions:	¿El sistema adapta el onboarding según el dispositivo utilizado? ¿Se consideran usuarios con diferentes niveles de alfabetización digital?
Issues:	Balance entre simplicidad y recolección de información necesaria, accesibilidad para usuarios con discapacidades

4.2. Strategic-Level Domain-Driven Design

4.2.1. EventStorming

EventStorming es una técnica de modelado colaborativo e iterativo que permite analizar a fondo problemas complejos y de gran escala, ayudando a descubrir una amplia variedad de detalles y desafíos involucrados.



Enlace del Miroo para verlo completo: https://miro.com/app/board/uXjVIBJ9674=/?share_link_id=539583810560

4.2.2. Candidate Context Discovery

Step 1: Unstructured Exploration:

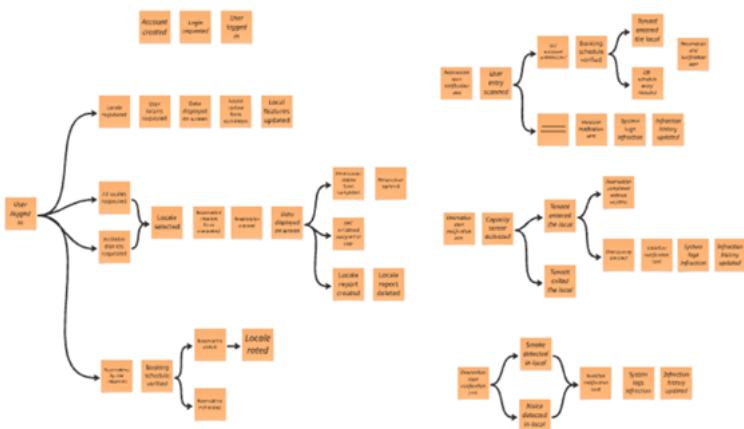
En la etapa inicial del EventStorming, se lleva a cabo una sesión de lluvia de ideas destinada a descubrir los eventos clave del dominio vinculados al negocio analizado. Es importante redactar estos eventos utilizando verbos en pasado, ya que deben reflejar acciones que ya han sucedido dentro del sistema o proceso.

Account created	User logged in	Locale update form completed	Reservation ended	Reservation not ended	Reservation update form completed	Reservation creation form completed	Locale selected
Login requested	Locale registered	Local features updated	Reservation updated	Reservation created	Reservations by user requested	User entry scanned	Locale report created
NFC wristband authenticated	Off-schedule entry detected	Tenant entered the local	Tenant exited the local	Capacity sensor activated	Oversaturation detected	Smoke detected in local	Noise detected in local
Data displayed on screen	System logs infraction	Infraction history updated	Reservation start notification sent	Reservation end notification sent	Reservation completed without incident	Locale rated	Violation notification sent
All locales requested	Locale report deleted	Locale report deleted	Available districts requested	User locales requested	Ownership verified	NFC wristband assigned to user	Wristband unlinked from user

Realizamos una sesión de lluvia de ideas para identificar todos los eventos relevantes que el sistema de reservas de locales con tecnología IoT debe manejar. En esta etapa inicial del Event Storming se mapearon eventos clave como la creación de cuentas, registro y actualización de locales, postulaciones y confirmación de reservas, así como la entrada y salida del arrendatario mediante pulsera NFC. También se incluyeron eventos generados por sensores, como detección de humo, sobreaforo y accesos no autorizados, fundamentales para garantizar el cumplimiento de las normas del local. Se consideraron alertas automáticas enviadas tanto al arrendador como al arrendatario en caso de infracciones, así como acciones relacionadas con la gestión de reportes, historial de asistencia e interacciones con la app móvil. Este mapeo inicial permite tener una visión integral de todos los comportamientos que el sistema debe registrar y reaccionar para brindar una solución de monitoreo y control eficaz para ambas partes involucradas.

Step 2: Timelines:

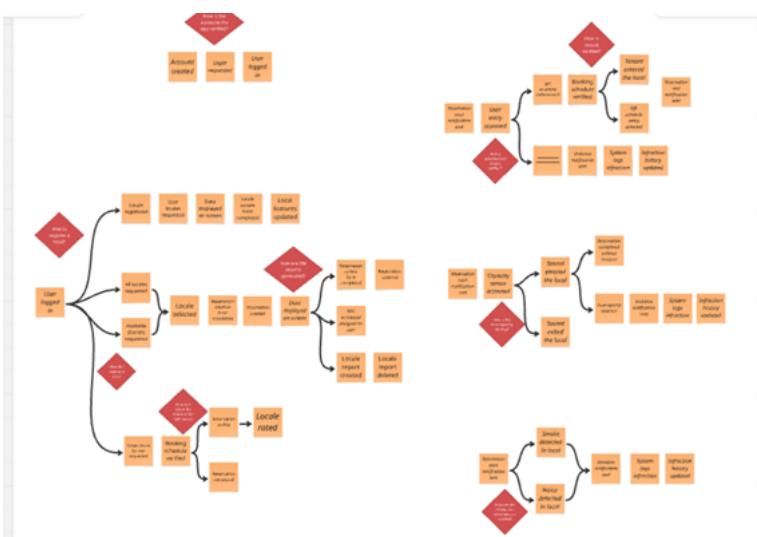
En esta segunda etapa, se toman los eventos de dominio identificados y se ordenan cronológicamente según su secuencia natural dentro del proceso. El primer enfoque consiste en construir un "camino feliz" (happy path), es decir, una secuencia ideal en la que todas las acciones se ejecutan correctamente y el flujo del sistema se desarrolla sin inconvenientes. Una vez definido este escenario óptimo, se incorporan variaciones o situaciones alternativas que representen posibles errores, fallos o condiciones excepcionales dentro del flujo operativo.



En esta segunda fase del Event Storming, organizamos los eventos de dominio identificados previamente en flujos secuenciales que reflejan cómo se comporta el sistema ante diferentes escenarios relacionados con el proceso de reserva, ingreso y monitoreo del uso del local. El flujo principal sigue un "camino feliz" donde el usuario inicia sesión, selecciona un local, crea una reserva, asiste con su pulsera NFC, y completa la experiencia sin incidentes, incluyendo la generación de reportes y la calificación del local. Además, se contemplan trayectorias alternativas que incluyen detecciones de humo, ruido o exceso de aforo, así como ingresos fuera de horario o con pulseras no autorizadas. Estas situaciones activan mecanismos de control como el registro de infracciones, el envío de notificaciones al arrendador o arrendatario, y la actualización del historial de eventos. Esta organización permite visualizar cómo el sistema reacciona ante distintos contextos operativos, sentando las bases para diseñar una lógica robusta, trazable y automatizada tanto en la app móvil como en la plataforma web.

Step 3: Paint Points:

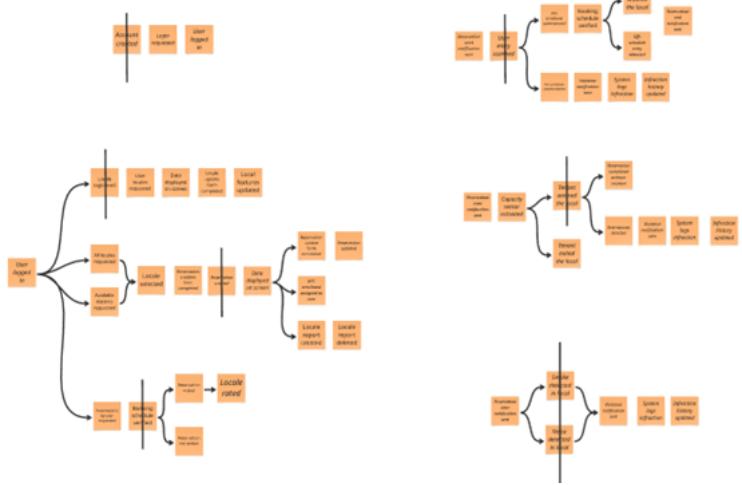
Una vez que los eventos fueron organizados en una secuencia temporal, utilizamos esta visión global del proceso para detectar áreas clave que requieren atención. Estos puntos críticos pueden manifestarse como cuellos de botella operativos, tareas manuales susceptibles de automatización, ausencia de documentación relevante o vacíos en el entendimiento del dominio. Esta revisión permite anticipar mejoras y definir prioridades para el diseño del sistema.



En esta fase del Event Storming, se identificaron los puntos críticos del sistema y se documentaron preguntas clave que deben ser respondidas para garantizar el correcto funcionamiento del proceso de reservas, ingreso y monitoreo de locales. Estas preguntas se integran directamente en los flujos de eventos y permiten explorar escenarios como la verificación de acceso mediante pulseras NFC, la validación del cronograma de reservas o la forma en que se detectan eventos de sobreaforo, humo o ruido durante el uso del local.

Step 4: Pivotal Points

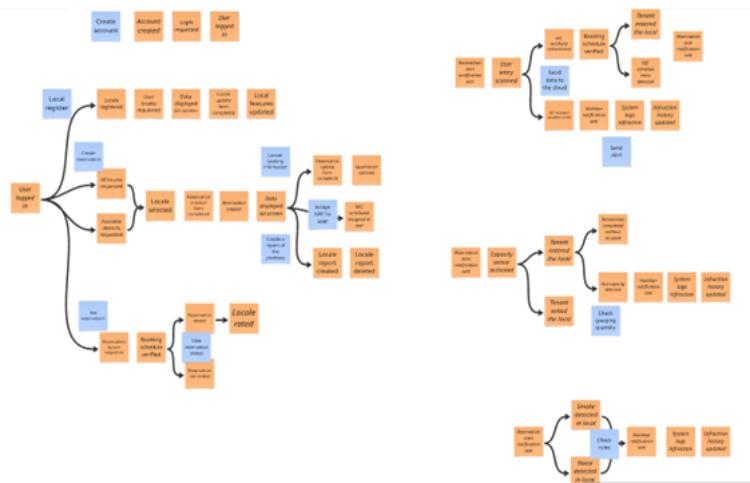
Una vez que se ha construido la línea de eventos completa, incluyendo los puntos de dolor, se procede a identificar aquellos eventos relevantes que representan un cambio significativo en el contexto o en la etapa del proceso. Estos se conocen como eventos principales, y se utilizan como puntos de corte dentro del flujo, marcando una transición clara entre lo que ocurre antes y lo que sucede después. Para visualizarlos, se incorpora una barra vertical que divide los eventos previos de los posteriores a dicho hito.



Hemos identificado los eventos principales (pivotal points) que marcan transiciones clave dentro del sistema de reservas y monitoreo de locales con tecnología IoT. Estos eventos delimitan fases importantes del proceso y permiten establecer momentos en los que es necesario validar decisiones fundamentales para garantizar el correcto funcionamiento del sistema.

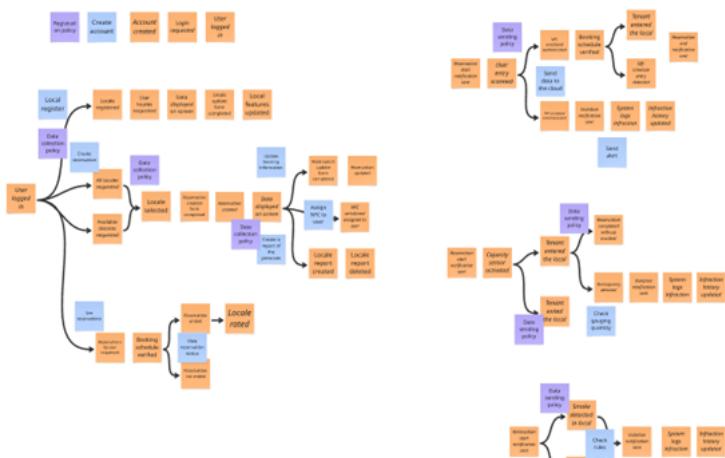
Step 5: Commands:

En esta etapa también incorporamos los comandos, los cuales representan las acciones que desencadenan uno o varios eventos dentro del sistema. A diferencia de los eventos de dominio, que describen hechos ya ocurridos, los comandos se expresan en forma imperativa, indicando las operaciones que deben llevarse a cabo en el sistema para provocar un cambio en su estado o avanzar en el flujo del proceso



Step 6: Policies:

En esta etapa se identifican las políticas de automatización, es decir, reglas que permiten que ciertos comandos se ejecuten automáticamente en respuesta a eventos del dominio. Esto implica que, al producirse un evento específico dentro del sistema, se dispare de forma inmediata la acción correspondiente, sin intervención manual. En otras palabras, se establece una relación directa entre eventos y comandos, permitiendo que el sistema reaccione de forma autónoma ante situaciones previamente definidas.

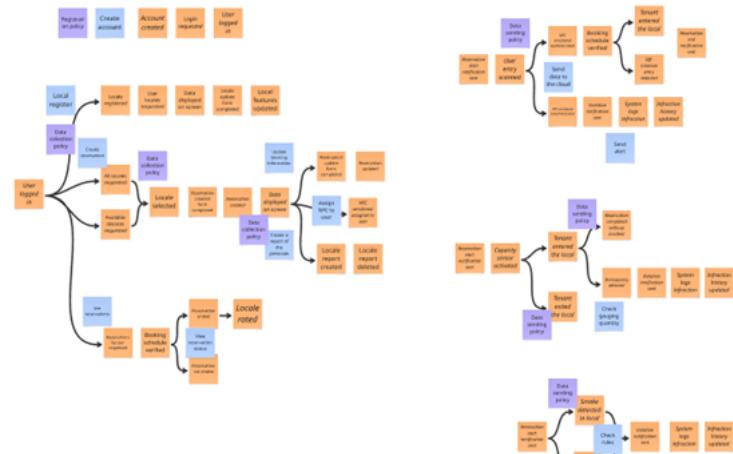


Definimos y organizamos las políticas de automatización que permiten ejecutar comandos de forma automática ante la ocurrencia de eventos clave dentro del sistema. Estas políticas cubren la recolección y envío de datos, la asignación de pulseras NFC, la validación de accesos, la generación de reportes del local, y el

monitoreo de condiciones críticas como sobreaforo, humo o ruido en el recinto. También se estructuraron reglas para el disparo automático de alertas al detectar eventos que representan infracciones, asegurando que el sistema reaccione de manera inmediata sin intervención manual.

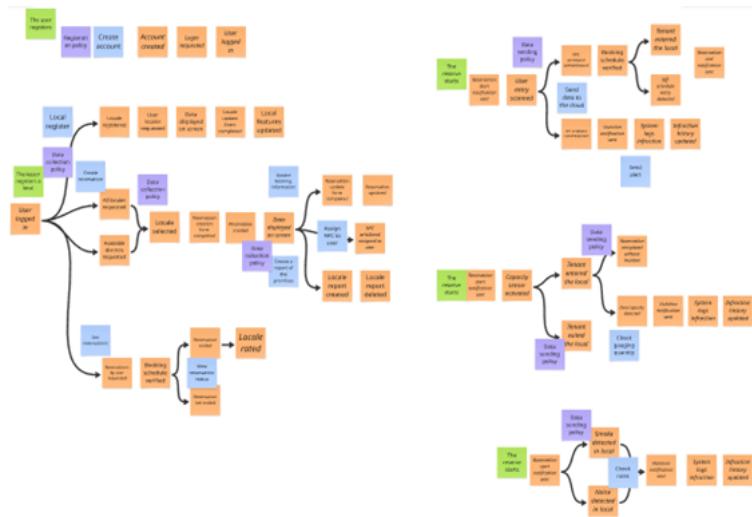
Step 7: Read models:

En esta etapa, incorporamos los modelos de lectura, que representan las vistas de datos del sistema que los agentes (ya sean usuarios o componentes del sistema) utilizan para decidir si deben ejecutar un comando. Esto implica definir visualizaciones específicas como reportes de locales, paneles de monitoreo de aforo, alertas en tiempo real por infracciones, estado de las reservas, historial de asistencia y validación de acceso mediante NFC.



Step 8: External Systems:

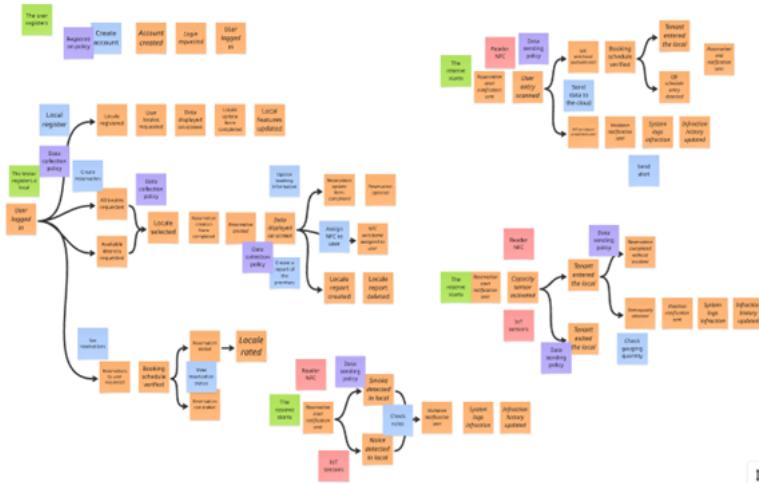
En este paso, integramos al modelo los sistemas externos, es decir, aquellos que están fuera del dominio principal del sistema que estamos diseñando. Estos sistemas pueden intervenir ejecutando comandos hacia nuestro sistema (como entradas) o bien recibir información generada por eventos del dominio (como salidas).



Incorporamos los sistemas externos que interactúan con el dominio principal, representados con etiquetas rojas. Se identificaron elementos clave como el lector NFC y los sensores IoT (aforo, humo, ruido), los cuales permiten recolectar datos del entorno físico y activan eventos automatizados dentro del sistema. Además, se detallaron las políticas de envío de datos, representadas en color morado, las cuales se encargan de transmitir información relevante a la nube o al backend en tiempo real.

Step 9: Aggregates:

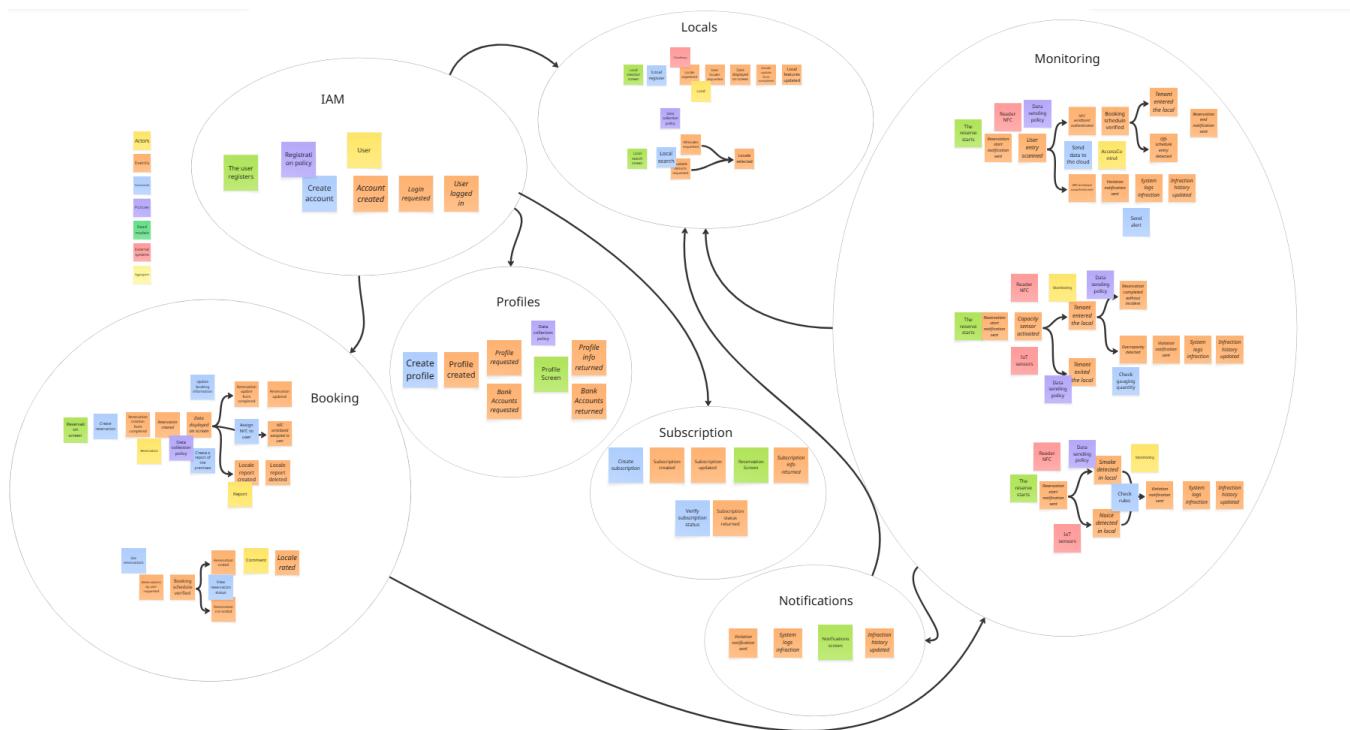
Después de haber definido todos los eventos y comandos del sistema, procedemos a agrupar aquellos conceptos que guardan una relación directa dentro de unidades llamadas agregados. Estos agregados representan componentes clave del dominio, ya que son los encargados de recibir comandos y, a partir de ellos, generar los eventos correspondientes que reflejan los cambios en el estado del sistema



Identificamos los agregados del dominio que encapsulan la lógica de negocio principal del sistema de reservas con IoT. Cada aggregate agrupa los comandos y eventos que afectan una misma unidad lógica, asegurando la coherencia de los datos y el cumplimiento de las reglas definidas. Se establecieron agregados como User, para el control de identidad y autenticación; Locale, encargado del ciclo de vida de los locales; Reservation, que gestiona las reservas y su estado; AccessControl, que regula el ingreso mediante pulseras NFC; y Monitoring, responsable de registrar condiciones detectadas por sensores IoT. Esta estructura permite organizar el dominio de forma modular y consistente, facilitando el mantenimiento y escalabilidad del sistema.

Step 10: Bounded Context:

Finalmente, agrupamos los agregados que mantienen una relación directa entre sí, ya sea porque comparten funciones estrechamente relacionadas o porque están conectados a través de ciertas políticas de negocio. Estos conjuntos de agregados constituyen candidatos naturales para establecer los contextos delimitados (Bounded Contexts) del sistema, permitiendo organizar el dominio en módulos coherentes y bien definidos según sus responsabilidades y reglas específicas.



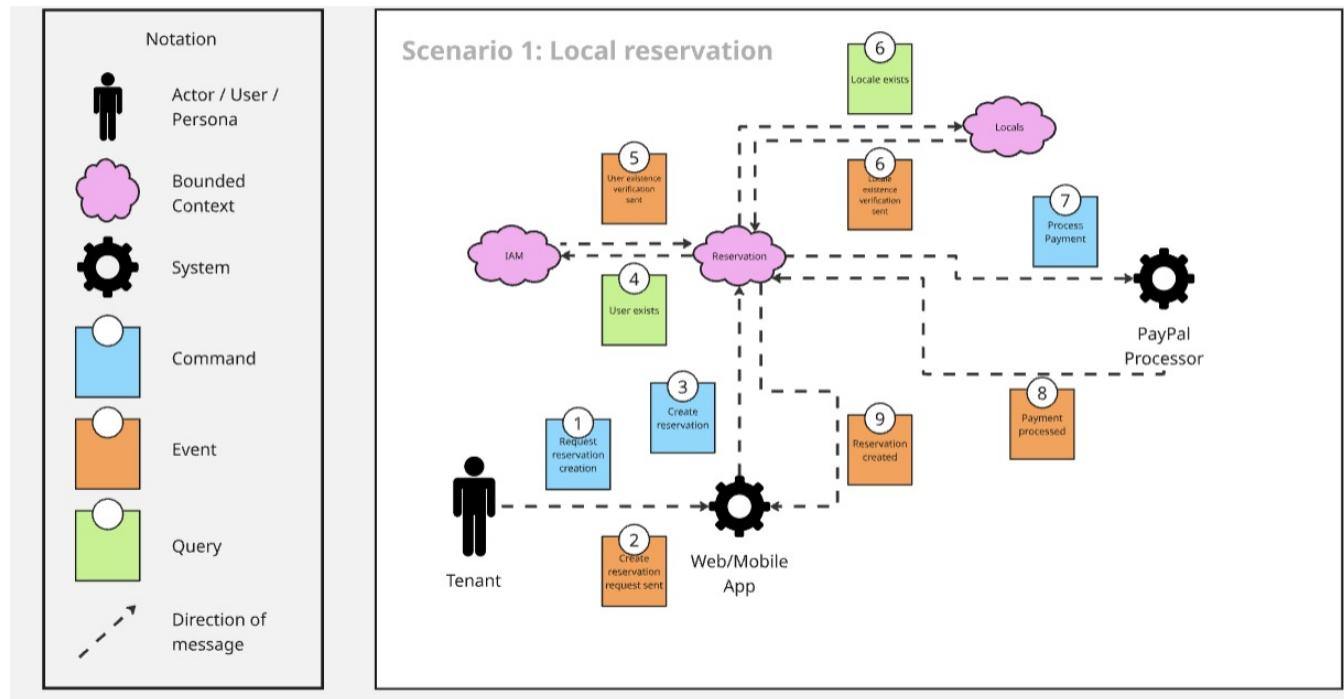
En esta etapa final, estructuramos el sistema en Bounded Contexts claramente definidos, agrupando los agregados que comparten responsabilidades, reglas de negocio o interacciones comunes. Esta segmentación permite organizar el dominio en módulos independientes, cada uno con un propósito específico y una lógica coherente. Se identificaron cinco contextos principales: IAM (Identity and Access Management): encargado de la gestión de identidad, registro, autenticación y control de acceso de los usuarios al sistema. Locals: responsable del registro, actualización, visualización y reporte de los locales disponibles para reserva. Booking: gestiona todo el ciclo de vida de las reservas, incluyendo su creación, modificación, finalización y calificación posterior. Monitoring: controla la recolección de datos desde sensores IoT (como aforo, humo o ruido), valida reglas del local y registra infracciones. Notifications: administra el envío de alertas a usuarios o arrendadores cuando se detectan condiciones críticas o eventos relevantes. Cada uno de estos contextos delimita claramente su modelo de datos, comandos y eventos, lo que permite mantener una arquitectura modular, trazable y preparada para escalar según las necesidades del sistema.

4.2.3. Domain Message Flows Modeling

Scenario 1: Local reservation:

La secuencia de reserva local fue diseñada para optimizar la experiencia del usuario manteniendo la máxima disponibilidad del sistema. La decisión de procesar inicialmente la reserva de forma local (steps 1-4) antes de involucrar sistemas externos refleja el principio de "optimistic UX": proporcionamos retroalimentación inmediata al usuario mientras procesamos las validaciones críticas en segundo plano. Este enfoque reduce significativamente el tiempo de respuesta percibido.

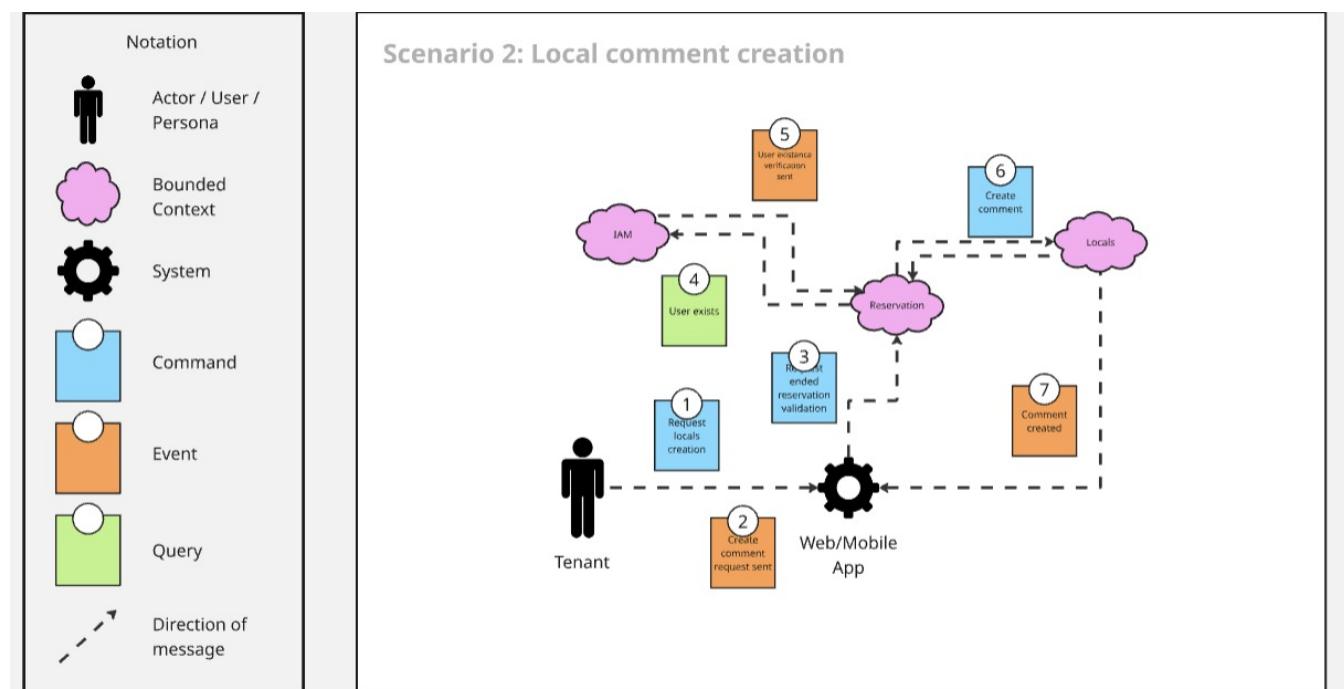
La integración con PayPal (steps 7-8) se posiciona estratégicamente después de la creación de la reserva, no antes, para evitar cargos por transacciones fallidas debido a problemas de disponibilidad de espacios. El flujo asíncrono entre IAM, Reservation y Locals bounded contexts demuestra cómo la arquitectura distribuida mantiene la consistencia eventual sin bloquear la experiencia del usuario.



Scenario 2: Local comment creation:

El flujo de comentarios fue diseñado con un enfoque de validación en capas que prioriza la integridad de datos y la experiencia del usuario. La decisión de validar primero la existencia del usuario (step 4) antes de permitir la creación del comentario (steps 1-3) previene comentarios huérfanos y mantiene la trazabilidad de la información.

La ausencia de un sistema de pago en este flujo es intencional: los comentarios son considerados contenido generado por usuarios que no requiere transacciones monetarias, simplificando significativamente el proceso. La comunicación directa entre bounded contexts (IAM → Reservation → Locals) optimiza el rendimiento al evitar saltos innecesarios entre servicios.

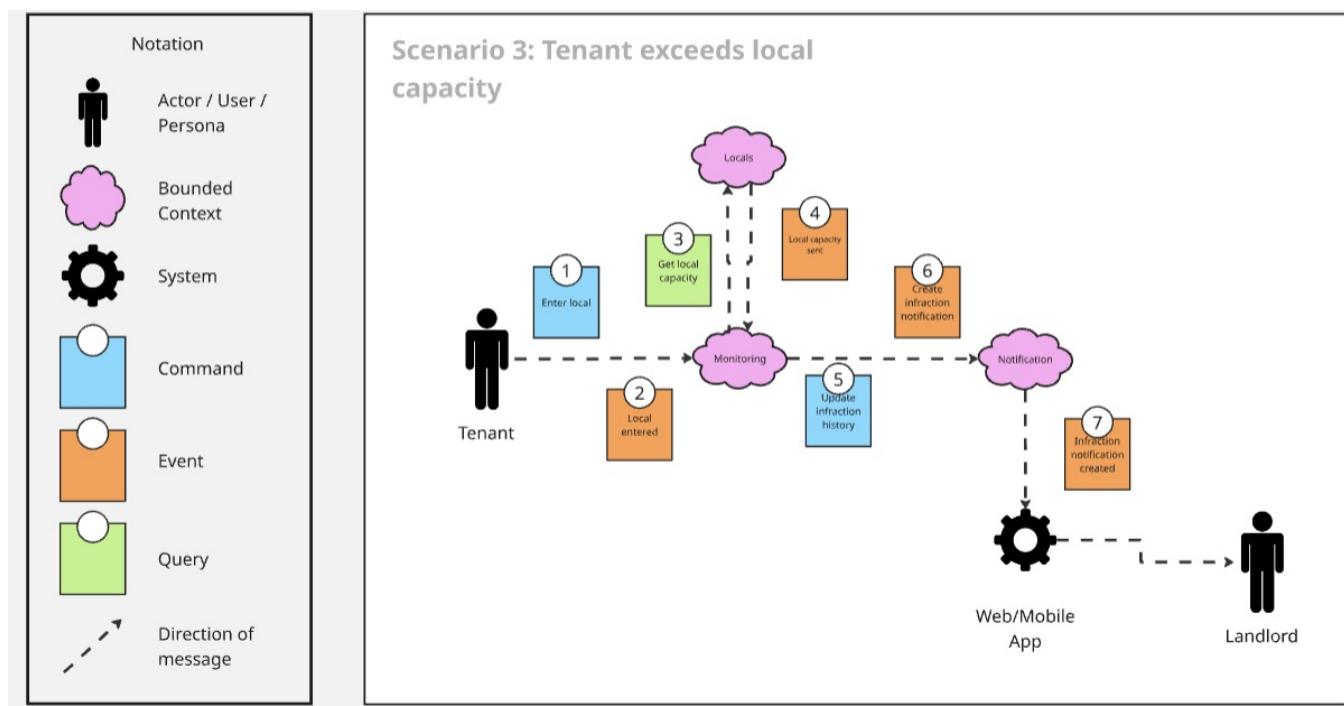


Scenario 3: Tenant exceeds local capacity:

Este escenario implementa un patrón de monitoreo proactivo que detecta condiciones de sobrecarga antes de que impacten la experiencia del usuario. La decisión de involucrar al Landlord (propietario) en el flujo (steps 6-7) reconoce que ciertos problemas operativos requieren intervención humana y no pueden ser

resueltos automáticamente por el sistema.

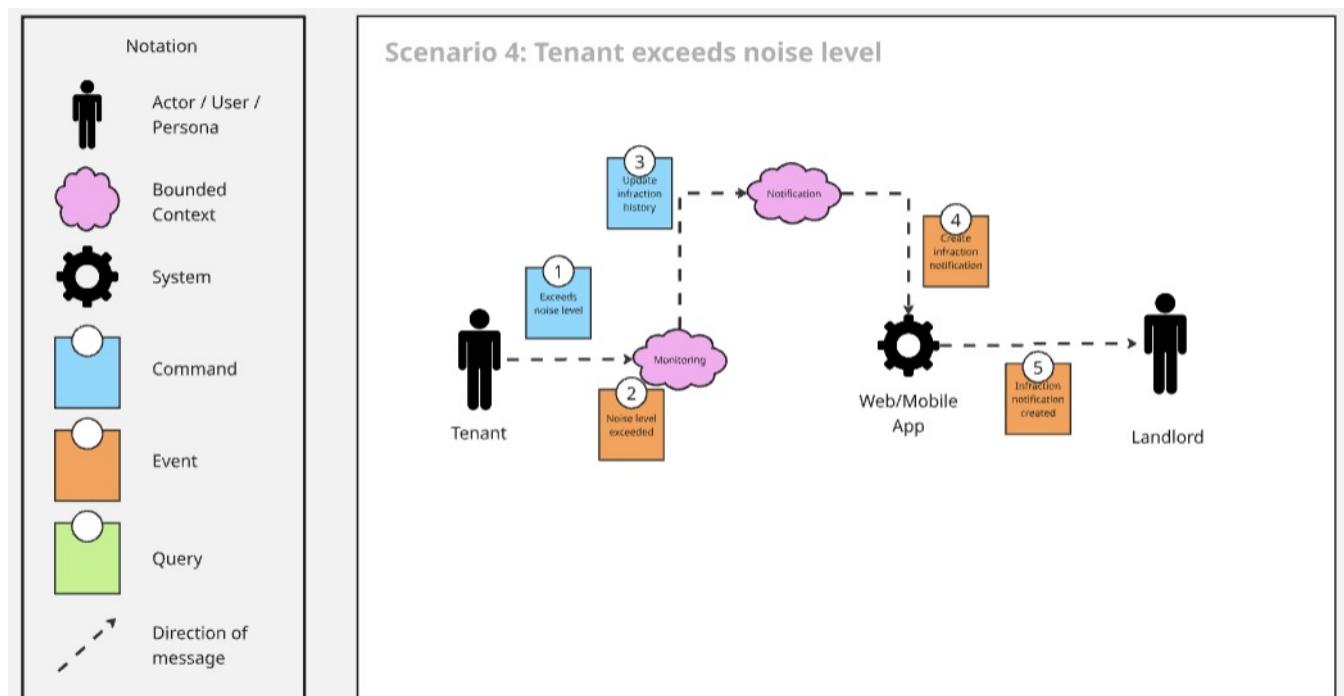
El comando "Enter local" (step 1) actúa como trigger para el sistema de monitoring, creando un registro de actividad que permite el análisis de patrones de uso. Esta aproximación event-driven facilita la escalabilidad horizontal del sistema de monitoreo sin impactar el rendimiento de las operaciones principales.



Scenario 4: Tenant exceeds noise level:

La arquitectura de detección de infracciones fue diseñada para balancear la automatización con la supervisión humana. El comando "Exceeds noise level" (step 1) desencadena un flujo que involucra tanto sistemas automatizados (Monitoring, Notification) como actores humanos (Landlord), reconociendo que las violaciones de políticas requieren tanto detección técnica como juicio humano.

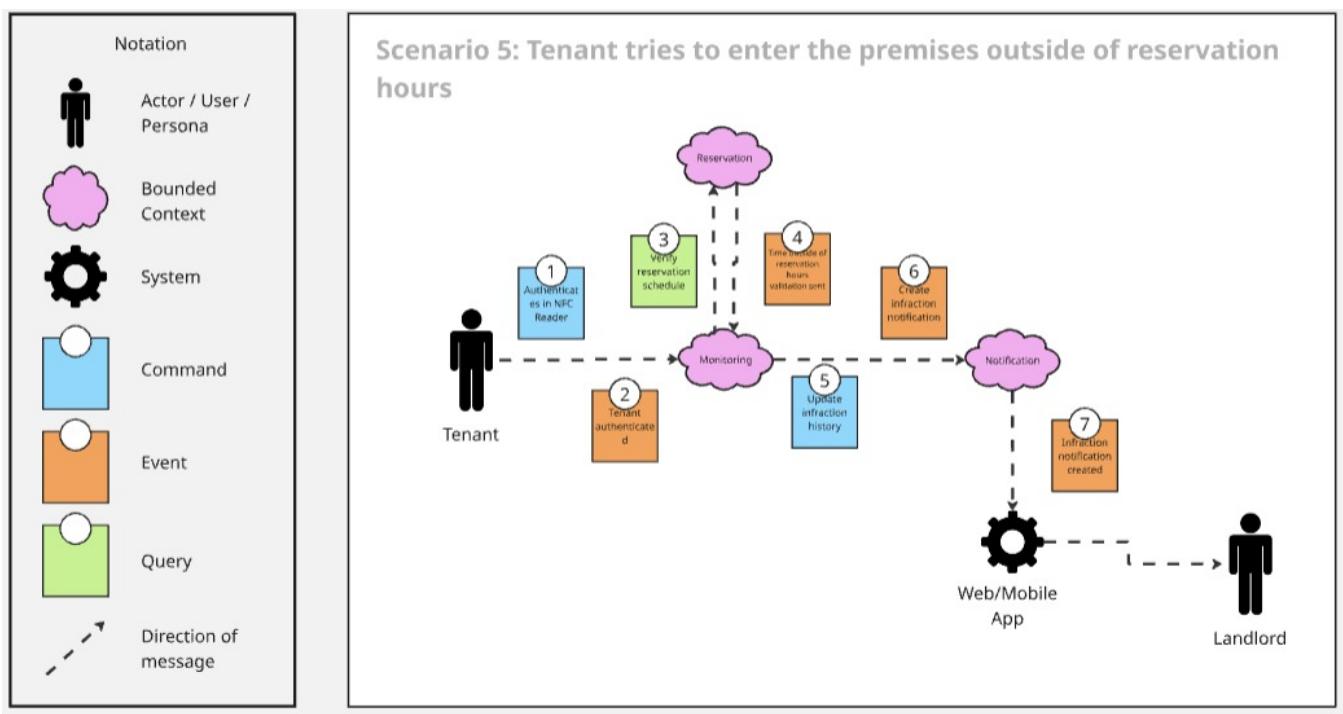
La simplicidad aparente del flujo (solo 5 steps) es intencional: las infracciones requieren respuesta rápida, y un flujo complejo podría retrasar las notificaciones críticas. La comunicación directa entre sistemas minimiza la latencia en situaciones que podrían requerir intervención inmediata.



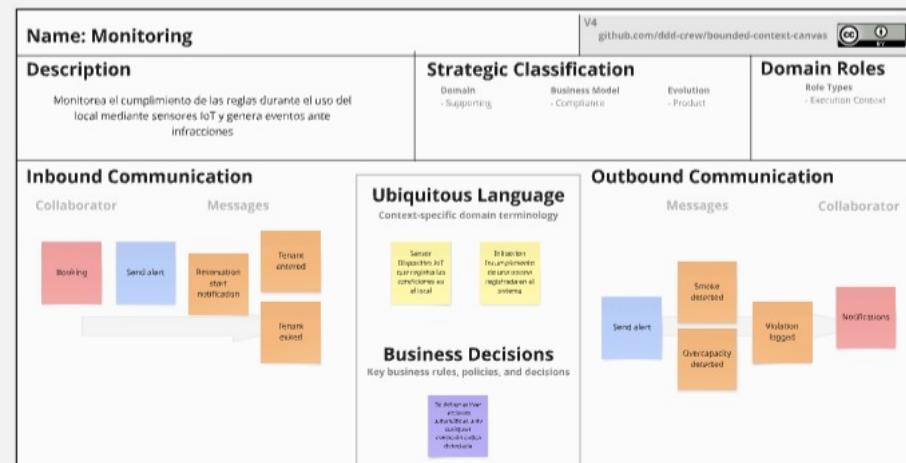
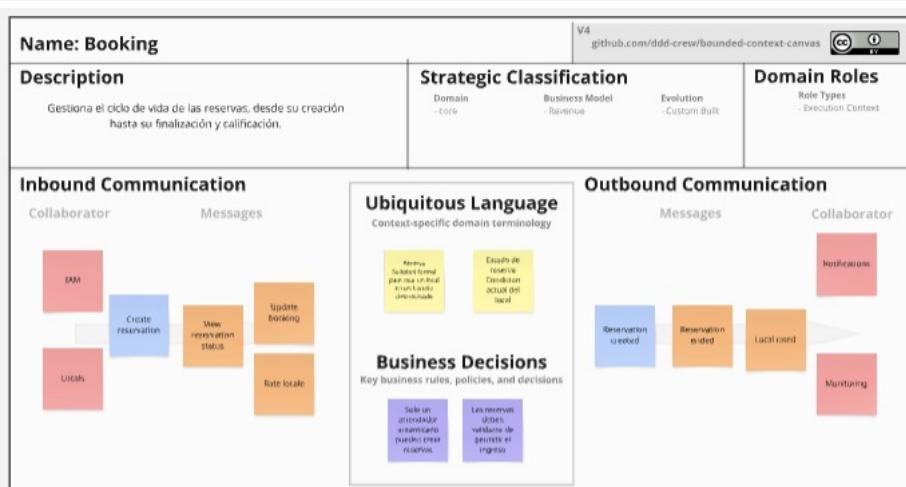
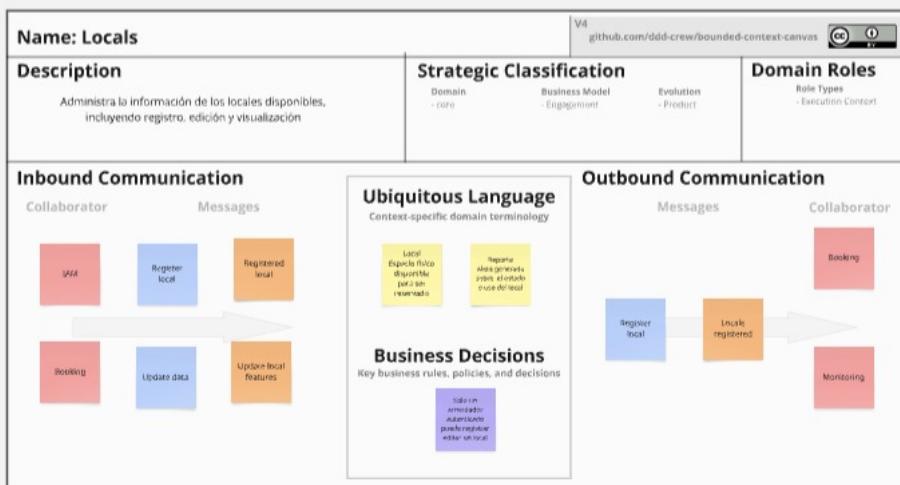
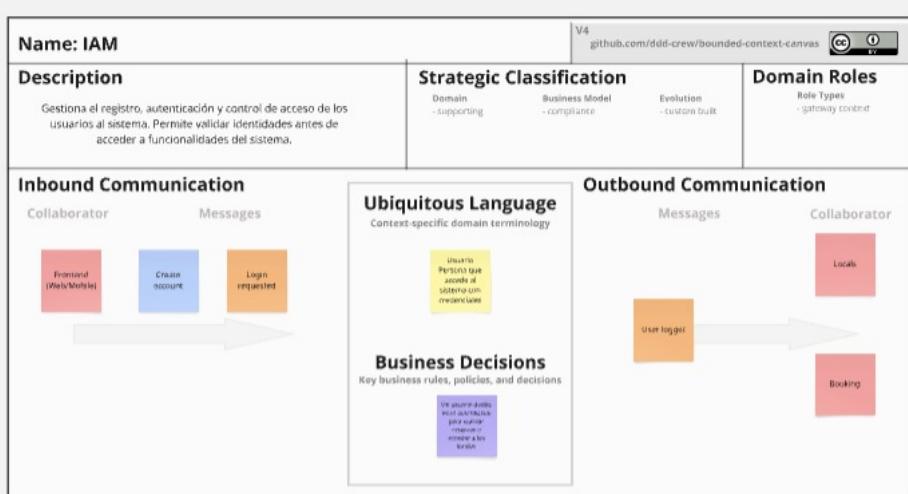
Scenario 5: Tenant tries to enter the premises outside of reservation hours

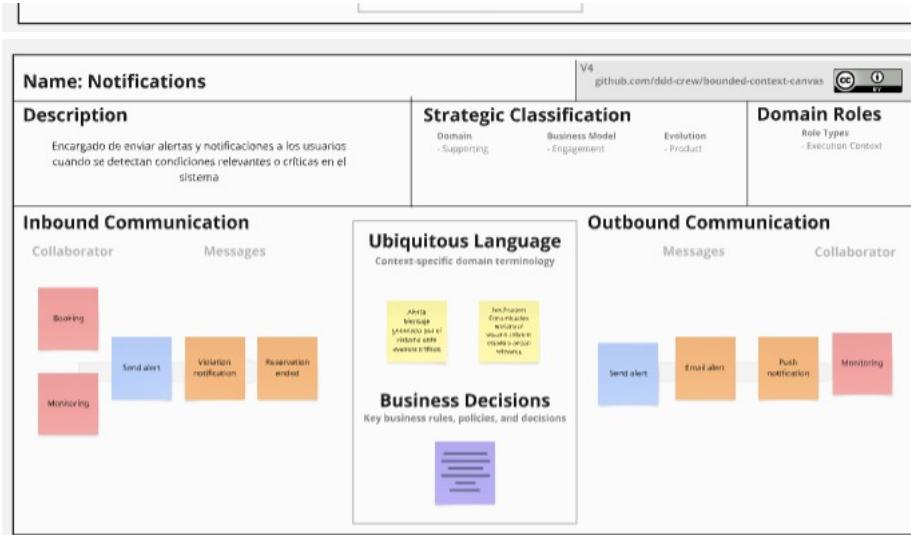
Este escenario implementa un sistema de validación temporal que protege tanto a propietarios como inquilinos. La consulta al schedule de reservas (step 3) añade una capa de validación lógica. Esta aproximación de doble validación (física + lógica) es fundamental en sistemas donde el acceso no autorizado puede tener consecuencias legales y financieras.

La decisión de mantener un historial de infracciones (step 5) facilita el análisis de patrones y la toma de decisiones informadas sobre políticas de acceso. El flujo relativamente extenso (7 steps) refleja la complejidad inherente en la gestión de accesos seguros y el cumplimiento de políticas.



4.2.4. Bounded Context Canvases





IAM (Identity and Access Management)

El bounded context de IAM fue diseñado como el núcleo de seguridad y control de acceso del sistema, manteniendo una separación clara entre autenticación y autorización. La decisión de centralizar la gestión de usuarios en este contexto refleja principios de seguridad consolidados: un punto único de control facilita auditorías, cumplimiento normativo y gestión de políticas de seguridad.

Las comunicaciones entrantes (Request via Webhook, Create User, Login Request) están diseñadas para manejar múltiples canales de autenticación, mientras que las salientes (User Logout, Locals) mantienen la integración con otros contextos. La clasificación estratégica como "Supporting" reconoce que, aunque crítico, IAM facilita las funcionalidades del negocio sin ser el core domain.

Locals

El contexto Locals encapsula toda la lógica relacionada con la gestión de espacios físicos y su información asociada. La decisión de separar este contexto del booking refleja el principio de separación de responsabilidades: Locals se enfoca en la información y características de los espacios, mientras que Booking maneja las transacciones y reservas.

Las múltiples comunicaciones entrantes (WM, Register Local, Update Data, etc.) demuestran la naturaleza central de este contexto en el ecosistema. La clasificación como "Core" refleja que la gestión de locales es fundamental para el modelo de negocio. El lenguaje ubicuo incluye términos específicos del dominio inmobiliario que aseguran consistencia en la comunicación entre stakeholders técnicos y de negocio.

Booking

El contexto de Booking gestiona el ciclo completo de vida de las reservas, desde la creación hasta la finalización. La decisión de mantener este contexto separado de Locals permite evolucionar independientemente las lógicas de reserva sin impactar la gestión de espacios. Esta separación facilita también la implementación de diferentes modelos de pricing y políticas de reserva.

Las comunicaciones con IAM, Locals y Monitoring reflejan las dependencias necesarias para un proceso de reserva completo: validación de usuarios, verificación de disponibilidad de espacios, y seguimiento de cumplimiento de políticas. La clasificación como "Core" junto con Locals forma el corazón del modelo de negocio de la plataforma.

Monitoring

El contexto de Monitoring implementa capacidades de observabilidad y cumplimiento en tiempo real. La decisión de crear un contexto dedicado para monitoring refleja la complejidad de gestionar espacios físicos donde el cumplimiento de reglas (ruido, ocupación, horarios) es crítico para el éxito del negocio.

Las comunicaciones entrantes desde Booking y otros sistemas, junto con las salientes hacia Notifications, crean un sistema de alertas proactivo que previene problemas antes de que escalen. La clasificación como "Supporting" reconoce que, aunque esencial para la operación, su función principal es facilitar y asegurar el cumplimiento de las operaciones core del negocio.

Notifications

El contexto de Notifications centraliza todas las comunicaciones hacia usuarios finales, implementando un patrón de messaging que desacopla la generación de eventos de su entrega. Esta separación permite implementar diferentes canales de comunicación (email, SMS, push notifications) sin impactar los contextos que generan los eventos.

La simplicidad aparente del contexto (pocas comunicaciones entrantes, múltiples salientes) refleja su naturaleza de hub de comunicaciones. La clasificación como "Generic" reconoce que las capacidades de notificación son reutilizables across múltiples dominios y no contienen lógica específica del negocio de alquiler de espacios.

4.2.5. Context Mapping

1. Pasos para Crear el Context Mapping

1.1. Identificación de los Bounded Contexts

- IAM (Identity and Access Management)
- Locals
- Booking
- Monitoring
- Notifications

1.2. Identificación de Relaciones Iniciales

- IAM ↔Booking: Relación de Customer/Supplier.
- IAM proporciona la autenticación y control de acceso, mientras Booking consume la identidad para permitir la creación de reservas.
- Locals →Booking: Relación de Customer/Supplier.
- Locals administra la disponibilidad y características de los espacios, Booking consulta esa información para registrar una reserva.
- Booking ↔Monitoring: Relación de Customer/Supplier.
- Booking genera el inicio y fin de la reserva, mientras Monitoring supervisa el cumplimiento de normas durante ese periodo.
- Monitoring ↔Notifications: Relación de Conformist.
- Notifications se adapta a los eventos generados por Monitoring para emitir alertas sin modificar la estructura de los datos.
- Booking ↔Notifications: Relación de Conformist.
- Booking puede generar eventos que Notifications transforma en mensajes para usuarios.

2. Análisis de Alternativas y Preguntas Clave

2.1. ¿Qué pasaría si movemos la gestión de reportes de locales desde Locals a Monitoring?

Impacto: Monitoring tendría control total del estado operacional del local.

Se perdería la separación entre información estática del local (Locals) y condiciones temporales (Monitoring).

Discusión:

No se recomienda el cambio. Mantener reportes en Locals permite independencia de la información operativa.

2.2. ¿Y si separáramos Monitoring en dos contexts: SensorEvents y RuleValidation?

Impacto:

Permitiría escalar el procesamiento de eventos por separado del motor de reglas.

Mayor complejidad y necesidad de sincronización.

Discusión:

Podría evaluarse si los volúmenes de datos lo justifican, pero por ahora es preferible mantenerlo unificado.

2.3. ¿Podría Notifications compartir un kernel con Booking?

Impacto:

Se podría optimizar el modelo de eventos compartidos.

Riesgo de acoplamiento excesivo.

Discusión:

Mejor mantener el contrato como evento público. No se recomienda un shared kernel.

2.4. ¿Sería viable mover la asignación de pulseras NFC de Booking a IAM?

Impacto:

IAM controlaría identidad + dispositivo asignado.

Aumenta la carga de IAM, mezcla responsabilidades de autenticación con gestión de dispositivos.

Discusión:

Se recomienda mantener la asignación en Booking para no contaminar IAM con lógica operacional.

3. Alternativas Recomendadas de Context Mapping

Mantener la separación entre datos estáticos (Locals) y monitoreo operacional (Monitoring).

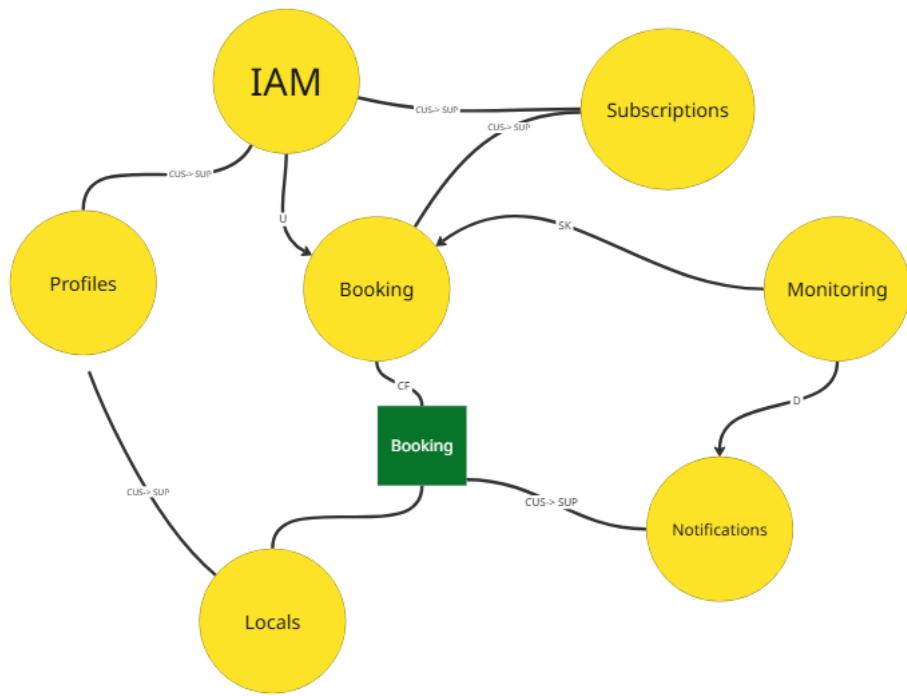
No dividir Monitoring aún, salvo que se presenten cuellos de botella o necesidades de escalabilidad por volumen de eventos.

Evitar compartir kernels entre contexts; priorizar integración por eventos públicos.

Mantener la asignación de dispositivos dentro de Booking.

4. Patrones de Relaciones Sugeridos

- Customer/Supplier: IAM ↔Booking, Locals ↔Booking, Booking ↔Monitoring
- Conformist: Monitoring ↔Notifications, Booking ↔Notifications
- Published Language: Monitoring publica eventos con estructura conocida para consumo externo (ej. por Notifications)
- Anti-Corruption Layer (ACL): Podría evaluarse entre Monitoring y cualquier motor externo de validación de reglas en el futuro

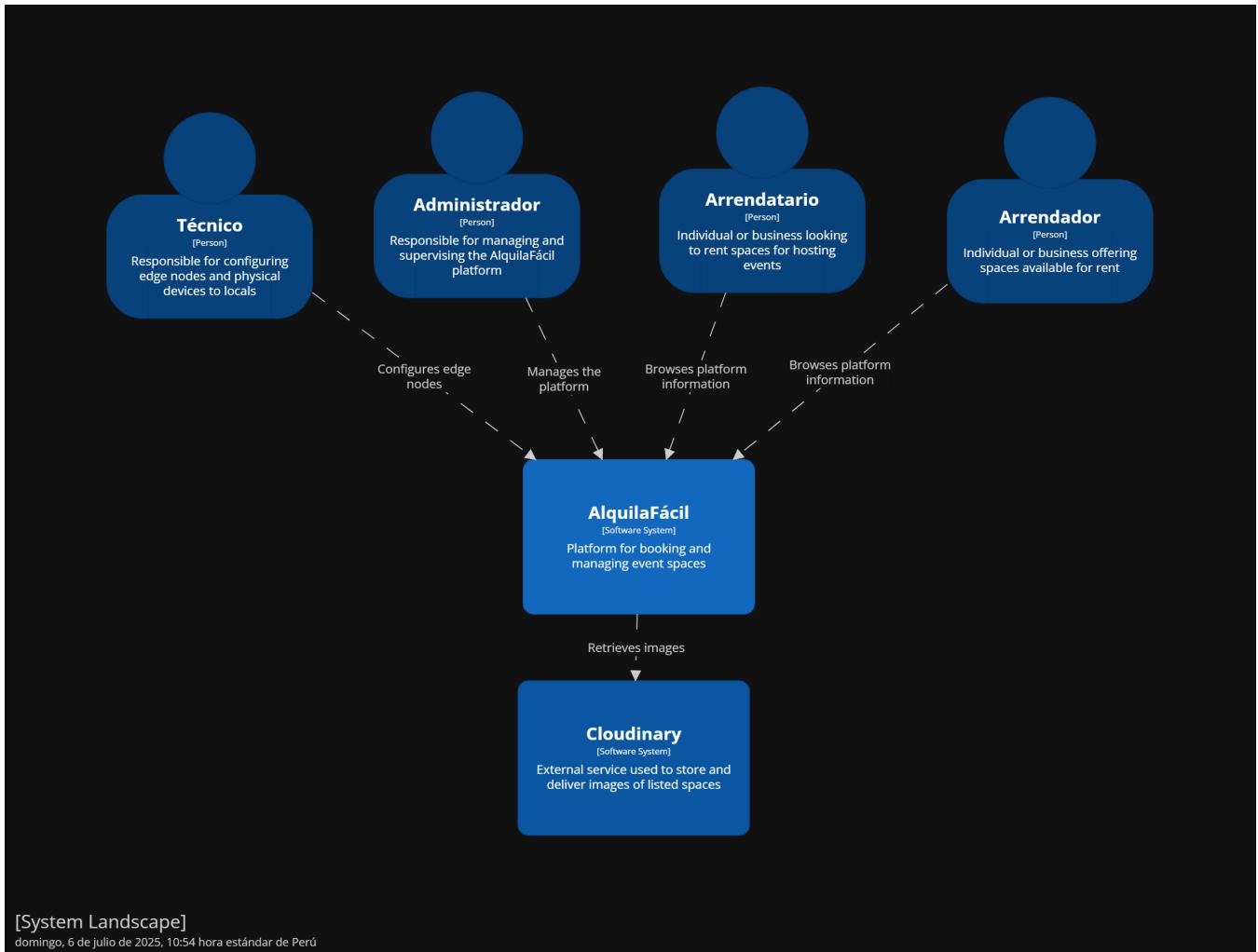


4.3. Software Architecture

4.3.1. Software Architecture System Landscape Diagram

El diagrama de landscape proporciona una vista estratégica de alto nivel que posiciona AlquilaFácil dentro del ecosistema tecnológico más amplio, mostrando cómo el sistema se integra con servicios externos críticos para su funcionamiento. La aparente similitud con el diagrama de contexto es intencional y beneficiosa: demuestra que el core del sistema mantiene su identidad y responsabilidades principales, independientemente de las integraciones externas.

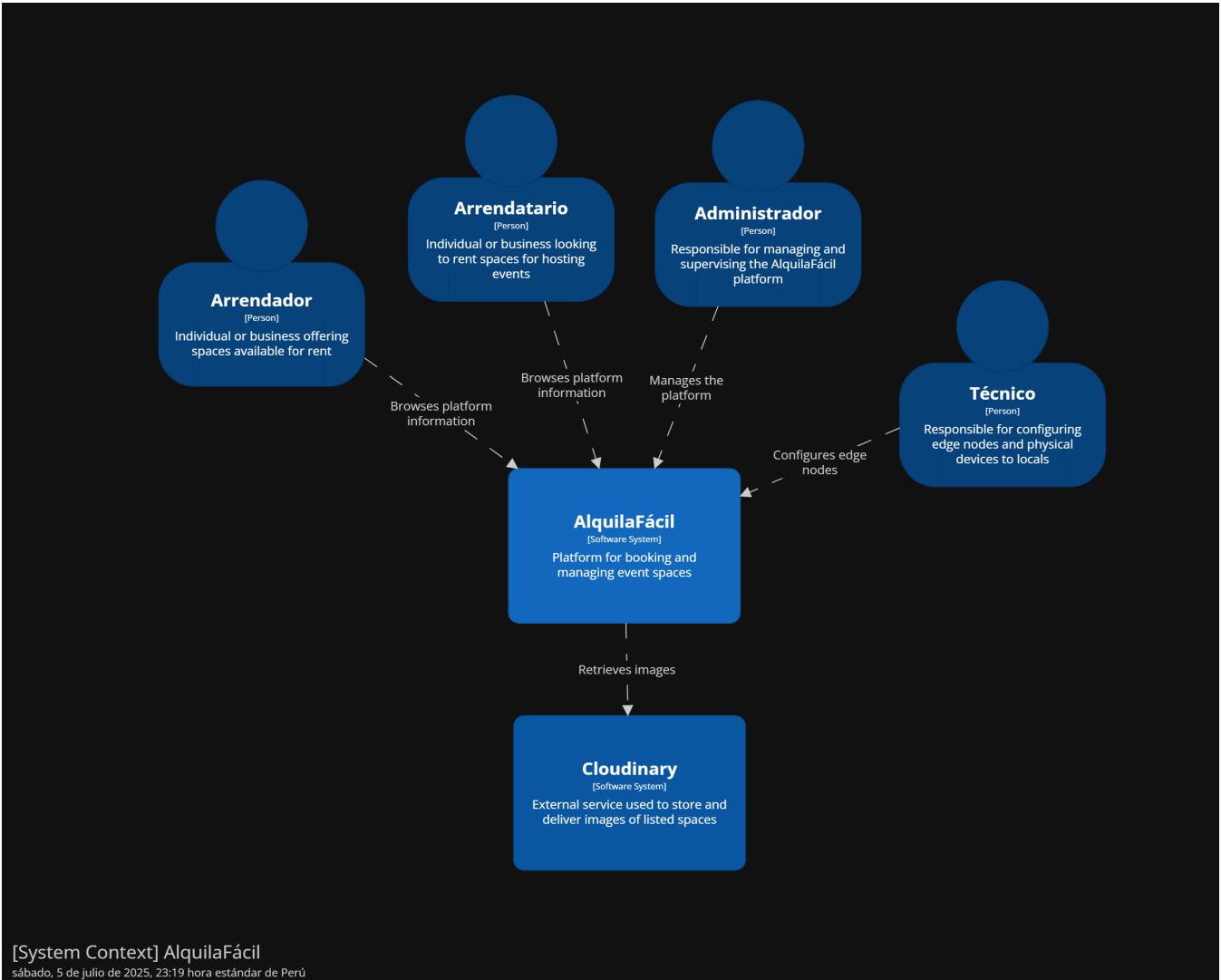
La decisión de mostrar únicamente Cloudinary como sistema externo refleja una filosofía de diseño "lean": mantener las dependencias externas al mínimo necesario para reducir la complejidad y los puntos de falla. Cloudinary fue seleccionado específicamente porque el manejo de imágenes (almacenamiento, transformación, optimización, CDN) requiere infraestructura altamente especializada que sería ineficiente desarrollar internamente. Esta vista landscape permite a los stakeholders comprender rápidamente las dependencias críticas del sistema y planificar adecuadamente la gestión de riesgos y la continuidad del negocio.



4.3.2. Software Architecture Context Level Diagrams

El diagrama de contexto presenta una vista simplificada e intencionalmente minimalista del sistema AlquilaFácil, enfocándose exclusivamente en los actores principales y sus interacciones directas con la plataforma. Esta decisión de simplicidad es estratégica: permite a los stakeholders no técnicos comprender rápidamente el propósito y alcance del sistema sin perderse en detalles técnicos.

La similitud aparente con el diagrama de landscape es completamente intencional y refleja una arquitectura coherente donde el sistema central mantiene las mismas responsabilidades fundamentales independientemente del nivel de detalle presentado. La diferencia clave radica en que el contexto se centra en QUÉ hace el sistema (booking y gestión de espacios), mientras que el landscape muestra CÓMO interactúa con servicios externos. Esta consistencia entre vistas valida la solidez del diseño arquitectónico.

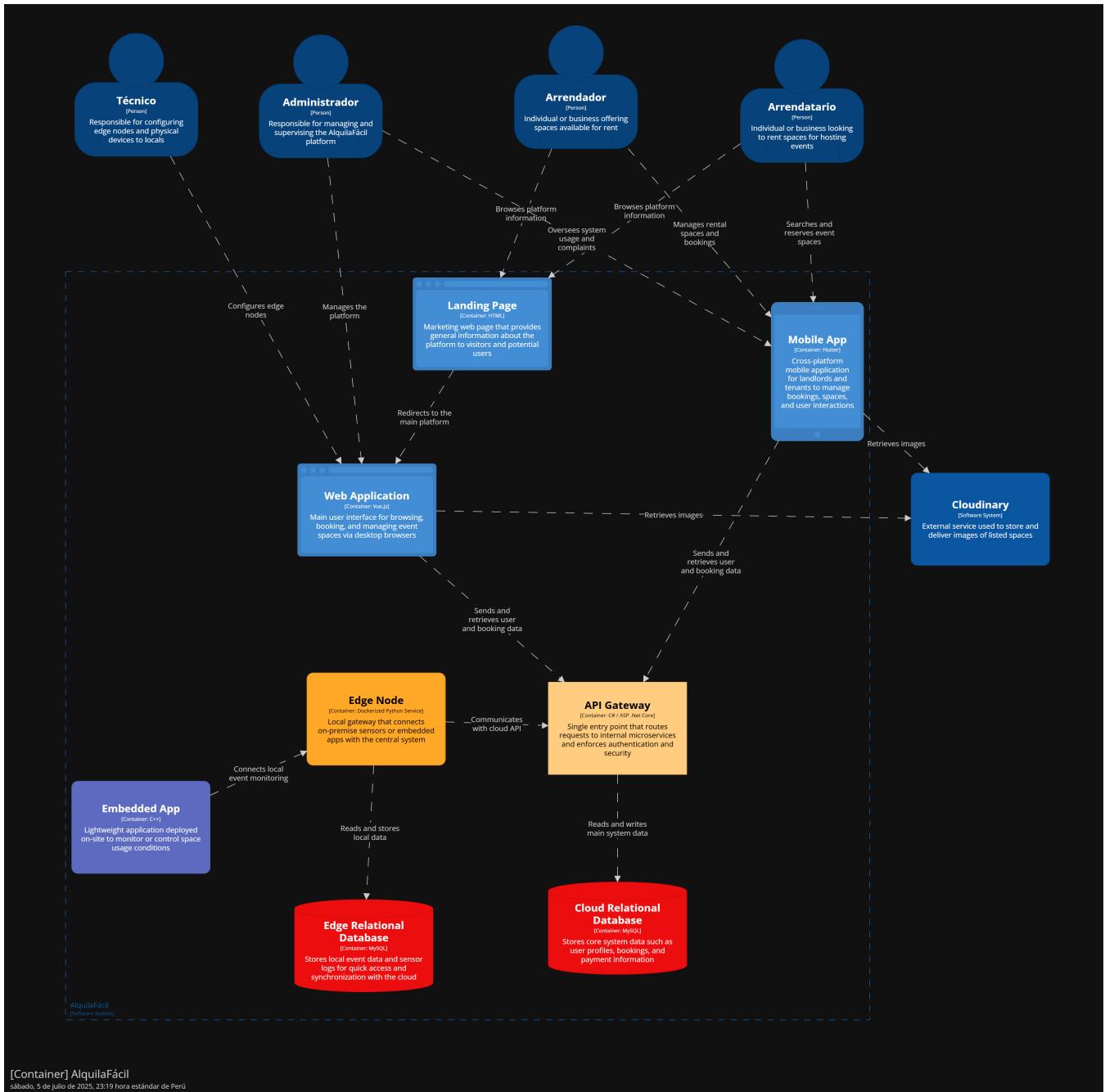


[System Context] AlquilaFácil
sábado, 5 de julio de 2025, 23:19 hora estándar de Perú

4.3.3. Software Architecture Container Level Diagrams

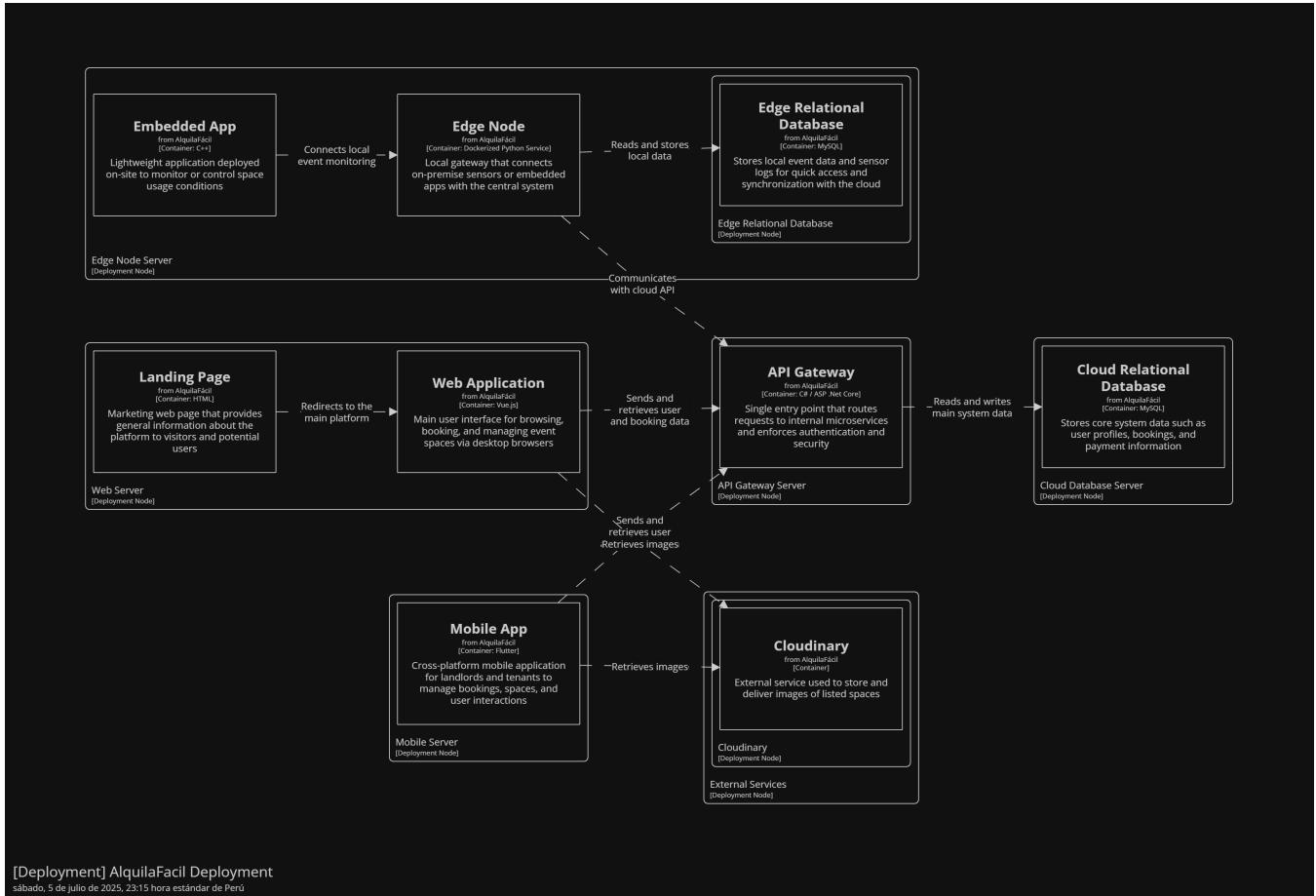
El diagrama de contenedor fue diseñado para mostrar la arquitectura técnica completa del sistema AlquilaFácil, destacando cómo cada componente tecnológico contribuye a satisfacer las necesidades de los diferentes tipos de usuarios. La decisión de incluir múltiples interfaces (Landing Page, Web Application, Mobile App) responde a la necesidad de proporcionar diferentes puntos de entrada según el contexto de uso: la landing page para atraer nuevos usuarios, la aplicación web para gestión completa desde escritorio, y la aplicación móvil para acceso rápido en movilidad.

La arquitectura distribuida con Edge Nodes refleja la realidad geográfica del negocio de alquiler de espacios, donde la latencia y la disponibilidad local son críticas. La separación entre bases de datos edge y cloud permite optimizar el rendimiento local mientras se mantiene la consistencia global de datos. El API Gateway centraliza la seguridad y el control de acceso, mientras que Cloudinary maneja específicamente las imágenes, reconociendo que el contenido visual es fundamental en el negocio de alquiler de espacios.



4.3.4. Software Architecture Deployment Diagrams

El diagrama de despliegue ilustra la estrategia de distribución física y lógica del sistema, diseñada para optimizar el rendimiento y la disponibilidad geográfica. La decisión de implementar una arquitectura edge-first responde directamente a los requisitos del negocio: los usuarios necesitan acceso rápido y confiable a información de espacios locales, independientemente de su ubicación geográfica. La separación clara entre deployment nodes (Edge Node Server, Web Server, API Gateway Server, etc.) refleja principios de separación de responsabilidades y escalabilidad horizontal. Cada servidor tiene un propósito específico, permitiendo escalamiento independiente según la demanda. La ubicación de Cloudinary como servicio externo reconoce que el almacenamiento y procesamiento de imágenes requiere infraestructura especializada que es más eficiente como servicio managed. Esta arquitectura también facilita el mantenimiento y las actualizaciones, ya que cada componente puede ser gestionado independientemente.



Capítulo V: Tactical-Level Software Design

5.1. Bounded Context: IAM Context

5.1.1. Domain Layer

El **Domain Layer** representa el núcleo del bounded context IAM, donde se encuentran las entidades, value objects, agregados y contratos de repositorio que encapsulan la lógica de negocio.

- **Aggregate Root**
 - **User**: entidad principal que representa a un usuario del sistema. Contiene atributos como **Id**, **Username**, **Email** y **PasswordHash**. Define operaciones para actualizar nombre de usuario y contraseña.
- **Entities**
 - **Role**: define los roles disponibles en el sistema (**ADMIN**, **CLIENT**).
 - **UserRole**: entidad de unión que materializa la relación muchos a muchos entre **User** y **Role**.
- **Value Objects**
 - **UserId**: encapsula el identificador del usuario.
 - **Email**: asegura el formato correcto y unicidad del correo electrónico.
 - **EncryptedPassword**: garantiza que la contraseña siempre se maneje en formato hasheado.
- **Enumeraciones**
 - **EUserRoles**: enum que define los roles base del sistema.
- **Interfaces (Repositories)**
 - **IUserRepository**: abstracción para persistencia de usuarios (**FindByEmail**, **ExistsByUsername**, **GetUsernameById**, etc.).
 - **IUserRoleRepository**: valida la existencia de roles de usuario.

El dominio establece las **reglas clave**: cada **User** debe tener email único, username único y contraseñas siempre encriptadas. La relación **User** ↔ **Role** es de tipo M:N mediante **UserRole**.

5.1.2. Interface Layer

El **Interface Layer** corresponde a la capa de presentación (exposición de API REST).

- **Controllers**
 - **AuthenticationController**: expone los endpoints de autenticación (**/auth/sign-in**, **/auth/sign-up**). Recibe comandos, los transforma en objetos de aplicación y retorna un **ActionResult**.

- o **UserController**: expone endpoints relacionados con la gestión de usuarios (`/users/{id}`, `/users`, `/users/{id}/username`). Permite obtener información y actualizar el nombre de usuario.

Su responsabilidad es recibir solicitudes HTTP, validar datos básicos, mapear a comandos/queries y retornar respuestas adecuadas. No contiene lógica de negocio.

5.1.3. Application Layer

El **Application Layer** coordina los casos de uso del bounded context IAM. Se implementa a través de servicios de comandos y queries.

- **Command Handlers**

- o **UserCommandService**: maneja flujos de `SignIn`, `SignUp` y `UpdateUsername`.
 - `SignUp`: crea un nuevo usuario, valida unicidad, encripta la contraseña y genera token.
 - `SignIn`: valida credenciales contra la base de datos y devuelve un `AuthResult`.
 - `UpdateUsername`: actualiza el nombre de usuario asegurando unicidad.
- o **SeedUserRolesCommandService**: inicializa roles base en la base de datos.

- **Query Handlers**

- o **UserQueryService**: implementa consultas (`GetUserById`, `GetAllUsers`, `GetUserByEmail`, `GetUsernameById`, `UserExists`).

Los comandos y queries se representan como DTOs inmutables (`SignInCommand`, `SignUpCommand`, `UpdateUsernameCommand`, etc.), facilitando la separación CQRS.

5.1.4. Infrastructure Layer

El **Infrastructure Layer** contiene implementaciones concretas de los contratos definidos en el dominio y la interacción con servicios externos.

- **Repositorios**

- o **UserRepository**: implementación de `IUserRepository` sobre PostgreSQL.
- o **UserRoleRepository**: implementación de `IUserRoleRepository`.

- **Servicios externos**

- o **JwtTokenService**: implementación de `ITokenService` para generar y validar JWT.
- o **BcryptHashingService**: implementación de `IHashingService` para hashear y verificar contraseñas.

- **Base de datos (PostgreSQL)**

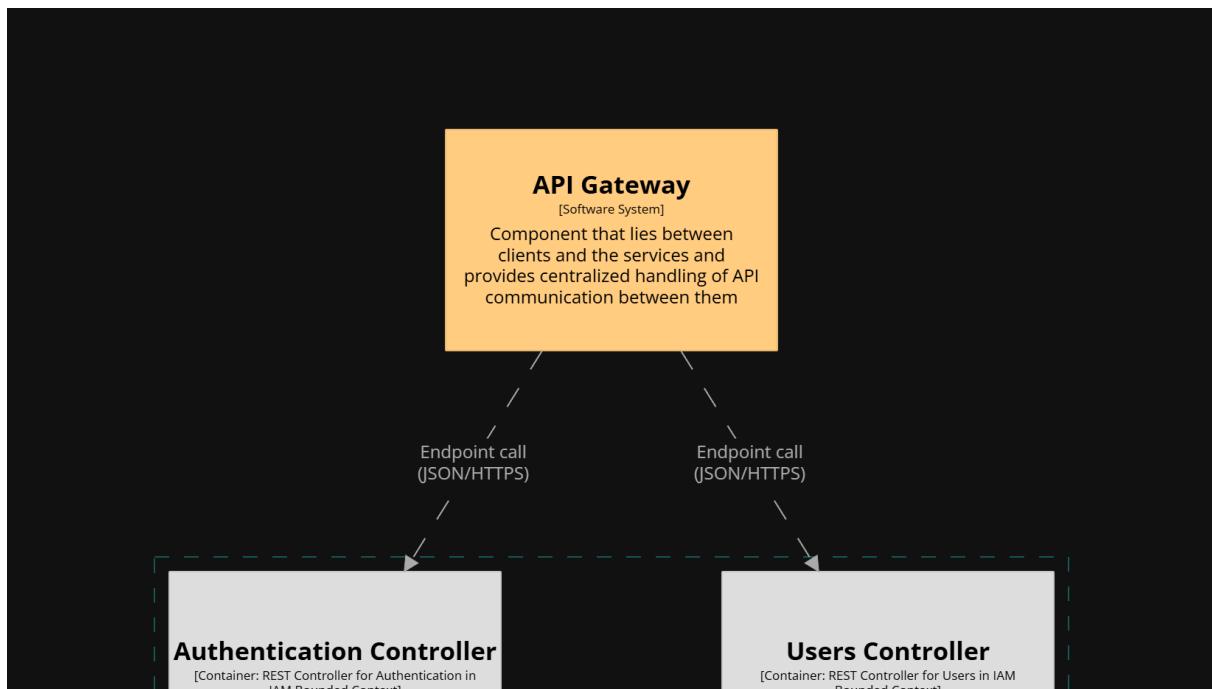
- o Tablas: `users`, `roles`, `user_roles`.
- o Persisten las entidades del dominio y soportan las relaciones M:N.

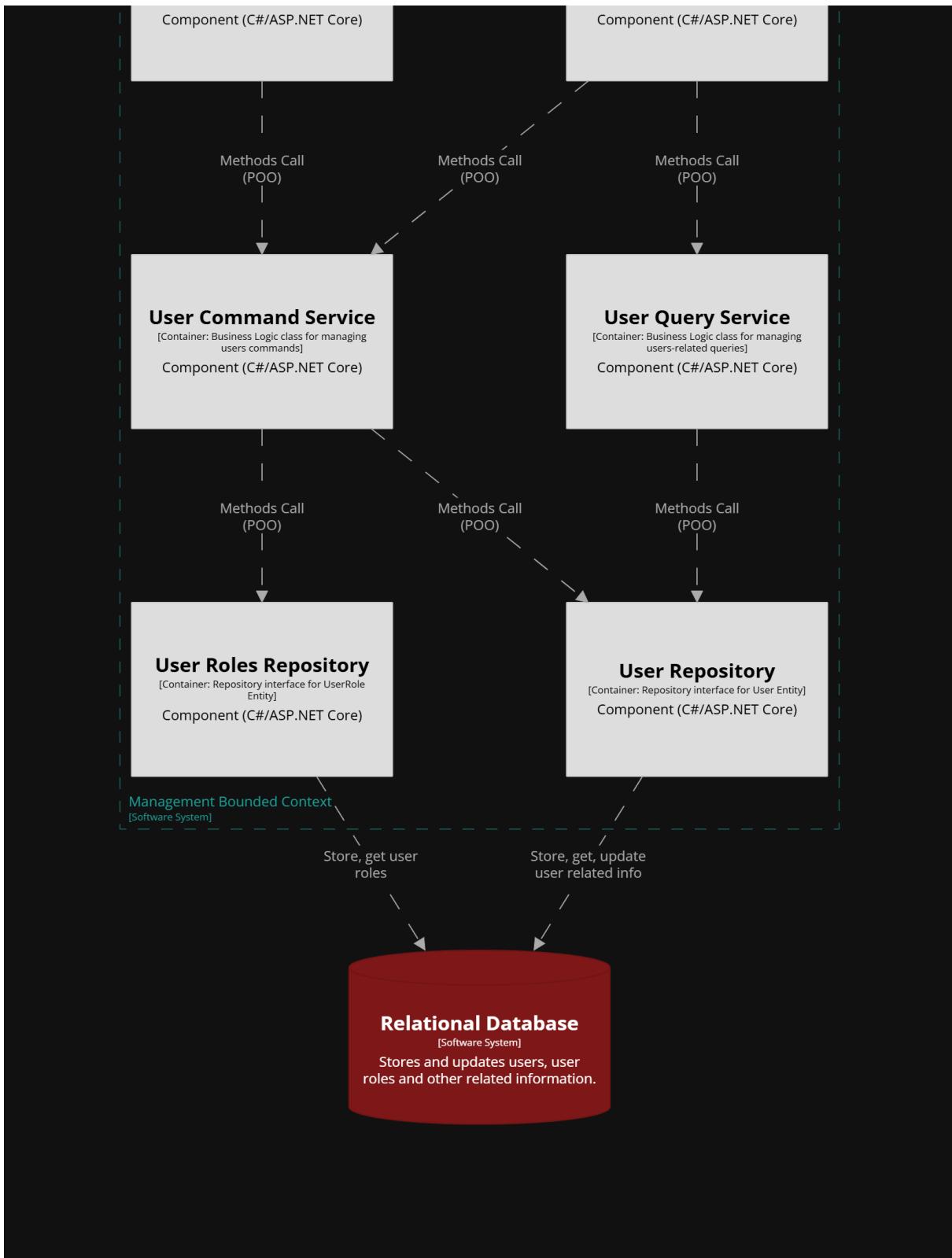
5.1.6. Bounded Context Software Architecture Component Level Diagrams

El bounded context IAM se despliega como un contenedor **IAM API** en el modelo C4. Sus componentes principales son:

- **Controllers**: `AuthenticationController`, `UserController`.
- **Application Services**: `UserCommandService`, `UserQueryService`, `SeedUserRolesCommandService`.
- **Domain**: entidades (`User`, `Role`, `UserRole`), value objects (`UserId`, `Email`, `EncryptedPassword`) y enums (`EUserRoles`).
- **Infrastructure Adapters**: `UserRepository`, `UserRoleRepository` (adaptadores DB), `JwtTokenService` (adaptador JWT), `BcryptHashingService` (adaptador de hash).
- **Base de datos PostgreSQL**: persistencia de usuarios y roles.

El flujo de interacción es: Controllers → Application Services → Domain (Entities, VOs, Repositories) → Adapters (DB/JWT/Hash).

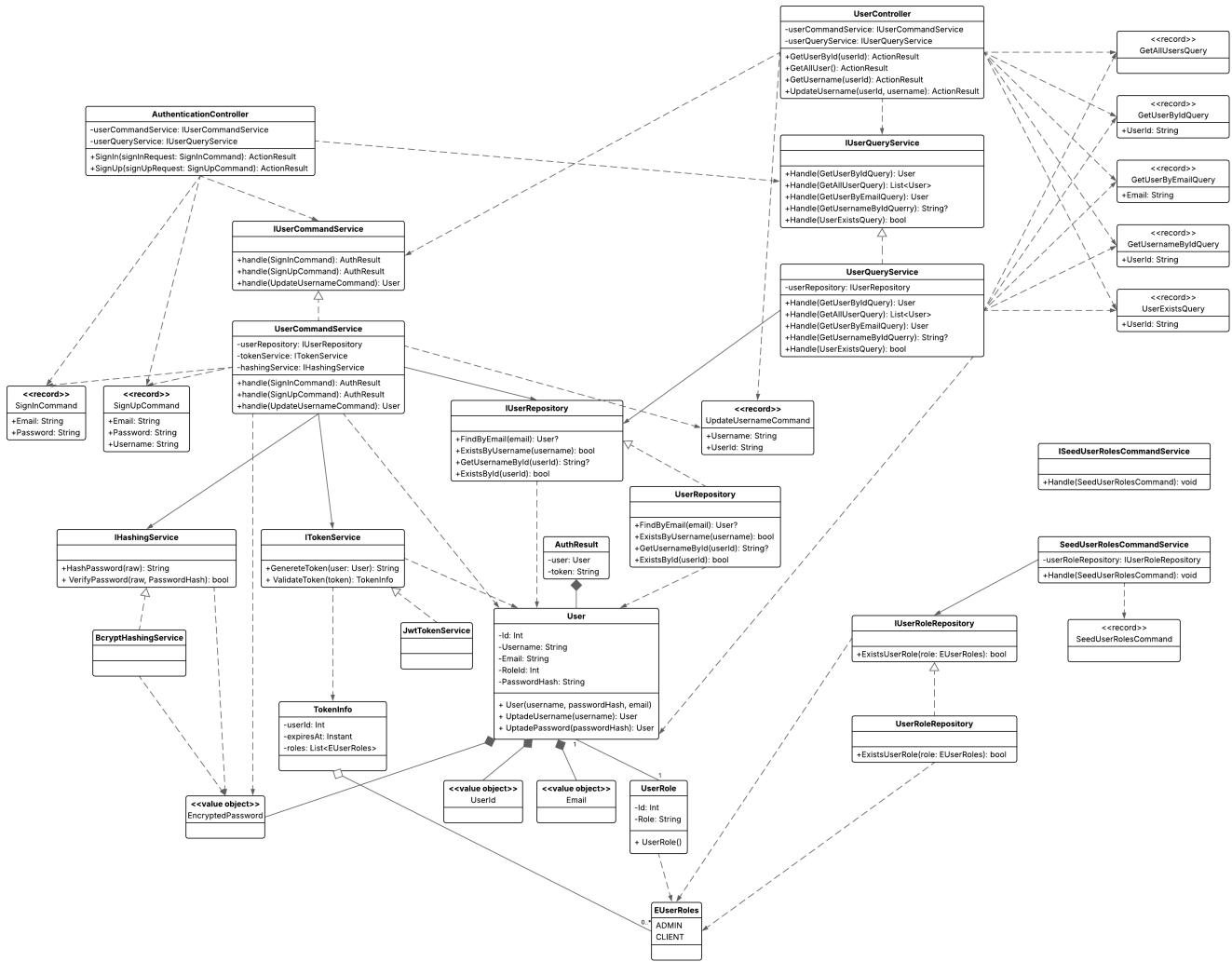




5.1.7. Bounded Context Software Architecture Code Level Diagrams

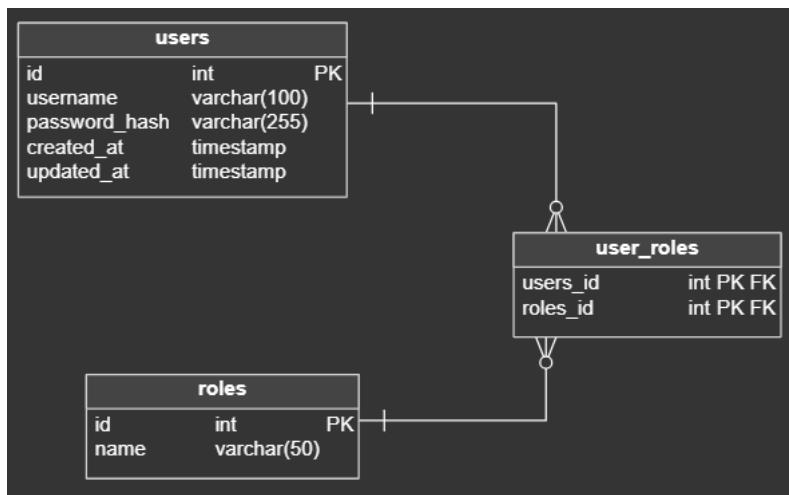
5.1.7.1. Bounded Context Domain Layer Class Diagrams

El siguiente diagrama de clases representa la capa de dominio del bounded context IAM, mostrando las relaciones entre las entidades User, Role y UserRole, así como los value objects (UserId, Email, EncryptedPassword) y las interfaces de los repositorios que gestionan la persistencia de usuarios y roles.



5.1.7.2. Bounded Context Database Design Diagram

El diagrama de base de datos del bounded context IAM muestra el diseño físico de las tablas users, roles y user_roles en PostgreSQL, incluyendo los tipos de datos, claves primarias, claves foráneas y las relaciones muchos a muchos entre usuarios y roles.



5.2. Bounded Context: Profile Context

5.2.1. Domain Layer

El **Domain Layer** del bounded context Profile encapsula las reglas de negocio relacionadas con la gestión de perfiles de usuario. Aquí se definen los agregados, entidades, value objects y repositorios que representan el núcleo del dominio.

- **Aggregate Root**

- **Profile:** representa el perfil de un usuario. Contiene información personal, documento de identidad, fecha de nacimiento, teléfono, cuentas bancarias y estado de suscripción. Expone métodos de factoría y actualización (`Profile.CreateProfileCommand`, `Update(UpdateProfileCommand)`).

- **Entities**

- **Profile** (aggregate root).

- **Value Objects**

- **PersonName:** compone `Name`, `FatherName` y `MotherName`.
- **DateOfBirth:** encapsula la fecha de nacimiento (`BirthDate`).
- **Phone:** encapsula el número de teléfono (`PhoneNumber`).
- **DocumentNumber:** representa el número de documento (`NumberDocument`).
- **BankAccount:** representa una cuenta bancaria, con `AccountNumber` e `InterbankAccountNumber`.
- **SubscriptionStatus:** indica el estado de suscripción del perfil (ej. `ACTIVE`, `INACTIVE`, `PENDING`).

- **Interfaces (Repositories)**

- **IProfileRepository:** abstracción para la persistencia de perfiles. Expone `FindByUserId(int userId)`.

El dominio asegura reglas como: cada perfil está asociado a un único `UserId`, el `DocumentNumber` es único, y cada perfil puede tener múltiples cuentas bancarias y registros de estado de suscripción.

5.2.2. Interface Layer

El **Interface Layer** expone la funcionalidad de Profile a través de controladores REST (no mostrados en el diagrama, pero implícitos en la arquitectura).

- **Responsabilidades**

- Recibir solicitudes HTTP para creación, actualización y consulta de perfiles.
- Mapear los requests a `CreateProfileCommand`, `UpdateProfileCommand` o queries específicas.
- Retornar respuestas con recursos de perfil y estados de operación.

Ejemplo de endpoints:

- `POST /profiles` → crear un perfil.
 - `PUT /profiles/{id}` → actualizar un perfil.
 - `GET /profiles/{userId}` → obtener perfil por usuario.
 - `GET /profiles/{userId}/bank-accounts` → obtener cuentas bancarias.
 - `GET /profiles/{userId}/subscription-status` → obtener estado de suscripción.
-

5.2.3. Application Layer

El **Application Layer** coordina los casos de uso del bounded context Profile mediante servicios de comandos y queries.

- **Command Handlers**

- **ProfileCommandService:** maneja la creación y actualización de perfiles.
 - `Handle(CreateProfileCommand)` → crea un nuevo perfil y lo persiste.
 - `Handle(UpdateProfileCommand)` → actualiza información personal, documento, teléfono o cuentas bancarias de un perfil.

- **Query Handlers**

- **ProfileQueryService:** maneja las consultas asociadas al perfil.
 - `Handle(GetProfileByUserIdQuery)` → retorna el perfil por `UserId`.
 - `Handle(GetSubscriptionStatusByUserIdQuery)` → retorna el estado de suscripción como VO.
 - `Handle(GetProfileBankAccountsByUserIdQuery)` → retorna la lista de cuentas bancarias.

- **DTOs (Commands y Queries)**

- **Commands:** `CreateProfileCommand`, `UpdateProfileCommand`.
 - **Queries:** `GetProfileByUserIdQuery`, `GetSubscriptionStatusByUserIdQuery`, `GetProfileBankAccountsByUserIdQuery`.
-

5.2.4. Infrastructure Layer

El **Infrastructure Layer** contiene las implementaciones concretas y la interacción con la base de datos.

- **Repositorios**

- **ProfileRepository:** implementación de `IProfileRepository` sobre PostgreSQL. Permite persistir y recuperar perfiles.

- **Base de datos (PostgreSQL)**

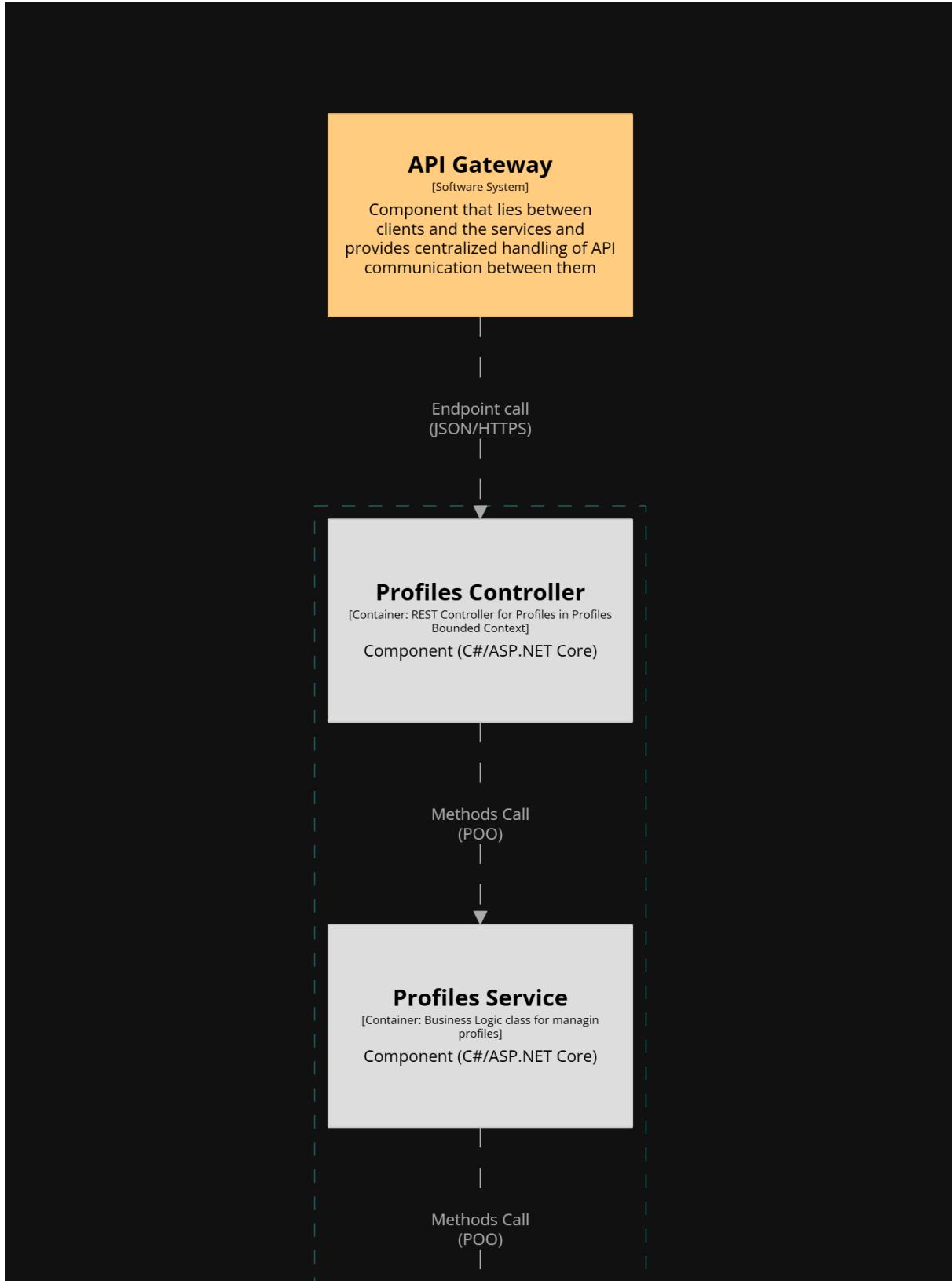
- **profiles:** almacena los datos principales del perfil (nombre, documento, fecha de nacimiento, teléfono).
 - **bank_accounts:** almacena las cuentas bancarias asociadas a cada perfil.
 - **subscription_statuses:** guarda el estado de suscripción asociado a cada perfil.
-

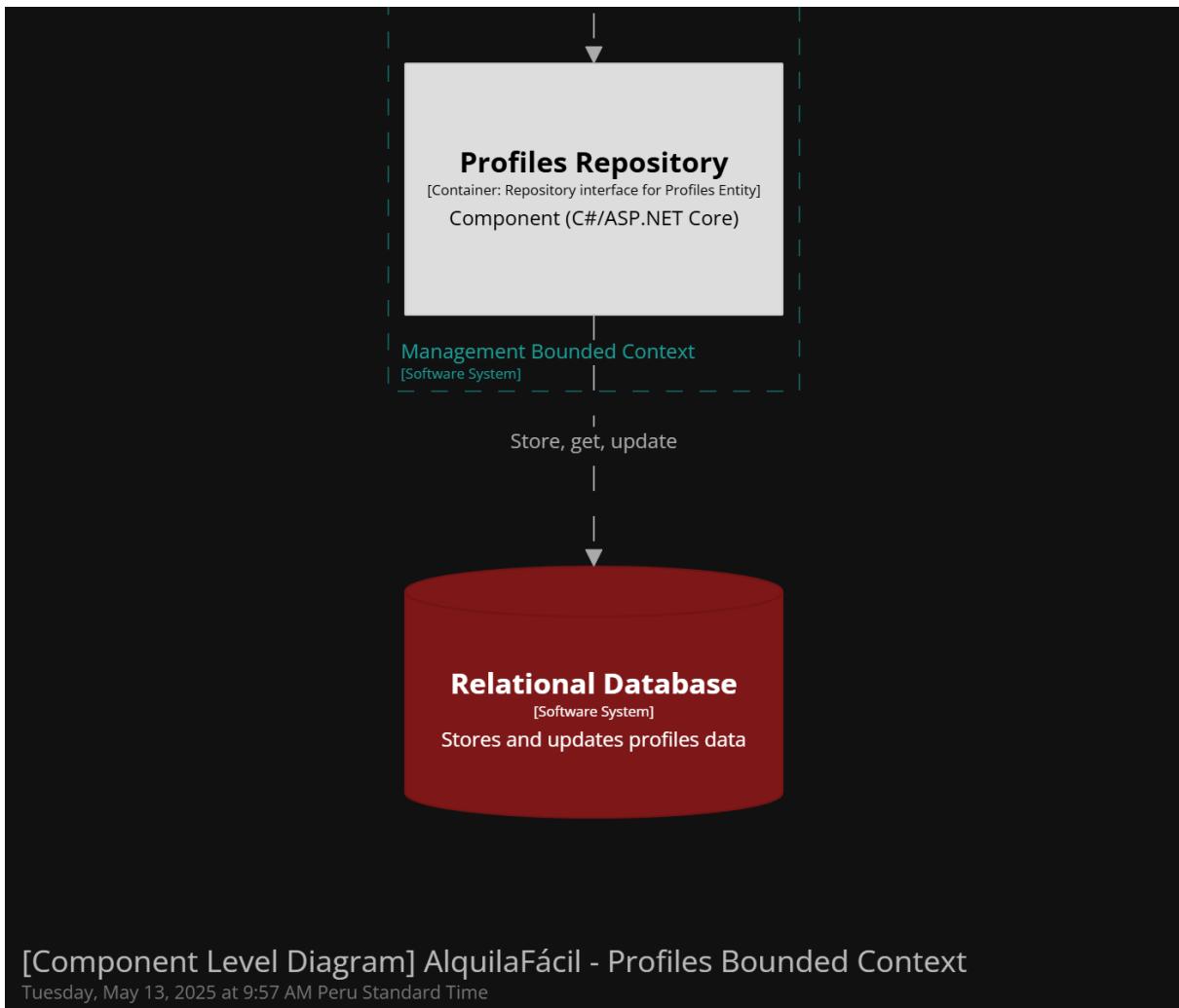
5.2.6. Bounded Context Software Architecture Component Level Diagrams

El bounded context Profile se despliega como un contenedor **Profile API** en el modelo C4. Sus componentes principales son:

- **Controllers:** exponen endpoints para crear, actualizar y consultar perfiles.
- **Application Services:** `ProfileCommandService`, `ProfileQueryService`.
- **Domain:** entidades (`Profile`) y value objects (`PersonName`, `DateOfBirth`, `Phone`, `DocumentNumber`, `BankAccount`, `SubscriptionStatus`).
- **Infrastructure Adapters:** `ProfileRepository` como adaptador de persistencia.
- **Base de datos PostgreSQL:** almacenamiento de perfiles, cuentas bancarias y estados de suscripción.

El flujo: Controllers → Application Services → Domain (Aggregate Profile + VO) → Repository → DB.

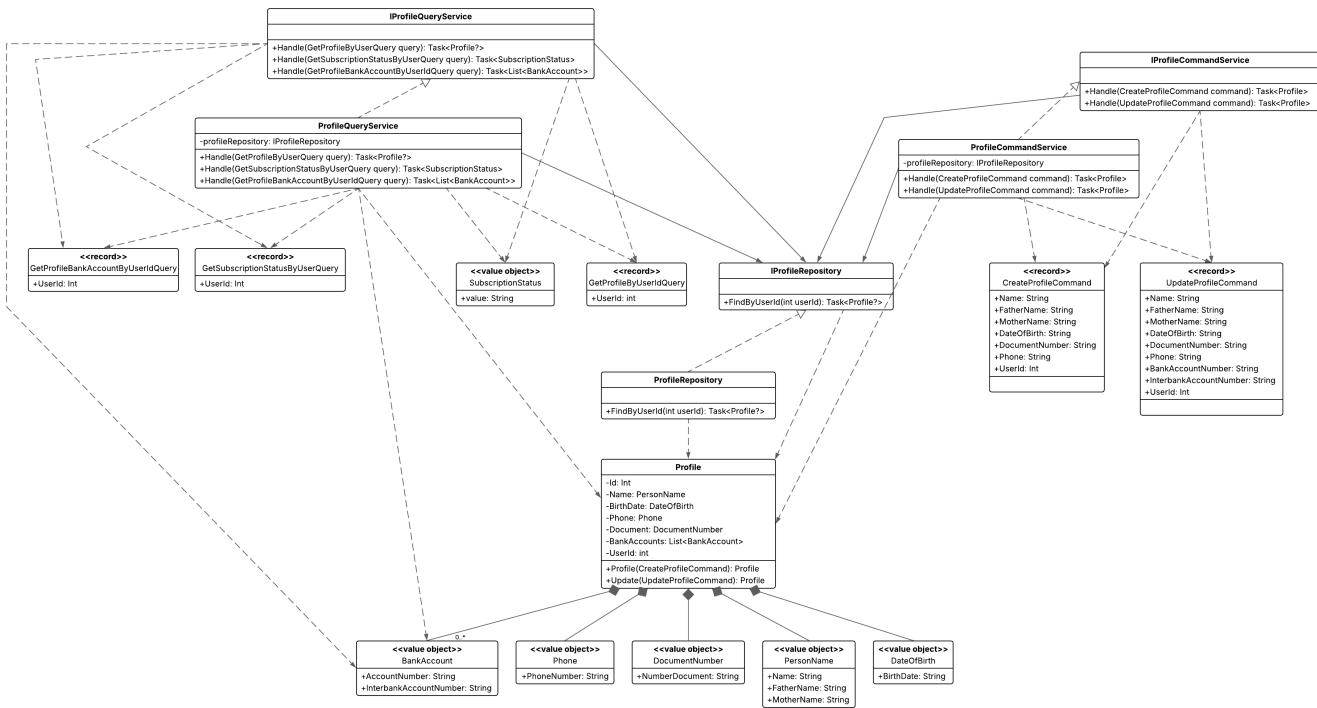




5.2.7. Bounded Context Software Architecture Code Level Diagrams

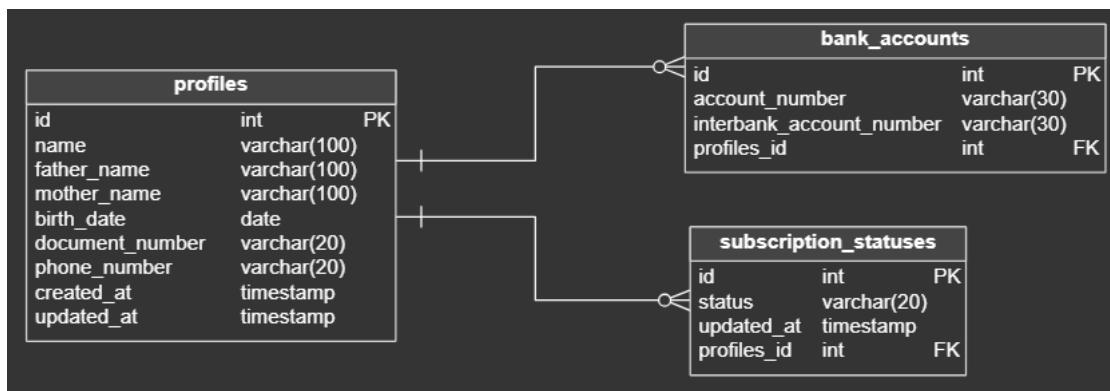
5.2.7.1. Bounded Context Domain Layer Class Diagrams

El diagrama de clases del bounded context Profile ilustra el aggregate root Profile y sus value objects asociados (PersonName, DateOfBirth, Phone, DocumentNumber, BankAccount, SubscriptionStatus), así como la interfaz del repositorio IProfileRepository que define los contratos de persistencia.



5.2.7.2. Bounded Context Database Design Diagram

El diagrama de base de datos del bounded context Profile presenta el diseño de las tablas profiles, bank_accounts y subscription_statuses, mostrando sus atributos, tipos de datos, claves y las relaciones uno a muchos entre el perfil y sus cuentas bancarias y estados de suscripción.



5.3. Bounded Context: Locals Context

5.3.1. Domain Layer

Aggregates

1. Local

- **Descripción:** Representa un espacio físico disponible para ser arrendado, incluyendo características como ubicación, precio, tipo y capacidad.
- **Atributos:**
 - **Id:** Identificador único del local.
 - **Features:** Características especiales del local.
 - **Capacity:** Capacidad máxima de personas.
 - **LocalName:** Tipo o nombre del local (objeto de valor **LocalName**).
 - **NightPrice:** Precio por noche (objeto de valor **NightPrice**).
 - **PhotoUrl:** URL de la foto principal del local (objeto de valor **PhotoUrl**).
 - **StreetAddress:** Dirección completa (objeto de valor **StreetAddress**).
 - **CityPlace:** Ciudad y país (objeto de valor **CityPlace**).
 - **DescriptionMessage:** Descripción del local (objeto de valor **DescriptionMessage**).
 - **LocalCategoryId:** Identificador de categoría del local.
 - **UserId:** Identificador del propietario del local.

2. Comment

- **Descripción:** Representa un comentario realizado por un usuario sobre un local, incluyendo la valoración y el texto del comentario.

- **Atributos:**

- ***Id*:** Identificador único del comentario.
- ***UserId*:** ID del usuario que realiza el comentario.
- ***LocalId*:** ID del local sobre el que se comenta.
- ***Text*:** Texto del comentario (objeto de valor **TextComment**).
- ***Rating*:** Valoración del local (objeto de valor **RatingComment**).
- ***CommentText*:** Texto plano del comentario.
- ***CommentRating*:** Valor numérico de la calificación.

3. Report

- **Descripción:** Representa un reporte o queja generado sobre un local, realizado por un usuario.

- **Atributos:**

- ***Id*:** Identificador único del reporte.
- ***LocalId*:** ID del local reportado.
- ***Title*:** Título del reporte.
- ***UserId*:** ID del usuario que genera el reporte.
- ***Description*:** Descripción del problema o situación reportada.
- ***CreatedAt*:** Fecha de creación del reporte.

Entities

1. LocalCategory

- **Descripción:** Representa una categoría de locales, como auditorio, salón de eventos, oficina, etc.

- **Atributos:**

- ***Id*:** Identificador único de la categoría.
- ***Name*:** Nombre de la categoría (por ejemplo, "Auditorio", "Coworking", "Salón de Eventos").
- ***PhotoUrl*:** URL de una imagen representativa de la categoría.

Value Objects

1. CityPlace

- **Descripción:** Representa el país y la ciudad donde se ubica un local.

- **Atributos:**

- ***Country*:** País del local.
- ***City*:** Ciudad del local.
- ***FullCityPlace*:** Representación completa como texto ("País, Ciudad").

2. DescriptionMessage

- **Descripción:** Representa el mensaje descriptivo o detalle del local.

- **Atributos:**

- ***MessageDescription*:** Texto descriptivo del local.

3. LocalName

- **Descripción:** Representa el tipo o nombre asignado a un local.

- **Atributos:**

- ***TypeLocal*:** Nombre o tipo del local (ej: "Auditorio", "Sala de Reuniones").

4. NightPrice

- **Descripción:** Representa el precio por noche de alquiler del local.

- **Atributos:**

- ***PriceNight*:** Valor numérico del precio.

5. PhotoUrl

- **Descripción:** Representa la URL de la foto principal del local.

- **Atributos:**

- ***PhotoUrlLink*:** Enlace de la imagen.

6. RatingComment

- **Descripción:** Representa la calificación numérica otorgada a un local en un comentario.

- **Atributos:**

- ***Rating*:** Puntuación del comentario (ej: de 1 a 5 estrellas).

7. StreetAddress

- **Descripción:** Representa la dirección geográfica del local.

- **Atributos:**

- **District**: Distrito donde se ubica el local.
- **Street**: Calle específica del local.
- **FullAddress**: Representación completa como texto ("Distrito, Calle").

8. TextComment

- **Descripción:** Representa el texto escrito de un comentario sobre un local.
- **Atributos:**
 - **Text**: Contenido textual del comentario.

Commands

1. CreateLocalCommand

- **Descripción:** Permite registrar un nuevo local con todos sus datos estructurados.
- **Parámetros:**
 - **District**: Distrito donde se ubica el local.
 - **Street**: Calle del local.
 - **LocalType**: Tipo o categoría del local.
 - **Country**: País.
 - **City**: Ciudad.
 - **Price**: Precio por noche.
 - **PhotoUrl**: Enlace de la imagen principal.
 - **DescriptionMessage**: Descripción del local.
 - **LocalCategoryId**: ID de la categoría.
 - **UserId**: ID del arrendador que registra el local.
 - **Features**: Características adicionales del local.
 - **Capacity**: Capacidad máxima de personas.

2. UpdateLocalCommand

- **Descripción:** Actualiza los atributos de un local existente.
- **Parámetros:**
 - **Id**: ID del local a actualizar.
 - (Resto de parámetros iguales a [CreateLocalCommand](#))

3. CreateCommentCommand

- **Descripción:** Registra un nuevo comentario para un local.
- **Parámetros:**
 - **UserId**: ID del usuario que comenta.
 - **LocalId**: ID del local.
 - **Text**: Texto del comentario.
 - **Rating**: Puntuación asignada al local.

4. CreateReportCommand

- **Descripción:** Permite crear un reporte operativo o queja sobre un local.
- **Parámetros:**
 - **LocalId**: ID del local reportado.
 - **Title**: Título del reporte.
 - **UserId**: ID del usuario que genera el reporte.
 - **Description**: Contenido del reporte.

5. DeleteReportCommand

- **Descripción:** Elimina un reporte existente en base a su ID.
- **Parámetros:**
 - **Id**: ID del reporte a eliminar.

6. SeedLocalCategoriesCommand

- **Descripción:** Permite poblar la base de datos con un conjunto inicial de categorías de locales.
- **Parámetros:** (*Ninguno*)

Queries

1. GetAllCommentsByLocalIdQuery

- **Descripción:** Obtiene todos los comentarios asociados a un local específico.
- **Parámetros:**
 - **LocalId**: ID del local.

2. GetAllLocalCategoriesQuery

- **Descripción:** Obtiene todas las categorías de locales disponibles.
- **Parámetros:** (Ninguno)

3. GetAllLocalDistrictsQuery

- **Descripción:** Obtiene todos los distritos donde existen locales registrados.
- **Parámetros:** (Ninguno)

4. GetAllLocalsByLocalCategoryIdQuery

- **Descripción:** Obtiene todos los locales filtrados por una categoría específica.
- **Parámetros:**
 - `LocalCategoryId`: ID de la categoría.

5. GetAllLocalsQuery

- **Descripción:** Obtiene todos los locales registrados en el sistema.
- **Parámetros:** (Ninguno)

6. GetLocalByIdQuery

- **Descripción:** Consulta los detalles de un local específico por su ID.
- **Parámetros:**
 - `LocalId`: ID del local.

7. GetLocalCategoryByIdQuery

- **Descripción:** Consulta los detalles de una categoría específica de locales.
- **Parámetros:**
 - `Id`: ID de la categoría.

8. GetLocalsByCategoryIdAndCapacityRangeQuery

- **Descripción:** Obtiene locales que pertenecen a una categoría específica y cuyo rango de capacidad está dentro de los límites indicados.
- **Parámetros:**
 - `LocalCategoryId`: ID de la categoría.
 - `MinCapacity`: Capacidad mínima.
 - `MaxCapacity`: Capacidad máxima.

9. GetLocalsByUserIdQuery

- **Descripción:** Obtiene todos los locales registrados por un usuario específico (arrendador).
- **Parámetros:**
 - `UserId`: ID del usuario.

10. GetReportsByLocalIdQuery

- **Descripción:** Obtiene todos los reportes asociados a un local específico.
- **Parámetros:**
 - `LocalId`: ID del local.

11. GetReportsByUserIdQuery

- **Descripción:** Obtiene todos los reportes generados por un usuario específico.
- **Parámetros:**
 - `UserId`: ID del usuario.

12. IsLocalOwnerQuery

- **Descripción:** Verifica si un usuario específico es propietario de un local.
- **Parámetros:**
 - `UserId`: ID del usuario.
 - `LocalId`: ID del local.

Repositories (Interfaces)

1. ICommentRepository

- **Descripción:** Interfaz para gestionar los comentarios asociados a locales.
- **Métodos:**
 - `GetAllCommentsByLocalId(int localId)`: Obtiene todos los comentarios de un local específico.

2. ILocalCategoryRepository

- **Descripción:** Interfaz para gestionar las categorías de locales.
- **Métodos:**
 - `ExistsLocalCategory(EALocalCategoryTypes type)`: Verifica si existe una categoría específica.

- `GetAllLocalCategories()`: Obtiene todas las categorías de locales.

3. ILocalRepository

- **Descripción:** Interfaz para la gestión de locales registrados en el sistema.
- **Métodos:**
 - `GetAllDistrictsAsync()`: Obtiene un listado de todos los distritos donde hay locales.
 - `GetLocalsByCategoryIdAndCapacityRange(int categoryId, int minCapacity, int maxCapacity)`: Filtra locales por categoría y rango de capacidad.
 - `GetLocalsByUserIdAsync(int userId)`: Obtiene los locales de un usuario específico (arrendador).
 - `IsOwnerAsync(int userId, int localId)`: Verifica si un usuario es dueño de un local.

4. IReportRepository

- **Descripción:** Interfaz para la gestión de reportes generados sobre locales.
- **Métodos:**
 - `GetReportsByLocalId(int localId)`: Obtiene los reportes asociados a un local.
 - `GetReportsByUserId(int userId)`: Obtiene los reportes generados por un usuario específico.

5.3.2. Interface Layer

Controllers

1. CommentController

- **Descripción:** Maneja operaciones relacionadas con los comentarios de locales.
- **Métodos:**
 - `GetAllCommentsByLocalId(int localId)`: Obtiene todos los comentarios realizados sobre un local específico.
 - `CreateComment(CreateCommentResource resource)`: Registra un nuevo comentario para un local.

2. LocalCategoriesController

- **Descripción:** Expone operaciones para consultar las categorías de locales disponibles.
- **Métodos:**
 - `GetAllLocalCategories()`: Obtiene todas las categorías de locales.

3. LocalsController

- **Descripción:** Maneja las operaciones CRUD principales sobre locales.
- **Métodos:**
 - `CreateLocal(CreateLocalResource resource)`: Registra un nuevo local.
 - `GetAllLocals()`: Obtiene todos los locales registrados en el sistema.
 - `GetLocalById(int localId)`: Consulta los detalles de un local específico.
 - `UpdateLocal(int localId, UpdateLocalResource resource)`: Actualiza la información de un local existente.
 - `SearchByCategoryIdAndCapacityRange(int categoryId, int minCapacity, int maxCapacity)`: Filtra locales por categoría y rango de capacidad.
 - `GetAllDistricts()`: Obtiene todos los distritos donde hay locales registrados.
 - `GetUserLocals(int userId)`: Obtiene todos los locales registrados por un arrendador específico.

4. ReportController

- **Descripción:** Maneja operaciones relacionadas con la creación, consulta y eliminación de reportes de locales.
- **Métodos:**
 - `CreateReport(CreateReportResource createReportResource)`: Registra un nuevo reporte de estado o queja sobre un local.
 - `GetReportsByUserId(int userId)`: Obtiene todos los reportes generados por un usuario.
 - `GetReportsByLocalId(int localId)`: Obtiene todos los reportes asociados a un local específico.
 - `DeleteReport(int reportId)`: Elimina un reporte existente por su ID.

5.3.3. Application Layer

Command Services

1. CommentCommandService

- **Descripción:** Maneja la lógica de negocio para registrar comentarios en locales.
- **Dependencias:** `ICommentRepository`, `ILocalRepository`, `IUserCommentExternalService`, `IUnitOfWork`.
- **Métodos:**
 - `Handle(CreateCommentCommand command)`: Valida existencia del local y usuario, crea el comentario y lo persiste.

2. LocalCommandService

- **Descripción:** Maneja la creación y actualización de locales por parte de los arrendadores.
- **Dependencias:** `ILocalRepository`, `ILocalCategoryRepository`, `IUnitOfWork`.
- **Métodos:**
 - `Handle(CreateLocalCommand command)`: Crea un nuevo local validando categoría y precio.

- `Handle(UpdateLocalCommand command)`: Actualiza un local existente validando entidad y datos.

3. LocalCategoryCommandService

- **Descripción:** Inicializa las categorías de locales disponibles en el sistema.
- **Dependencias:** `ILocalCategoryRepository`, `IUnitOfWork`.
- **Métodos:**
 - `Handle(SeedLocalCategoriesCommand command)`: Registra categorías predefinidas si no existen.

4. ReportCommandService

- **Descripción:** Maneja la creación y eliminación de reportes asociados a locales.
- **Dependencias:** `IReportRepository`, `IUnitOfWork`.
- **Métodos:**
 - `Handle(CreateReportCommand command)`: Crea y guarda un nuevo reporte.
 - `Handle>DeleteReportCommand command)`: Elimina un reporte existente validando su existencia.

Query Services

1. CommentQueryService

- **Descripción:** Permite consultar los comentarios hechos sobre un local.
- **Dependencias:** `ICommentRepository`.
- **Métodos:**
 - `Handle(GetAllCommentsByLocalIdQuery query)`: Devuelve todos los comentarios asociados a un local.

2. LocalQueryService

- **Descripción:** Permite consultar información de locales y sus filtros asociados.
- **Dependencias:** `ILocalRepository`.
- **Métodos:**
 - `Handle GetAllLocalsQuery query)`: Lista todos los locales del sistema.
 - `Handle GetLocalByIdQuery query)`: Devuelve los detalles de un local específico.
 - `Handle GetAllLocalDistrictsQuery query)`: Devuelve todos los distritos donde hay locales registrados.
 - `Handle GetLocalsByUserIdQuery query)`: Lista los locales registrados por un arrendador.
 - `Handle GetLocalsByCategoryIdAndCapacityRangeQuery query)`: Devuelve locales filtrados por categoría y rango de capacidad.
 - `Handle IsLocalOwnerQuery query)`: Verifica si un usuario es dueño del local.

3. LocalCategoryQueryService

- **Descripción:** Permite consultar la lista de categorías de locales.
- **Dependencias:** `ILocalCategoryRepository`.
- **Métodos:**
 - `Handle GetAllLocalCategoriesQuery query)`: Retorna todas las categorías existentes.

4. ReportQueryService

- **Descripción:** Permite obtener reportes asociados a locales o usuarios.
- **Dependencias:** `IReportRepository`.
- **Métodos:**
 - `Handle GetReportsByLocalIdQuery query)`: Lista los reportes generados para un local.
 - `Handle GetReportsByUserIdQuery query)`: Lista los reportes generados por un usuario.

5.3.4. Infrastructure Layer

Repositories (Implementaciones)

1. CommentRepository

- **Descripción:** Implementación del repositorio para gestionar comentarios sobre locales.
- **Hereda de:** `BaseRepository<Comment>`
- **Métodos implementados:**
 - `GetAllCommentsByLocalId(int localId)`: Devuelve todos los comentarios asociados a un local específico.

2. LocalCategoryRepository

- **Descripción:** Implementación del repositorio para gestionar categorías de locales.
- **Hereda de:** `BaseRepository<LocalCategory>`
- **Métodos implementados:**
 - `ExistsLocalCategory(EALocalCategoryTypes type)`: Verifica si existe una categoría específica.
 - `GetAllLocalCategories()`: Devuelve todas las categorías de locales disponibles.

3. LocalRepository

- **Descripción:** Implementación del repositorio para gestionar entidades `Local`.

- **Hereda de:** BaseRepository<Local>

- **Métodos implementados:**

- `GetAllDistrictsAsync()`: Obtiene todos los distritos registrados donde hay locales, combinando ubicación con dirección.
- `GetLocalsByCategoryIdAndCapacityrange(int categoryId, int minCapacity, int maxCapacity)`: Devuelve locales filtrados por categoría y rango de capacidad.
- `GetLocalsByUserIdAsync(int userId)`: Obtiene todos los locales de un usuario específico.
- `IsOwnerAsync(int userId, int localId)`: Verifica si un usuario es dueño de un local determinado.
- `GetLocalByUserId(int userId, int localId)`: Obtiene un local específico por ID si pertenece al usuario.

4. ReportRepository

- **Descripción:** Implementación del repositorio para gestionar reportes relacionados a locales.

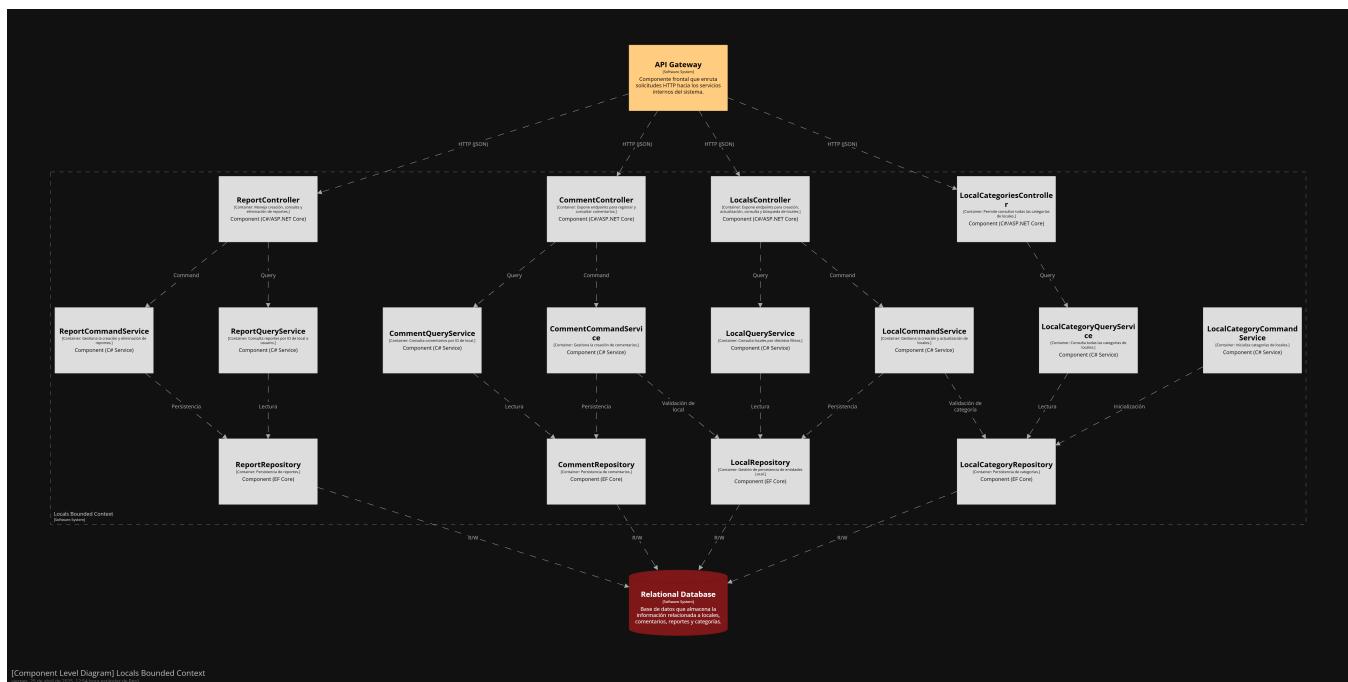
- **Hereda de:** BaseRepository<Report>

- **Métodos implementados:**

- `GetReportsByLocalId(int localId)`: Devuelve todos los reportes asociados a un local.
- `GetReportsByUserId(int userId)`: Devuelve todos los reportes realizados por un usuario específico.

5.3.6. Bounded Context Software Architecture Component Level Diagrams

El diagrama de componentes C4 del bounded context Locals muestra la arquitectura interna del contenedor Locals API, incluyendo los controladores, servicios de aplicación (comandos y consultas), repositorios y su interacción con la base de datos relacional para gestionar locales, comentarios, reportes y categorías.



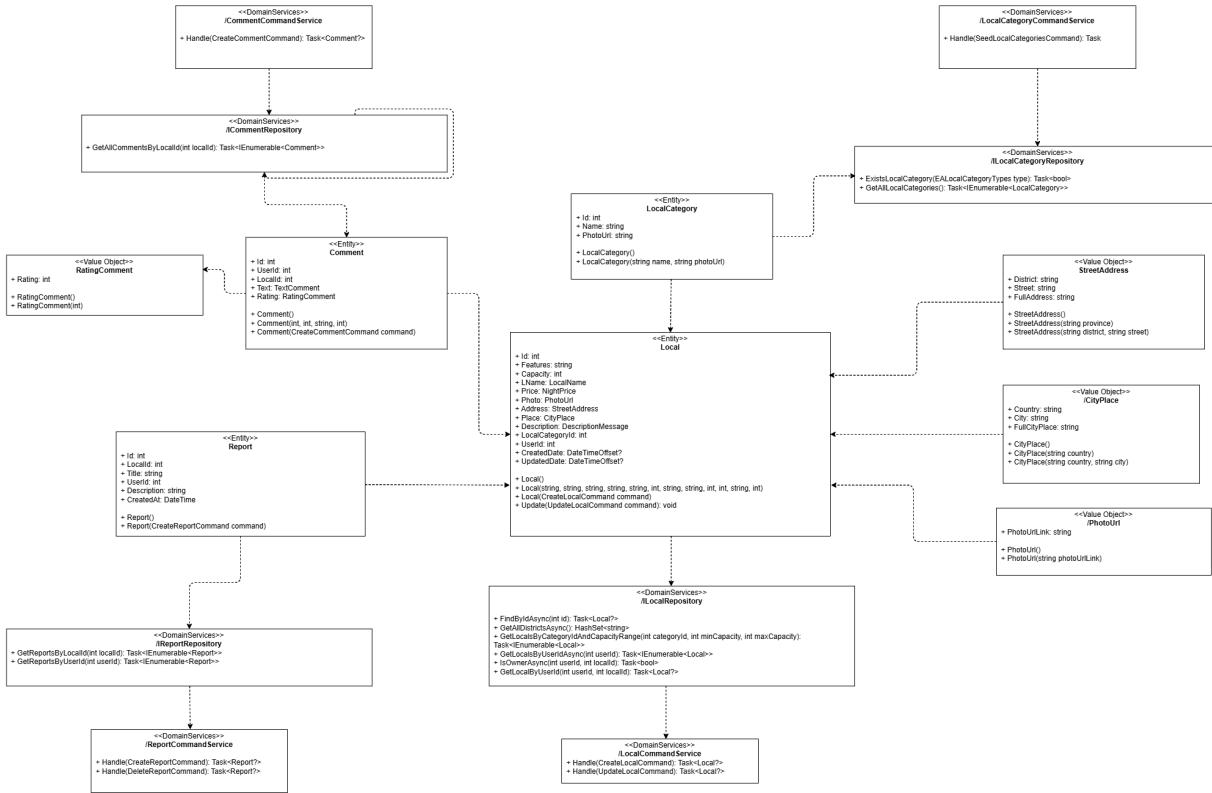
El bounded context **Locals** se despliega como un contenedor **Locals API** en el modelo C4. Sus componentes principales son:

- **Controllers:** exponen endpoints HTTP para gestionar locales, comentarios, reportes y categorías de locales a través de **LocalsController**, **CommentController**, **ReportController** y **LocalCategoriesController**.
- **Application Services:**
 - **Command Services:** **LocalCommandService**, **CommentCommandService**, **ReportCommandService** y **LocalCategoryCommandService** para operaciones de escritura.
 - **Query Services:** **LocalQueryService**, **CommentQueryService**, **ReportQueryService** y **LocalCategoryQueryService** para operaciones de lectura.
- **Infrastructure Adapters:** **LocalRepository**, **CommentRepository**, **ReportRepository** y **LocalCategoryRepository** como adaptadores de persistencia que implementan las interfaces del dominio.
- **Base de datos Relacional (Relational Database):** almacenamiento centralizado de locales, comentarios, reportes y categorías, gestionando la información relacionada con espacios arrendables y sus interacciones con usuarios.

5.3.7. Bounded Context Software Architecture Code Level Diagrams

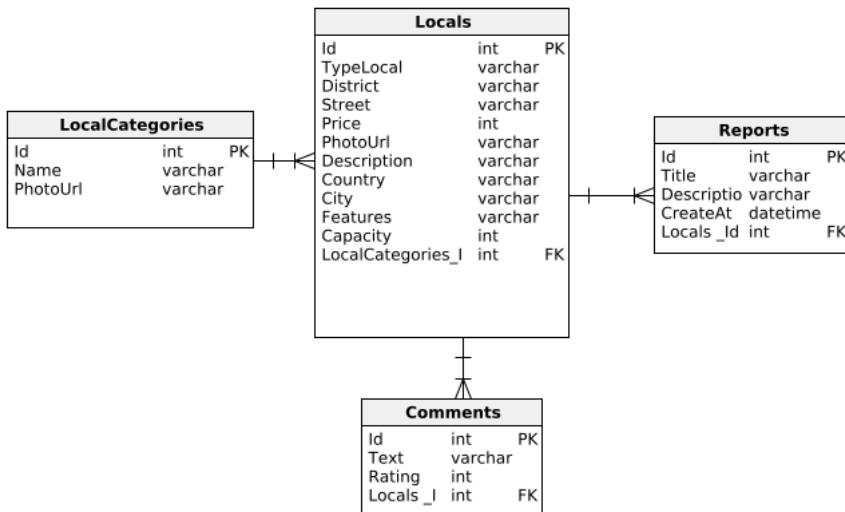
5.3.7.1. Bounded Context Domain Layer Class Diagrams

El diagrama de clases del bounded context Locals representa las entidades principales del dominio: Local (aggregate root), Comment, Report y LocalCategory, junto con sus value objects (Location, Address, TimeRange, Price) y las interfaces de los repositorios correspondientes.



5.3.7.2. Bounded Context Database Design Diagram

El diagrama de base de datos del bounded context Locals muestra el diseño de las tablas locales, comentarios, reportes y categorías de locales, incluyendo sus relaciones, claves primarias, claves foráneas y los tipos de datos de cada atributo.



5.4 Bounded Context: Booking Context

5.4.1. Domain Layer

Aggregates

1. Reservation

- **Id:** Identificador único de la reserva.
- **StartDate:** Fecha y hora de inicio de la reserva
- **EndDate:** Fecha y hora del fin de la reserva.

- **UserId**: Identificador único del usuario.
- **LocalId**: Identificador único del local.

Commands

1. CreateReservationCommand:

- **Descripción:** Crear una nueva reserva con la fecha y hora de inicio, fecha y hora de fin, userId y LocalId.

2. DeleteReservationCommand:

- **Descripción:** Elimina una reserva con el Id de la misma.

3. UpdateReservationDateCommand:

- **Descripción:** Permite cambiar la fecha y hora de inicio y la fecha y hora de fin.

Queries

1. GetReservationByEndDate:

- **Descripción:** Obtener todas las reservas que finalizan en una fecha específica.

2. GetReservationByStartDate:

- **Descripción:** Obtener todas las reservas que inician en una fecha específica.

3. GetReservationByLocalIdQuery:

- **Descripción:** Obtener todas las reservas asociadas a un local específico.

4. GetReservationByOwnerIdQuery:

- **Descripción:** Obtener todas las reservas realizadas en locales pertenecientes a un propietario específico.

5. GetReservationByUserId:

- **Descripción:** Obtener todas las reservas realizadas por un usuario específico.

Repositories (interfaces)

1. ReservationRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de reservas.
- **Métodos:**
 - **GetReservationsByUserIdAsync(int userId)**: Obtiene todas las reservas realizadas por un usuario específico.
 - **GetReservationByStartDateAsync(DateTime startDate)**: Obtiene todas las reservas que inician en una fecha específica.
 - **GetReservationByEndDateAsync(DateTime endDate)**: Obtiene todas las reservas que finalizan en una fecha específica.
 - **GetReservationsByLocalIdAsync(List<int> localId)**: Obtiene todas las reservas asociadas a uno o más locales específicos.

5.4.2. Interface Layer

Controllers

1. ReservationController:

- **Descripción:** Expone endpoints para la gestión de reservas, incluyendo su creación, actualización, eliminación y consultas por distintos criterios.
- **Métodos:**
 - **CreateReservationAsync(CreateReservationResource resource)**: Endpoint para crear una nueva reserva.
 - **UpdateReservationAsync(int id, UpdateReservationResource resource)**: Endpoint para actualizar las fechas de una reserva existente.
 - **DeleteReservationAsync(int id)**: Endpoint para eliminar una reserva por su ID.
 - **GetReservationsByUserIdAsync(int userId)**: Endpoint para obtener todas las reservas hechas por un usuario específico.
 - **GetReservationUserDetailsAsync(int userId)**: Endpoint que obtiene reservas y detalles de suscriptores asociadas al usuario dueño del local.
 - **GetReservationByStartDateAsync(DateTime startDate)**: Endpoint para obtener todas las reservas que comienzan en una fecha específica.
 - **GetReservationByEndDateAsync(DateTime endDate)**: Endpoint para obtener todas las reservas que terminan en una fecha específica.

5.4.3. Application Layer

Command Services

1. ReservationCommandService:

- **Descripción:** Servicio encargado de manejar los comandos relacionados a la creación, actualización y eliminación de reservas, aplicando reglas de negocio y validaciones necesarias.
- **Métodos:**

- `Handle(CreateReservationCommand reservation)`: Crea una nueva reserva validando existencia de usuario, local, fechas correctas y propiedad del local.
- `Handle(UpdateReservationDateCommand reservation)`: Actualiza las fechas de una reserva existente tras validar que las fechas sean válidas y futuras.
- `Handle>DeleteReservationCommand reservation`): Elimina una reserva por su ID si esta existe.

Query Services

2. ReservationQueryService:

- **Descripción:** Servicio encargado de consultar reservas desde la base de datos, permitiendo filtros por usuario, fechas o propiedad del local.
- **Métodos:**
 - `GetReservationsByUserIdAsync(GetReservationsByUserId query)`: Obtiene todas las reservas asociadas a un usuario dado su ID.
 - `GetReservationByStartDateAsync(GetReservationByStartDate query)`: Obtiene todas las reservas que comienzan en una fecha específica.
 - `GetReservationByEndDateAsync(GetReservationByEndDate query)`: Obtiene todas las reservas que finalizan en una fecha específica.
 - `GetReservationsByOwnerIdAsync(GetReservationsByOwnerIdQuery query)`: Obtiene todas las reservas de los locales propiedad de un usuario específico.

5.4.4. Application Layer

Repositories (implementaciones)

ReservationRepository

Descripción:

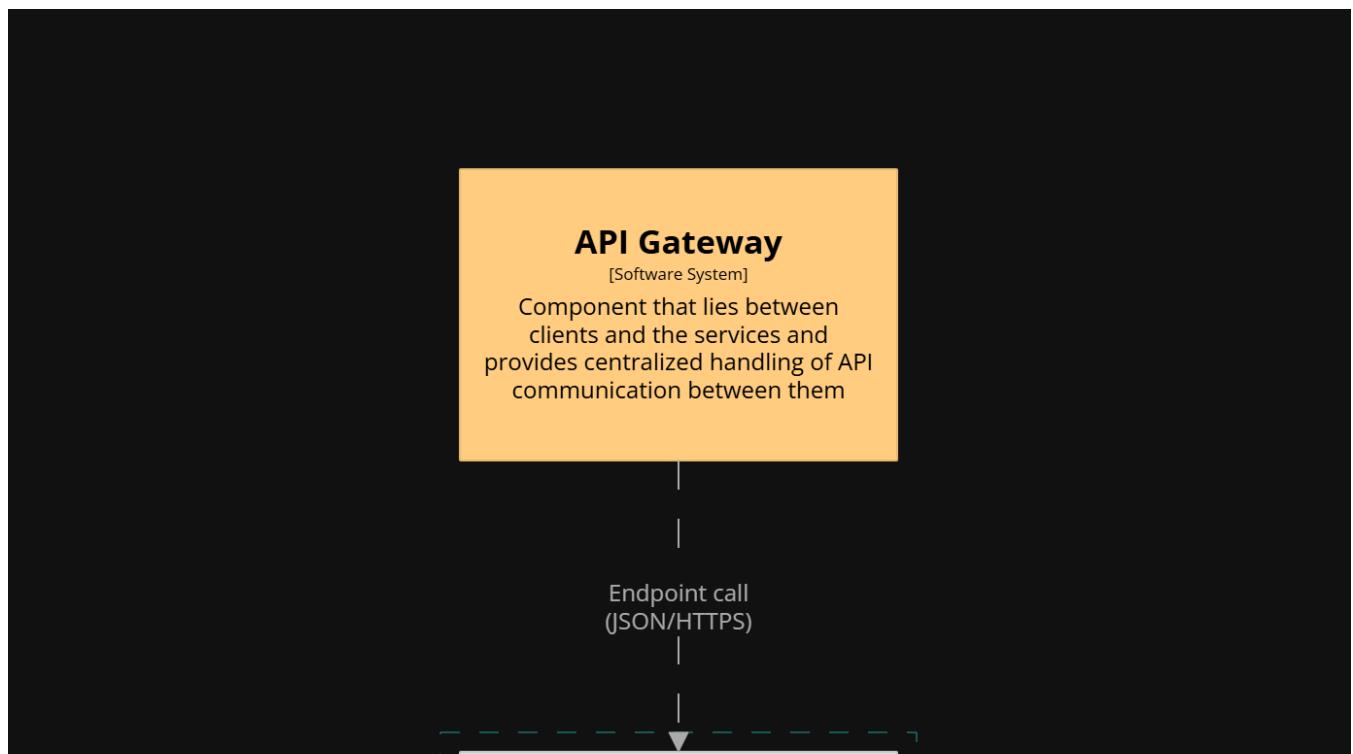
Implementación del repositorio para interactuar con la base de datos de reservas, utilizando Entity Framework Core.

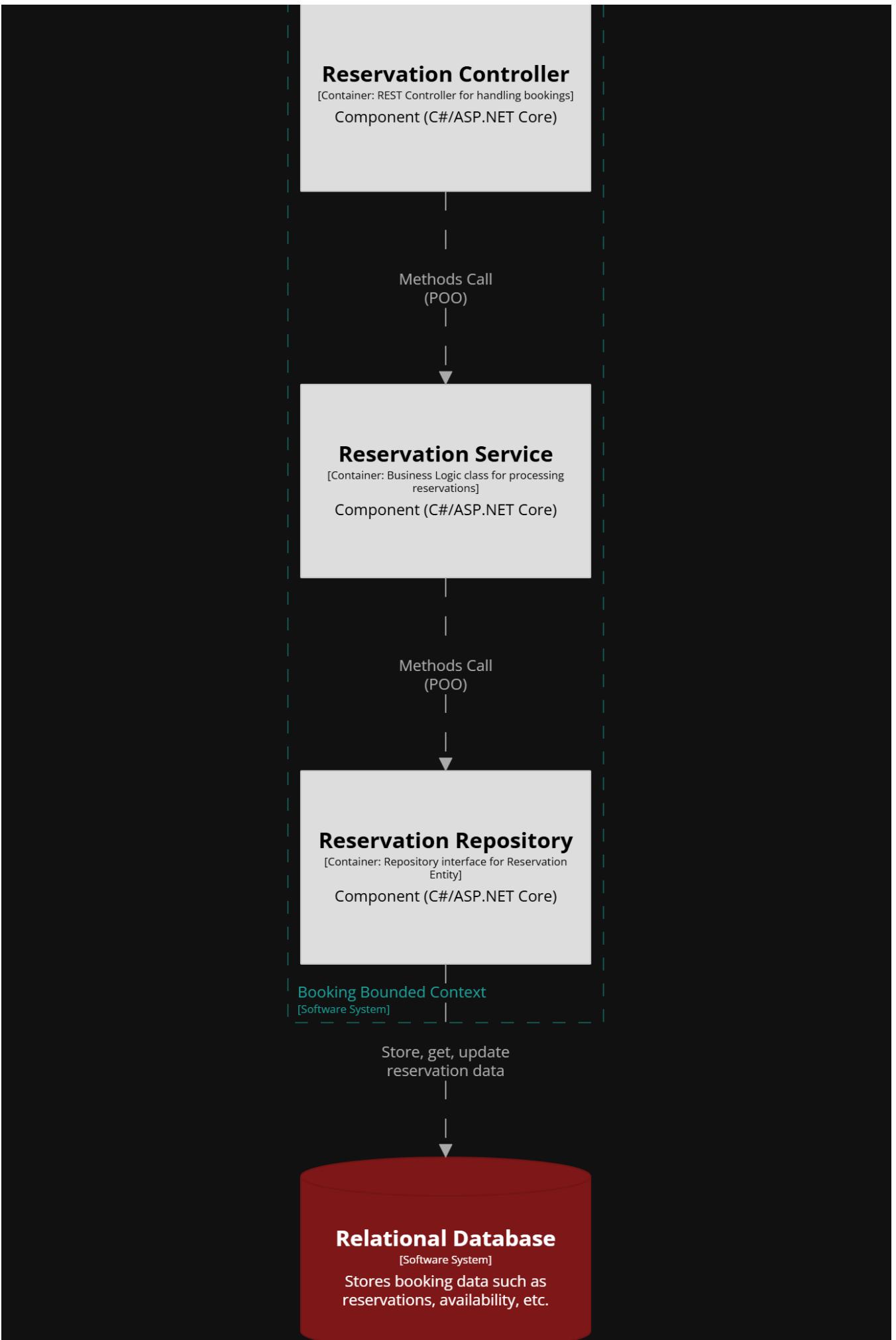
Métodos:

- `GetReservationsByUserIdAsync(int userId)`
Obtiene todas las reservas realizadas por un usuario específico.
- `GetReservationByStartDateAsync(DateTime startDate)`
Obtiene todas las reservas que comienzan en una fecha específica.
- `GetReservationByEndDateAsync(DateTime endDate)`
Obtiene todas las reservas que terminan en una fecha específica.
- `GetReservationsByLocalIdAsync(List<int> localId)`
Obtiene todas las reservas asociadas a una lista de IDs de locales.

5.4.6. Bounded Context Software Architecture Component Level Diagrams

El diagrama de componentes C4 del bounded context Booking muestra la arquitectura del contenedor Booking API, incluyendo el controlador de reservas, los servicios de comandos y consultas, el repositorio de reservas y su conexión con la base de datos para gestionar las reservaciones de locales.





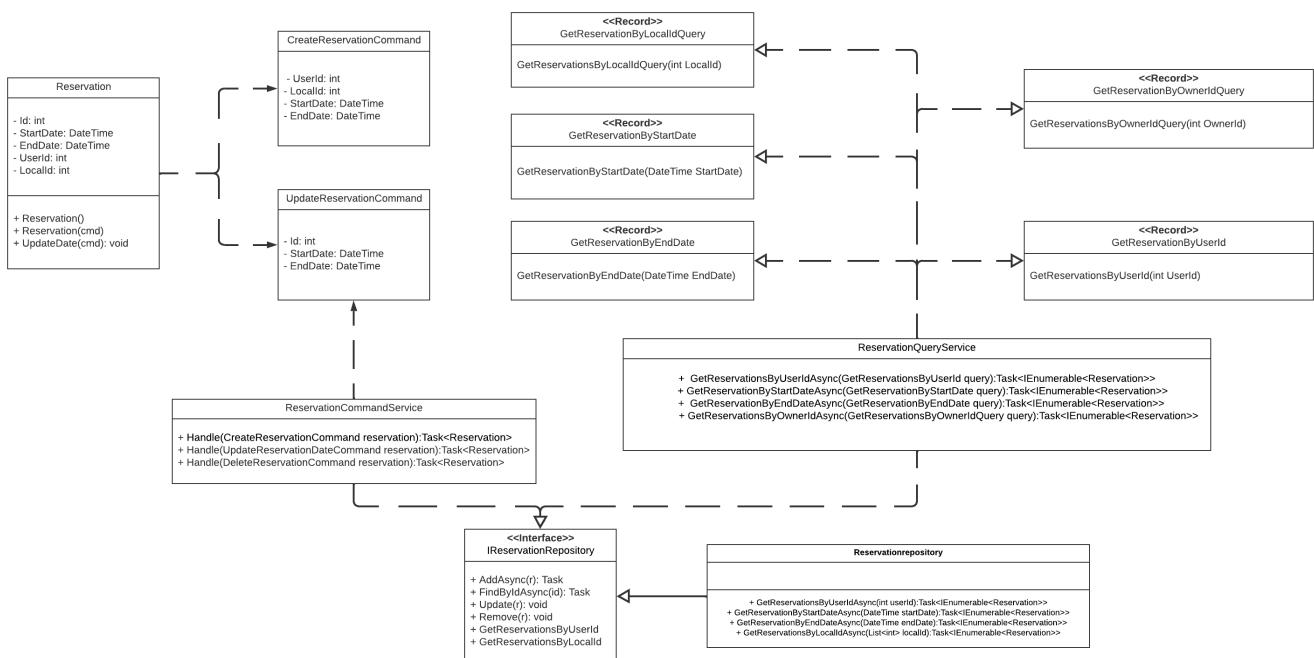
[Component Level Diagram] AlquilaFácil - Booking Bounded Context

jueves, 24 de abril de 2025, 9:08 p.m. hora estándar de Perú

5.4.7. Bounded Context Software Architecture Code Level Diagrams

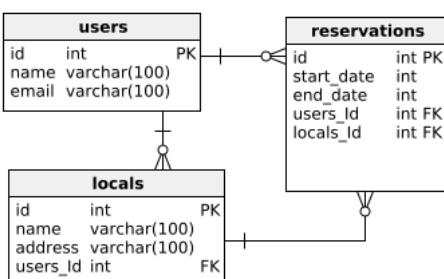
5.4.7.1. Bounded Context Domain Layer Class Diagrams

El diagrama de clases del bounded context Booking representa el aggregate root Reservation con sus atributos (StartDate, EndDate, UserId, LocalId), los comandos para crear, actualizar y eliminar reservas, las consultas para obtener reservaciones, y la interfaz del repositorio IReservationRepository.



5.4.7.2. Bounded Context Database Design Diagram

El diagrama de base de datos del bounded context Booking muestra la tabla reservations con sus atributos (id, start_date, end_date, user_id, local_id), sus tipos de datos, claves primarias y las claves foráneas que establecen relaciones con las tablas de usuarios y locales.



5.5. Bounded Context: Notifications Context

5.5.1. Domain Layer

Aggregates

◊ Notification

- **Descripción:** Representa una notificación enviada a un usuario dentro del sistema.

- **Atributos:**

- **Id:** Identificador único de la notificación.
- **UserId:** Usuario destinatario.
- **Title:** Título de la notificación.
- **Message:** Contenido textual.
- **Type:** Tipo de notificación (información, advertencia, error).
- **CreationDate:** Fecha y hora de creación.
- **State:** Estado de la notificación (**enviada**, **leída**).

Value Objects

- **NotificationType:** Enum de tipo de notificación **INFO**, **WARN** y **ERROR**.
- **NotificationStatus:** Enum de estado **SENT** y **READ**.
- **UserId:** encapsula el id del usuario (**userId**).

5.5.2. Interface Layer

Controllers

- **NotificationController**

- **Descripción:** Expone endpoints para gestionar las notificaciones del sistema.
- **Métodos:**
 - **GetAllNotificationsByUserId(int userId):** Retorna todas las notificaciones asociadas a un usuario.
 - **GetAllNotificationsByUserIdAndType(int userId, string type):** Lista las notificaciones no leídas del usuario.
 - **MarkAsRead(int notificationId):** Marca una notificación como leída.

5.5.3. Application Layer

Command Services

- **NotificationCommandService**

- **Descripción:** Contiene la lógica para gestionar comandos relacionados con el envío de notificaciones.
- **Métodos:**
 - **Handle(SendNotificationCommand command):** Valida y envía una nueva notificación.
 - **Handle(MarkAsReadCommand command):** Cambia el estado de una notificación a leída.

Query Services

- **NotificationQueryService**

- **Descripción:** Permite consultar notificaciones relacionadas a un usuario.
- **Métodos:**
 - **Handle(GetAllNotificationsByUserIdQuery query):** Devuelve todas las notificaciones del usuario.
 - **Handle(GetAllNotificationsByUserIdAndType query):** Devuelve notificaciones por usuario y tipo.

5.5.4. Infrastructure Layer

Repositories (Implementaciones)

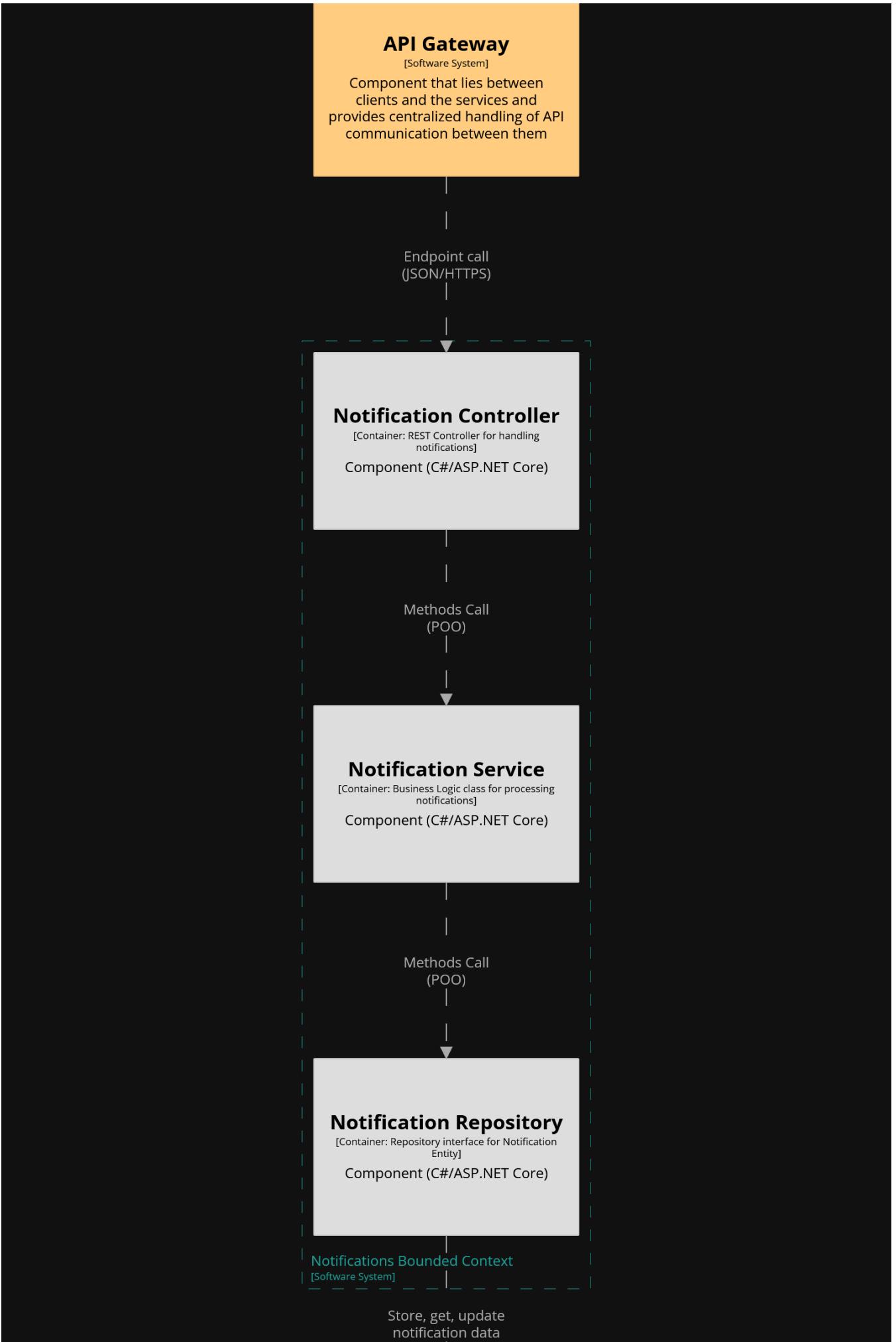
- **NotificationRepository**

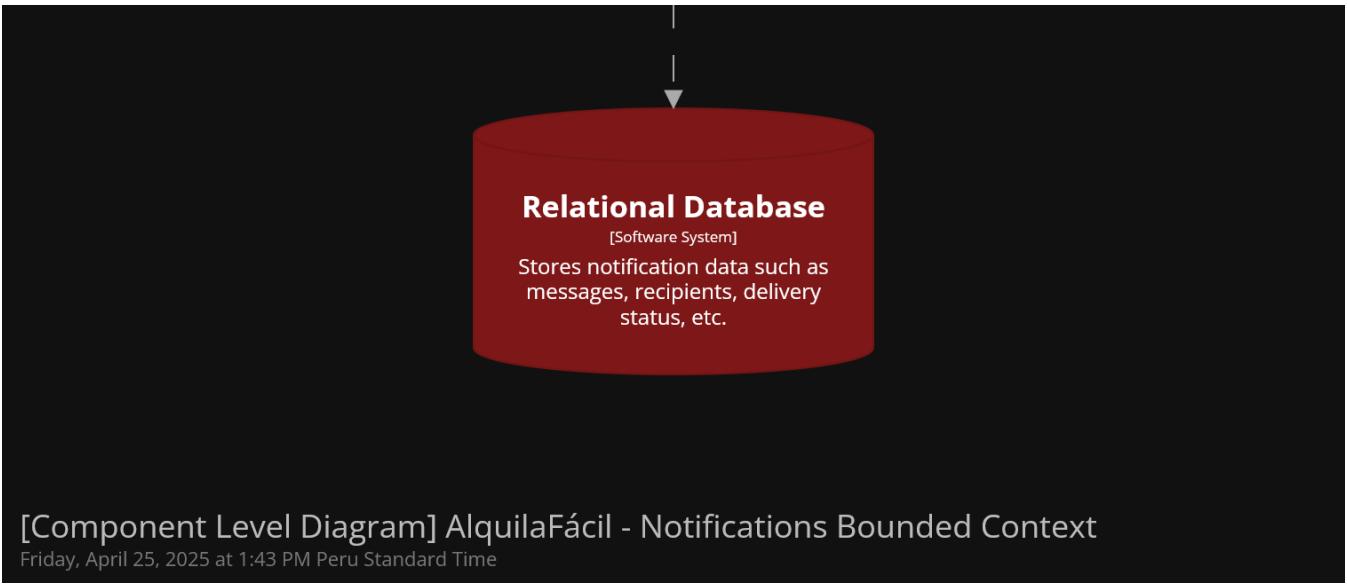
- **Descripción:** Implementa acceso a la base de datos de notificaciones.
- **Métodos:**
 - **findById(userId):** Lista de notificaciones por usuario.
 - **findByIdAndType(userId, type):** Lista de notificaciones por usuario y tipo.
 - **save(notification):** Guarda una nueva notificación.

5.5.6. Bounded Context Software Architecture Component Level Diagrams

El diagrama de componentes C4 del bounded context Notifications presenta la arquitectura del contenedor Notifications API, mostrando el controlador de notificaciones, los servicios de comandos y consultas, el repositorio de notificaciones y su integración con la base de datos y servicios externos de mensajería.







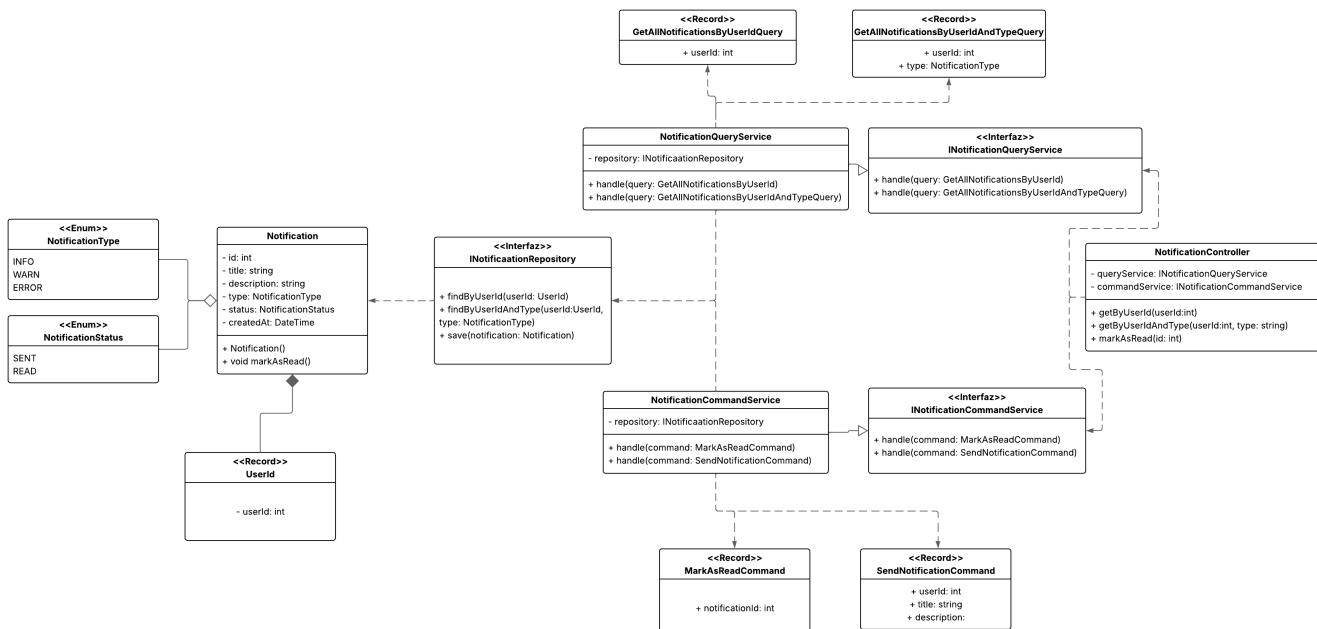
[Component Level Diagram] AlquilaFácil - Notifications Bounded Context

Friday, April 25, 2025 at 1:43 PM Peru Standard Time

5.5.7. Bounded Context Software Architecture Code Level Diagrams

5.5.7.1. Bounded Context Domain Layer Class Diagrams

El diagrama de clases del bounded context Notifications muestra el aggregate root Notification con sus atributos (Message, Type, SentDate, UserId), los value objects relacionados, los comandos para enviar notificaciones, las consultas para obtenerlas y la interfaz del repositorio INotificationRepository.



5.5.7.2. Bounded Context Database Design Diagram

El diagrama de base de datos del bounded context Notifications muestra la tabla notifications con sus campos (id, message, type, sent_date, user_id), tipos de datos, claves primarias y las claves foráneas que relacionan las notificaciones con los usuarios destinatarios.

notification		
id	int	PK
title	varchar(255)	
descripti	varchar(50)	
type	varchar(100)	
status	varchar(100)	
created_	timestamp	



5.6. Bounded Context: Subscription Context

5.6.1. Domain Layer

Aggregates

1. Invoice:

- **Descripción:** Representa la compra de una suscripción en la aplicación
- **Atributos:**
 - **Id:** Identificador de la compra
 - **Amount:** Monto de la compra
 - **Date:** Fecha de la compra
 - **SubscriptionId:** Id de la suscripción comprada

2. Plan:

- **Descripción:** Representa un plan de suscripción en la aplicación
- **Atributos:**
 - **Id:** Identificador del plan de suscripción.
 - **Name:** Nombre del plan
 - **Service:** Servicio que ofrece el plan
 - **Price:** Precio del plan

3. Subscription:

- **Descripción:** Representa una suscripción en la aplicación
- **Atributos:**
 - **Id:** Identificador de la suscripción.
 - **UserId:** Identificador del usuario que compro la suscripción.
 - **SubscriptionStatusId:** Identificador del estado de la suscripción.
 - **PlanId:** Identificador del plan de la suscripción
 - **VoucherImageUrl:** Imagen del voucher de la suscripción

4. SubscriptionAudit:

- **Description:** Registro de las fechas de creación o actualización de las suscripciones.
- **Atributos:**
 - **CreatedDate:** Fecha de creación de la suscripción
 - **UpdatedDate:** Fecha de actualización de la suscripción

Entities

1. SubscriptionStatus:

- **Descripción:** Estado de la suscripción
- **Atributos:**
 - **Id:** Identificador único del estado de la suscripción.
 - **Status:** Nombre del estado de la descripción.

Value Objects

1. ESubscriptionStatus:

- **Descripción:** Enumerador para obtener los estados de la suscripción.

Commands

1. ActiveSubscriptionStatusCommand:

- **Descripción:** Comando para modificar el estado de una suscripción a activa.

2. CreateInvoiceCommand:

- **Descripción:** Comando para crear una compra de una suscripción.

3. CreatePlanCommand:

- **Descripción:** Comando para crear un plan de suscripción

4. CreateSubscriptionCommand:

- **Descripción:** Comando para crear una suscripción

5. CreateSubscriptionPaymentCommand:

- **Descripción:** Comando para crear un pago de una subscripción

6. **SeedSubscriptionPlanCommand:**

- **Descripción:** Comando para inicializar los planes de suscripción

7. **SeedSubscriptionStatusCommand:**

- **Descripción:** Comando para inicializar los estados de suscripción

Queries

1. **GetAllInvoicesQuery:**

- **Descripción:** Consulta para obtener todas las compras de suscripción.

2. **GetAllPlans:**

- **Descripción:** Consulta para obtener todos los planes de suscripción.

3. **GetAllSubscriptionPayments:**

- **Descripción:** Consulta para obtener todos los pagos de suscripción.

4. **GetInvoiceByIdQuery:**

- **Descripción:** Consulta para obtener una compra de suscripción dado su id.

5. **GetPlanByIdQuery:**

- **Descripción:** Consulta para obtener un plan de suscripción dado su id.

6. **GetSubscriptionByIdQuery:**

- **Descripción:** Consulta para obtener una suscripción dado su id.

7. **GetSubscriptionByUserIdQuery:**

- **Descripción:** Consulta para obtener una suscripción dado el id del usuario

8. **GetSubscriptionPaymentByIdQuery:**

- **Descripción:** Consulta para obtener el pago de una suscripción dado su id

9. **GetSubscriptionsByUserIdQuery:**

- **Descripción:** Consulta para obtener todas las suscripciones de un solo usuario.

Repositories

1. **IInvoiceRepository:**

- **Descripción:** Permite interactuar con la base de datos para almacenar u obtener datos de las compras.

2. **IPlanRepository:**

- **Descripción:** Permite interactuar con la base de datos para almacenar u obtener datos de los planes de suscripción.

3. **ISubscriptionRepository:**

- **Descripción:** Permite interactuar con la base de datos para almacenar u obtener datos de las suscripciones.

• **Métodos:**

- `FindByUserIdAsync(int userId)`: Obtiene una suscripción a través del id del usuario
- `FindByUserIdsListAsync`: Obtiene una lista de suscripciones a través de una lista de ids de usuarios.

4. **ISubscriptionStatusRepository:**

- **Descripción:** Permite interactuar con la base de datos para almacenar u obtener datos de los estados de suscripción.

• **Métodos:**

- `ExistsBySubscriptionStatus(ESubscriptionStatus subscriptionStatus)`: Verifica si existe un estado.

Services

1. **IInvoiceCommandService:**

- **Descripción:** Maneja los comandos para la creación de las compras

• **Métodos:**

- `Handle(CreateInvoiceCommand command)`: Maneja el comando para crear una compra.

2. **IInvoiceQueryService:**

- **Descripción:** Maneja las consultas de las compras desde la base de datos
- **Métodos:**
 - `Handle(GetInvoiceByIdQuery query)`: Maneja la consulta para obtener una compra a través de su id.
 - `Handle(GetAllInvoicesQuery query)`: Maneja la consulta para obtener todas las compras registradas.

3. IPlanCommandService

- **Descripción:** Maneja los comandos para la creación de los planes
- **Métodos:**
 - `Handle(CreatePlanCommand command)`: Maneja el comando para la creación de planes de suscripción.

4. IPlanQueryService

- **Descripción:** Maneja las consultas de los planes desde la base de datos
- **Métodos:**
 - `Handle(GetAllPlansQuery query)`: Maneja la consulta para obtener todos los planes de suscripción.

5. ISeedSubscriptionPlanCommandService

- **Descripción:** Permite el manejo del comando para la inicialización de los planes de suscripción.
- **Métodos:**
 - `Handle(SeedSubscriptionPlanCommand command)`: Maneja el comando para la inicialización de los planes de suscripción.

6. ISubscriptionCommandService

- **Descripción:** Permite el manejo de comandos para la creación y activación de las suscripciones
- **Métodos:**
 - `Handle(CreateSubscriptionCommand command)`: Maneja el comando para la creación de suscripciones.
 - `Handle(ActiveSubscriptionStatusCommand command)`: Maneja el comando para la activación de una suscripción.

7. ISubscriptionQueryService

- **Descripción:** Permite el manejo de consultas de las suscripciones desde la base de datos.
- **Métodos:**
 - `Handle(GetSubscriptionByIdQuery query)`: Maneja la consulta para obtener una suscripción en base a su id.
 - `Handle(GetAllSubscriptionsQuery query)`: Maneja la consulta para obtener todas las suscripciones.
 - `Handle(GetSubscriptionByUserIdQuery query)`: Maneja la consulta para obtener una suscripción en base a la id de su usuario.
 - `Handle(GetSubscriptionsByUserIdQuery query)`: Maneja la consulta para obtener todas las suscripciones de un usuario en base a su id.

8. ISubscriptionStatusCommandService

- **Descripción:** Permite el manejo de comandos para la inicialización de estados.
- **Métodos:**
 - `Handle(SeedSubscriptionStatusCommand command)`: Maneja la inicialización de los estados.

5.6.2. Interface Layer

Facades

1. ISubscriptionContextFacade

- **Descripción:** Permite el uso de métodos del bounded context de subscriptions en otros bounded context.
- **Métodos:**
 - `GetSubscriptionByUserIdsList(List<int> userIdsList)`: Obtiene las suscripciones de varios usuarios a través de sus ids.
 - `GetSubscriptionStatusByUserId(int userId)`: Obtiene los estados de la suscripción de un usuario a través de su id.

Controller

1. InvoiceController:

- **Descripción:** Expone endpoints para el manejo de las compras dentro de la aplicación
- **Métodos:**
 - `CreateInvoice([CreateInvoiceResource createInvoiceResource])`: Endpoint para crear una compra
 - `GetInvoices()`: Endpoint para obtener todas las compras.
 - `GetInvoiceById(int invoiceId)`: Endpoint para obtener una compra en base a su id.

2. PlanController:

- **Descripción:** Exponer endpoints para el manejo de los planes de suscripción dentro de la aplicación
- **Métodos:**
 - `GetAllPlans()`: Endpoint para obtener todos los planes de suscripción dentro de la aplicación

3. SubscriptionsController:

- **Descripción:** Exponer endpoints para el manejo de las suscripciones dentro de la aplicación.
- **Métodos:**

- `CreateSubscription(CreateSubscriptionResource createSubscriptionResource)`: Endpoint para la creación de suscripciones dentro de la aplicación
- `GetAllSubscriptions()`: Endpoint para obtener todas las suscripciones dentro de la aplicación
- `GetSubscriptionById(int subscriptionId)`: Endpoint para obtener suscripciones en base a su id
- `ActiveSubscriptionStatus(int subscriptionId)`: Endpoint para activar una suscripción en base a su id

5.6.3. Application Layer

Command Services (Implementation)

1. InvoiceCommandService:

- **Descripción:** Maneja los comandos para la creación de las compras
- **Métodos:**
 - `Handle(CreateInvoiceCommand command)`: Maneja el comando para crear una compra.

2. PlanCommandService

- **Descripción:** Maneja los comandos para la creación de los planes
- **Métodos:**
 - `Handle(CreatePlanCommand command)`: Maneja el comando para la creación de planes de suscripción.

3. SeedSubscriptionPlanCommandService

- **Descripción:** Permite el manejo del comando para la inicialización de los planes de suscripción.
- **Métodos:**
 - `Handle(SeedSubscriptionPlanCommand command)`: Maneja el comando para la inicialización de los planes de suscripción.

4. SubscriptionCommandService

- **Descripción:** Permite el manejo de comandos para la creación y activación de las suscripciones
- **Métodos:**
 - `Handle(CreateSubscriptionCommand command)`: Maneja el comando para la creación de suscripciones.
 - `Handle(ActiveSubscriptionStatusCommand command)`: Maneja el comando para la activación de una suscripciones.

5. SubscriptionStatusCommandService

- **Descripción:** Permite el manejo de comandos para la inicialización de estados.
- **Métodos:**
 - `Handle(SeedSubscriptionStatusCommand command)`: Maneja la inicialización de los estados.

Query Services (Implementación)

1. InvoiceQueryService:

- **Descripción:** Maneja las consultas de las compras desde la base de datos
- **Métodos:**
 - `Handle(GetInvoiceByIdQuery query)`: Maneja la consulta para obtener una compra a través de su id.
 - `Handle(GetAllInvoicesQuery query)`: Maneja la consulta para obtener todas las compras registradas.

2. PlanQueryService

- **Descripción:** Maneja las consultas de los planes desde la base de datos
- **Métodos:**
 - `Handle(GetAllPlansQuery query)`: Maneja la consulta para obtener todos los planes de suscripción.

3. SubscriptionQueryService

- **Descripción:** Permite el manejo de consultas de las suscripciones desde la base de datos.
- **Métodos:**
 - `Handle(GetSubscriptionByIdQuery query)`: Maneja la consulta para obtener una suscripción en base a su id.
 - `Handle(GetAllSubscriptionsQuery query)`: Maneja la consulta para obtener todas las suscripciones.
 - `Handle(GetSubscriptionByUserIdQuery query)`: Maneja la consulta para obtener una suscripción en base a la id de su usuario.
 - `Handle(GetSubscriptionsByUserIdQuery query)`: Maneja la consulta para obtener todas las suscripciones de un usuario en base a su id.

OutboundServices

1. ExternalUserWithSubscriptionService

- **Descripción:** Permite utilizar métodos del bounded context de usuarios.
- **Métodos:**
 - `UserExists(int id)`: Permite validar si un usuario existe dada su id.

5.6.4. Infrastructure Layer

Repositories (Implementación)

1. **IInvoiceRepository**:

- **Descripción:** Permite interactuar con la base de datos para almacenar u obtener datos de las compras.

2. **IPlanRepository**:

- **Descripción:** Permite interactuar con la base de datos para almacenar u obtener datos de los planes de suscripción.

3. **ISubscriptionRepository**:

- **Descripción:** Permite interactuar con la base de datos para almacenar u obtener datos de las suscripciones.

• **Métodos:**

- `FindByUserIdAsync(int userId)`: Obtiene una suscripción a través del id de su usuario
- `FindByUserIdsListAsync`: Obtiene una lista de suscripciones a través de una lista de ids de usuarios.

4. **ISubscriptionStatusRepository**:

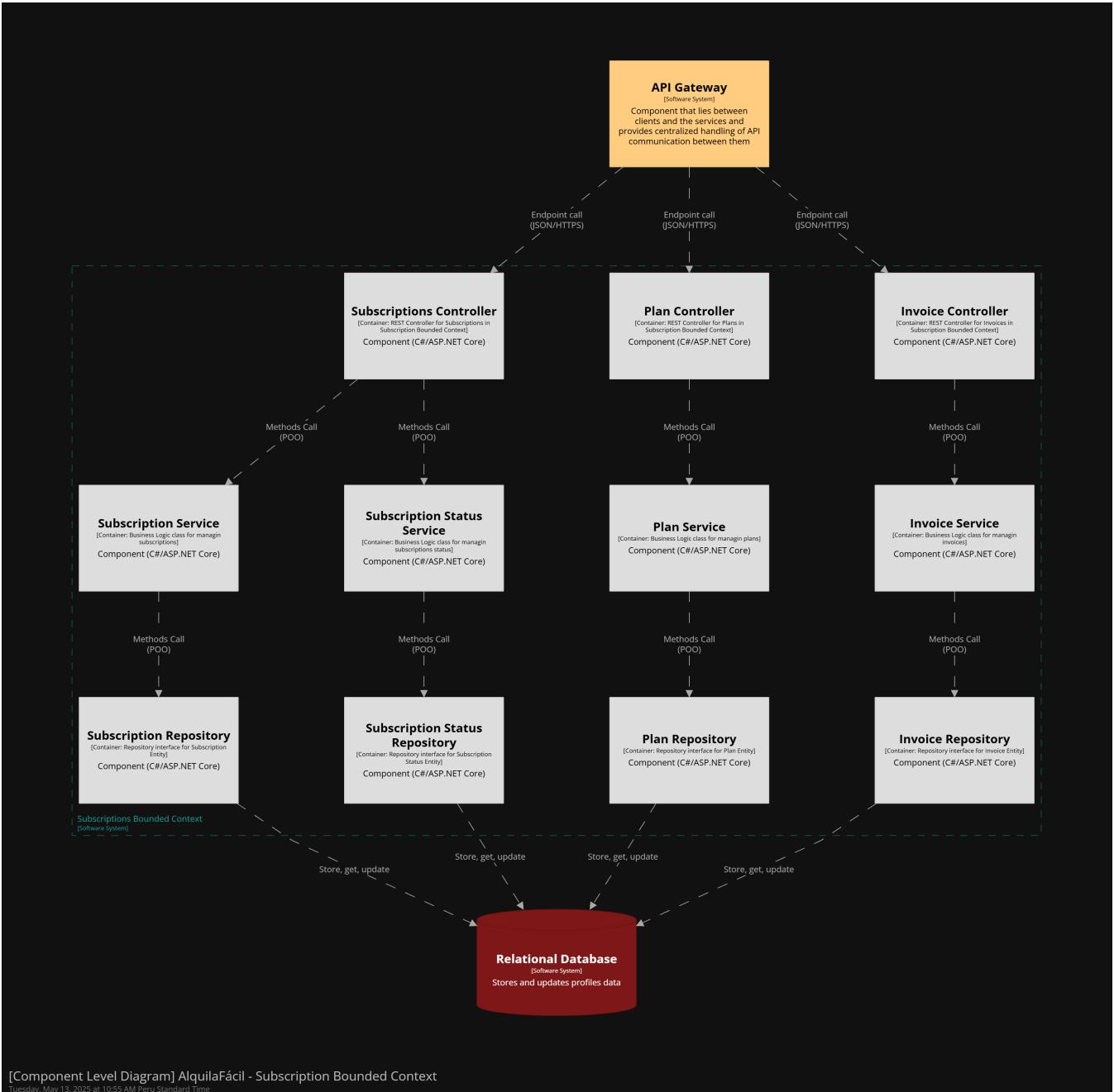
- **Descripción:** Permite interactuar con la base de datos para almacenar u obtener datos de los estados de suscripción.

• **Métodos:**

- `ExistsBySubscriptionStatus(ESubscriptionStatus subscriptionStatus)`: Verifica si existe un estado.

5.6.6. Bounded Context Software Architecture Component Level Diagrams

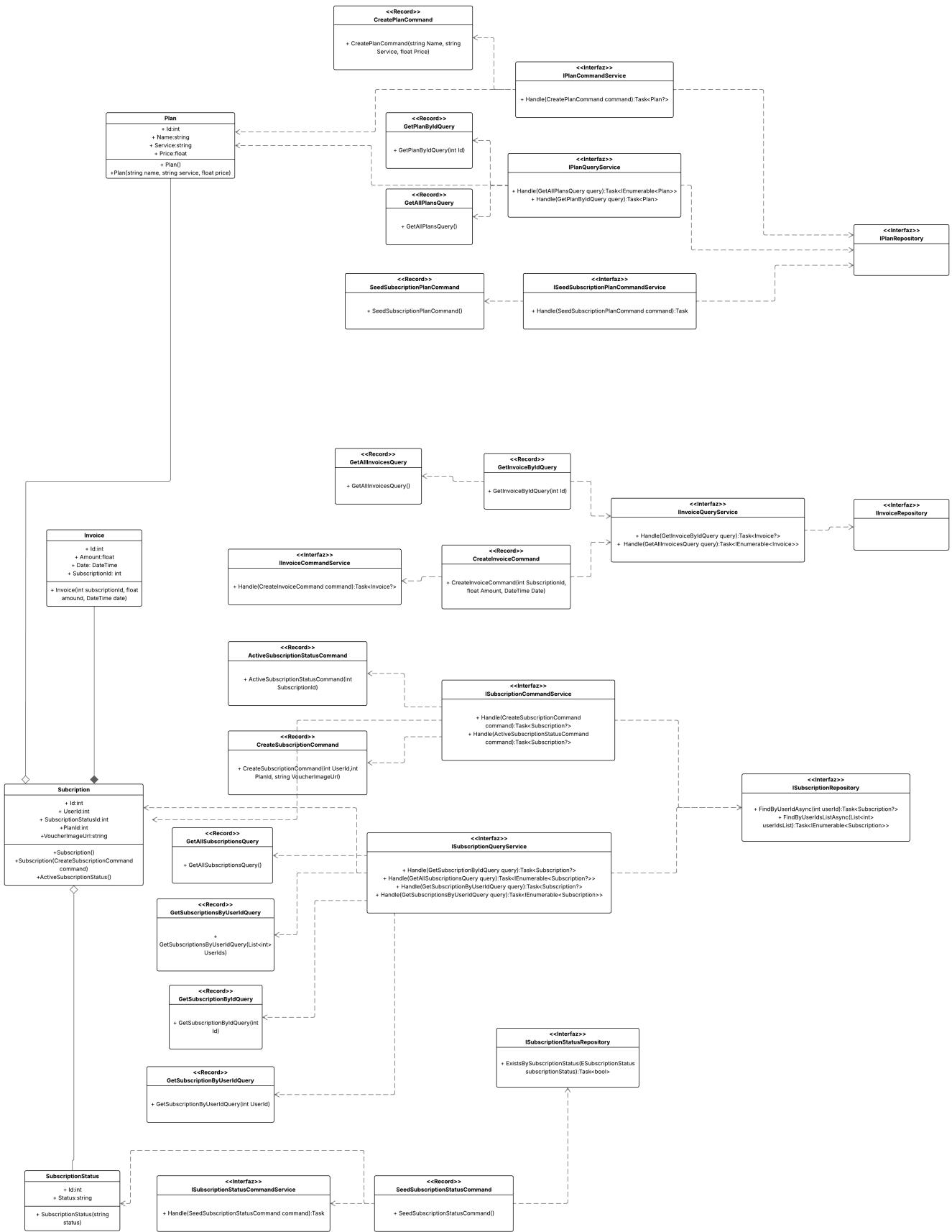
El diagrama de componentes C4 del bounded context Subscription ilustra la arquitectura del contenedor Subscription API, mostrando los controladores para gestionar suscripciones, planes y facturas, los servicios de aplicación, los repositorios correspondientes y su conexión con la base de datos y servicios externos de usuarios.



5.6.7. Bounded Context Software Architecture Code Level Diagrams

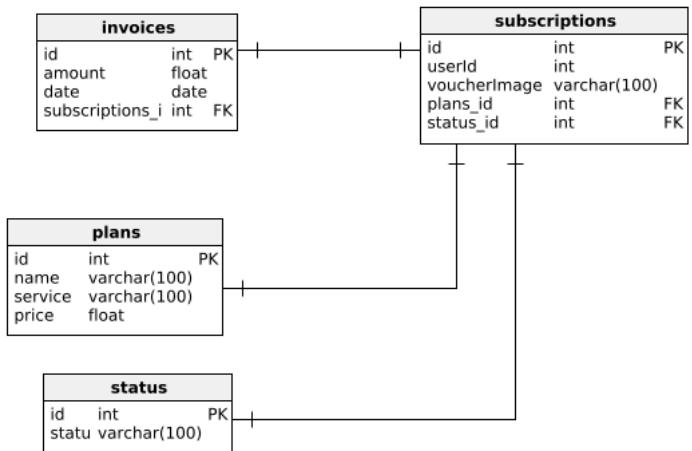
5.6.7.1. Bounded Context Domain Layer Class Diagrams

El diagrama de clases del bounded context Subscription presenta los aggregates Invoice, Plan y Subscription con sus atributos respectivos, los value objects (SubscriptionStatus, PlanDuration), los comandos para crear y gestionar subscripciones, las consultas y las interfaces de los repositorios (IInvoiceRepository, IPlanRepository, ISubscriptionRepository).



5.6.7.2. Bounded Context Database Design Diagram

El diagrama de base de datos del bounded context Subscription muestra las tablas invoices, plans, subscriptions y subscription_statuses con sus respectivos campos, tipos de datos, claves primarias y las relaciones entre ellas, estableciendo cómo se almacenan las subscripciones, planes y facturas en la base de datos.



5.7. Bounded Context: Monitoring Context

5.7.1. Domain Layer

Aggregates

1. Sensor:

- **Descripción:** Representa un sensor IoT desplegado dentro de un espacio monitoreado (por ejemplo, para ruido o humo).
- **Atributos:**
 - **Id:** Identificador único del sensor.
 - **Code:** Código físico o serial único del sensor.
 - **Type:** Tipo del sensor: NOISE, SMOKE, TEMP, etc.
 - **State:** Estado del sensor (ACTIVE, INACTIVE, MAINTENANCE).
 - **Location:** Ubicación física dentro de la propiedad.
 - **LocalId:** Identificador del local al que pertenece.
 - **Readings:** Colección de lecturas registradas por el sensor.

2. NFCTag

- **Descripción:** Representa una tarjeta NFC registrada y asociada a un usuario o propósito.
- **Atributos:**
 - **Id:** Identificador único del tag.
 - **Code:** Código físico del tag NFC.
 - **Alias:** Nombre opcional para identificar la tarjeta.
 - **IsActive:** Si está activa o desactivada.
 - **ExpirationDate:** Fecha opcional de expiración del tag.

Entities

1. Reading:

- **Descripción:** Representa un dato capturado por un sensor en un momento específico.
- **Atributos:**
 - **Id:** Identificador único de la lectura.
 - **SensorId:** ID del sensor que envió la lectura.
 - **Timestamp:** Fecha y hora de la lectura.
 - **Value:** Valor medidor por el sensor.
 - **Unit:** Unidad de medida (°C, %, etc.).

2. NFCDetectionEvent:

- **Descripción:** Representa un evento de detección de un tag NFC en un lector.
- **Atributos:**
 - **Id:** Identificador único de la configuración.
 - **DetectedAt:** Momento de la detección.
 - **TagId:** Identificador del NFCTag detectado.
 - **SensorId:** Identificador de sensor que detecta.

Value Objects

1. SensorType:

- **Descripción:** Objeto de valor que representa el tipo de un sensor. Asegura el control del tipo de infracción respectivo.
- **Atributos:**
 - **SensorType:** Tipo del sensor: (NOISE, SMOKE, TEMP, etc).

2. SensorStatus:

- **Descripción:** Objeto de valor que representa el estado de un sensor. Asegura que el estado cumpla con ciertas reglas de negocio (por ejemplo, solo valores como 'activo' o 'inactivo').
- **Atributos:**
 - **Status:** Estado del dispositivo (activo, inactivo, mantenimiento).

3. Unit:

- **Descripción:** Objeto de valor que representa la unidad de valor de una lectura en el sensor.
- **Atributos:**
 - **Unit:** Unidad del valor (ej: dB, ppm, °C).

Commands

Sensor

1. CreateSensorCommand:

- **Descripción:** Crear un nuevo sensor con tipo, ubicación y estado inicial

2. CreateReadingCommand:

- **Descripción:** Crear una lectura individual desde un sensor para registrarla

3. UpdateSensorCommand:

- **Descripción:** Activar o desactivar un sensor específico

NFCTag

1. CreateNFCTagCommand:

- **Descripción:** Registra una nueva tarjeta NFC con UID y alias opcional

2. UpdateNFCTagCommand:

- **Descripción:** Desactiva una tarjeta NFC existente

3. CreateNFCDetectionEvent:

- **Descripción:** Registra un evento de detección de un tag NFC en un lector

Queries

Sensor

1. GetAllSensorsByLocalIdQuery:

- **Descripción:** Obtener todos los sensores registrados para un id de local dado.

2. GetSensorByIdQuery:

- **Descripción:** Obtener detalles de un sensor específico

3. GetReadingsBySensorIdQuery:

- **Descripción:** Obtener todas las lecturas de un sensor

NFCTag

1. GetNFCTagByIdQuery:

- **Descripción:** Obtener detalles de un tag NFC específico

2. GetAllTagDetectionEventsBySensorIdQuery:

- **Descripción:** Obtener eventos de detección para un sensor específico

Repositories (Interfaces)

1. SensorRepository:

- **Descripción:** Interfaz para interactuar con la base de datos de sensores.
- **Métodos:**
 - `FindAllByLocalId(int id)`: Busca todos los sensores por ID de local.
 - `FindByID(int id)`: Busca un sensor por su ID.
 - `Save(Sensor sensor)`: Guarda o actualiza un sensor en la base de datos.

2. **ReadingRepository:**

- **Descripción:** Interfaz para interactuar con la base de datos de lecturas de sensores.
- **Métodos:**
 - `FindAllBySensorId(int id)`: Busca todas las lecturas por ID de sensor.
 - `Save(Reading reading)`: Guarda o actualiza una lectura en la base de datos.

3. **NFCTagRepository:**

- **Descripción:** Interfaz para interactuar con la base de datos de tags NFC.
- **Métodos:**
 - `FindByID(int id)`: Busca un tag NFC por su ID.
 - `Save(NFCTag nfcTag)`: Guarda o actualiza un tag NFC en la base de datos.

4. **NFCDetectionEventRepository:**

- **Descripción:** Interfaz para interactuar con la base de datos de detecciones NFC.
- **Métodos:**
 - `FindAllBySensorId(int id)`: Busca todas las detecciones por ID de sensor.
 - `Save(NFCDetectionEvent nfcDetectionEvent)`: Guarda o actualiza una detección en la base de datos.

5.7.2. Interface Layer

Controllers

1. **SensorsController:**

- **Descripción:** Expone endpoints para la gestión de sensores, incluyendo la creación, actualización y recuperación de lecturas.
- **Métodos:**
 - `RegisterSensor(Sensor sensor)`: Endpoint para registrar un nuevo sensor.
 - `RegisterReading(Reading reading)`: Endpoint para registrar una lectura de sensor.
 - `UpdateSensorStatus(int id, String status)`: Endpoint para actualizar el estado de un dispositivo.
 - `GetAllSensorsByLocalId(int id)`: Endpoint para obtener todos los sensores dado un ID de local.
 - `GetSensorById(int id)`: Endpoint para obtener los detalles de un sensor por su ID.
 - `GetAllReadingsBySensorId(int id)`: Endpoint para obtener todas las lecturas dado un ID de sensor.

2. **NFCTagsController:**

- **Descripción:** Expone endpoints para la gestión de sensores, incluyendo la creación, actualización y recuperación de dispositivos.
- **Métodos:**
 - `RegisterNFCTag(NFCTag nfcTag)`: Endpoint para registrar un nuevo tag NFC.
 - `RegisterNFCDetectionEvent(NFCDetectionEvent nfcDetectionEvent)`: Endpoint para registrar una detección de evento NFC.
 - `UpdateNFCTagStatus(int id, String status)`: Endpoint para actualizar el estado de un tag NFC.
 - `GetNFCTagById(int id)`: Endpoint para obtener los detalles de un tag NFC por su ID.
 - `GetAllNFCDetectionEventsBySensorId(int id)`: Endpoint para obtener todas las detecciones de eventos NFC dado un ID de sensor.

5.7.3. Application Layer

Command Services

1. **SensorCommandService:**

- **Descripción:** Maneja comandos para crear y actualizar sensores con sus lecturas.
- **Métodos:**
 - `Handle(CreateSensorCommand command)`: Valida y aplica el comando para crear un sensor.
 - `Handle(CreateReadingCommand command)`: Valida y aplica el comando para crear una lectura.
 - `Handle(UpdateSensorCommand command)`: Valida y aplica el comando para actualizar el estado de un sensor.

2. **NFCTagCommandService:**

- **Descripción:** Maneja comandos para crear y actualizar tags NFC con sus detecciones de eventos.
- **Métodos:**
 - `Handle(CreateNFCTagCommand command)`: Valida y aplica el comando para crear un tag NFC.
 - `Handle(CreateNFCDetectionEventCommand command)`: Valida y aplica el comando para crear una detección de evento NFC.
 - `Handle(UpdateNFCTagCommand command)`: Valida y aplica el comando para actualizar el estado de un tag NFC.

Query Services

1. SensorQueryService:

- **Descripción:** Ofrece consultas para obtener información sobre los sensores y sus lecturas.
- **Métodos:**
 - `Handle(GetAllSensorsByLocalIdQuery query)`: Valida y aplica el query para obtener todos los sensores dado un ID de local.
 - `Handle(GetSensorByIdQuery query)`: Valida y aplica el query para obtener la información de un sensor dado su ID.
 - `Handle(GetReadingsBySensorIdQuery query)`: Valida y aplica el query para obtener las lecturas dado un ID de sensor.

2. NFCTagQueryService:

- **Descripción:** Ofrece consultas para obtener información sobre los tag NFC y sus detecciones de eventos.
- **Métodos:**
 - `Handle(GetNFCTagByIdQuery query)`: Valida y aplica el query para obtener la información de un tag NFC.
 - `Handle(GetAllTagDetectionEventsBySensorIdQuery query)`: Valida y aplica el query para obtener todas las detecciones de eventos dado un ID de sensor.

5.7.4. Infrastructure Layer

Repositories (Implementaciones)

1. SensorRepository:

- **Descripción:** Implementación para interactuar con la base de datos de sensores.
- **Métodos:**
 - `FindAllByLocalId(int id)`: Busca todos los sensores por ID de local.
 - `FindById(int id)`: Busca un sensor por su ID.
 - `Save(Sensor sensor)`: Guarda o actualiza un sensor en la base de datos.

2. ReadingRepository:

- **Descripción:** Implementación para interactuar con la base de datos de lecturas de sensores.
- **Métodos:**
 - `FindAllBySensorId(int id)`: Busca todas las lecturas por ID de sensor.
 - `Save(Reading reading)`: Guarda o actualiza una lectura en la base de datos.

3. NFCTagRepository:

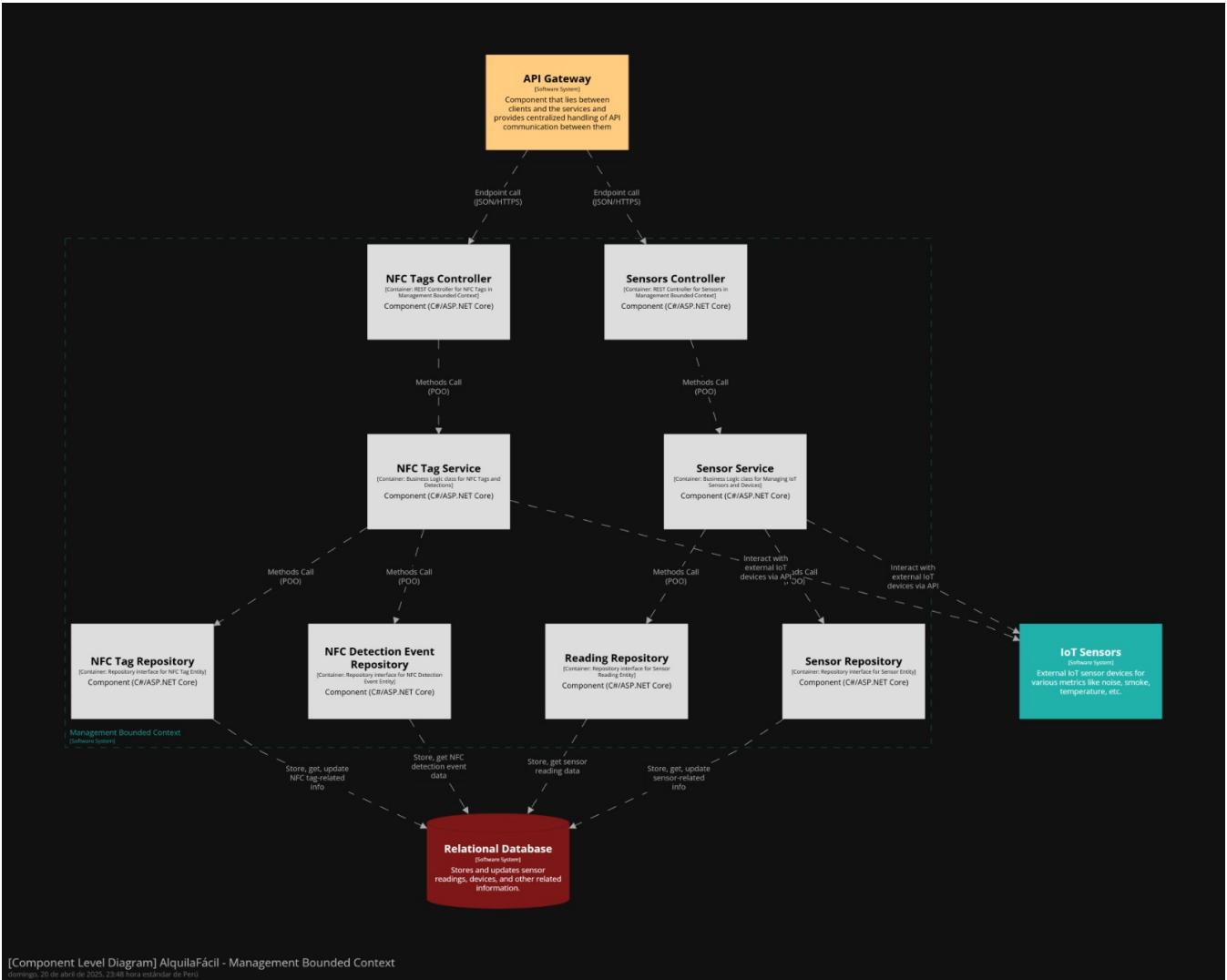
- **Descripción:** Implementación para interactuar con la base de datos de tags NFC.
- **Métodos:**
 - `FindById(int id)`: Busca un tag NFC por su ID.
 - `Save(Sensor sensor)`: Guarda o actualiza un tag NFC en la base de datos.

4. NFCDetectionEventRepository:

- **Descripción:** Implementación para interactuar con la base de datos de detecciones NFC.
- **Métodos:**
 - `FindAllBySensorId(int id)`: Busca todas las detecciones por ID de sensor.
 - `Save(NFCDetectionEvent nfcDetectionEvent)`: Guarda o actualiza una detección en la base de datos.

5.7.6. Bounded Context Software Architecture Component Level Diagrams

El diagrama de componentes C4 del bounded context Monitoring muestra la arquitectura del contenedor Monitoring API, incluyendo los controladores para gestionar sensores y tags NFC, los servicios de comandos y consultas, los repositorios (SensorRepository, ReadingRepository, NFCTagRepository, NFCDetectionEventRepository) y su integración con la base de datos para monitorear eventos en los locales.

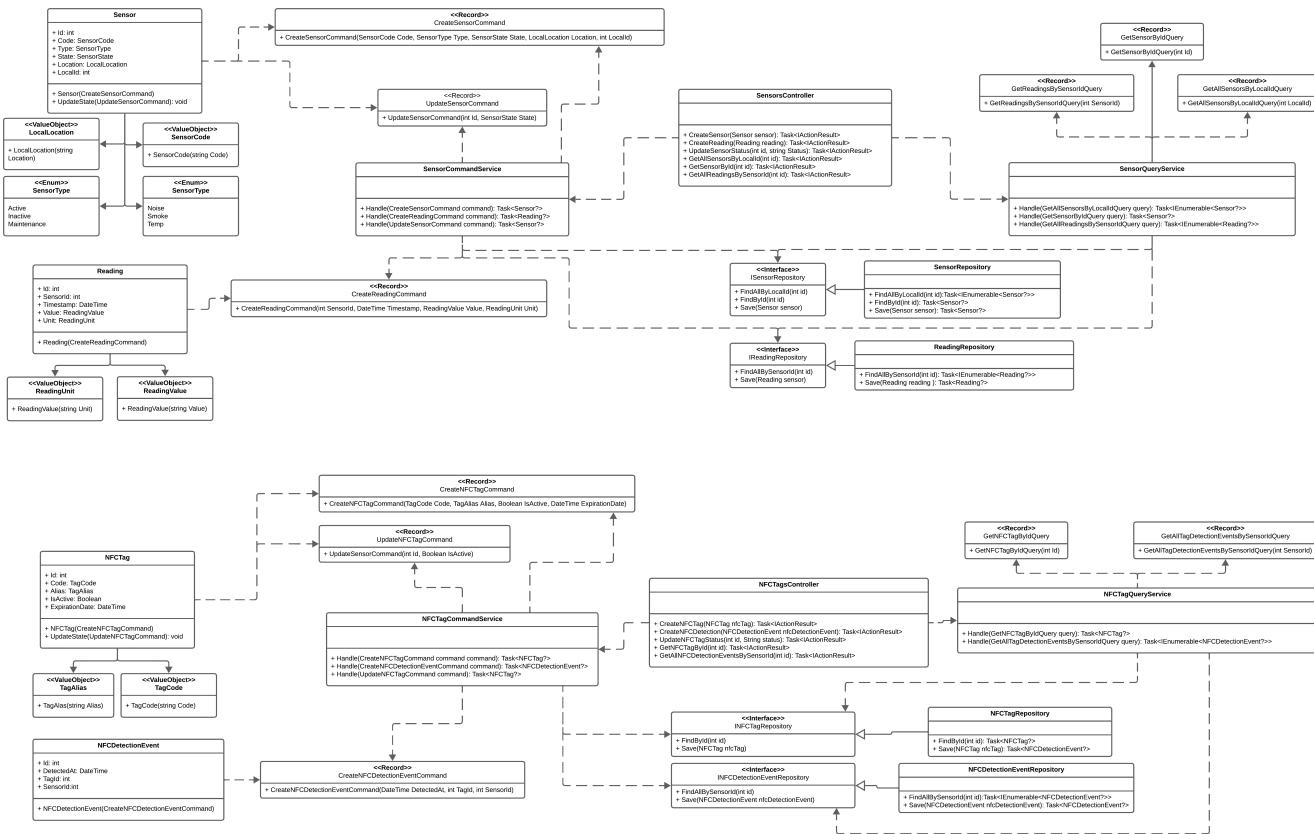


[Component Level Diagram] AlquilaFácil - Management Bounded Context
domingo, 20 de abril de 2025, 23:48 Hora estándar de Perú

5.7.7. Bounded Context Software Architecture Code Level Diagrams

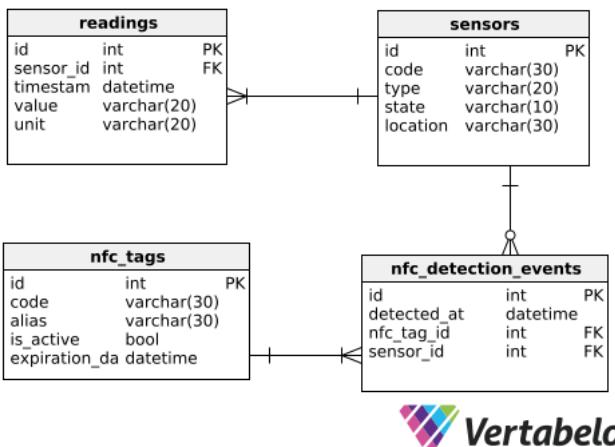
5.7.7.1. Bounded Context Domain Layer Class Diagrams

El diagrama de clases del bounded context Monitoring presenta los aggregates Sensor, Reading, NFCTag y NFCDetectionEvent con sus respectivos atributos, los value objects relacionados con tipos de sensores y estados, los comandos para registrar sensores y lecturas, las consultas para obtener información de monitoreo y las interfaces de los repositorios correspondientes.



5.7.7.2. Bounded Context Database Design Diagram

El diagrama de base de datos del bounded context Monitoring muestra las tablas sensors, readings, nfc_tags y nfc_detection_events con sus campos respectivos, tipos de datos, claves primarias y claves foráneas, estableciendo las relaciones entre sensores, sus lecturas, tags NFC y eventos de detección para el monitoreo de locales.



Capítulo VI: Solution UX Design

6.1. Style Guidelines

Un Style Guideline es un conjunto de reglas y normas que definen cómo se debe redactar, diseñar o presentar documentos, contenido web, software u otros trabajos creativos. A continuación, se detallan las especificaciones de los parámetros implementados en la estructura del proyecto.

6.1.1. General Style Guidelines

Overview:

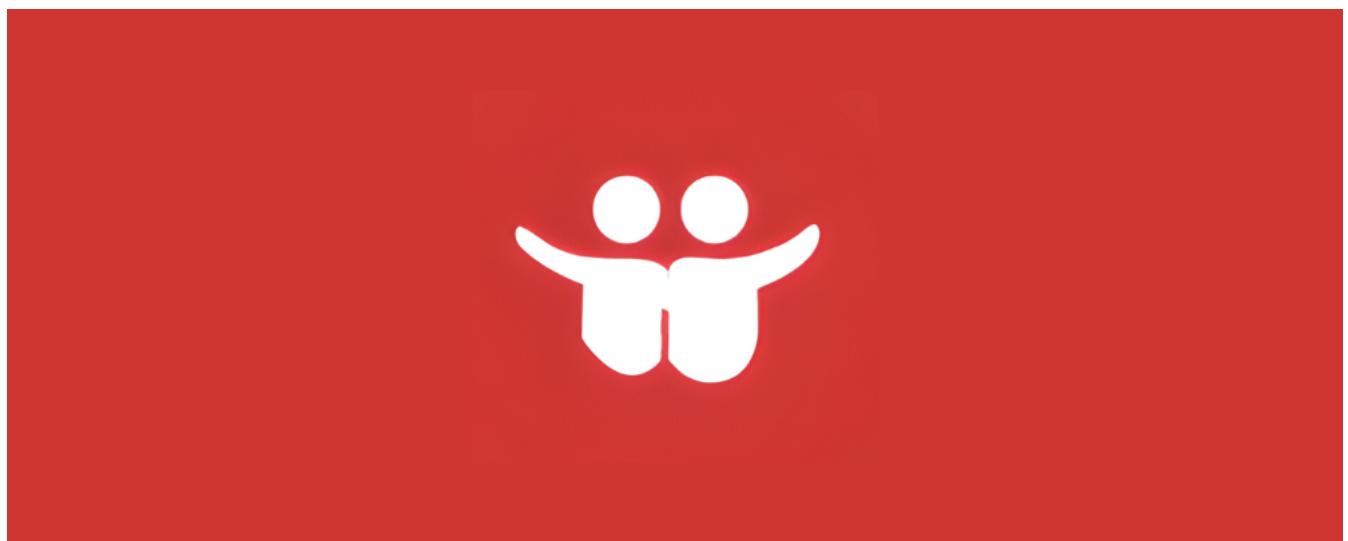
Deseamos capturar la atención del usuario desde el inicio mediante la creación y el diseño de una perspectiva del producto que establezca una conexión inmediata y reconocible.

Brand Overview:

AlquilaFácil es un startup tecnológico fundada en el 2024 por un grupo de estudiantes compuesta por un grupo de estudiantes de la carrera de Ingeniería de Software. Nuestro startup está dedicada a simplificar el proceso de alquiler de espacios para eventos, brindando una plataforma innovadora y accesible para propietarios y organizadores por igual.

Brand Name:

Como el startup se trata de alquilar centros para eventos y también para publicitar centros que podrían ser alquilados, se vio conveniente llamarlo "AlquilaFácil".

**Typography:**

El tamaño elegido para la tipografía por el equipo fue:

Auto Style Sheet

Typography

Text - 100px

Line Height: AUTO / Letter Spacing: 0 %

Text - 60px

Line Height: AUTO / Letter Spacing: 0 %

Text - 50px

Line Height: AUTO / Letter Spacing: 0 %

Text - 30px

Line Height: AUTO / Letter Spacing: 0 %

Text - 25px

Line Height: AUTO / Letter Spacing: 0 %

Text - 20px

Line Height: AUTO / Letter Spacing: 0 %

Text - 16px

Line Height: AUTO / Letter Spacing: 0 px

Colors:

La gama de colores elegidos son:

Auto Style Sheet

Colors

Used Background Colors

#FFFFFF	#000000	#FB9E49	#D13333
---------	---------	---------	---------

Used Font Colors

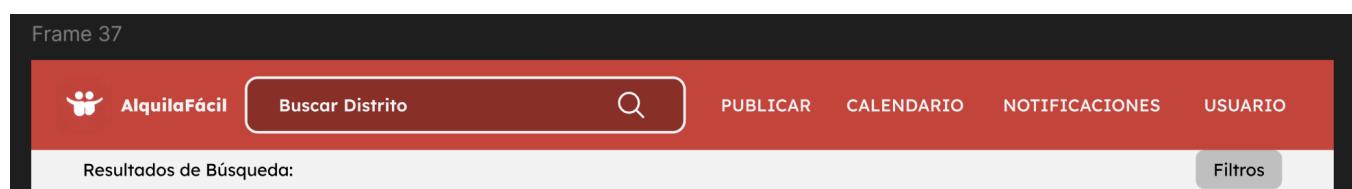
Text #000000	Text #FB9E49
Text #D13333	Text #FFFFFF

6.1.2. Web, Mobile & Devices Style Guidelines

Teniendo en cuenta que tanto para la Web App se han utilizado los mismo estilos base ya predeterminados, como lo son: La tipografía, los colores y el brand del producto AlquilaFacil, hay ciertos aspectos que se diferencian dentro de la versión Mobile y Web. Por ejemplo:

Toolbar:

Una de las principales notoriedades es la Toolbar, al tratarse de una plataforma Web a diferencia de la contraparte Mobile, se tiene la posibilidad de ser mucho más amplia y tener el contenido más disperso.

**Inputs/Buttons:**

Al tener mucho más espacio en la pantalla, los inputs y botones de la Web App son mucho más grandes y con mayor espacio.

Opciones

[Ver comentarios >](#)

[Reportar Espacio >](#)

Fecha y hora de inicio:

dd/mm/aaaa --:--



Fecha y hora de fin:

dd/mm/aaaa --:--



Reservar

Espaciado

Siguiendo lo mencionado antes, teniendo en cuenta que el tamaño de la pantalla de una Web App es mucho más amplio, se puede tener un espacio mejor entre los componentes y elementos para una visibilidad más intuitiva.

The screenshot shows the AlquilaFácil mobile application interface. At the top, there is a red header bar with the logo "AlquilaFácil", a search bar labeled "Buscar Distrito", and a "PUBLICAR" button. A small orange box in the top right corner contains the number "97". Below the header, a section titled "Detalles del local:" is displayed, featuring a large image of a decorated event space with tables and chairs. To the left of the image, there is a summary of the local details:

Local 1
Perú, Lima
Aforo: 10 personas
Propietario: 10 personas
Descripción:
10 personas
Características

Mobile Guidelines

Entre las plataformas de AlquilaFacil Mobile, existe una característica que comparten ambas plataformas del App Mobile, el cuál es la NavBar del aplicativo.

Navbar:

En comparación con la versión de escritorio, la barra de navegación es mucho más reducida y sintetizada a íconos alusivos a las funcionalidades principales de AlquilaFacil

The image displays four screenshots of the AlquilaFácil mobile application illustrating the navigation bar across different screens:

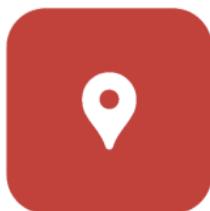
- US15**: Shows the initial step of the process: "Empezar a usar AlquilaFácil es muy sencillo." It includes three steps: 1. Describe tu espacio, 2. Haz que destaque, and 3. Terminar y publicar. A "Comencemos" button is at the bottom.
- US15**: Shows the first step of describing the space: "Paso 1: Describe tu espacio". It features a house icon and a brief description: "En este paso, te preguntaremos qué tipo de propiedad tienes. A continuación, indicarás la ubicación y algunas cosas más." Navigation buttons "Anterior" and "Siguiente" are at the bottom.
- US15**: Shows the second step of describing the space: "¿Cuál de estas opciones describe mejor tu espacio?". It lists four options with icons: "Casa de playa", "Casa urbana", "Salones elegantes", and "Casa de campo". Navigation buttons "Anterior" and "Siguiente" are at the bottom.
- US15**: Shows the confirmation step: "Confirma tu dirección". It includes fields for País (Perú), Departamento (Lima), Distrito (San Isidro), and Número, edificio, apartamento (Validizan 202). Navigation buttons "Anterior" and "Siguiente" are at the bottom.

Buttons:

Ambas versiones tanto como para Android e iOS comparten similitudes en todos los aspectos, sin embargo hay ciertos botones por ejemplo que en la versión de iOS son enfocados más en colores Blancos o Negros, como por ejemplo:



Realizar búsqueda



San Isidro, Departamento de Lima



San Miguel, Departamento de Lima

IoT Guidelines

En esta parte se describen los lineamientos técnicos para la implementación de sensores de movimiento, humo y ruido en un entorno simulado utilizando Wokwi. El objetivo es establecer una base sólida para el desarrollo, simulación y posterior migración a dispositivos físicos reales, asegurando buenas prácticas en diseño, seguridad y escalabilidad.

Sensores Utilizados

Tipo de Sensor	Modelo Sugerido	Tipo de Señal	Función
Movimiento	PIR HC-SR501	Digital	Detecta presencia o movimiento
Humo/Gases	MQ-2	Analógica	Mide concentración de gases
Ruido/Ambiente Sonoro	Micrófono analógico	Analógica	Detecta intensidad del sonido

Librerías y Dependencias

Sensor	Librería Recomendada	Propósito
MQ-2	MQUnifiedsensor	Calibración, regresión y lectura de ppm
PIR	No requiere	Lectura digital directa
Micrófono	No requiere	Lectura analógica directa

En Wokwi, las simulaciones suelen funcionar sin necesidad de instalar librerías, pero se recomienda su uso para mantener fidelidad con hardware real.

Lineamientos Técnicos (IoT Guidelines)

Diseño Modular

- Separar la lógica de cada sensor en funciones o bloques independientes.
- Facilitar el mantenimiento y futuras expansiones del sistema.

Buenas Prácticas

- Evitar el uso de funciones bloqueantes (`delay()`), emplear temporizadores no bloqueantes (`millis()`).
- Definir umbrales personalizables para cada tipo de sensor.
- Utilizar estructuras de datos para representar lecturas de sensores de forma estandarizada (ej. JSON).

Comunicación de Datos

- Para simulación: enviar datos por consola serial en formato estructurado (ej. JSON).
- Para implementación real: considerar protocolos como MQTT o HTTP para transmisión a la nube.
- Preparar el sistema para emitir alertas basadas en condiciones específicas (ej. detección de gas o ruido elevado).

Seguridad Básica (aplicable a entorno físico)

- Uso de identificadores únicos por dispositivo.
- Encriptación de datos en tránsito (cuando se conecten a la red).
- Control de acceso a interfaces y configuraciones.

Configuración en Wokwi

Sensor	Pin sugerido	Tipo de Lectura
PIR	D2	Digital (<code>HIGH/LOW</code>)
MQ-2	A0	Analógica (ppm)
Micrófono	A1	Analógica (nivel)

Se recomienda utilizar resistencias de pull-down o pull-up según el comportamiento observado en la simulación para evitar fluctuaciones no deseadas.

6.2. Information Architecture

En esta sección, se va a presentar la estructura del software según cada segmento objetivo. Además, los elementos que emplearan para la navegación de esta.

6.2.2. Labeling Systems

El sistema de etiquetado o Labelling System de AlquilaFácil está diseñado para proporcionar claridad y coherencia en la presentación de información, facilitando la comprensión rápida y eficiente de los contenidos por parte del usuario. Este sistema emplea etiquetas visuales y textuales en diferentes áreas clave de la aplicación:

• **Etiquetas de búsqueda y filtro:**

En la parte superior de la interfaz, los campos de búsqueda y filtro están claramente etiquetados con íconos universales, como una lupa para la búsqueda y un embudo para los filtros. Estos íconos, junto con textos descriptivos, permiten que los usuarios identifiquen rápidamente las funciones y realicen acciones sin confusión.

• **Etiquetas en tarjetas de espacios:**

Cada tarjeta de espacio cuenta con etiquetas descriptivas que destacan la ubicación y el precio, las cuales están claramente diferenciadas en la interfaz. Estas etiquetas ayudan a los usuarios a identificar los detalles más relevantes de un espacio sin necesidad de interactuar más allá del listado inicial.

• **Iconos como Etiquetas Visuales:**

La aplicación utiliza iconos ampliamente reconocidos, como la estrella para "puntuaciones" y el botón de "más" para añadir nuevos espacios, que actúan como etiquetas visuales. Estos iconos, en combinación con sus posiciones estratégicas, ofrecen una experiencia intuitiva, reduciendo la necesidad de explicaciones textuales adicionales.

• **Categorías de espacios:**

Los espacios se pueden clasificar por diferentes tipos o características, lo que permite el uso de etiquetas de categoría. Estas etiquetas aparecen en los resultados de búsqueda para que los usuarios identifiquen rápidamente el tipo de espacio que están viendo (por ejemplo, oficinas, espacios de coworking, etc.).

6.2.3. Searching Systems

El motor de búsqueda es fundamental para que los usuarios encuentren rápidamente detalles específicos

Características claves:

- **Búsqueda por ubicación:**
Los usuarios podrán buscar centros para eventos cercanos a su ubicación actual o especificar una ubicación deseada.
- **Búsqueda por características:**
Los usuarios podrán buscar características específicas, como piscina, parrilla, pet friendly, vigilancia.
- **Filtros avanzados:**
Se proporcionarán filtros para refinar la búsqueda, como calificaciones, precios y disponibilidad.
- **Resultados relevantes:**
El sistema de búsqueda mostrará resultados relevantes y ordenados de acuerdo con la ubicación y otros criterios.

6.2.4. SEO Tags and Meta Tags

Las meta-etiquetas nos permiten codificar y especificar metadatos en una página web. Aunque no son visibles para los usuarios, los navegadores y rastreadores web las leen. Estas etiquetas facilitan el análisis de archivos HTML y ayudan en el mantenimiento del contenido. Además, influyen en el posicionamiento de nuestra página en los motores de búsqueda.

- **Título:**
Las meta etiquetas nos permiten codificar y especificar metadatos en una página web. Aunque no son visibles para los usuarios, los navegadores y rastreadores web las leen. Estas etiquetas facilitan el análisis de archivos HTML y ayudan en el mantenimiento del contenido. Además, influyen en el posicionamiento de nuestra página en los motores de búsqueda.

...

```
<title>Register your processes with AlquilaFacil</title>
```

...

- **Codificación de caracteres:**
Se decidió usar el utf-8 por la eficiencia de memoria. Es más eficiente en términos de memoria para caracteres del BMP (Plano Multilingüe Básico, que incluye la mayoría de los caracteres comunes).

...

```
<meta charset="utf-8">
```

...

- **Descripción:**
Esta etiqueta meta nos permite ofrecer un resumen del contenido de la página web. En ella, proporcionamos una breve descripción de lo que los usuarios pueden esperar visualizar en la página.

...

```
<meta name="description" content="AlquilaFacil is a web application focused on publish and rent centers for events"/>
```

...

- **Palabras clave:**
En esta etiqueta se pone las palabras claves relacionadas con el tema o contenido de la página web.

...

```
<meta name="keywords" content="publish, rent, management, application, announcements, centers"/>
```

...

- **Autor y derechos de autor:**
Se utiliza para registrar la información del autor de la página web y la propiedad y derechos de autor.

...

```
<meta name="author" content="AlquilaFacil"/>
```

...

6.2.5. Navigation Systems

El Navigation System de AlquilaFácil está diseñado para proporcionar una experiencia de usuario fluida, intuitiva y eficiente, permitiendo a los usuarios moverse por la aplicación de manera rápida y sin complicaciones. Este sistema incluye varios elementos clave que optimizan el acceso a las distintas funciones y secciones de la aplicación:

- **Barra de Navegación Inferior:**

La aplicación cuenta con una barra de navegación fija en la parte inferior de la pantalla que agrupa las principales funcionalidades. Los iconos incluidos, como la lupa (búsqueda), la campana (notificaciones), el botón de "+" (añadir nuevo espacio), el calendario (gestión de eventos o reservas) y el perfil (opciones del usuario), permiten un acceso directo a cada sección, facilitando la navegación entre las funciones esenciales sin necesidad de menús desplegables o múltiples clics.

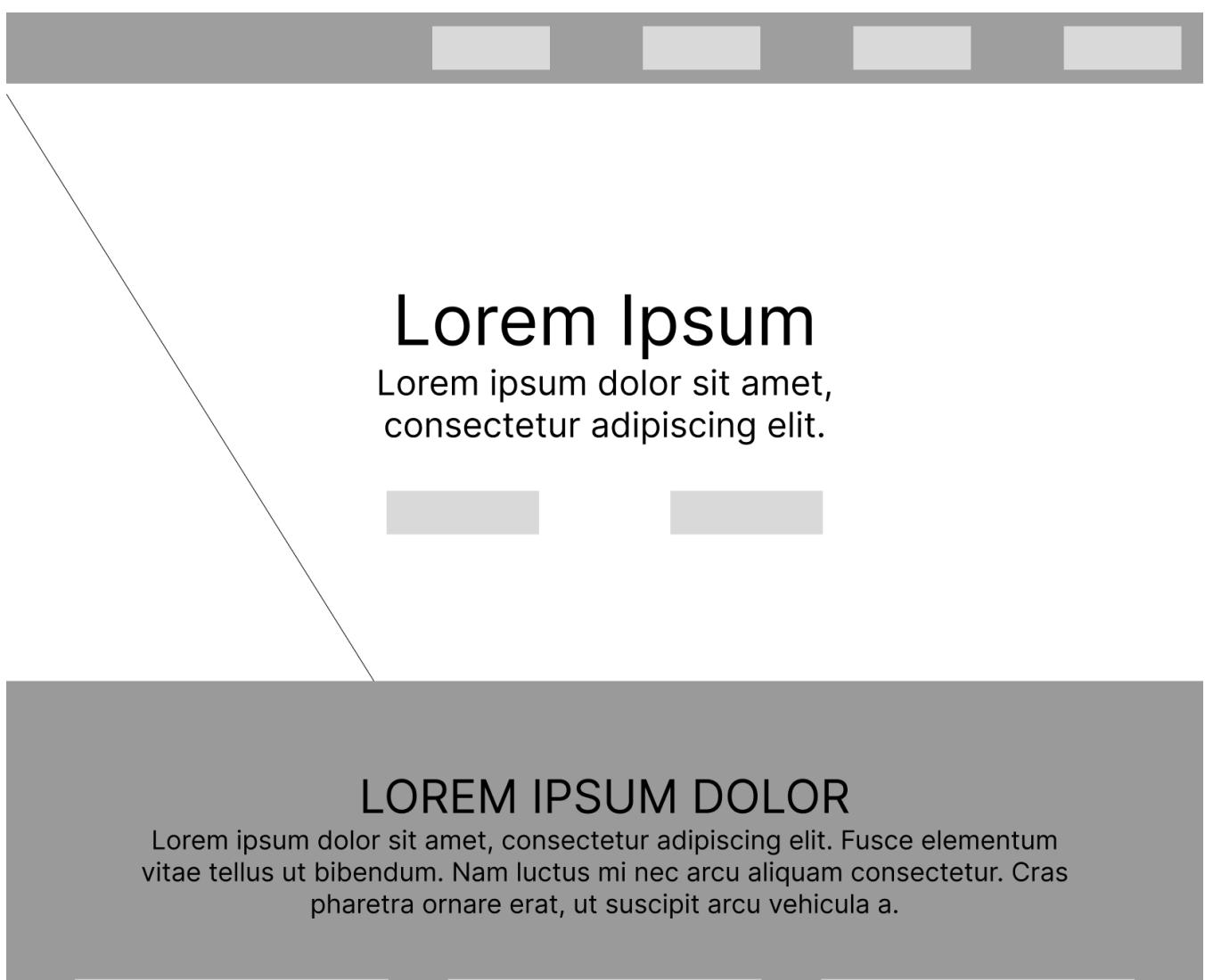
- **Navegación Jerárquica:**

AlquilaFácil implementa un sistema de navegación jerárquica en el que los usuarios pueden profundizar en los detalles de un espacio o acción, pero siempre tienen la opción de volver a la pantalla anterior de manera sencilla. Este enfoque evita que los usuarios se pierdan en la aplicación, manteniendo siempre claro en qué nivel de la jerarquía se encuentran.

6.3. Landing Page UI Design

6.3.1. Landing Page Wireframe

A continuación, se presentan los wireframes de la Landing Page de AlquilaFácil, los cuales muestran la estructura y disposición de los elementos principales de la página sin incluir detalles visuales finales. Estos wireframes sirven como guía inicial para el diseño y desarrollo de la interfaz.

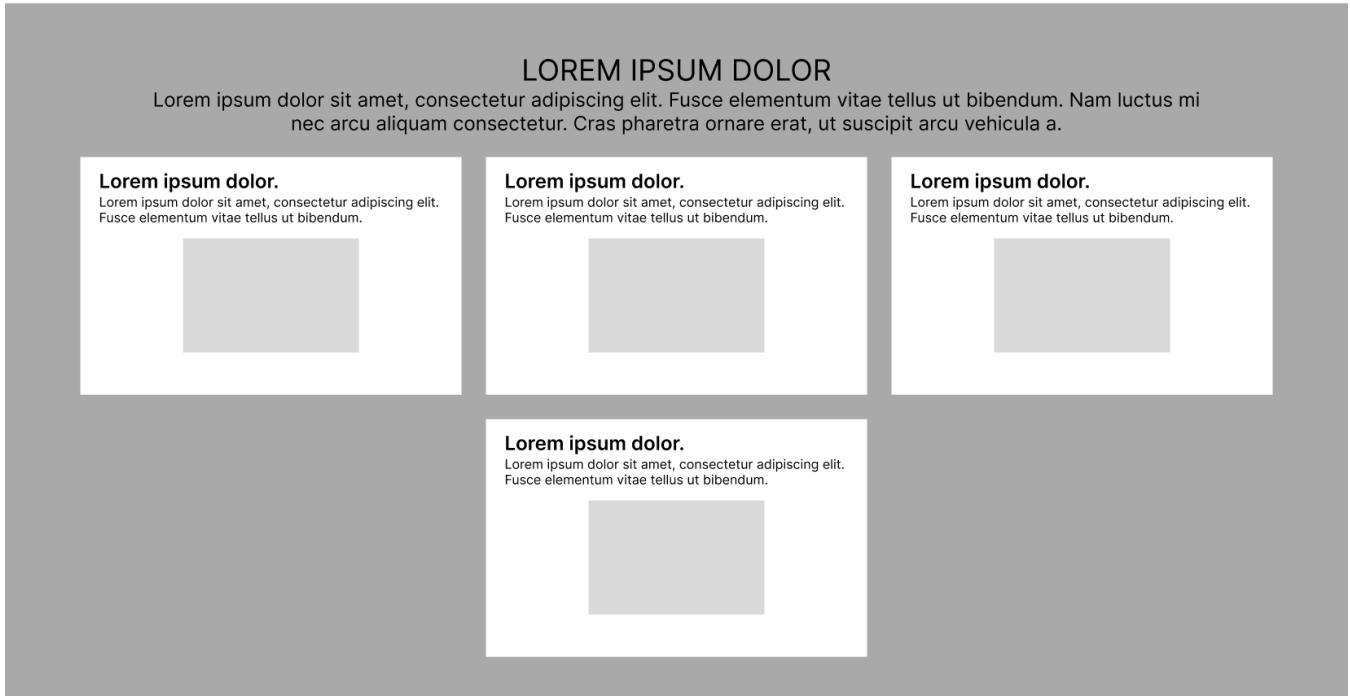




LOREM IPSUM DOLOR

Consectetur adipiscing elit. Fusce elementum vitae tellus ut bibendum. Nam luctus mi nec arcu aliquam consectetur. Cras pharetra ornare erat, ut suscipit arcu vehicula a.

Section 1



LOREM IPSUM DOLOR

Fusce elementum vitae.
Fusce elementum vitae.

LOREM IPSUM DOLOR
Fusce.
Fusce .

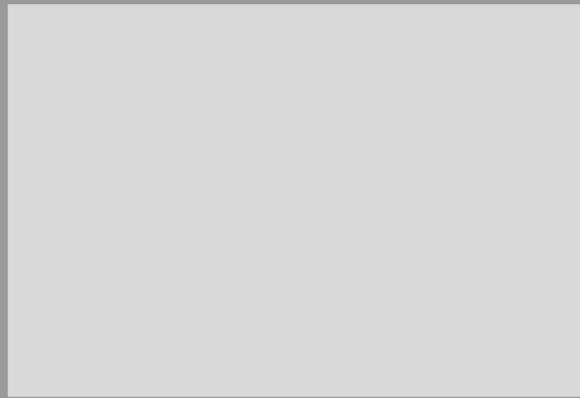
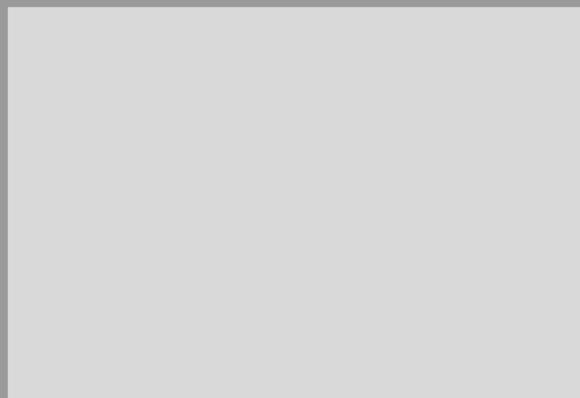
LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. FUSCE ELEMENTUM VITAE TELLUS UT BIBENDUM.

Lorem Ipsum

**Lorem ipsum dolor sit amet,
 consectetur adipiscing elit.**

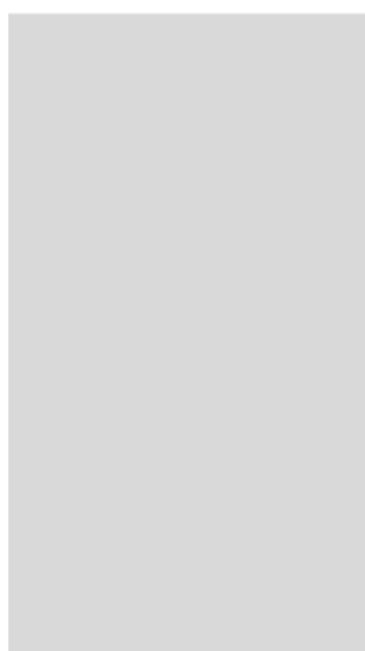
LOREM IPSUM DOLOR

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce elementum vitae tellus ut bibendum. Nam luctus mi nec arcu aliquam consectetur. Cras pharetra ornare erat, ut suscipit arcu vehicula a.



LOREM IPSUM DOLOR

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce elementum vitae tellus ut bibendum. Nam luctus mi nec arcu aliquam consectetur. Cras pharetra ornare erat, ut suscipit arcu vehicula a.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce elementum vitae tellus ut bibendum. Nam luctus mi nec arcu aliquam consectetur. Cras pharetra ornare erat, ut suscipit arcu vehicula a.

Lorem ipsum dolor sit amet



LOREM IPSUM DOLOR

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce elementum vitae tellus ut bibendum. Nam luctus mi nec arcu aliquam consectetur. Cras pharetra ornare erat, ut suscipit arcu vehicula a.

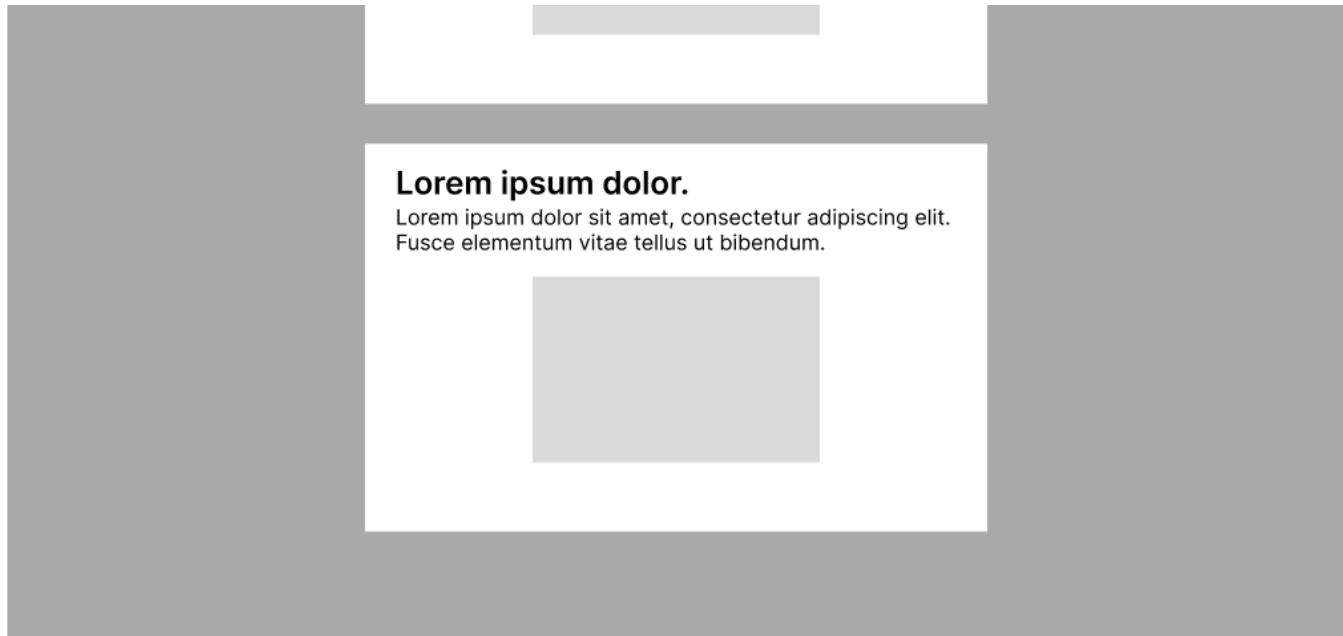
Lorem ipsum dolor.

Etiam ipsum dolor sit amet, consectetur adipiscing elit.
Fusce elementum vitae tellus ut bibendum.

Lorem ipsum dolor.

Etiam posuimus.
Lorem ipsum dolor sit amet, consectetur adipiscing elit.
Fusce elementum vitae tellus ut bibendum.

Text



LOREM IPSUM DOLOR.

Lore ipsum dolor sit amet, consectetur adipiscing elit.
Fusce elementum vitae tellus ut bibendum.



LOREM IPSUM DOLOR

Fusce elementum vitae.
Fusce elementum vitae.

LOREM IPSUM DOLOR

Fusce.
Fusce .

Lore ipsum dolor sit amet, consectetur adipiscing elit. Fusce elementum vitae tellus ut bibendum.

Enlace: <https://www.figma.com/design/Sk1zmVhp0zEdPv97Vlxvwm/Landing-Page?node-id=4-202&t=0W0h3rzLTOTQUsPF-1>

6.3.2. Landing Page Mock-up

Los mock-ups de la Landing Page presentan el diseño visual completo con colores, tipografía, imágenes y todos los elementos gráficos finales. Estos prototipos de alta fidelidad permiten visualizar exactamente cómo se verá la página cuando esté implementada.

AlquilaFácil

El espacio ideal para tus eventos, al alcance de tus manos

Consíguelo en el App Store

DISPONIBLE EN Google Play



DESCUBRE ESPACIOS ÚNICOS

Explora una amplia variedad de espacios para eventos, reuniones o momentos especiales. Filtra por ubicación, tamaño, precio, y encuentra el lugar perfecto que se adapte a tus necesidades.



Salón de eventos moderno

Capacidad: 150 personas

[Ver detalles](#)



Oficina compartida en el centro

Capacidad: 25 personas

[Ver detalles](#)



Terraza al aire libre

Capacidad: 50 personas

[Ver detalles](#)

PUBLICA TU ESPACIO

En **AlquilaFácil**, tienes la oportunidad de compartir tu espacio único con una amplia audiencia. Al registrarte en nuestra aplicación, podrás publicar tu espacio para eventos, reuniones o celebraciones y conectarte con personas que buscan el lugar perfecto.

Simplemente describe tu espacio, establece tu disponibilidad y precio, y comienza a recibir reservas. **AlquilaFácil** te proporciona la plataforma ideal para maximizar el uso de tu espacio y generar ingresos adicionales.

[Comienza a publicar](#)



Algunos de estos usuarios de **AlquilaFácil** están en busca de tu espacio:



¿Qué esperas para comenzar a alquilar tus espacios?

BENEFICIOS DE USAR ALQUILA FÁCIL

Con AlquilaFácil, tanto anfitriones como usuarios disfrutan de una experiencia única y enriquecedora. Aquí hay algunas ventajas clave:

Variedad de espacios

Accede a una amplia gama de lugares únicos que se adaptan a cualquier tipo de evento.

Sala recreacional 1
San Miguel, Perú
Aforo: 100 personas
Propietario: Italo Luna
Descripción: Sala con decoración moderna perfecta para actividades musicales.

Búsqueda personalizada

Filtrá los espacios según ubicación, tamaño y precio para encontrar lo que realmente necesitas.

Comentarios de reservas

La confianza es clave al reservar un espacio. Aquí podrás leer las experiencias de otros usuarios que han utilizado AlquilaFácil, ayudándote a tomar decisiones informadas.

Gestión sencilla

Los anfitriones pueden gestionar sus reservas y disponibilidad de manera sencilla desde la aplicación.

Contáctanos
Email: info@alquilafacil.com
Teléfono: +51 991 042 243

Síguenos

© 2025 AlquilaFácil. Todos los derechos reservados. | [Política de Privacidad](#) | [Términos de Servicio](#)

AlquilaFácil

EMPEZAR

AlquilaFácil

El espacio ideal para...

El espacio ideal para
tus
eventos, al alcance de
tus manos



DESCUBRE ESPACIOS ÚNICOS

Explora una amplia variedad de espacios para eventos, reuniones o momentos especiales. Filtra por ubicación, tamaño, precio, y encuentra el lugar perfecto que se adapte a tus necesidades.



Salón de eventos moderno

Capacidad: 150 personas

[Ver detalles](#)





Oficina compartida en el centro

Capacidad: 25 personas

[Ver detalles](#)



Terraza al aire libre

Capacidad: 50 personas

[Ver detalles](#)

PUBLICA TU ESPACIO

En **AlquilaFácil**, tienes la oportunidad de compartir tu espacio único con una amplia audiencia. Al registrarte en nuestra aplicación, podrás publicar tu espacio para eventos, reuniones o celebraciones y conectarte con personas que buscan el lugar perfecto.

Simplemente describe tu espacio, establece tu disponibilidad y precio, y comienza a recibir reservas. **AlquilaFácil** te proporciona la plataforma ideal para maximizar el uso de tu espacio y generar ingresos adicionales.

Comienza a publicar



Algunos de estos usuarios de **AlquilaFácil** están en busca de tu espacio:



¿Qué esperas para comenzar a alquilar tus espacios?

BENEFICIOS DE USAR ALQUILAFÁCIL

Con **AlquilaFácil**, tanto anfitriones como usuarios disfrutan de una experiencia única y enriquecedora.
Aquí hay algunas ventajas clave:

Variedad de espacios

Accede a una amplia gama de lugares únicos que se adaptan a cualquier tipo de evento.



Sala recreacional 1

San Miguel, Perú

Aforo: 100 personas

Propietario: Italo Luna

Descripción:

Sala con decoración moderna perfecta para actividades sociales.

Búsqueda personalizada

Filtra los espacios según ubicación, tamaño y precio para encontrar lo que realmente necesitas.



Filtros

Tipo de espacio



Casa de playa



Casa urbana



Salones elegantes



Casa de campo

Comentarios de reservas

La confianza es clave al reservar un espacio. Aquí podrás leer las experiencias de otros usuarios que han utilizado AlquilaFácil, ayudándote a tomar decisiones informadas.



Comentarios



Angello Sosa



Me gustó mucho el interior del local, es muy espacioso



Fernando Salgado



El espacio estaba sucio cuando ingresé

Gestión sencilla

Los anfitriones pueden gestionar sus reservas y disponibilidad de manera sencilla desde la aplicación.

Toca en una fecha para ver mas detalles o agendar un nuevo evento

Lu	Ma	Mi	Ju	Vi	Sa	Do
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

Contáctanos

Email: info@alquilafacil.com

Teléfono: +51 991 042 243

Síguenos



© 2025 AlquilaFácil. Todos los derechos reservados. |
[Política de Privacidad](#) | [Términos de Servicio](#)

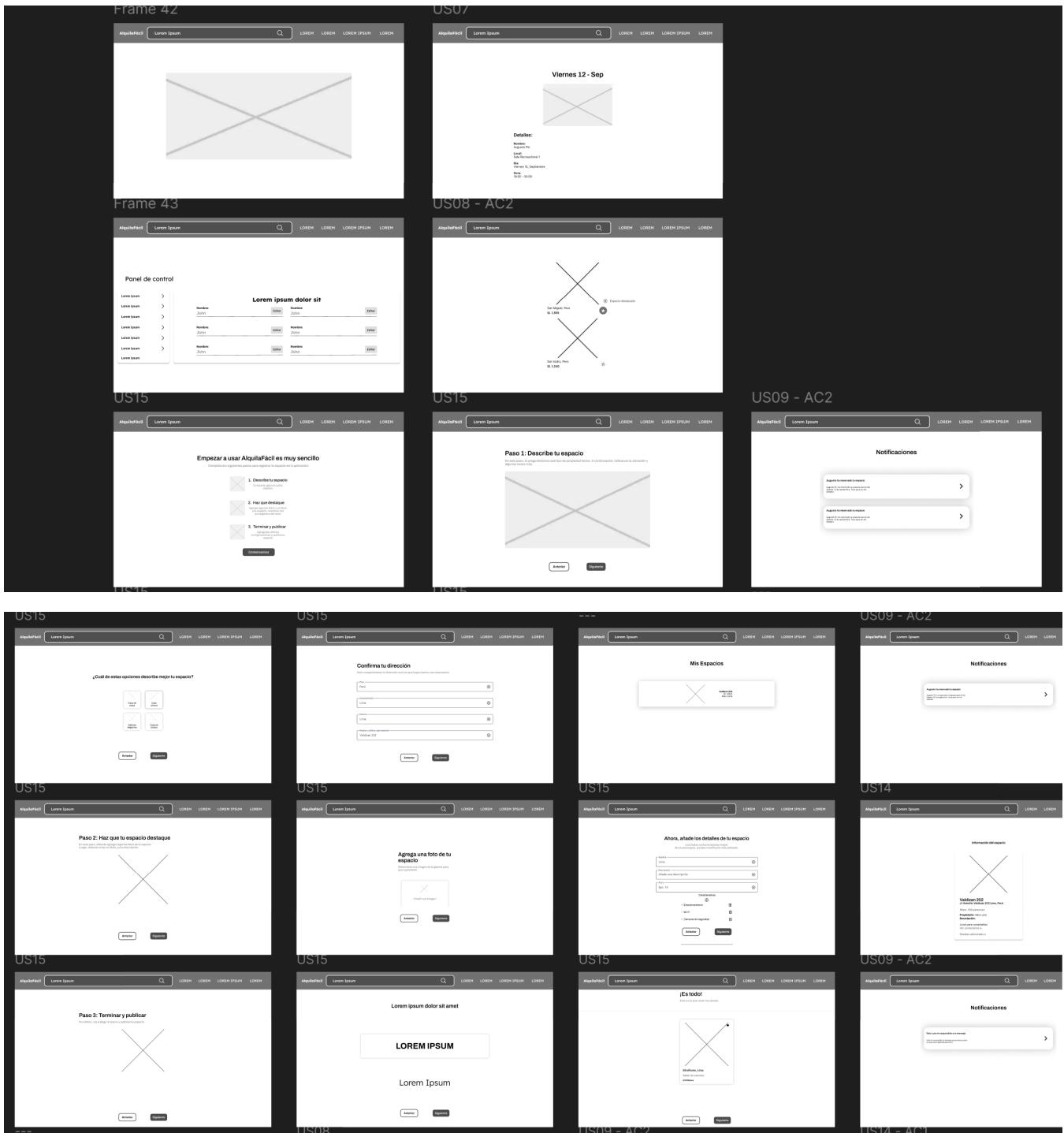
Enlace: <https://www.figma.com/design/Sk1zmVhp0zEdPv97Vlxvwm/Landing-Page?node-id=4-422&p=f&t=kZ23Arq5BMQvyN7O-0>

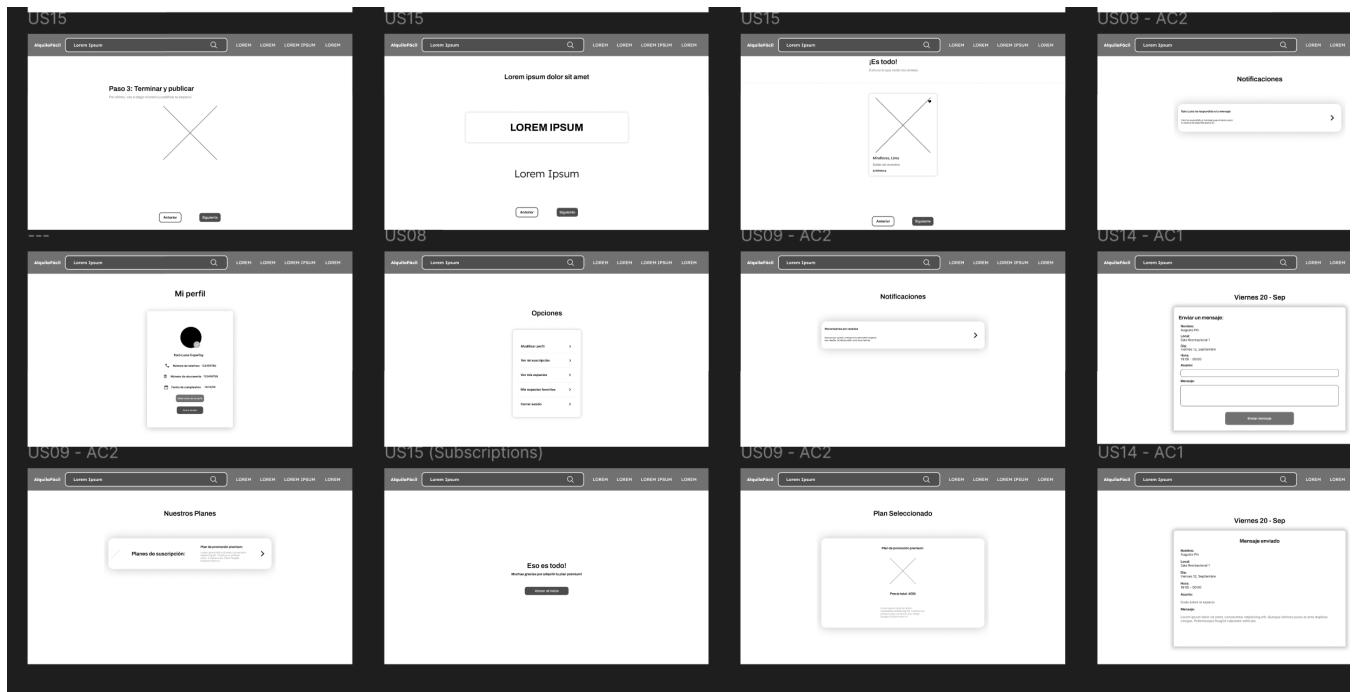
6.4. Applications UX/UI Design

6.4.1. Applications Wireframes.

Los wireframes de la aplicación web muestran la estructura y organización de las diferentes pantallas y funcionalidades del sistema. Estos diseños de baja fidelidad permiten visualizar la arquitectura de la información y el flujo de navegación antes de aplicar el diseño visual final.

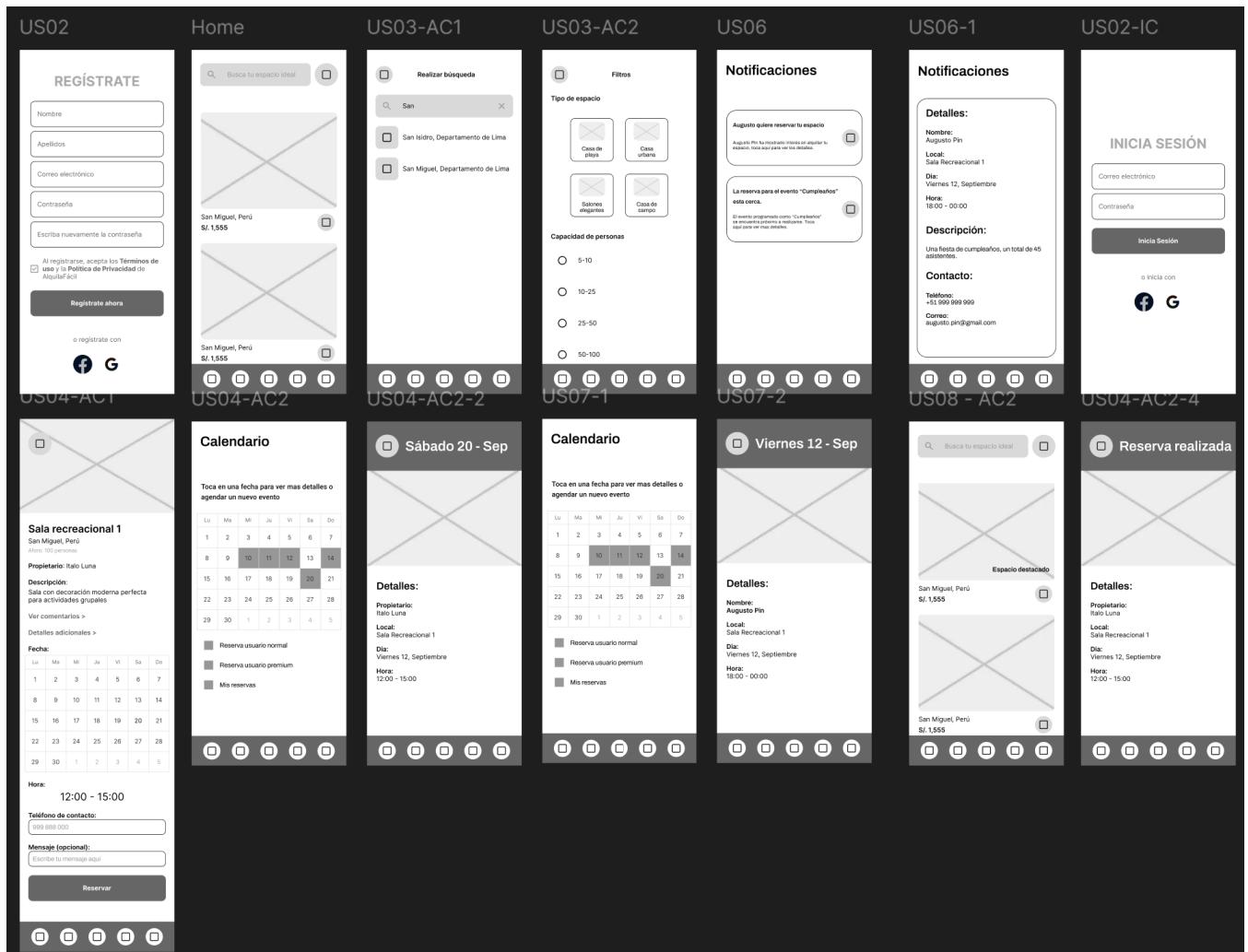
The image displays six wireframe prototypes for a web application named 'AlquilaFácil'. The prototypes are arranged in two columns and three rows. The first row shows the 'NavBar' (top navigation bar) and 'US01-Wireframe' (login screen). The second row shows 'Frame 37' (search results), 'Frame 341' (local details), and 'US02' (registration screen). The third row shows 'US03-AC1' (a dashboard or account center) and 'US03-AC2' (a settings or preferences page). Each wireframe includes placeholder text ('LOREM ipsum') and various UI elements like buttons, dropdowns, and icons.





Enlace WebPage: <https://www.figma.com/design/8e7OrbLVZLRQHAgSK4pDYD/Web-App?node-id=1-207&t=hZfD2aBdxlDEywVP-1>

Los wireframes de la aplicación móvil presentan el diseño estructural adaptado a dispositivos móviles, considerando las particularidades de la interfaz táctil y las limitaciones de espacio en pantalla. Estos esquemas permiten planificar la experiencia de usuario en dispositivos iOS y Android.



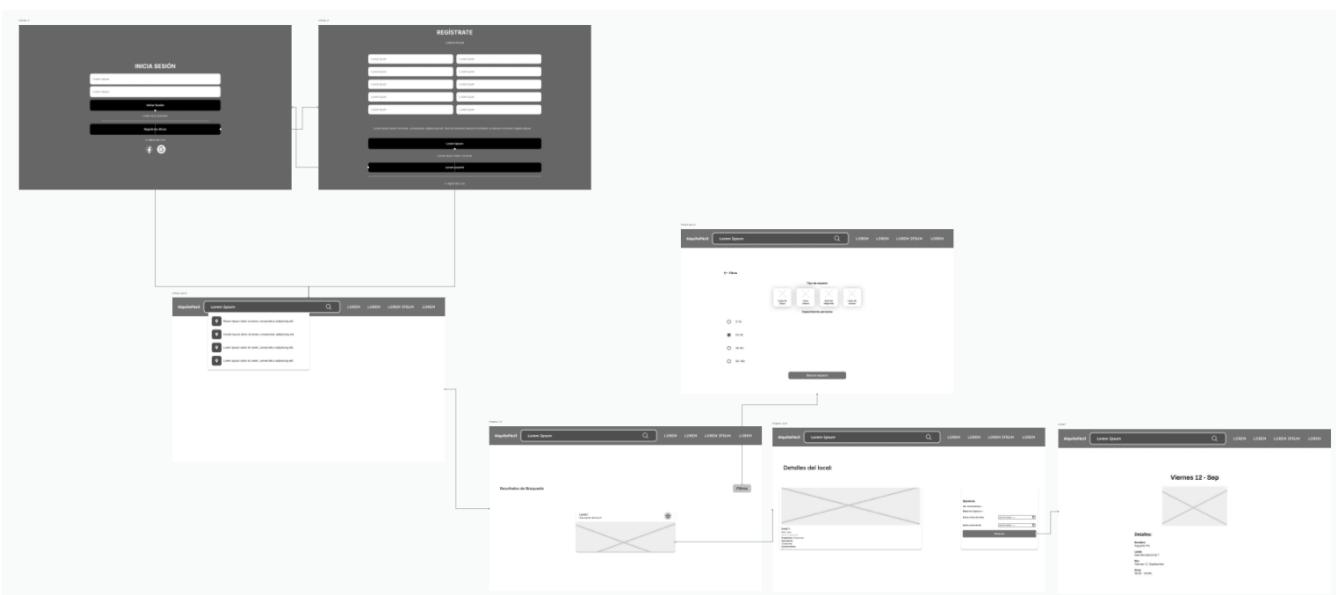
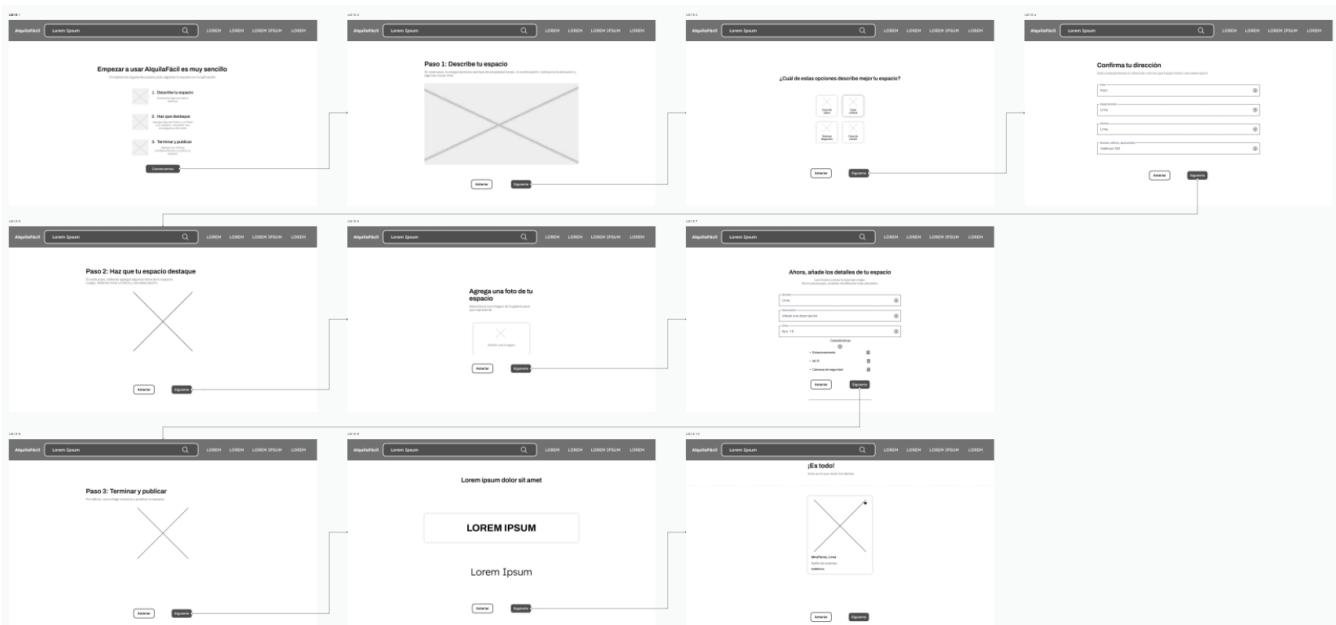


Enlace Mobile: <https://www.figma.com/design/ZgcH4GPA3oBLZf4SvVs52Z/Mobile-App?node-id=1-207&t=hZfD2aBdxlDEywVP-1>

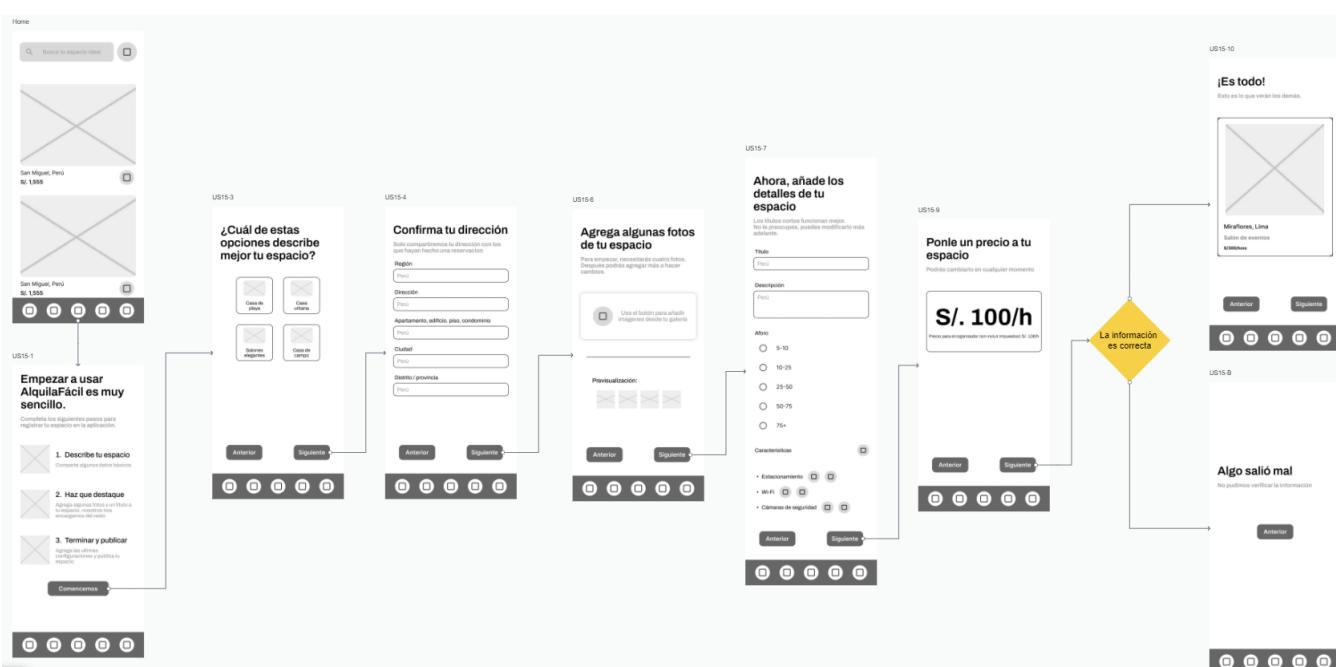
6.4.2. Applications Wireflow Diagrams

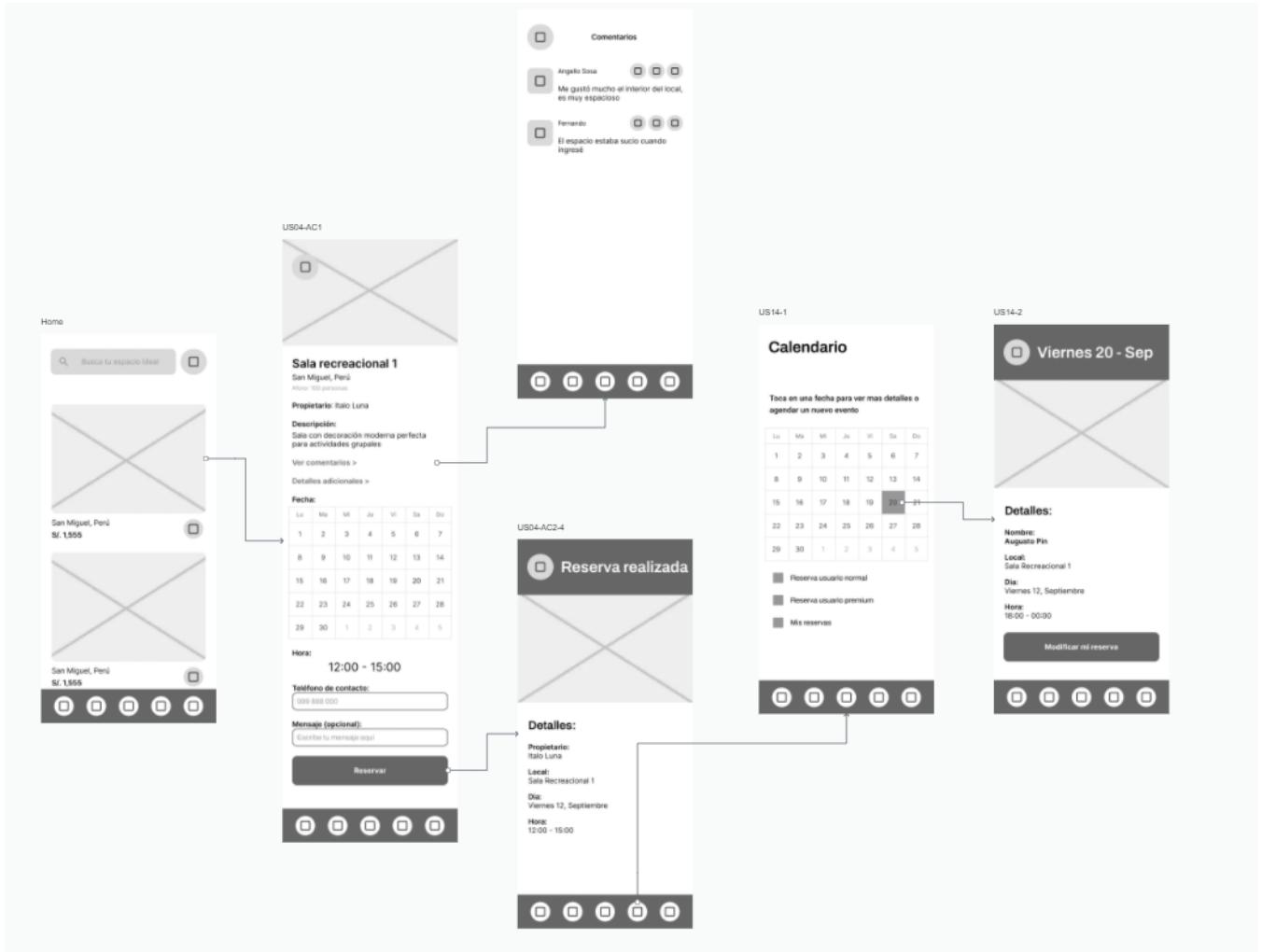
Los wireflows combinan los wireframes con los flujos de navegación para representar cómo el usuario interactúa con la aplicación paso a paso. Permiten visualizar las pantallas en secuencia junto con las conexiones entre ellas, mostrando las posibles rutas y decisiones que puede tomar el usuario. De esta forma, facilitan la comprensión del comportamiento del sistema y la lógica detrás de cada interacción.

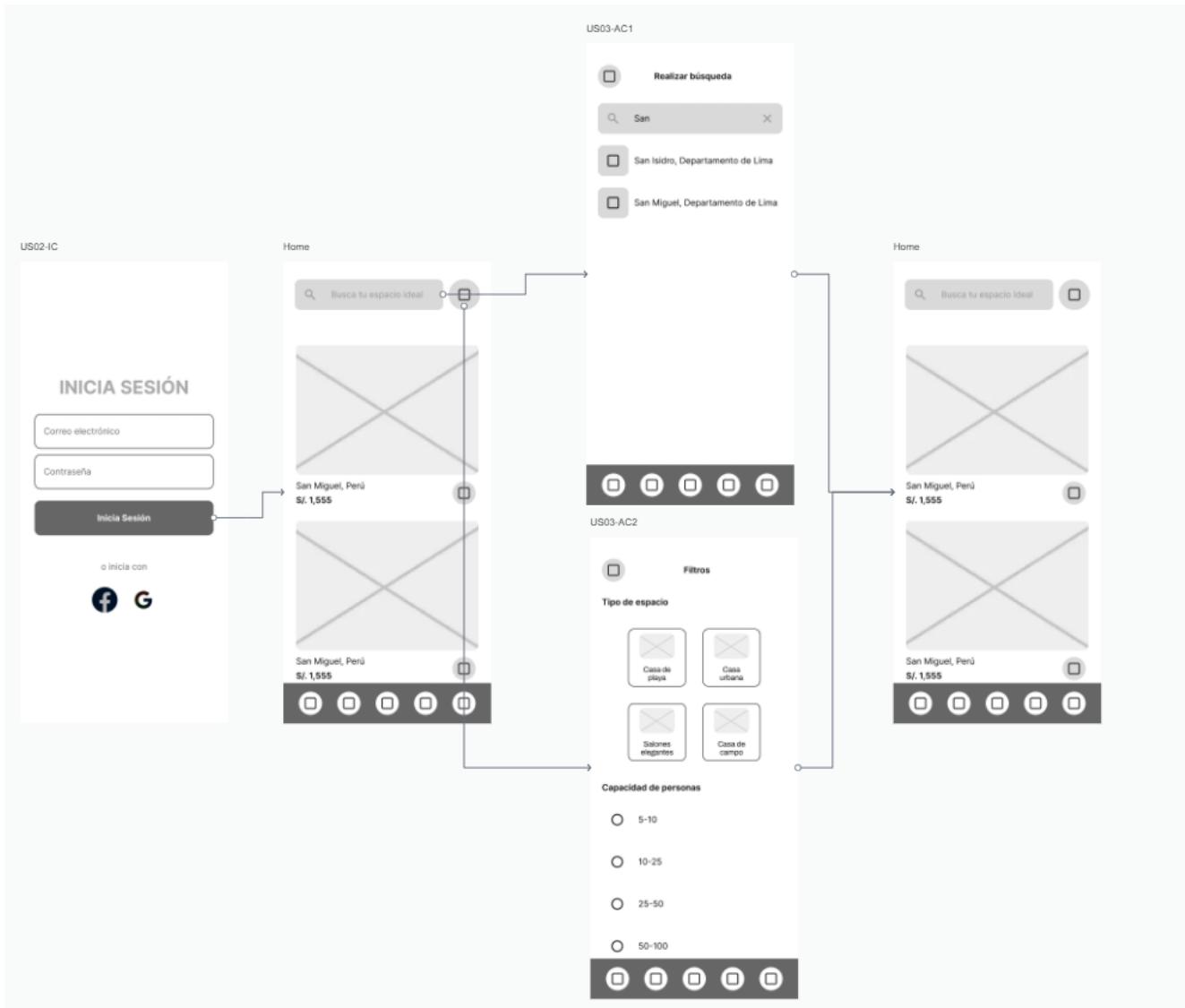
Web



Mobile







Enlace de los wireflows: <https://overflow.io/s/1IDNN3EF>

6.4.3. Applications Mock-ups

Los mock-ups de la aplicación web presentan el diseño visual completo de las diferentes pantallas del sistema, incluyendo colores corporativos, tipografía, iconografía y todos los elementos gráficos de acuerdo a las guías de estilo establecidas. Estos prototipos de alta fidelidad representan fielmente la apariencia final de la aplicación.

US02-1

Frame 37

US02-2

US03-AC1

Frame 341

US03-AC2

Frame 42

US07

US08 - AC2

US05

US05 - AC2

US05

US09 - AC2

US05

US09 - AC2

US05

US05

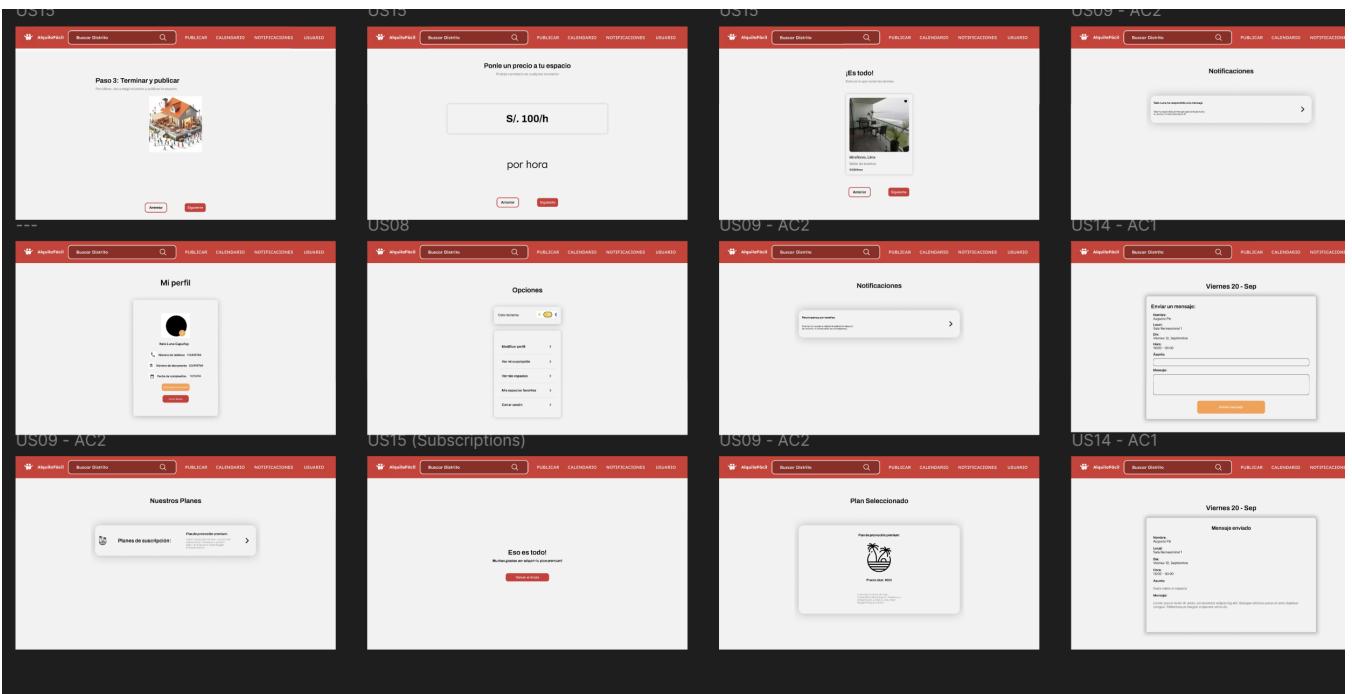
US05

US09 - AC2

US05

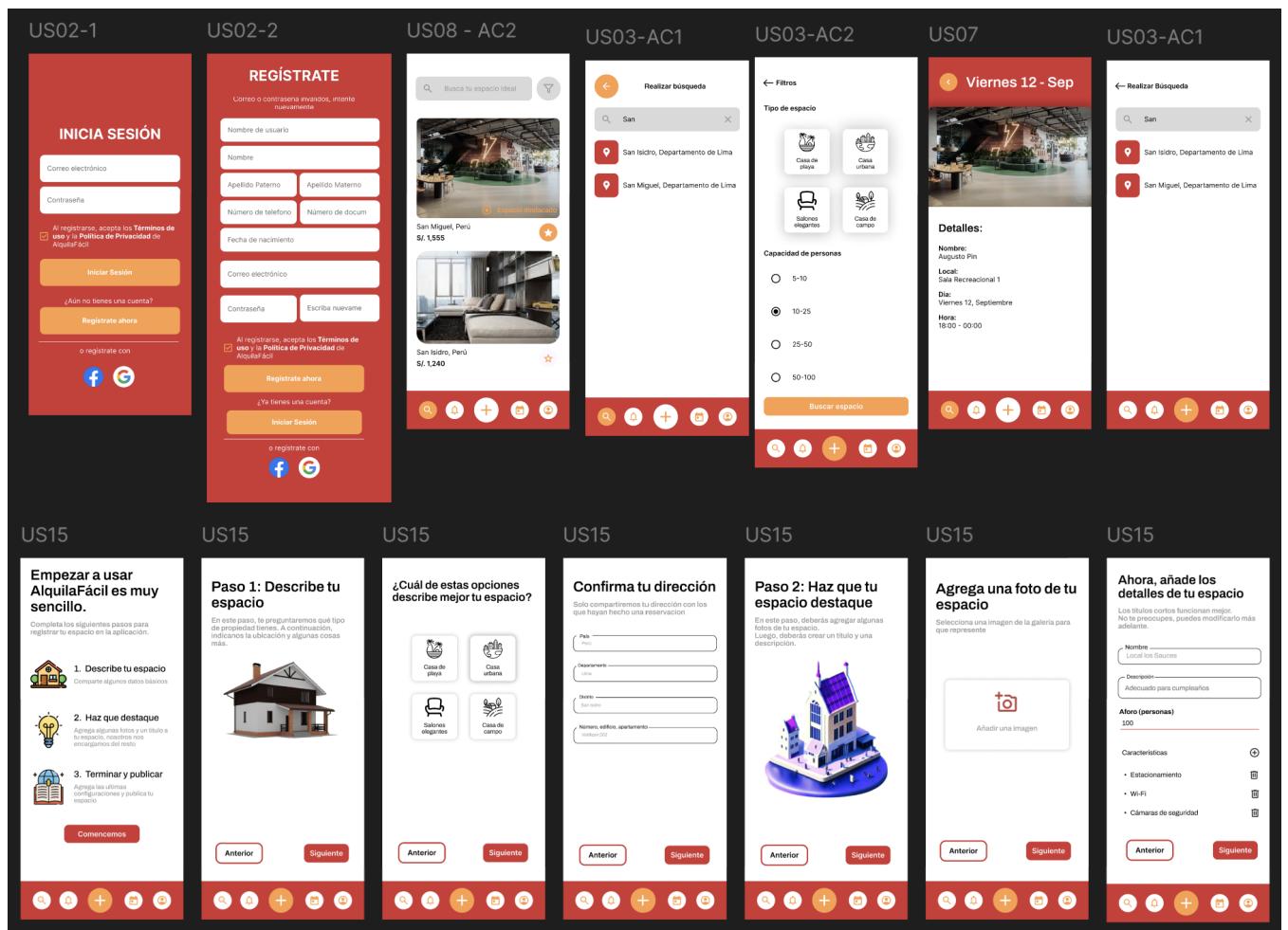
US05

US09 - AC2



Enlace WebPage: https://www.figma.com/design/8e7OrbLVZLRQHAgSK4pDYD/Web-App?node_id=1-206&t=hZfD2aBdxlDEywVP-1

Los mock-ups de la aplicación móvil muestran el diseño final optimizado para dispositivos móviles, incorporando todos los elementos visuales, patrones de interacción táctil y adaptaciones necesarias para ofrecer una experiencia de usuario óptima en smartphones y tablets, tanto para iOS como para Android.



This image displays a grid of 20 mobile application screens, likely from a Figma design, illustrating various user interface components and flow states. The screens are organized into several rows:

- Row 1:** US15 (4 screens), US15 (4 screens), US15 (4 screens), US15 B (4 screens).
- Row 2:** US8 (1 screen), US11-AC01 (1 screen), US14 (1 screen), US5-AC1 (1 screen), US9 - AC2 (1 screen), US14 - AC1 (1 screen).
- Row 3:** US14 - AC1 (1 screen), US14 - AC2 (1 screen), US19 (1 screen), US15 (Subscription) (2 screens), US15 (Subscription) (2 screens), US15 (Subscription) (2 screens), US15 (Subscription) (2 screens).
- Row 4:** Home (1 screen), US06 (1 screen), US04-AC1 (1 screen), US04-AC2 (1 screen), US03-AC2 (1 screen), US14 (1 screen).

The screens show various features such as:

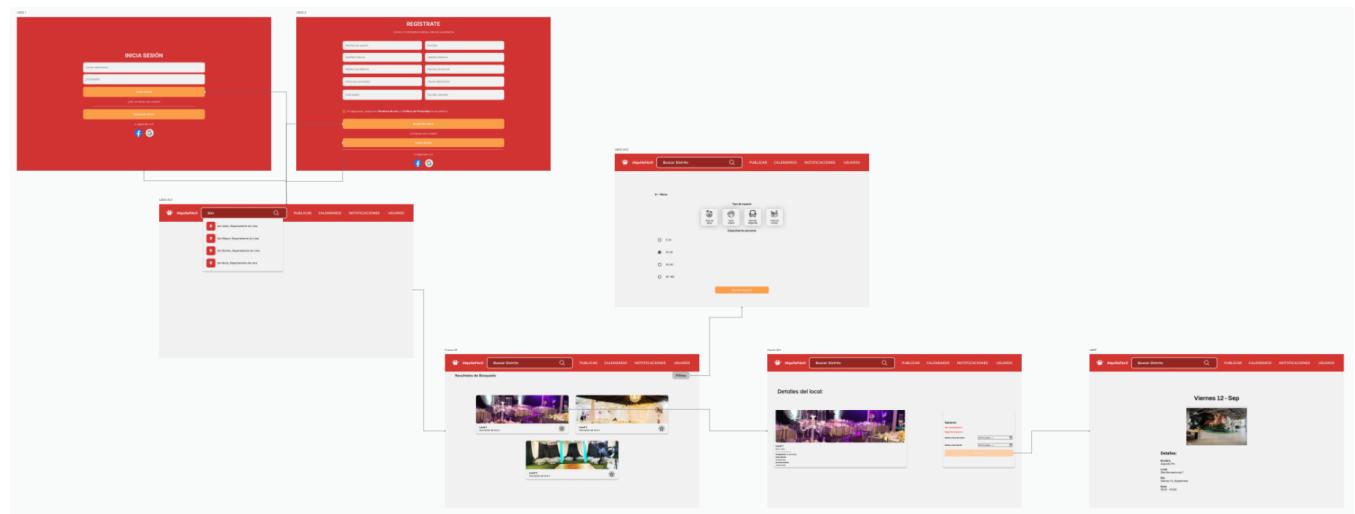
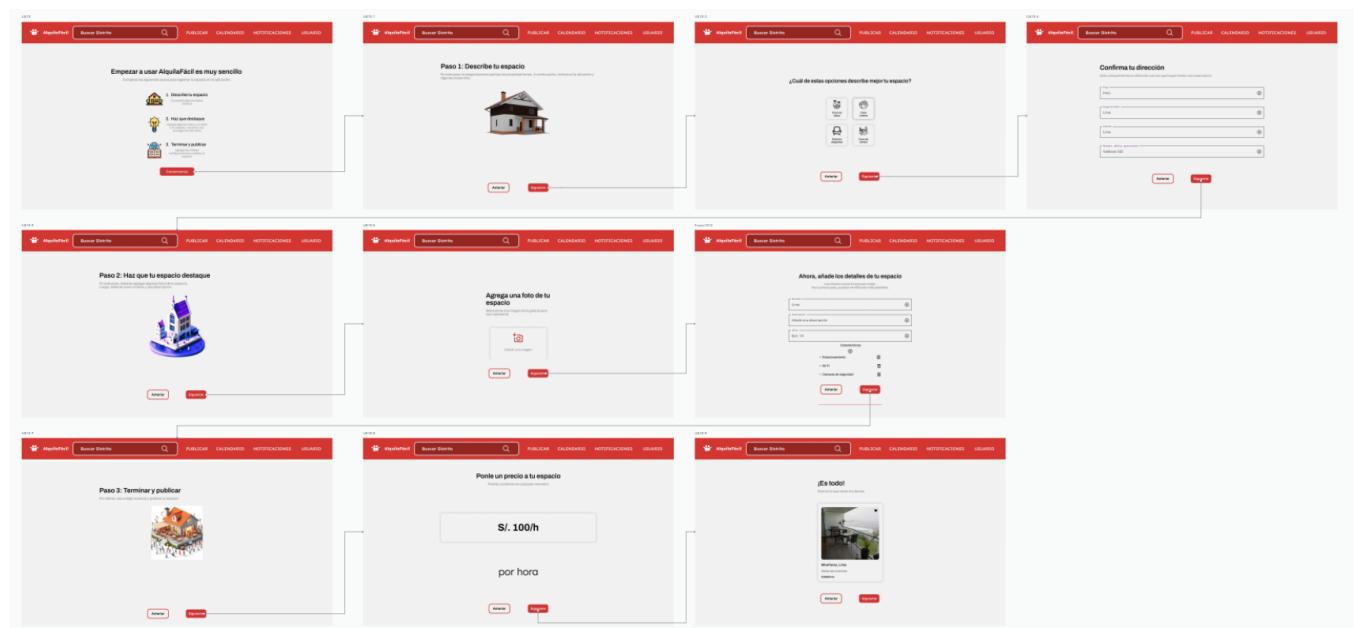
- Paso 3: Terminar y publicar**: Step 3 to finish and publish.
- Ponle un precio a tu espacio**: Set a price for your space.
- ¡Es todo!**: That's it! Confirmation message.
- Algo salió mal**: Something went wrong.
- Mi perfil**: My profile settings.
- Panel de control**: Control panel with navigation links.
- Calendario**: Calendar view showing availability and bookings.
- Comentarios**: Comments section with reviews.
- Notificaciones**: Notifications center.
- Mis espacios favoritos**: Favorite spaces.
- Enviar un mensaje**: Send a message.
- Mensaje enviado**: Message sent confirmation.
- Notificaciones**: Notifications center.
- Adquiere tu suscripción**: Purchase subscription.
- Información del plan**: Plan information.
- Elegir un método de pago**: Choose payment method.
- Precio total: \$500**: Total price.
- Notificaciones**: Notifications center.
- Calendario**: Calendar view.
- Filtros**: Filters for space search.
- Información del espacio**: Space information.

Enlace Mobile: <https://www.figma.com/design/ZgcH4GPA3oBLzf4SvVs5Z/Mobile-App?node-id=1-206&t=hZfD2aBdxlDEywVP-1>

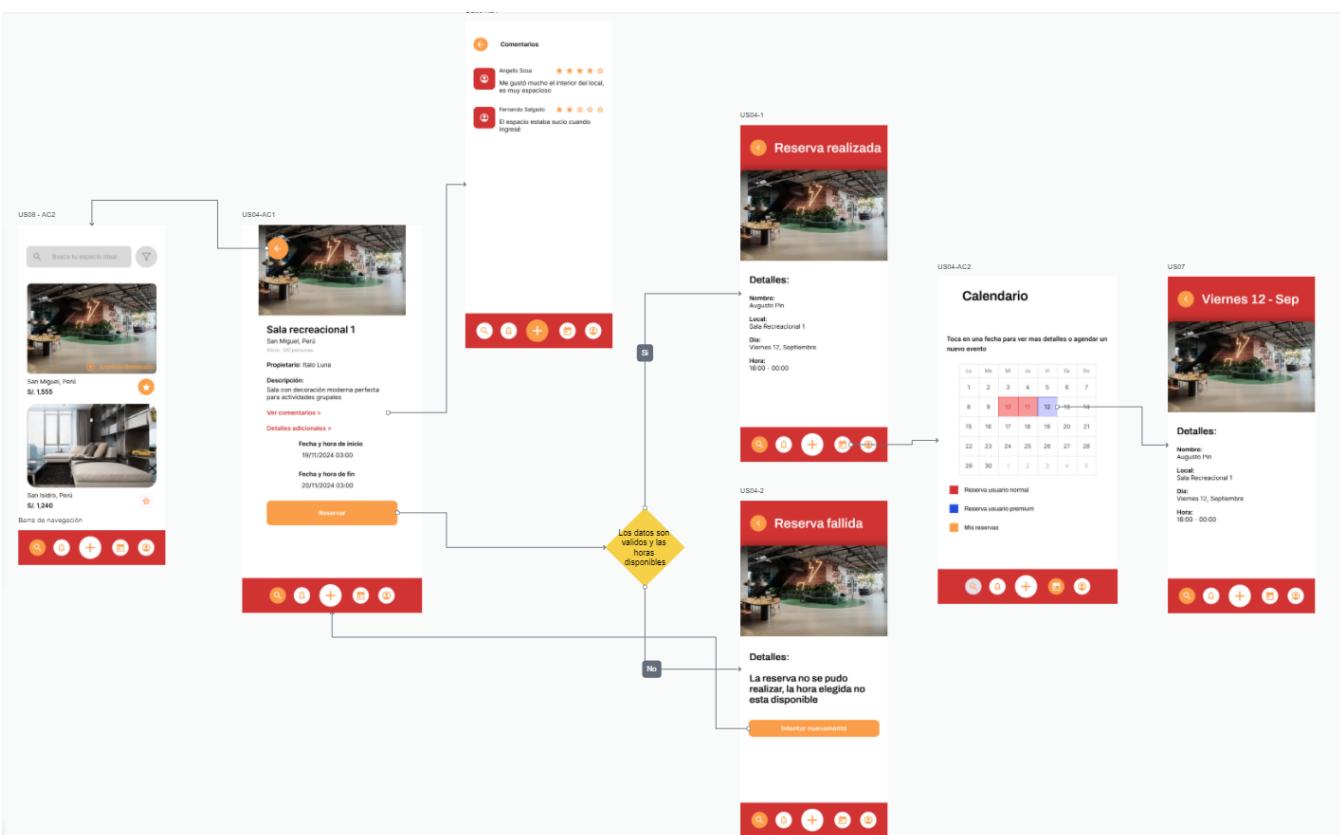
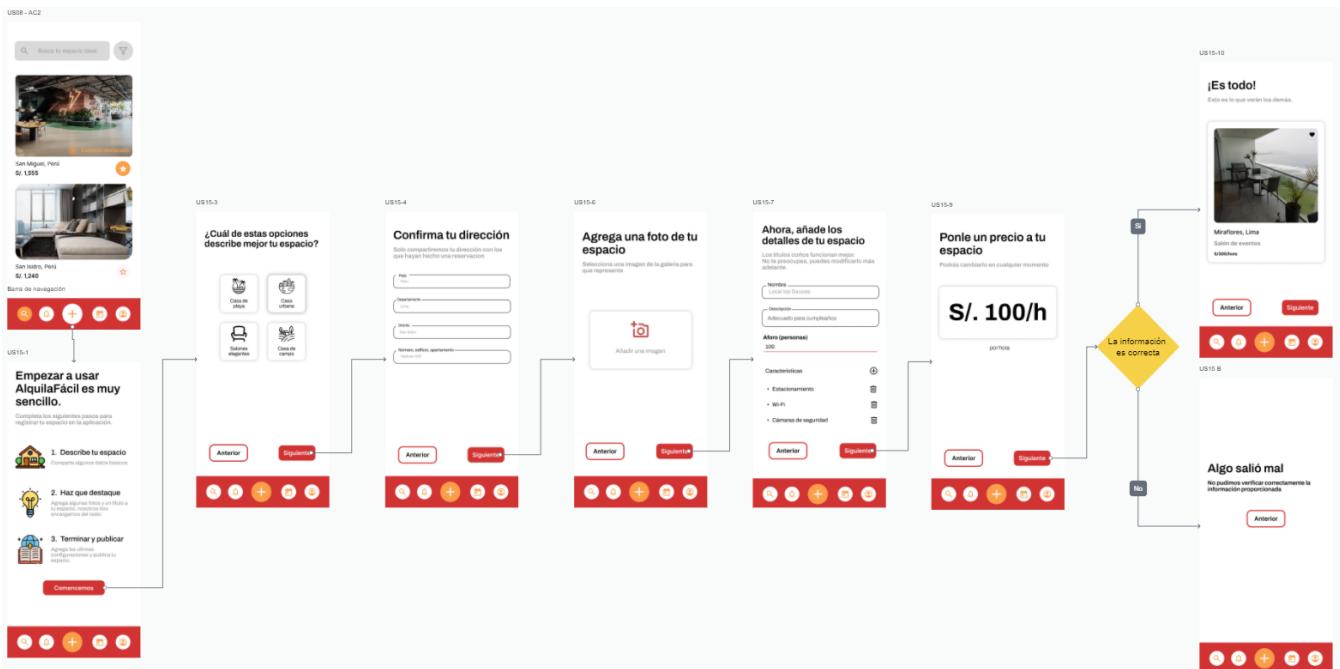
6.4.4. Applications User Flow Diagrams

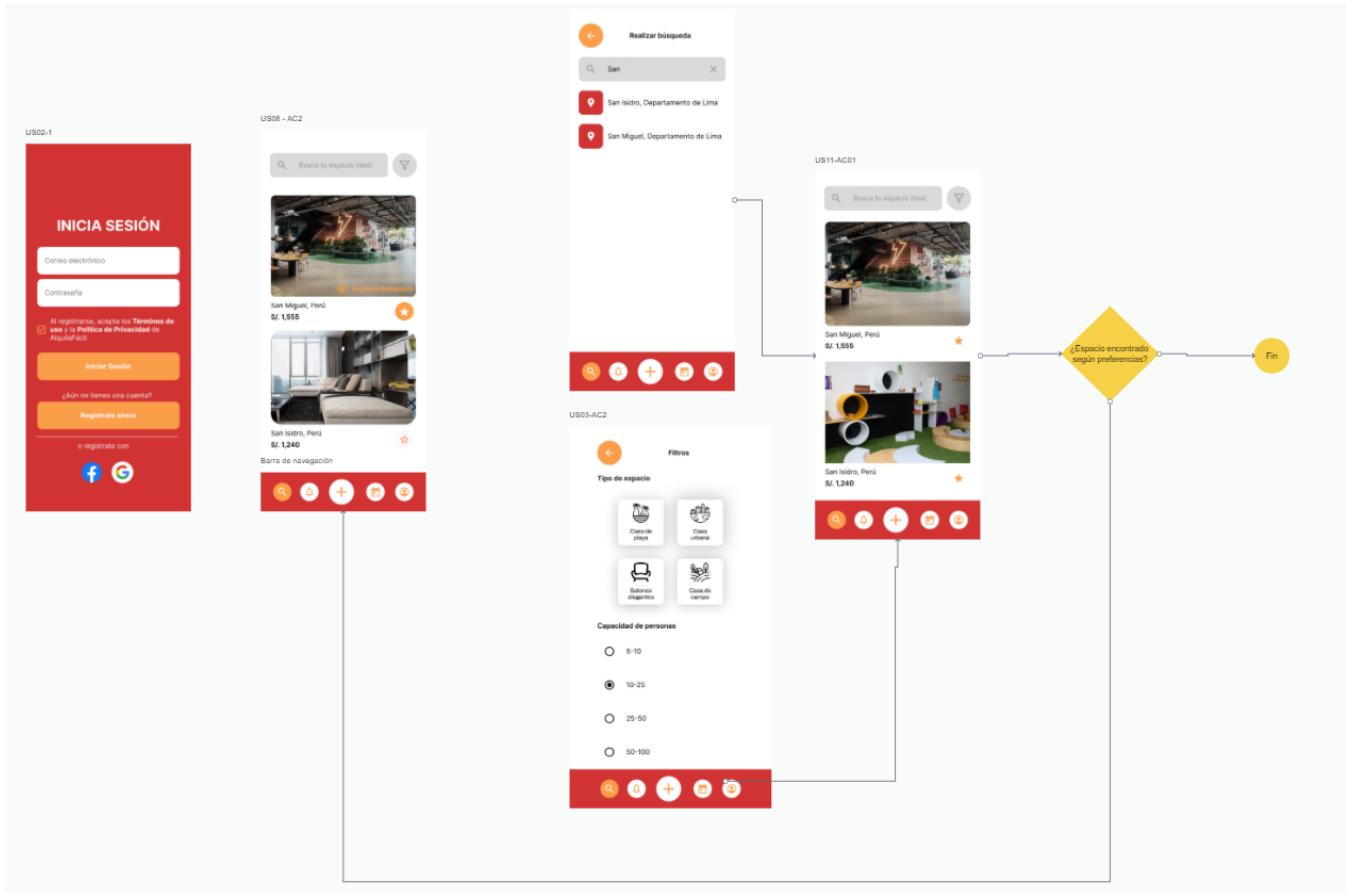
Los userflows describen gráficamente el recorrido que sigue el usuario dentro de la aplicación para cumplir un objetivo específico. Representan las decisiones, acciones y pantallas involucradas en cada paso del proceso, ayudando a identificar puntos críticos de la experiencia y optimizar la usabilidad. Estos diagramas permiten entender de manera clara cómo el usuario se desplaza por el sistema desde el inicio hasta la finalización de una tarea.

Web



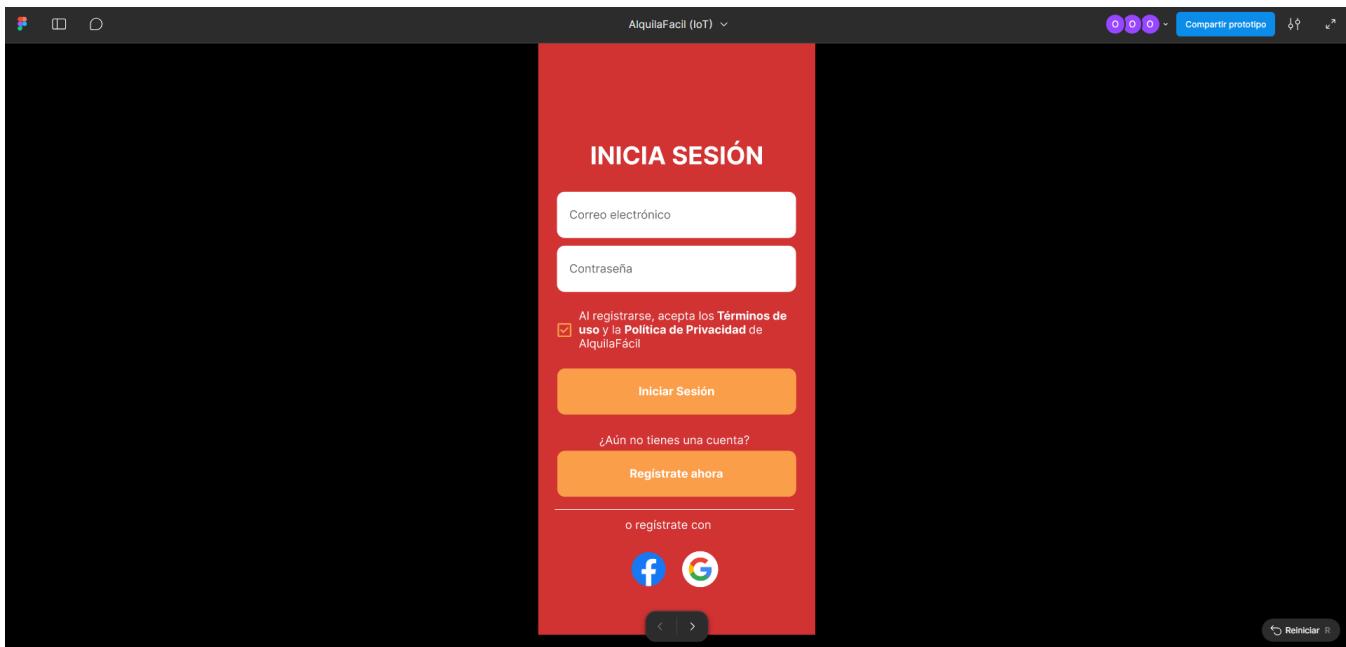
Mobile





6.5. Applications Prototyping

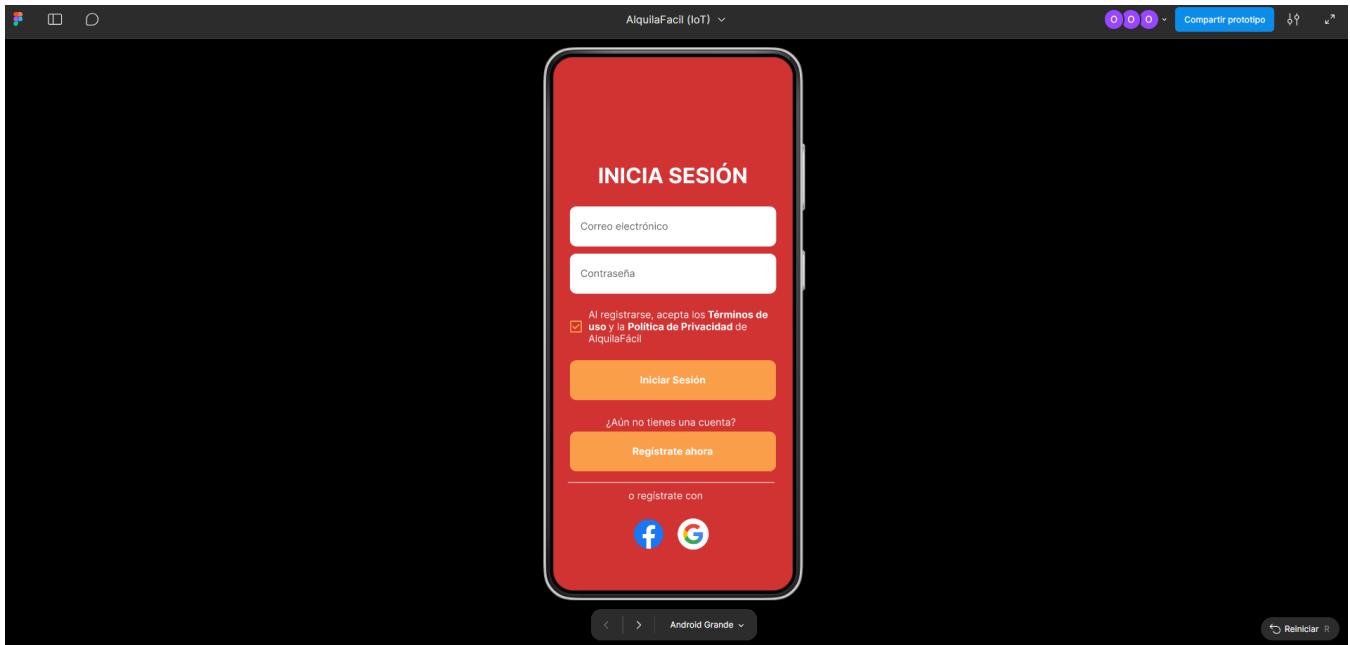
Mobile Android



Prototipo interactivo diseñado para dispositivos Android, optimizado para la experiencia móvil con navegación táctil y diseño adaptado a las guías de Material Design.

[Ver prototipo Android](#)

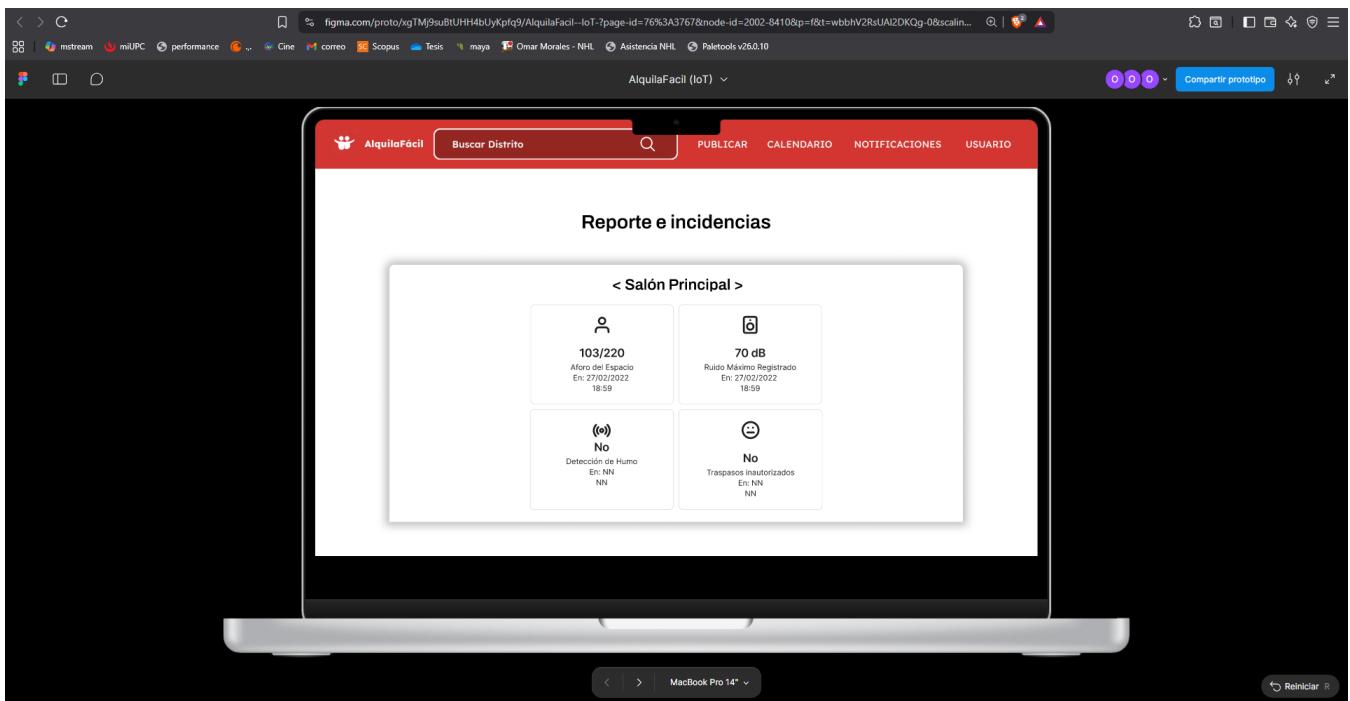
Mobile iOS



Prototipo interactivo diseñado específicamente para dispositivos iOS, siguiendo las directrices de Human Interface Guidelines de Apple para una experiencia nativa en iPhone y iPad.

[Ver prototipo iOS](#)

WebPage



Prototipo de la versión web de la plataforma, diseñado para navegadores de escritorio y tablets, con interfaz responsive y funcionalidades completas para la gestión de alquileres IoT.

[Ver prototipo Web](#)

Capítulo VII: Product Implementation, Validation & Deployment.

7.1. Software Configuration Management.

7.1.1. Software Development Environment Configuration.

Project Management	Microsoft 365 Utilizado como repositorio en la nube para almacenar videos de entrevistas, presentaciones de prototipos y otros recursos vinculados al desarrollo del proyecto.

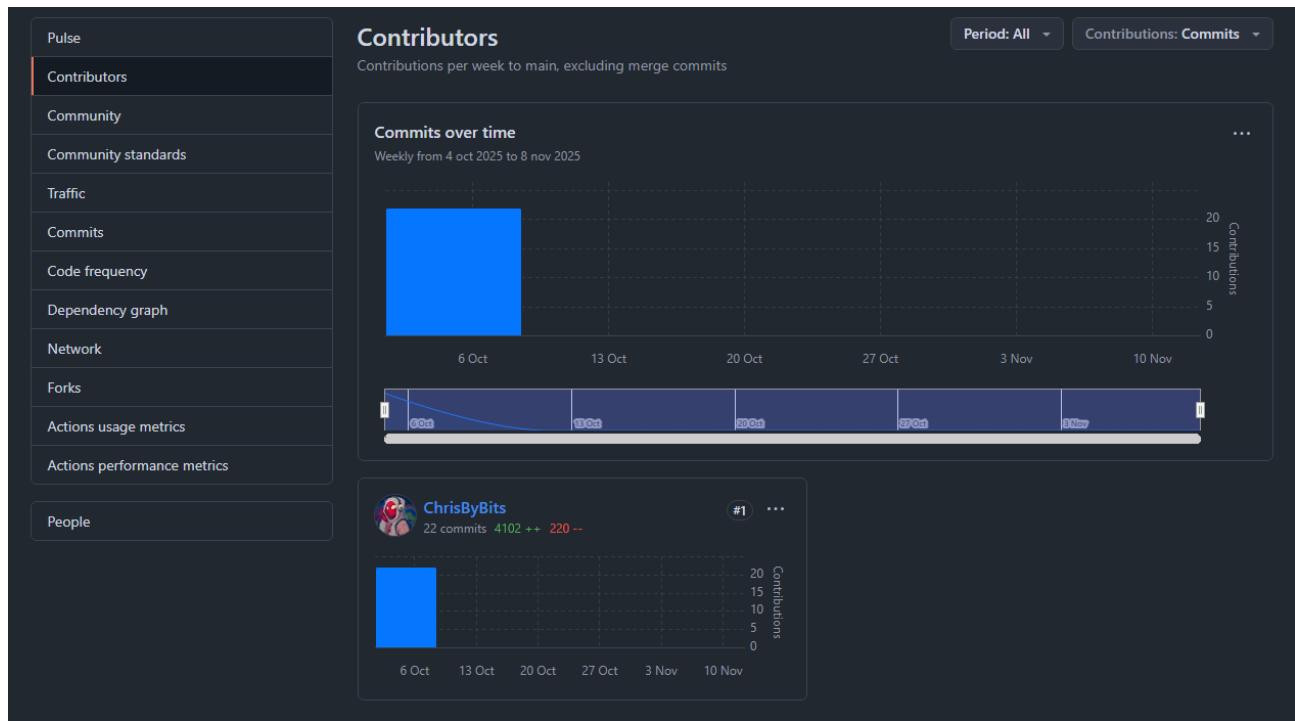
	Discord	Herramienta de comunicación grupal empleada para coordinar tareas, realizar reuniones virtuales y facilitar el trabajo colaborativo.
	WhatsApp	Aplicación de mensajería empleada principalmente para enviar recordatorios de reuniones y comunicaciones rápidas.
	Trello	Plataforma de gestión visual de tareas que permitió organizar, asignar y dar seguimiento al avance del proyecto.
Requirements Management	Structurizr	Software de modelado que facilitó la elaboración de diagramas C4 para representar la arquitectura del sistema.
	Vertabelo	Herramienta para el diseño y modelado visual de bases de datos relacionales.
	Lucidchart	Herramienta de diagramación online empleada para crear modelos UML como casos de uso, clases y secuencias.
	Miro	Plataforma de pizarra colaborativa usada para diseñar escenarios actuales (As-Is) y propuestos (To-Be) del sistema.
Product UX/UI Design	Figma	Aplicación de diseño utilizada para construir wireframes, interfaces gráficas y prototipos navegables del sistema.
Software Development	Git	Sistema de control de versiones empleado para gestionar el código fuente y asegurar un trabajo colaborativo eficaz.
	GitHub	Plataforma basada en Git que se usó para alojar el repositorio del proyecto y coordinar el trabajo en equipo.
	HTML5	Lenguaje de marcado utilizado para estructurar el contenido de la interfaz web.
	CSS	Lenguaje de estilos empleado para dar formato visual a las páginas HTML desarrolladas.
	JavaScript	Lenguaje de programación orientado a la web utilizado para incorporar funcionalidades e interacciones en la interfaz.
	Vue.js	Framework progresivo para construir interfaces de usuario dinámicas y componentes reutilizables en el frontend.
	C#	Lenguaje de programación moderno y orientado a objetos, usado principalmente en aplicaciones empresariales y en desarrollo con tecnologías Microsoft.
	ASP.NET	Framework de desarrollo web de Microsoft que permite crear aplicaciones web robustas y escalables con C#.
	Python	Lenguaje de programación versátil y legible, utilizado para el desarrollo rápido de aplicaciones, scripts, y servicios backend.
	FastAPI	Framework moderno y de alto rendimiento para construir APIs con Python, ideal para microservicios y servicios RESTful.
	C++	Lenguaje de programación de alto rendimiento orientado a sistemas embebidos, controladores y aplicaciones donde el uso eficiente de recursos es crítico.
Software Deployment	Netlify	Servicio en la nube que permitió desplegar la landing page y Web Application del sistema de forma automática, continua y con URL pública.
	Render	Servicio en la nube que permitió desplegar el Web Service y Edge Node de forma automática, continua y con URL pública.
	Wokwi	Plataforma en línea que permitió desplegar la Embedded App con URL pública.

7.1.2. Source Code Management.

Se empleó la metodología Git Flow con el objetivo de mantener un mayor control sobre la gestión del proyecto y permitir el desarrollo simultáneo tanto de la implementación de la aplicación como de la elaboración del informe.

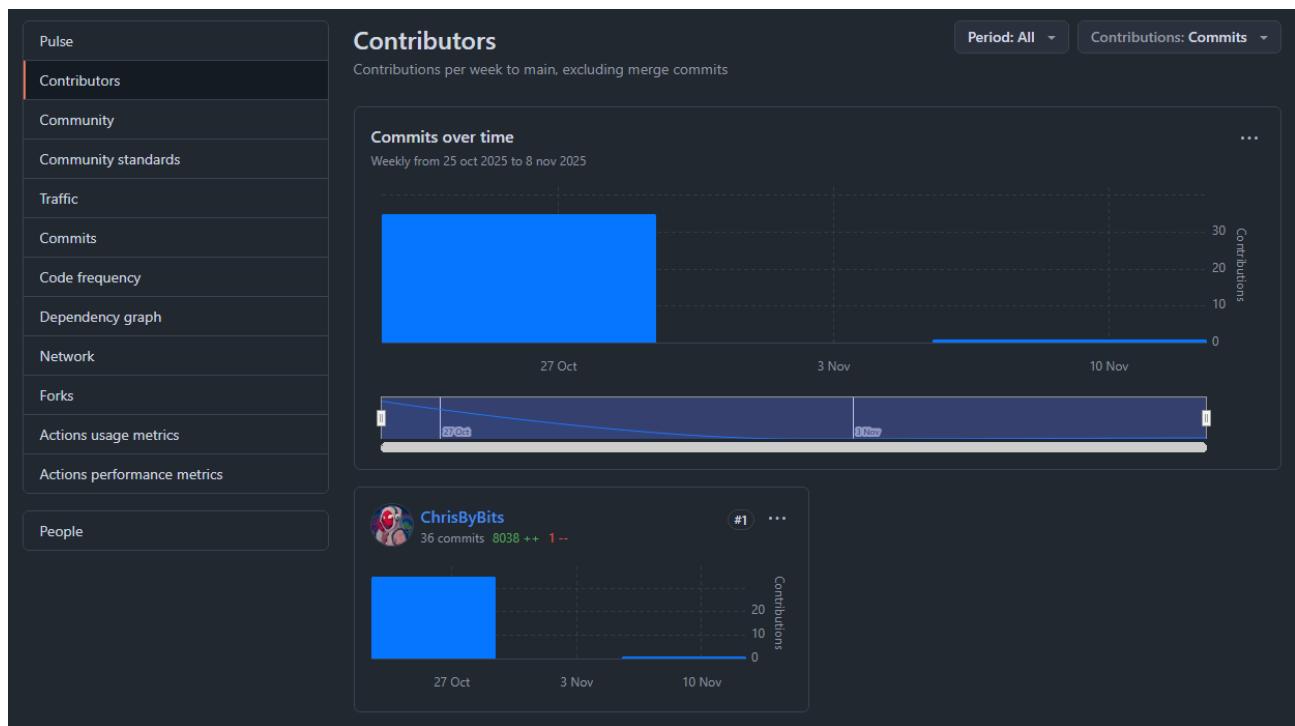
Esta estrategia facilitó la organización del trabajo mediante distintas ramas: una rama principal (main), que contiene la versión estable y lista para presentación, libre de errores; una rama de desarrollo (develop), que actúa como punto de integración de los avances realizados en otras ramas; además de ramas específicas creadas para implementar funcionalidades importantes del sistema y para el desarrollo de cada capítulo del reporte final.

- **Landing Page**



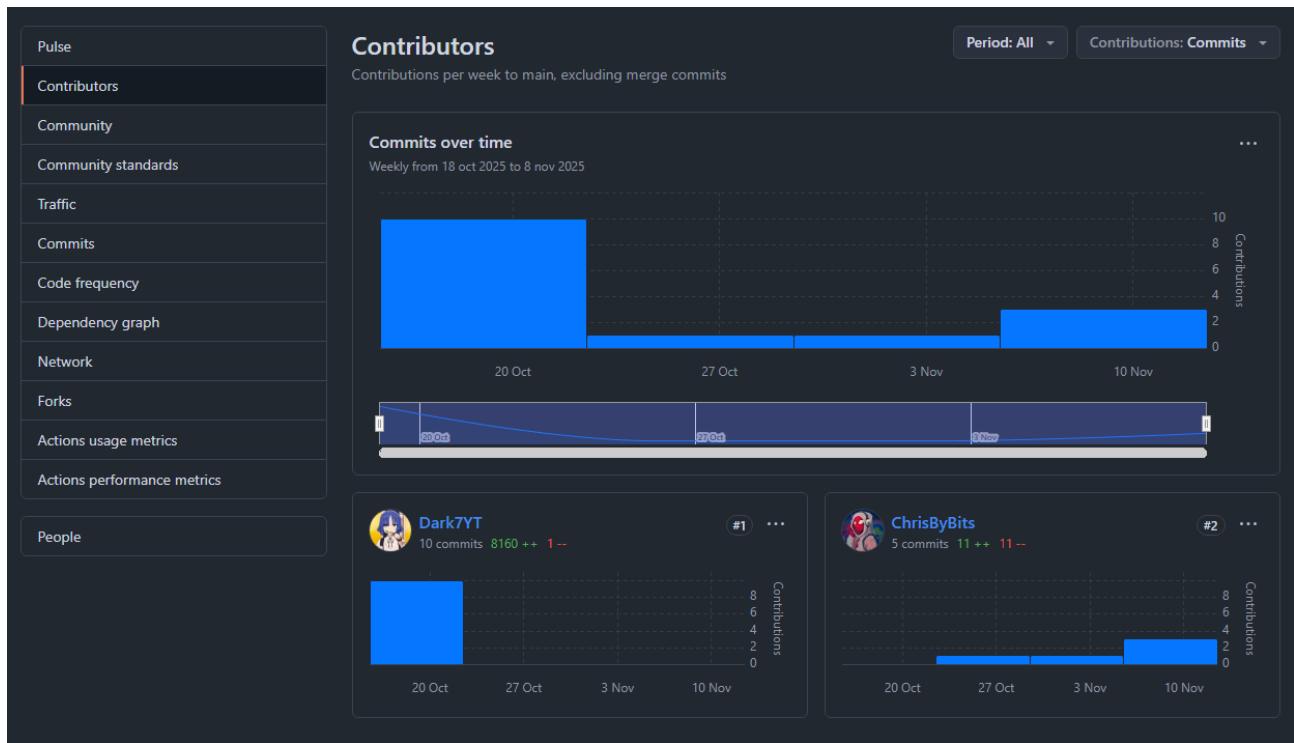
[Enlace de Repositorio de Landing Page](#)

- **Web Application**



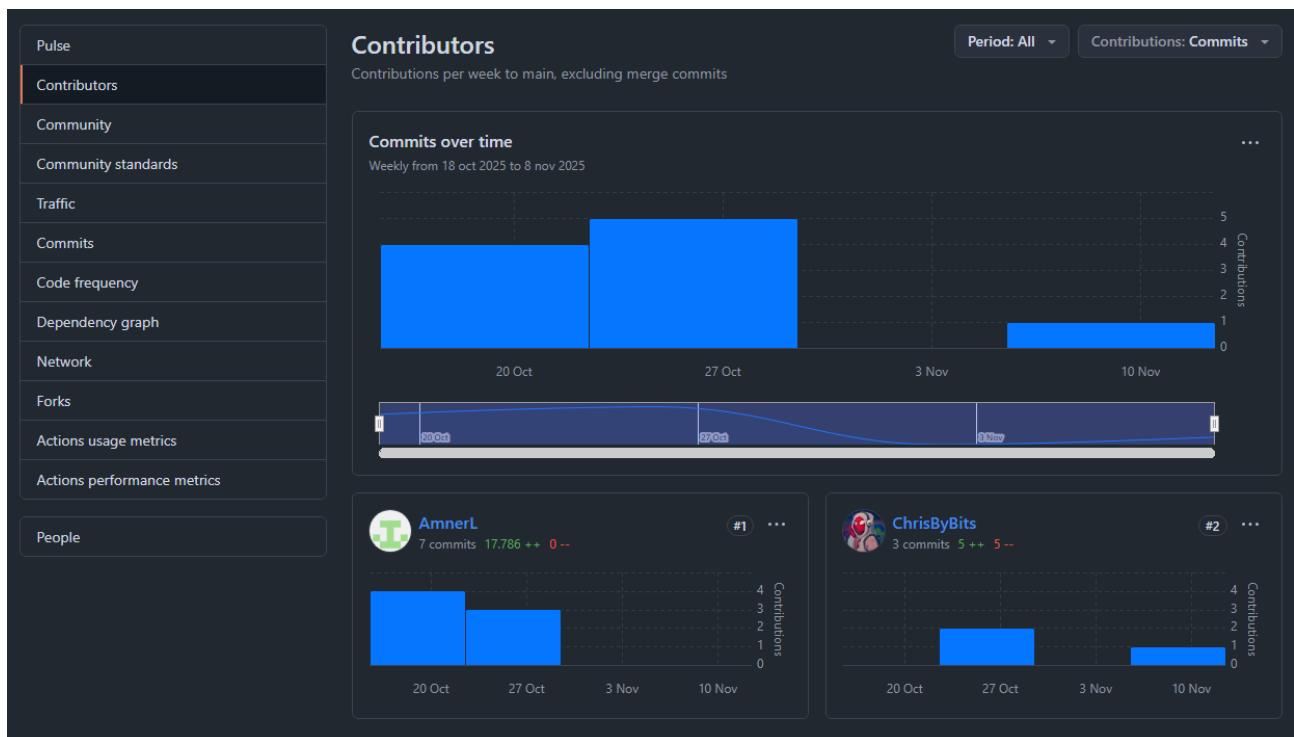
[Enlace de Repositorio de Landing Page](#)

- **Web Service**



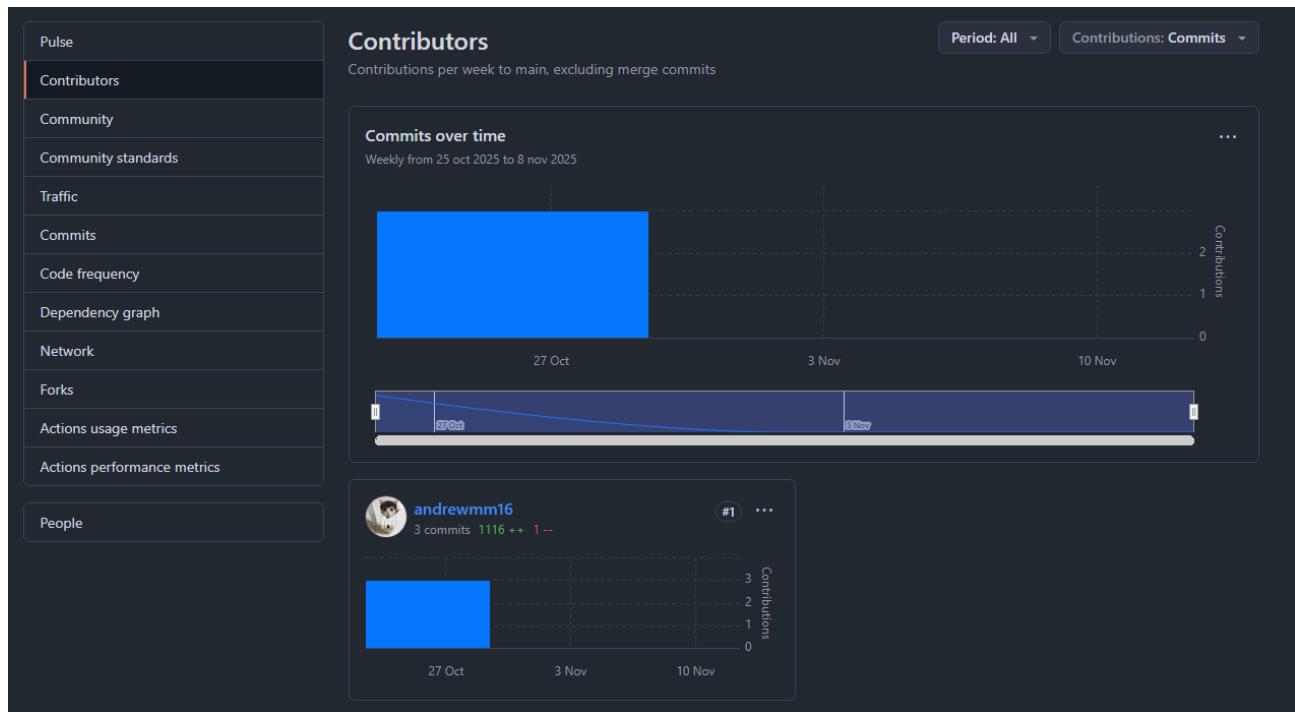
[Enlace de Repositorio de Landing Page](#)

- **Mobile Application**



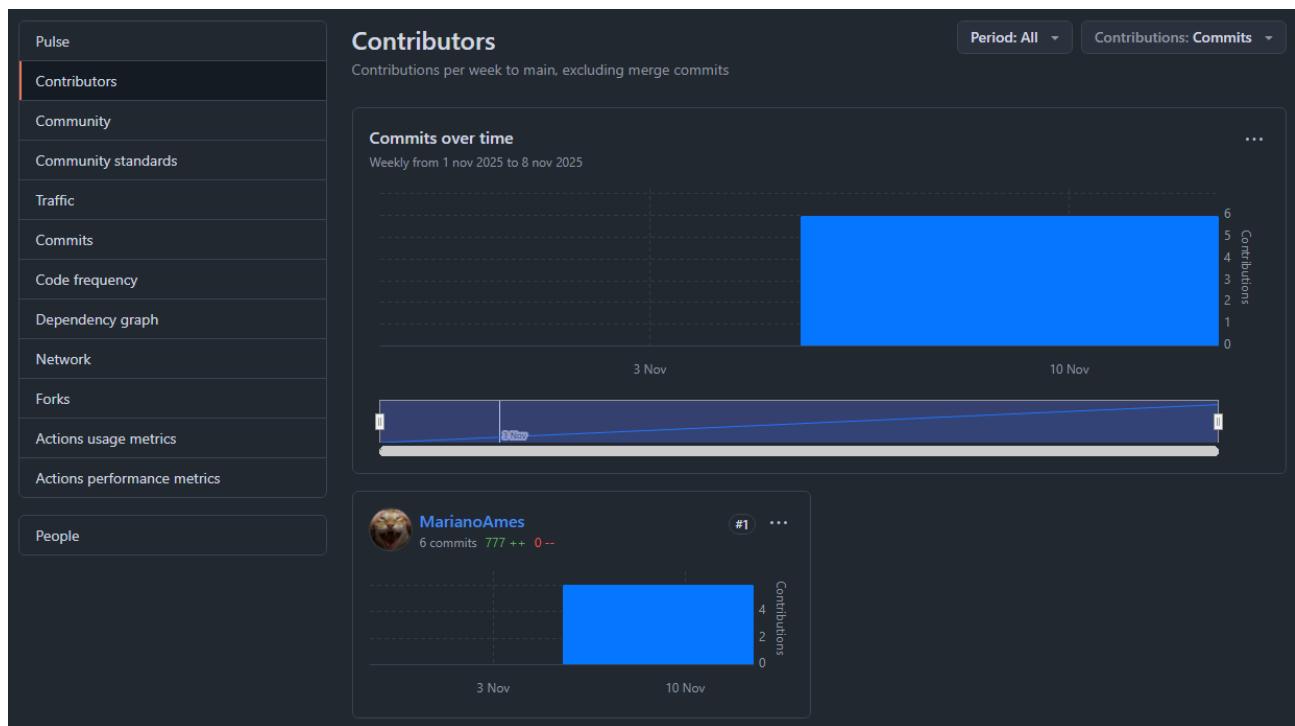
[Enlace de Repositorio de Landing Page](#)

- **Edge Node**



[Enlace de Repositorio de Landing Page](#)

- **Embedded Application**



[Enlace de Repositorio de Landing Page](#)

7.1.3. Source Code Style Guide & Conventions.

Lenguaje Markdown

Referencias Adoptadas	Explicación y Convenciones
The Markdown Guide: https://www.markdownguide.org/	Consideraremos guiones de la guía oficial de markdown para tener mayor noción de trabajo con el uso de esta tecnología. Usaremos el archivo .md para todo lo que es descripción o información del proyecto.

Lenguaje Gherkin

Referencias Adoptadas	Explicación y Convenciones
-----------------------	----------------------------

Referencias Adoptadas	Explicación y Convenciones
Gherkin Conventions for Readable Specifications: https://specflow.org/gherkin/gherkin-conventions-for-readable-specifications/	Consideramos guiarnos y aprender de la documentación situada en la referencia indicada, porque necesitamos tener ideas de cómo trabajar bajo ese lenguaje y poder usar de manera adecuada las herramientas.
Lenguaje HTML y CSS	
Referencias Adoptadas	Explicación y Convenciones
Html Style Guide and Coding Conventions: https://google.github.io/styleguide/htmlcssguide.html	Consideramos que será beneficioso guiarnos y aprender de la documentación de la referencia indicada, ya que contiene una gran cantidad de información sobre las etiquetas HTML. Esto es fundamental para su uso en nuestros proyectos.
Lenguaje C#	
Referencias Adoptadas	Explicación y Convenciones
C# Coding Conventions: https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions	Consideramos en utilizar la documentación de Microsoft para aprender buenas prácticas y sintaxis de C#, puesto que lo utilizaremos en el desarrollo del Back-end.
Lenguaje Dart	
Referencias Adoptadas	Explicación y Convenciones
Dart Style Guide: https://dart.dev/effective-dart/style	Dart ofrece en su documentación, algunas prácticas, estilos y nomenclaturas para tanto la declaración de funciones, carpetas, clases, etc.
Lenguaje Python	
Referencias Adoptadas	Explicación y Convenciones
Python Style Guide: https://www.python.org/doc/essays/styleguide/	Usamos las convenciones determinadas por la organización detrás de Python, para una amplia noción de las nomenclaturas y buenas prácticas para la elaboración de la Api.
Lenguaje C++	
Referencias Adoptadas	Explicación y Convenciones
C++ Core Guidelines: https://isocpp.github.io/CppCoreGuidelines/	Aplicamos las pautas modernas recomendadas por la comunidad C++, enfocándonos en el uso seguro de punteros, manejo eficiente de memoria, programación orientada a objetos y separación clara entre interfaz y lógica de implementación, especialmente en el desarrollo de componentes embebidos.
7.1.4. Software Deployment Configuration.	
En esta sección se especifica la configuración del despliegue de la solución IoT AlquilaFácil, incluyendo los pasos necesarios para que, a partir de los repositorios de código fuente, se pueda lograr el despliegue o publicación satisfactorio de cada uno de los productos digitales en la solución.	
Herramientas de Gestión y Control de Versiones	
Para llevar a cabo el despliegue de todos los componentes de la solución, se utilizaron las siguientes herramientas:	
<ul style="list-style-type: none"> Git: Sistema de control de versiones utilizado para registrar los cambios realizados en todos los proyectos y facilitar la colaboración entre los integrantes del equipo. GitHub: Plataforma en la que se alojaron los repositorios del proyecto. Allí se almacenaron todas las versiones del código de cada componente, asegurando acceso centralizado y controlado. Git Flow: Metodología aplicada para estructurar y organizar el desarrollo mediante distintas ramas, lo que facilitó la coordinación, integración y seguimiento del trabajo colaborativo. 	
Configuración de Despliegue por Componente	
1. Landing Page	
Plataforma de Despliegue: GitHub Pages	
Repositorio: https://github.com/Emergentez/alquilafacil-landing-page	
Tecnologías:	
<ul style="list-style-type: none"> HTML5, CSS3, JavaScript vanilla Tailwind CSS v3.4.13 para estilos AOS (Animate On Scroll) para animaciones 	
Pasos de Despliegue:	

1. El código fuente está desarrollado con HTML, CSS y JavaScript con Tailwind CSS.

2. Se ejecuta el build de Tailwind CSS:

```
npm run build
```

Este comando ejecuta: `npx tailwindcss -i ./src/input.css -o ./src/output.css --minify`

3. En el repositorio de GitHub, se accede a Settings > Pages.

4. En la sección "Source", se selecciona la rama `main`.

5. GitHub Pages genera automáticamente la URL de despliegue.

6. La landing page queda publicada en: <https://emergentez.github.io/alquilafacil-landing-page/>

Automatización: Cada push a la rama principal activa el despliegue automático en GitHub Pages mediante GitHub Actions.

2. Web Application

Plataforma de Despliegue: Netlify

Repositorio: <https://github.com/Emergentez/alquilafacil-web-app>

Tecnologías:

- Vue.js 3.5.13 (Composition API)
- Vite 6.2.0 como build tool
- Vue Router 4.5.0 para navegación
- Pinia 3.0.2 para manejo de estado
- Axios 1.8.4 para llamadas HTTP
- Tailwind CSS 4.1.3 para estilos
- Firebase 11.6.0 para autenticación y notificaciones
- SignalR 8.0.7 para comunicación en tiempo real
- Schedule-X para calendario de reservas

Pasos de Despliegue:

1. La aplicación web está desarrollada en Vue.js 3 con Vite.

2. Se conecta el repositorio de GitHub con Netlify.

3. Se configuran los comandos de build en Netlify:

- Build command: `npm run build`
- Publish directory: `dist`

4. Se configuran las variables de entorno en Netlify:

- `VITE_API_URL`: URL del Web Service backend

5. Netlify ejecuta `vite build` automáticamente con cada push a la rama principal.

6. Los archivos estáticos generados en `/dist` se despliegan en el CDN de Netlify.

7. La aplicación queda publicada con un dominio de Netlify.

Automatización: Continuous Deployment activado - cada commit en la rama principal despliega automáticamente.

3. Web Service (Backend API)

Plataforma de Despliegue: Render

Repositorio: <https://github.com/Emergentez/alquilafacil-web-service>

Tecnologías:

- ASP.NET Core 8.0 (C# / .NET 8)
- Entity Framework Core 8.0.7 para ORM
- MySQL con Pomelo.EntityFrameworkCore.MySql 8.0.2
- JWT Bearer Authentication para seguridad
- Swagger/Swashbuckle 6.5.0 para documentación de API
- BCrypt.Net para hashing de contraseñas
- SignalR para comunicación en tiempo real
- Docker para containerización

Arquitectura:

El proyecto sigue Domain-Driven Design (DDD) con los siguientes bounded contexts:

- IAM (Identity and Access Management)
- Booking (Reservas)
- Locals (Gestión de espacios)
- Management (Gestión IoT y sensores)
- Notifications (Notificaciones)
- Profiles (Perfiles de usuario)

- Subscriptions (Suscripciones)

Dockerfile:

```

FROM mcr.microsoft.com/dotnet/aspnet:8.0 AS base
WORKDIR /app
EXPOSE 8080
EXPOSE 8081

FROM mcr.microsoft.com/dotnet/sdk:8.0 AS build
ARG BUILD_CONFIGURATION=Release
WORKDIR /src
COPY ["AlquilaFacilPlatform/AlquilaFacilPlatform.csproj", "AlquilaFacilPlatform/"]
RUN dotnet restore "AlquilaFacilPlatform/AlquilaFacilPlatform.csproj"
COPY . .
WORKDIR "/src/AlquilaFacilPlatform"
RUN dotnet build "AlquilaFacilPlatform.csproj" -c $BUILD_CONFIGURATION -o /app/build

FROM build AS publish
RUN dotnet publish "AlquilaFacilPlatform.csproj" -c $BUILD_CONFIGURATION -o /app/publish

FROM base AS final
WORKDIR /app
COPY --from=publish /app/publish .
ENTRYPOINT ["dotnet", "AlquilaFacilPlatform.dll"]

```

Pasos de Despliegue en Render:

1. Se crea un nuevo Web Service en Render conectado al repositorio de GitHub.
2. Se selecciona "Docker" como entorno de despliegue.
3. Render detecta automáticamente el Dockerfile en el proyecto.
4. Se configuran las variables de entorno:
 - ConnectionStrings__DefaultConnection: URL de conexión a Railway MySQL
 - JWT__Secret: Secret key para tokens JWT
 - JWT__Issuer y JWT__Audience: Configuración JWT
 - ASPNETCORE_ENVIRONMENT: Production
5. Render ejecuta el build del contenedor Docker automáticamente.
6. El servicio queda expuesto en los puertos 8080 (HTTP) y 8081 (HTTPS).
7. Se obtiene una URL pública para el API REST (ej: <https://alquilafacil-api.onrender.com>).

Automatización: Auto-deploy activado - cada push a la rama principal construye y despliega el contenedor Docker automáticamente.

4. Edge Node (Servidor Edge/IoT Gateway)

Plataforma de Despliegue: Render

Repositorio: <https://github.com/Emergentez/alquilafacil-edge-node>

Tecnologías:

- Python 3.9.23
- FastAPI 0.115.12 para API REST
- Uvicorn 0.34.3 como servidor ASGI
- Peewee 3.18.1 como ORM (base de datos SQLite local)
- Pydantic 2.11.7 para validación de datos
- HTTP/WebSocket para comunicación con sensores y backend
- Docker para containerización

Funcionalidad:

El Edge Node actúa como gateway IoT que:

- Recibe datos de sensores embebidos (humo, ruido, movimiento, aforo)
- Procesa y valida datos en tiempo real
- Almacena temporalmente eventos en SQLite local
- Envía datos procesados al Web Service backend
- Gestiona la comunicación bidireccional con dispositivos IoT

Bounded Contexts:

- IAM: Gestión de autenticación del edge node
- Locals: Información de espacios monitoreados
- Management: Procesamiento de datos de sensores IoT
- Shared: Componentes compartidos

Dockerfile:

```
FROM python:3.9.23-bookworm
WORKDIR /app

COPY .

RUN pip install --upgrade pip && pip install --no-cache-dir -r requirements.txt

EXPOSE 80

CMD ["uvicorn", "main:app", "--host", "0.0.0.0", "--port", "80", "--reload"]
```

Pasos de Despliegue en Render:

1. Se crea un nuevo Web Service en Render conectado al repositorio de GitHub.
2. Se selecciona "Docker" como entorno de despliegue.
3. Render detecta automáticamente el **Dockerfile** en el proyecto.
4. Se configuran variables de entorno:
 - **BACKEND_API_URL**: URL del Web Service backend
 - **DATABASE_PATH**: Ruta a la base de datos SQLite local
 - **WEBSOCKET_ENDPOINT**: Endpoint para comunicación con sensores
5. Render construye la imagen Docker e instala dependencias de **requirements.txt**.
6. El servicio FastAPI se ejecuta con Uvicorn en el puerto 80.
7. Se obtiene una URL pública (ej: <https://alquilafacil-edge.onrender.com>).

Dependencias principales (requirements.txt):

- fastapi==0.115.12
- uvicorn==0.34.3
- peewee==3.18.1
- pydantic==2.11.7
- httpx==0.28.1
- python-dotenv==1.1.0

Automatización: Auto-deploy activado - cada push a la rama principal reconstruye y despliega el contenedor Docker automáticamente.

5. Mobile Application

Plataforma de Distribución: APK con Flutter

Repositorio: <https://github.com/Emergentez/alquilafacil-mobile-app>

Tecnologías:

- Flutter SDK 3.4.3+
- Dart SDK >=3.4.3 <4.0.0
- Provider 6.1.2 para gestión de estado
- HTTP 1.2.2 y Dio 5.7.0 para llamadas a APIs
- Firebase Core 3.7.0 y Firebase Auth 5.3.2 para autenticación
- Google Sign-In 6.2.2 y Facebook Auth 4.3.4 para SSO
- Table Calendar 3.1.2 para calendario de reservas
- Image Picker 1.1.2 para carga de imágenes
- Shared Preferences 2.0.10 para almacenamiento local
- WebSocket Channel 2.4.0 para comunicación en tiempo real

Características Principales:

- Gestión de reservas de espacios
- Calendario interactivo de disponibilidad
- Sistema de calificaciones y reseñas
- Notificaciones en tiempo real
- Chat/comunicación con arrendadores
- Perfil de usuario personalizable

Pasos de Generación y Distribución del APK:

1. Configuración Previa:

- Configurar **android/app/build.gradle** con:
 - Version code y version name
 - Nombre del paquete (applicationId)
 - Min SDK version y Target SDK version

- Permisos (Internet, Cámara, Almacenamiento)
- Configurar Firebase en `android/app/google-services.json`
- Configurar signing key en `android/key.properties`

2. Build del APK:

```
flutter clean
flutter pub get
flutter build apk --release
```

3. Build de APK separados por ABI (opcional, para optimizar tamaño):

```
flutter build apk --split-per-abi
```

4. Ubicación de los APKs generados:

- APK universal: `build/app/outputs/flutter-apk/app-release.apk`
- APKs por ABI:
 - `build/app/outputs/flutter-apk/app-armeabi-v7a-release.apk`
 - `build/app/outputs/flutter-apk/app-arm64-v8a-release.apk`
 - `build/app/outputs/flutter-apk/app-x86_64-release.apk`

5. Distribución:

- **GitHub Releases:** Se sube el APK como asset en la sección de Releases
- **Firebase App Distribution:** Para distribución beta a testers
- **Descarga directa:** Link directo al APK en documentación
- **Google Play Store (futuro):** Publicación mediante Google Play Console

Testing:

```
flutter test
```

6. Embedded Application (Simulación IoT)

Plataforma de Simulación: Wokwi

Repositorio: <https://github.com/Emergentez/alquilafacil-embedded-app>

Pasos de Simulación y Configuración:

1. El código embebido está desarrollado en C++ para Arduino/ESP32.
2. Se utiliza Wokwi (<https://wokwi.com>) como simulador de hardware IoT.
3. Configuración en Wokwi:
 - Se carga el código desde el repositorio o directamente en el editor
 - Se configura el archivo `diagram.json` con los componentes (sensores de humo, ruido, movimiento, etc.)
 - Se configuran las conexiones WiFi para comunicación con el Edge Node
4. El simulador permite:
 - Probar el comportamiento de sensores sin hardware físico
 - Validar la lógica de detección de eventos
 - Verificar la comunicación con el backend/edge node
5. Para deployment en hardware real:
 - Se compila el código con Arduino IDE o PlatformIO
 - Se flashea en dispositivos ESP32/Arduino físicos

7. Base de Datos

Plataforma de Hosting: Railway (MySQL)

Configuración:

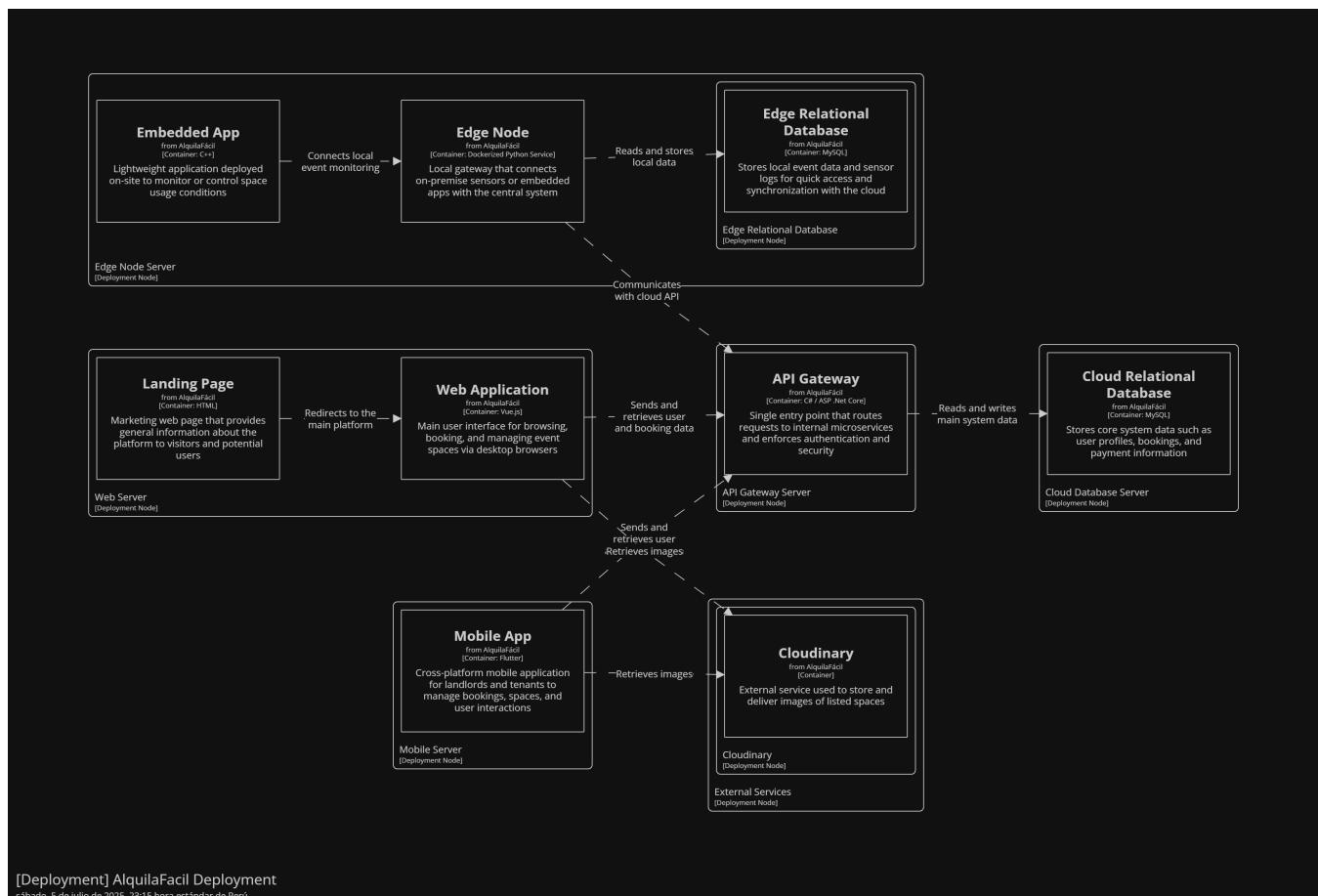
1. Se crea una nueva base de datos MySQL en Railway.
2. Railway proporciona automáticamente:
 - Host URL
 - Puerto
 - Usuario y contraseña
 - Nombre de la base de datos
3. Se ejecutan los scripts de migración/inicialización:

- Creación de tablas (usuarios, espacios, reservas, sensores, eventos)
 - Inserción de datos iniciales si es necesario
4. Se configuran las credenciales en las variables de entorno del Web Service y Edge Node.
5. Railway proporciona:
- Backups automáticos
 - Monitoreo de uso
 - Escalado según necesidad

Conexión: El Web Service y Edge Node se conectan usando las credenciales proporcionadas por Railway.

Deployment Diagram (C4 Model)

A continuación se presenta el Deployment Diagram que ilustra la arquitectura de despliegue de la solución AlquilaFácil:



Descripción del Diagrama:

- **GitHub Pages:** Hospeda la Landing Page estática accesible públicamente.
- **Netlify:** Despliega la Web Application (SPA) con CDN global.
- **Render (Web Service):** Ejecuta el backend API REST que gestiona la lógica de negocio.
- **Render (Edge Node):** Procesa datos de sensores IoT en tiempo real.
- **Railway MySQL:** Almacena todos los datos persistentes de la aplicación.
- **APK Flutter:** Distribuido a dispositivos Android de usuarios finales.
- **Wokwi:** Simula dispositivos embebidos IoT para desarrollo y testing.

Flujo de Comunicación:

1. Usuarios acceden a Landing Page (GitHub Pages) → redirigen a Web App (Netlify)
2. Web App y Mobile App consumen APIs del Web Service (Render)
3. Dispositivos IoT (simulados en Wokwi) envían datos al Edge Node (Render)
4. Edge Node procesa y envía datos al Web Service
5. Web Service almacena/consulta datos en Railway MySQL
6. Notificaciones push se envían desde Web Service a Mobile App

7.2. Solution Implementation.

7.2.1. Sprint 1

7.2.1.1. Sprint Planning 1

Sprint #	Sprint 1
Sprint Planning Background	
Date	01/11/2025
Time	12:00
Location	Modalidad remota por Discord
Prepared By	Lecca Choccare Christopher Bryan
Attendees (to planning meeting)	<ul style="list-style-type: none"> - Ames Oviedo, Mariano Jose - Cachis Gonzales, Sebastian Nicolas - Lecca Choccare Christopher Bryan - Llamo Sánchez, Amner Levi - Morales Montalvo, Omar Andrew
Sprint Goal & User Stories	
Sprint 1 Goal	<p>Our focus is on developing and deploying a complete IoT-enabled platform for AlquilaFácil, including the landing page, web application, mobile application, backend web service, edge node for IoT gateway, and embedded application for sensor simulation. This comprehensive approach aims to deliver an end-to-end solution that connects users, spaces, and IoT monitoring capabilities.</p> <p>We believe it delivers a fully functional ecosystem where users can discover the platform through the landing page, manage spaces and reservations via web and mobile applications, while IoT sensors monitor events in real-time through the edge node and embedded systems, all powered by a robust backend service with MySQL database on Railway.</p> <p>This will be confirmed through successful deployment of all six applications, complete integration testing across all components demonstrating real-time IoT data flow from sensors to applications, and internal team testing verifying all 25 user stories functionality from authentication to IoT monitoring features.</p>
Sprint 1 Velocity	103
Sum of Story Points	103

Aspect Leaders and Collaborators

La siguiente tabla resume los líderes (L) y colaboradores (C) para cada aspecto técnico del Sprint:

Team Member (Last Name, First Name)	GitHub Username	Landing Page Leader (L) / Colaborator (C)	Web Application Leader (L) / Colaborator (C)	Web Service Leader (L) / Colaborator (C)	Mobile App Leader (L) / Colaborator (C)	Edge Node Leader (L) / Colaborator (C)	Embedded App Leader (L) / Colaborator (C)
Ames Oviedo, Mariano José	MarianoAmes	C	C	C	C	C	L
Cachis Gonzales, Sebastian Nicolas	Dark7YT	C	C	L	C	C	C
Lecca Choccare, Christopher Bryan	ChrisByBits	L	L	C	C	C	C
Llamo Sánchez, Amner Levi	AmnerL	C	C	C	L	C	C
Morales Montalvo, Omar Andrew	andrewmm16	C	C	C	C	L	C

7.2.1.2. Sprint Backlog 1.

En esta sección se especifica el detalle del Sprint Backlog 1, la cual es una lista de tareas designadas para esta primera fase del proyecto:



Enlace del tablero: <https://trello.com/b/63Zub4fa/product-backlog-iot>

User Story ID	US Title	Task ID	Task Title	Description	Estimation (Hours)	Assigned To	Status
US01	Registrar usuario	T-01-1	Diseñar interfaz de formulario de registro	Diseñar y maquetar la interfaz del formulario de registro con todos los campos requeridos.	3	Mariano Ames	Done
US01	Registrar usuario	T-01-2	Implementar lógica de validación y registro	Implementar la lógica de validación de datos y conexión con el backend para registro de usuarios.	3	Mariano Ames	Done
US02	Iniciar sesión	T-02-1	Crear componente de formulario de login	Crear el componente visual del formulario de inicio de sesión con campos de email y contraseña.	3	Sebastian Cachis	Done
US02	Iniciar sesión	T-02-2	Implementar autenticación y manejo de sesiones	Implementar la lógica de autenticación, validación de credenciales y gestión de tokens de sesión.	3	Sebastian Cachis	Done
US03	Registrar espacios	T-03-1	Diseñar formulario multipaso de registro	Diseñar y estructurar el formulario multipaso con todos los campos organizados por secciones.	8	Omar Morales	Done
US03	Registrar espacios	T-03-2	Implementar validaciones y envío de datos	Implementar validaciones por cada paso y lógica de envío completo al backend.	8	Omar Morales	Done
US04	Buscar espacios disponibles	T-04-1	Crear componente de barra de búsqueda	Crear el componente visual de la barra de búsqueda con campo de texto e ícono.	3	Christopher Lecca	Done
US04	Buscar espacios disponibles	T-04-2	Implementar lógica de búsqueda y resultados	Implementar la lógica de búsqueda por ubicación y visualización de resultados filtrados.	3	Christopher Lecca	Done
US05	Filtrar espacios disponibles	T-05-1	Diseñar panel de filtros avanzados	Diseñar interfaz de filtros con opciones de capacidad y categoría.	2	Amner Llamo	Done
US05	Filtrar espacios disponibles	T-05-2	Implementar lógica de filtrado	Implementar la lógica de aplicación de múltiples filtros y actualización de resultados.	3	Amner Llamo	Done
US06	Visualizar información del espacio	T-06-1	Diseñar página de detalles del espacio	Diseñar el layout completo de la página de detalles con todas las secciones informativas.	2	Mariano Ames	Done
US06	Visualizar información del espacio	T-06-2	Cargar y renderizar datos del espacio	Implementar la carga de datos desde el backend y renderizado dinámico de toda la información.	3	Mariano Ames	Done
US07	Reservar espacios	T-07-1	Crear formulario de reserva	Diseñar y crear el formulario de reserva con selección de fecha y datos del evento.	3	Omar Morales	Done
US07	Reservar espacios	T-07-2	Implementar proceso de pago y confirmación	Integrar sistema de pago y lógica de confirmación de reserva exitosa.	3	Omar Morales	Done
US08	Gestionar calendario de reservas	T-08-1	Implementar componente de calendario	Crear el componente visual del calendario con capacidad de mostrar fechas y eventos.	3	Christopher Lecca	Done
US08	Gestionar calendario de reservas	T-08-2	Integrar datos de reservas en calendario	Implementar la lógica para cargar reservas y mostrarlas en el calendario con códigos de color.	3	Christopher Lecca	Done
US09	Calificar y comentar sobre espacios	T-09-1	Diseñar interfaz de calificaciones y reseñas	Diseñar el componente de calificación con estrellas y área de texto para comentarios.	2	Sebastian Cachis	Done
US09	Calificar y comentar sobre espacios	T-09-2	Implementar envío y visualización de reseñas	Implementar lógica para enviar reseñas al backend y mostrar reseñas existentes.	3	Sebastian Cachis	Done
US10	Notificar sobre actividades	T-10-1	Implementar sistema de notificaciones push	Configurar sistema de notificaciones en tiempo real para eventos de espacios.	3	Amner Llamo	Done
US10	Notificar sobre actividades	T-10-2	Crear panel de visualización de notificaciones	Diseñar e implementar panel donde se muestren todas las notificaciones recibidas.	2	Amner Llamo	Done
US11	Controlar espacios favoritos	T-11-1	Implementar funcionalidad de agregar favoritos	Crear lógica para agregar y quitar espacios de la lista de favoritos del usuario.	2	Mariano Ames	Done

User Story ID	US Title	Task ID	Task Title	Description	Estimation (Hours)	Assigned To	Status
US11	Controlar espacios favoritos	T-11-2	Diseñar vista de espacios favoritos	Diseñar e implementar la página que muestra todos los espacios marcados como favoritos.	3	Mariano Ames	Done
US12	Visualizar espacios propios	T-12-1	Crear vista de listado de espacios propios	Diseñar interfaz que muestre todos los espacios publicados por el arrendador.	2	Omar Morales	Done
US12	Visualizar espacios propios	T-12-2	Implementar carga de datos de espacios propios	Implementar lógica para cargar y filtrar espacios según el propietario actual.	3	Omar Morales	Done
US13	Modificar espacios publicados	T-13-1	Habilitar edición de campos del espacio	Implementar formulario de edición con campos precargados del espacio seleccionado.	3	Christopher Lecca	Done
US13	Modificar espacios publicados	T-13-2	Guardar cambios y actualizar información	Implementar lógica para enviar cambios al backend y actualizar la información del espacio.	3	Christopher Lecca	Done
US14	Actualizar perfil y cerrar sesión	T-14-1	Crear página de edición de perfil	Diseñar e implementar formulario de edición de perfil con datos del usuario.	2	Sebastian Cachis	Done
US14	Actualizar perfil y cerrar sesión	T-14-2	Implementar funcionalidad de cierre de sesión	Implementar lógica para cerrar sesión, limpiar tokens y redirigir al usuario.	3	Sebastian Cachis	Done
US15	Reportar espacios inseguros	T-15-1	Crear formulario de reporte de espacios	Diseñar formulario con campos para describir el problema de seguridad del espacio.	2	Amner Llamo	Done
US15	Reportar espacios inseguros	T-15-2	Implementar envío y almacenamiento de reportes	Implementar lógica para enviar reportes al backend y notificar a administradores.	3	Amner Llamo	Done
US16	Visualizar espacios reportados	T-16-1	Diseñar vista de reportes realizados	Crear interfaz que muestre lista de todos los reportes realizados por el usuario.	2	Mariano Ames	Done
US16	Visualizar espacios reportados	T-16-2	Cargar y mostrar detalles de reportes	Implementar carga de reportes desde el backend y visualización de su estado.	3	Mariano Ames	Done
US17	Acceder a información de ayuda	T-17-1	Diseñar sección de preguntas frecuentes	Crear diseño y estructura de la página de FAQ con categorías de preguntas.	3	Omar Morales	Done
US17	Acceder a información de ayuda	T-17-2	Implementar información de contacto	Implementar sección de contacto con formulario y datos de soporte técnico.	2	Omar Morales	Done
US18	Control de aforo	T-18-1	Integrar sensores de conteo de personas	Configurar integración con sensores IoT para recibir datos de aforo en tiempo real.	4	Christopher Lecca	Done
US18	Control de aforo	T-18-2	Implementar alertas de aforo	Crear sistema de alertas que notifique cuando se aproxime o exceda el aforo permitido.	3	Christopher Lecca	Done
US19	Monitoreo de niveles de ruido	T-19-1	Integrar sensores de nivel de ruido	Configurar conexión con sensores de ruido para recibir mediciones en tiempo real.	3	Sebastian Cachis	Done
US19	Monitoreo de niveles de ruido	T-19-2	Crear sistema de notificaciones de ruido	Implementar lógica de comparación con umbral y envío de notificaciones al arrendador.	3	Sebastian Cachis	Done
US20	Detección de humo	T-20-1	Integrar sensores de detección de humo	Configurar integración con sensores de humo del sistema IoT.	3	Amner Llamo	Done
US20	Detección de humo	T-20-2	Implementar alertas automáticas de humo	Crear sistema de alertas automáticas que notifique inmediatamente al detectar humo.	4	Amner Llamo	Done
US21	Movimiento en áreas no autorizadas	T-21-1	Integrar sensores de movimiento	Configurar sensores de movimiento en zonas restringidas del sistema IoT.	3	Mariano Ames	Done
US21	Movimiento en áreas no autorizadas	T-21-2	Implementar notificaciones de acceso restringido	Crear lógica que envíe notificaciones al detectar movimiento en áreas no permitidas.	4	Mariano Ames	Done
US22	Registro de incidentes	T-22-1	Diseñar historial de eventos de sensores	Crear interfaz para visualizar cronológicamente todos los eventos registrados.	3	Omar Morales	Done

User Story ID	US Title	Task ID	Task Title	Description	Estimation (Hours)	Assigned To	Status
US22	Registro de incidentes	T-22-2	Implementar almacenamiento y filtrado	Implementar sistema de almacenamiento de eventos y opciones de filtrado por tipo y fecha.	4	Omar Morales	Done
US23	Explorar landing informativa	T-23-1	Diseñar estructura y secciones del landing	Crear diseño responsive del landing page con todas las secciones principales.	5	Christopher Lecca	Done
US23	Explorar landing informativa	T-23-2	Implementar contenido y animaciones	Implementar contenido, imágenes y animaciones para una experiencia atractiva.	5	Christopher Lecca	Done
US24	Navegar por secciones desde el header	T-24-1	Diseñar menú de navegación responsive	Crear diseño del header con menú de navegación adaptable a dispositivos móviles.	5	Sebastian Cachis	Done
US24	Navegar por secciones desde el header	T-24-2	Implementar scroll suave entre secciones	Implementar funcionalidad de navegación con scroll suave hacia las secciones del landing.	5	Sebastian Cachis	Done
US25	Ver beneficios diferenciados	T-25-1	Diseñar tarjetas de beneficios	Crear diseño de tarjetas informativas para arrendadores y arrendatarios con iconografía.	5	Amner Llamo	Done
US25	Ver beneficios diferenciados	T-25-2	Implementar sección de beneficios	Implementar sección con todas las tarjetas de beneficios organizadas por tipo de usuario.	5	Amner Llamo	Done

7.2.1.3. Development Evidence for Sprint Review.

En esta sección se presentan los commits más relevantes realizados en el repositorio de GitHub, donde se evidencia el trabajo progresivo en el desarrollo y despliegue, tanto como la Landing Page y Web Application. Se incluyen fusiones de ramas, configuraciones de despliegue, implementación de componentes y mejoras de contenido visual.

Landing Page

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
Emergentez/alquilafacil-landing-page	/main	c13b0e2	refactor: update sign-in link to new URL	ChrisByBits	12/11/2025
Emergentez/alquilafacil-landing-page	/main	eb3d7e5	styles: change switcher position	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	d039d57	feat: add i18n	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	7af54c3	feat: add i18n	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	d4ba850	refactor: fix formatting in main.yml	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	e16a440	fix: delete .github/workflows/develop	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	6053ec9	feat: add CI workflow for main branch	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	497c4a7	Merge branch 'develop'	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	b0026d4	feat: add main.yml	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	fb4c6f9	Merge hotfix/fix-css-link into develop	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	b25b7a8	Hotfix: improve gitignore patterns	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	114ef8e	fix: improve gitignore patterns	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	9454dea	Merge release/v1.0.0 back into develop	ChrisByBits	07/10/2025

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
Emergentez/alquilafacil-landing-page	/main	8469bd2	Release v1.0.0	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	57e9075	chore: add production build script to package.json	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	11e4d4b	ci: add GitHub Actions deployment workflow	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	17f7e10	Merge feature/showcase into develop	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	f97e6a7	build: add compiled Tailwind CSS output	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	e6fa8d5	feat: add event venue images for gallery	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	9817c13	feat: add feature screenshots	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	f305999	feat: add mobile app mockups for showcase	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	bbae926	Merge feature/legal-pages into develop	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	c2dbe45	docs: add terms of service page	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	cd8e41f	docs: add privacy policy page	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	bfb3be	Merge feature/sensors into develop	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	5eb767f	feat: add IoT sensors images for monitoring section	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	3ea8c4b	Merge feature/hero-section into develop	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	3de4d33	feat: add landing page HTML structure	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	82f48e4	feat: add hero section images	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	b30aea9	feat: add brand logo asset	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	966b0db	style: add base CSS and Tailwind input files	ChrisByBits	07/10/2025
Emergentez/alquilafacil-landing-page	/main	494e9ec	feat: add project initial configuration	ChrisByBits	07/10/2025

Web Application

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
Emergentez/alquilafacil-web-app	/main	901dadd	refactor: fix reading event	ChrisByBits	10/11/2025
Emergentez/alquilafacil-web-app	/main	529fb51	release: version 1.0.0	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	e6a6b19	merge: complete application integration	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	4255caa	feat: integrate all modules in main app	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	0d58137	style: add global styles and tailwind configuration	ChrisByBits	29/10/2025

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
Emergentez/alquilafacil-web-app	/main	df79430	merge: integrate IoT monitoring module into develop	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	4b3b749	feat(management): add local monitoring dashboard	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	d9b282b	feat(management): implement real-time monitoring services	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	d55d859	feat(management): add IoT monitoring models	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	1b4c34f	merge: integrate notifications module into develop	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	101f7e3	feat(notification): add notifications page	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	7a98684	feat(notification): implement notifications API service	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	25e2143	feat(notification): add notification model	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	bdb23cc	merge: integrate subscriptions system into develop	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	8854591	feat(subscriptions): add paypal integration and purchase page	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	622fe77	feat(subscriptions): implement subscriptions API services	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	4eaeca7	feat(subscriptions): add subscription models	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	e56bdb8	merge: integrate user profile module into develop	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	af005a3	feat(profile): add control panel with subscriptions management	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	91f4b73	feat(profile): add reports and support features	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	2be14d3	feat(profile): add calendar integration	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	42bfd45	feat(profile): add profile management components	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	1df7a77	feat(profile): implement profiles API service	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	5aa4df9	feat(profile): add profile models	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	2abcf81	merge: integrate booking system into develop	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	ae67098	feat(booking): add reservation details page	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	d446f22	feat(booking): implement reservations API service	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	c0eacc2	feat(booking): add reservation models and utilities	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	0bb10b8	merge: integrate locals management module into develop	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	26ca283	feat(locals): add multi-step local publication wizard	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	78d41b9	feat(locals): add reporting functionality	ChrisByBits	29/10/2025

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
Emergentez/alquilafacil-web-app	/main	9aa25f5	feat(locals): add local details and comments features	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	da40fb2	feat(locals): add local display components	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	ffff0bea	feat(locals): add search and filtering pages	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	34583ad	feat(locals): implement locals API services	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	4674536	feat(locals): add local models and data structures	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	7ee1d95	merge: integrate authentication module into develop	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	6ab41de	feat(auth): add authentication UI components and pages	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	5946dca	feat(auth): add authentication store with pinia	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	93bb0f3	feat(auth): implement authentication services and firebase integration	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	c441ce7	feat(auth): add authentication models and request/response structures	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	ea23191	merge: integrate project setup into develop	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	bafca219	feat(public): add public components and routing infrastructure	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	fd9b7c8	feat(shared): add core shared services and utilities	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	94c04d4	chore: initial project setup with vue, vite and tailwind	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-app	/main	2a32a44	chore: initial repository setup	ChrisByBits	29/10/2025

Web Service

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
Emergentez/alquilafacil-web-service	/main	a227e71	refactor: update url for deployment	MarianoAmes	13/11/2025
Emergentez/alquilafacil-web-service	/main	878441f	refactor: update production database connection string	ChrisByBits	12/11/2025
Emergentez/alquilafacil-web-service	/main	d27e12f	refactor: fix reading event	ChrisByBits	10/11/2025
Emergentez/alquilafacil-web-service	/main	ec7a695	refactor: change production database connection string	ChrisByBits	10/11/2025
Emergentez/alquilafacil-web-service	/main	b7c1b6d	Update production database connection string	ChrisByBits	03/11/2025
Emergentez/alquilafacil-web-service	/main	5514fea	Update production appsettings for connection and origins	ChrisByBits	29/10/2025
Emergentez/alquilafacil-web-service	/main	fa6e499	Add integration tests for Notification, Profile, Report, Reservation, Subscription, and User controllers	Dark7YT	22/10/2025
Emergentez/alquilafacil-web-service	/main	5f19fb2	feat: Add subscription management entities, repositories, services, and REST controller	Dark7YT	22/10/2025

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
Emergentez/alquilafacil-web-service	/main	ff31e44	feat: Implement external services and repositories for user, local, notification, profile, and subscription management	Dark7YT	22/10/2025
Emergentez/alquilafacil-web-service	/main	7e5dddc	feat: Implement Local Edge Node and Reading repositories, controllers, and SignalR hub	Dark7YT	22/10/2025
Emergentez/alquilafacil-web-service	/main	e8090b2	feat: Implement LocalCategory, Local, and Report repositories	Dark7YT	22/10/2025
Emergentez/alquilafacil-web-service	/main	1d4e93d	Add user management and comment functionality	Dark7YT	22/10/2025
Emergentez/alquilafacil-web-service	/main	13616a8	feat: Implement reservation commands and queries	Dark7YT	22/10/2025
Emergentez/alquilafacil-web-service	/main	ae1ba97b	feat: Initialize AlquilaFacilPlatform project structure with essential files and configurations	Dark7YT	22/10/2025
Emergentez/alquilafacil-web-service	/main	a0ed972	fix: ensure consistent formatting in README.md	Dark7YT	22/10/2025
Emergentez/alquilafacil-web-service	/main	986fbcd	first commit	Dark7YT	22/10/2025

Mobile Application

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
Emergentez/alquilafacil-mobile-app	/main	94b9aed	refactor: fix calendar date bug	ChrisByBits	10/11/2025
Emergentez/alquilafacil-mobile-app	/main	1e03a3d	Update WebSocket URL for monitoring screen	ChrisByBits	29/10/2025
Emergentez/alquilafacil-mobile-app	/main	65fb331	Update BASE_URL to use secure HTTPS protocol	ChrisByBits	29/10/2025
Emergentez/alquilafacil-mobile-app	/main	e2ab8dd	feat: Integrate subscription plans and payment logic	AmnerL	27/10/2025
Emergentez/alquilafacil-mobile-app	/main	7669947	feat: Implement alert notification service	AmnerL	27/10/2025
Emergentez/alquilafacil-mobile-app	/main	ff86e9f1	feat: Introduce reservation and management/monitoring modules	AmnerL	27/10/2025
Emergentez/alquilafacil-mobile-app	/main	0e438fb	feat: Implement spaces module and public data/UI components	AmnerL	25/10/2025
Emergentez/alquilafacil-mobile-app	/main	f73b25c	feat: Add user authentication (IAM) and profile management modules	AmnerL	25/10/2025
Emergentez/alquilafacil-mobile-app	/main	254fbe2	feat: add first configuration	AmnerL	25/10/2025
Emergentez/alquilafacil-mobile-app	/main	dfcf73a	Initial commit	AmnerL	25/10/2025

Edge Node

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
Emergentez/alquilafacil-edge-node	/main	fa8d9a4	feat: readme with instructions added	andrewmwm16	27/10/2025
Emergentez/alquilafacil-edge-node	/main	2f0a288	feat: edge node initial version added	andrewmwm16	27/10/2025
Emergentez/alquilafacil-edge-node	/main	8f7b106	Initial commit	andrewmwm16	27/10/2025

Embedded Application

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
Emergentez/alquilafacil-embedded-app	/main	f86c429	feat: add sketch ino	MarianoAmes	12/11/2025
Emergentez/alquilafacil-embedded-app	/main	3ab2a5f	feat: add device class and implementation	MarianoAmes	12/11/2025
Emergentez/alquilafacil-embedded-app	/main	e24f886	feat: add sound sensor class	MarianoAmes	12/11/2025
Emergentez/alquilafacil-embedded-app	/main	61c9aa8	feat: add smoke sensor class	MarianoAmes	12/11/2025
Emergentez/alquilafacil-embedded-app	/main	21733d2	feat: add movement sensors files	MarianoAmes	12/11/2025
Emergentez/alquilafacil-embedded-app	/main	d6c89e3	feat: create modestiot classes	MarianoAmes	12/11/2025

7.2.1.4. Testing Suite Evidence for Sprint Review.

A continuación, se muestra la tabla de commits relacionados al testing de la aplicación Web Service:

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
Emergentez/alquilafacil-web-service	/main	fa6a499	Add integration tests for Notification, Profile, Report, Reservation, Subscription, and User controllers	- Implemented tests for NotificationController to verify notification retrieval and deletion. - Added ProfileController tests for fetching and updating user profiles. - Created ReportController tests to ensure report creation and retrieval by user and local IDs. - Developed ReservationController tests for creating, updating, deleting, and retrieving reservations. - Added SubscriptionController tests for subscription creation, retrieval, and status updates. - Implemented UserController tests for user retrieval and updates. - Introduced new REST controllers and resources for handling plans and invoices. - Added transformation classes for converting resources to commands and entities.	22/10/2025

Resumen de Tests Automatizados

A continuación se presenta la relación completa de tests automatizados desarrollados para el Sprint 1, organizados por tipo de testing:

Unit Tests (Pruebas Unitarias)

Los Unit Tests se enfocan en probar entidades del dominio y sus comportamientos de manera aislada:

Test Class	Clase Probada	Comportamientos Probados	User Stories Relacionadas
UserTests	User (IAM.Domain.Model.Aggregates)	<ul style="list-style-type: none"> - Constructor con parámetros inicializa propiedades correctamente - UpdateUsername() actualiza el nombre de usuario - UpdatePasswordHash() actualiza el hash de contraseña 	US01, US02, US14
ProfileTests	Profile (Profiles.Domain.Model.Aggregates)	<ul style="list-style-type: none"> - Constructor inicializa propiedades del perfil - Actualización de información de perfil 	US14
LocalTests	Local (Locals.Domain.Model.Aggregates)	<ul style="list-style-type: none"> - Constructor inicializa datos del espacio - Validación de capacidad y categoría - Actualización de información del espacio 	US03, US06, US12, US13
ReservationTests	Reservation (Booking.Domain.Model.Aggregates)	<ul style="list-style-type: none"> - Creación de reserva con fechas válidas - Validación de conflictos de fechas - Actualización de estado de reserva 	US07, US08
CommentTests	Comment (Locals.Domain.Model.Entities)	<ul style="list-style-type: none"> - Creación de comentario con calificación - Validación de calificación (rango 1-5) - Asociación con espacio y usuario 	US09
NotificationTests	Notification (Notifications.Domain.Model.Aggregates)	<ul style="list-style-type: none"> - Creación de notificación - Estado de lectura/no lectura - Asociación con usuario 	US10

Test Class	Clase Probada	Comportamientos Probados	User Stories Relacionadas
ReportTests	Report (Management.Domain.Model.Aggregates)	- Creación de reporte de espacio - Validación de descripción de problema - Estado del reporte	US15, US16
SubscriptionTests	Subscription (Subscriptions.Domain.Model.Aggregates)	- Creación de suscripción - Estados de suscripción (activa/inactiva) - Fecha de renovación	Funcionalidad de suscripciones
Integration Tests (Pruebas de Integración)			
Los Integration Tests validan la interacción entre controladores y servicios del Web Service:			
Test Class	Controller Probado	Endpoints Probados	User Stories Relacionadas
AuthenticationControllerTests	AuthenticationController	- POST /api/v1/authentication/sign-in (credenciales válidas/inválidas) - POST /api/v1/authentication/sign-up (registro exitoso/fallido)	US01, US02
LocalControllerTests	LocalController	- GET /api/v1/locals (listar espacios) - GET /api/v1/locals/{id} (detalle de espacio) - POST /api/v1/locals (crear espacio) - PUT /api/v1/locals/{id} (actualizar espacio) - GET /api/v1/locals/search (búsqueda y filtrado)	US03, US04, US05, US06, US12, US13
ReservationControllerTests	ReservationController	- GET /api/v1/reservations (listar reservas) - POST /api/v1/reservations (crear reserva) - PUT /api/v1/reservations/{id} (actualizar reserva) - DELETE /api/v1/reservations/{id} (cancelar reserva) - GET /api/v1/reservations/calendar (calendario)	US07, US08
CommentControllerTests	CommentController	- GET /api/v1/comments/local/{localId} (comentarios de espacio) - POST /api/v1/comments (crear comentario y calificación) - GET /api/v1/comments/user/{userId} (comentarios de usuario)	US09
NotificationControllerTests	NotificationController	- GET /api/v1/notifications/user/{userId} (obtener notificaciones) - DELETE /api/v1/notifications/{id} (eliminar notificación) - PUT /api/v1/notifications/{id}/read (marcar como leída)	US10
ProfileControllerTests	ProfileController	- GET /api/v1/profiles/{userId} (obtener perfil) - PUT /api/v1/profiles/{userId} (actualizar perfil)	US14
ReportControllerTests	ReportController	- POST /api/v1/reports (crear reporte) - GET /api/v1/reports/user/{userId} (reportes de usuario) - GET /api/v1/reports/local/{localId} (reportes de espacio)	US15, US16
UserControllerTests	UserController	- GET /api/v1/users/{id} (obtener usuario) - PUT /api/v1/users/{id} (actualizar usuario)	US01, US02, US14
SubscriptionControllerTests	SubscriptionController	- GET /api/v1/subscriptions/{userId} (obtener suscripción) - POST /api/v1/subscriptions (crear suscripción) - PUT /api/v1/subscriptions/{id}/status (actualizar estado)	Funcionalidad de suscripciones
LocalCategoriesControllerTests	LocalCategoriesController	- GET /api/v1/local-categories (listar categorías) - Soporte para filtrado por categoría	US05

Nota: Los tests de IoT (US18-US22: Control de aforo, Monitoreo de ruido, Detección de humo, Movimiento, Registro de incidentes) se prueban a nivel de Edge Node con Python/FastAPI y simulación en Wokwi. Los tests de Landing Page (US23-US25) son pruebas E2E manuales de UI/UX.

Acceptance Tests (BDD con Gherkin)

En adición, se presentan los archivos .feature que contienen cada uno de los escenarios a validar para las features respectivas a las historias de usuario trabajadas en este Sprint.

US01:

```
Feature: Registro de usuario

Scenario: Registro exitoso
  Given Que un usuario desea registrar su espacio en AlquilaFácil
  When Completa el formulario de registro con la información requerida
  Then Recibe una confirmación de registro y puede acceder a su cuenta

Scenario: Validación de datos
  Given Que un usuario completa el formulario de registro en AlquilaFácil
  When Envía el formulario
  Then Los datos proporcionados se validan para garantizar la precisión y la autenticidad
```

US02:

```
Feature: Inicio de sesión

Scenario: Inicio de sesión exitoso
  Given Que un usuario registrado desea acceder a su cuenta en AlquilaFácil
  When Ingresa su correo electrónico y contraseña correctos en el formulario de inicio de sesión
  Then Es autenticado exitosamente y se le otorga acceso a su cuenta

Scenario: Error en el inicio de sesión por credenciales incorrectas
  Given Que un usuario registrado intenta acceder a su cuenta en AlquilaFácil
  When Ingresa una combinación incorrecta de correo electrónico o contraseña
  Then Se le muestra un mensaje de error indicando que las credenciales son incorrectas y se le solicita que intente nuevamente
```

US03:

```
Feature: Registro de espacios

Scenario: Registro de datos básicos
  Given Que un arrendador desea registrar su espacio en AlquilaFácil
  When Completa los campos organizados en varios pasos con información detallada sobre la propiedad
  Then Puede enviar el registro con éxito y recibir confirmación de su inclusión en la plataforma

Scenario: Validación de datos requeridos en cada paso del registro
  Given Que un arrendador está completando el registro de su espacio
  When Rellena todos los campos correspondientes a un paso específico
  Then Se activa el botón que le permite continuar al siguiente paso
```

US04:

```
Feature: Búsqueda de espacios disponibles

Scenario: Búsqueda principal por ubicación
  Given Que un arrendatario busca un espacio para eventos en una ubicación específica
  When Ingresa la ubicación deseada en el campo de búsqueda
  Then Se muestran los espacios disponibles en esa ubicación

Scenario: Búsqueda general de espacios
  Given Que un arrendatario no ingresa texto en la barra de búsqueda
  When Presiona sobre el ícono de búsqueda
  Then Se muestran todos los espacios disponibles
```

US05:

```
Feature: Filtrado de espacios disponibles

Scenario: Filtrado por capacidad
  Given Que un arrendatario desea un espacio con capacidad para un número específico de personas
```

When Aplica un filtro de capacidad en la búsqueda
Then Se muestran solo los espacios que cumplen con ese criterio

Scenario: Filtrado por categoría
Given Que un arrendatario desea un espacio de una categoría específica
When Aplica un filtro de categoría en la búsqueda
Then Se muestran solo los espacios que cumplen con ese criterio

US06:

Feature: Visualización de información del espacio

Scenario: Visualización de información
Given Que el arrendatario selecciona un espacio en AlquilaFácil
When Accede a la página de detalles del espacio
Then Puede visualizar información detallada como aforo máximo, descripción del espacio y servicios disponibles

Scenario: Visualización de reseñas del espacio
Given Que el arrendatario está revisando un espacio
When Accede a la página de comentarios
Then Puede visualizar las reseñas y calificaciones dejadas por otros usuarios sobre ese espacio

US07:

Feature: Reservar espacios

Scenario: Proceso de reserva
Given Que un arrendatario ha encontrado el espacio ideal en AlquilaFácil
When Selecciona un espacio y una fecha
Then Se muestra un formulario para completar los detalles de la reserva

Scenario: Pago de la reserva con vouchers
Given Que el usuario está a punto de culminar el proceso de reserva de un espacio
When Presiona el botón de realizar reserva
Then Realiza el pago de la reserva a través de vouchers

Scenario: Confirmación de reserva
Given Que un arrendatario ha realizado el pago de la reserva
When Es redirigido a la aplicación
Then Recibe una confirmación de reserva y los detalles se actualizan en su cuenta

US08:

Feature: Gestión del calendario de reservas

Scenario: Existencia de reserva de usuario normal
Given Que un arrendatario ha realizado una reserva de uno de mis espacios
When El propietario accede al calendario
Then Puede ver el día de la reserva resaltado en color rojo

Scenario: Existencia de reserva de usuario premium
Given Que un arrendatario con suscripción premium ha reservado uno de mis espacios
When El propietario accede al calendario
Then Puede ver el día de la reserva resaltado en color amarillo

Scenario: Existencia de reserva de espacio ajeno
Given Que un arrendatario ha realizado una reserva
When Accede al calendario
Then Puede ver el día de su reserva resaltado en color azul

US09:

Feature: Calificar y comentar sobre espacios

Scenario: Publicar reseña exitosamente
Given Que un arrendatario ha completado una reserva de un espacio
When Accede a la sección de reseñas del espacio
And Completa el formulario con calificación y comentario
Then Su reseña se publica exitosamente
And Otros usuarios pueden visualizar la reseña

Scenario: Validación de calificación
Given Que un arrendatario está escribiendo una reseña
When Intenta enviar la reseña sin seleccionar una calificación
Then El sistema muestra un mensaje de error
And No permite publicar la reseña hasta completar la calificación

US10:

Feature: Notificar sobre actividades de arrendatarios

Scenario: Recibir notificación de nueva reserva
Given Que un arrendatario ha realizado una reserva en mi espacio
When La reserva se confirma en el sistema
Then Recibo una notificación en tiempo real
And La notificación aparece en mi panel de notificaciones

Scenario: Recibir notificación de cancelación
Given Que tengo una reserva confirmada en mi espacio
When El arrendatario cancela la reserva
Then Recibo una notificación de cancelación
And El calendario se actualiza automáticamente

US11:

Feature: Controlar espacios favoritos

Scenario: Agregar espacio a favoritos
Given Que un arrendatario está visualizando un espacio
When Presiona el botón de "Agregar a favoritos"
Then El espacio se guarda en su lista de favoritos
And El ícono de favorito cambia de estado

Scenario: Eliminar espacio de favoritos
Given Que un arrendatario tiene espacios en su lista de favoritos
When Presiona el botón de "Quitar de favoritos"
Then El espacio se elimina de la lista
And Puede confirmar la eliminación en la vista de favoritos

US12:

Feature: Visualizar espacios propios publicados

Scenario: Listar todos los espacios publicados
Given Que soy un arrendador con espacios publicados
When Accedo a la sección "Mis Espacios"
Then Veo una lista con todos mis espacios publicados
And Cada espacio muestra su nombre, categoría y estado

Scenario: Ver detalles de espacio propio
Given Que estoy visualizando mi lista de espacios
When Seleccióno un espacio específico
Then Puedo ver todos los detalles completos del espacio
And Visualizo las reservas asociadas a ese espacio

US13:

Feature: Modificar espacios publicados

Scenario: Actualizar información del espacio
Given Que un arrendador desea modificar su espacio publicado
When Accede al formulario de edición
And Modifica campos como descripción, aforo o servicios
And Guarda los cambios
Then Los datos se actualizan exitosamente
And Los cambios son visibles inmediatamente

Scenario: Cambiar visibilidad temporal del espacio
Given Que un arrendador tiene un espacio publicado
When Cambia el estado de visibilidad a "No disponible temporalmente"

Then El espacio deja de aparecer en búsquedas públicas
And El arrendador puede reactivarlo posteriormente

US14:

Feature: Actualizar perfil y sesión de usuario

Scenario: Modificar nombre de usuario
Given Que un usuario está en su perfil
When Cambia su nombre de usuario
And Guarda los cambios
Then El nuevo nombre se actualiza en el sistema
And Se refleja en todas sus interacciones

Scenario: Cerrar sesión exitosamente
Given Que un usuario está autenticado en la aplicación
When Presiona el botón "Cerrar sesión"
Then El sistema cierra la sesión correctamente
And El usuario es redirigido a la pantalla de inicio de sesión

US15:

Feature: Reportar espacios inseguros

Scenario: Enviar reporte de espacio inseguro
Given Que un usuario identifica un espacio con problemas de seguridad
When Accede a la opción "Reportar espacio"
And Completa el formulario con descripción del problema
And Envía el reporte
Then El reporte se registra exitosamente
And El usuario recibe confirmación del envío

Scenario: Validación de campos requeridos en reporte
Given Que un usuario está creando un reporte
When Intenta enviar el reporte sin completar campos obligatorios
Then El sistema muestra mensajes de error
And No permite enviar el reporte hasta completar todos los campos

US16:

Feature: Visualizar espacios reportados

Scenario: Ver lista de reportes realizados
Given Que un usuario ha realizado reportes de espacios
When Accede a la sección "Mis Reportes"
Then Visualiza una lista con todos sus reportes
And Cada reporte muestra el espacio, fecha y estado

Scenario: Ver detalle de reporte específico
Given Que un usuario está viendo su lista de reportes
When Selecciona un reporte específico
Then Puede ver los detalles completos del reporte
And Visualiza el estado de resolución (pendiente, en revisión, resuelto)

US17:

Feature: Acceder a información de ayuda

Scenario: Consultar preguntas frecuentes
Given Que un usuario necesita ayuda
When Accede a la sección de "Ayuda"
Then Visualiza una lista de preguntas frecuentes organizadas por categorías
And Puede expandir cada pregunta para ver la respuesta completa

Scenario: Contactar con soporte
Given Que un usuario no encuentra respuesta en las FAQs
When Accede a la sección de "Contacto"
Then Visualiza las opciones de contacto (email, teléfono, formulario)
And Puede enviar un mensaje directamente al equipo de soporte

US18:

Feature: Control de aforo

Scenario: Monitoreo en tiempo real del aforo

Given Que un evento está en curso en mi local

When Los sensores detectan el número de personas presentes

Then Recibo información actualizada en tiempo real del aforo

And La información se muestra en el dashboard de monitoreo

Scenario: Alerta de aforo aproximándose al límite

Given Que el aforo actual está cerca del límite permitido

When El sistema detecta que se ha alcanzado el 90% de capacidad

Then Se envía una alerta preventiva al arrendador

And El sistema sugiere acciones a tomar

Scenario: Notificación de aforo excedido

Given Que el aforo máximo permitido es conocido

When Los sensores detectan que se ha superado el aforo

Then El sistema envía notificación inmediata por push y email

And Se registra el incidente en el historial

US19:

Feature: Monitoreo de niveles de ruido

Scenario: Monitoreo continuo de nivel de ruido

Given Que un evento está en curso

When Los sensores de ruido están activos

Then El sistema monitorea continuamente los decibeles

And Muestra el nivel actual en el dashboard

Scenario: Notificación de umbral de ruido superado

Given Que se ha configurado un umbral de ruido de 85 dB

When El sensor detecta niveles superiores a 85 dB

Then El arrendador recibe una notificación inmediata

And Se muestra la hora y duración del exceso

Scenario: Registro de eventos de ruido excesivo

Given Que el nivel de ruido ha superado el umbral

When El evento de ruido excesivo finaliza

Then El sistema registra el incidente con timestamp

And El registro queda disponible en el historial de eventos

US20:

Feature: Detección de humo

Scenario: Detección de humo activada

Given Que los sensores de humo están operativos

When Se detecta presencia de humo en el local

Then Se activa una alerta automática de emergencia

And El arrendador recibe notificación inmediata

Scenario: Escalamiento de alerta de humo

Given Que se ha detectado humo en el local

When La alerta es generada por el sistema

Then Se notifica al arrendador por múltiples canales (push, email, SMS)

And Se registra como incidente de alta prioridad

And Se sugiere contactar a servicios de emergencia

Scenario: Verificación post-alerta

Given Que se activó una alerta de humo

When El arrendador verifica la situación

Then Puede confirmar o descartar la emergencia en el sistema

And El incidente se actualiza con la resolución

US21:

Feature: Movimiento en áreas no autorizadas

Scenario: Detección de movimiento en zona restringida

Given Que existen áreas delimitadas como restringidas
When Los sensores detectan movimiento en una zona restringida
Then Se genera una alerta inmediata
And El arrendador recibe notificación con ubicación específica

Scenario: Registro de accesos no autorizados
Given Que se detectó movimiento en área restringida
When El evento es confirmado por los sensores
Then Se registra en el historial con timestamp y ubicación
And Se incluye captura del sensor si está disponible

Scenario: Configuración de zonas restringidas
Given Que soy arrendador de un espacio
When Accedo a la configuración de monitoreo
Then Puedo definir qué áreas son restringidas
And Configurar el nivel de sensibilidad de los sensores

US22:

Feature: Registro de incidentes

Scenario: Visualizar historial completo de eventos
Given Que soy arrendador de un espacio con sensores
When Accedo a la sección de "Registro de Incidentes"
Then Visualizo un historial cronológico de todos los eventos
And Cada evento muestra tipo, fecha, hora y detalles

Scenario: Filtrar eventos por tipo
Given Que estoy visualizando el historial de incidentes
When Aplico filtros por tipo (aforo, ruido, humo, movimiento)
Then Solo se muestran los eventos del tipo seleccionado
And Puedo combinar múltiples filtros

Scenario: Exportar evidencia de incidentes
Given Que necesito documentación de incidentes
When Selecciono un rango de fechas
And Presiono el botón "Exportar"
Then El sistema genera un reporte PDF con todos los eventos
And El reporte incluye gráficos y estadísticas

US23:

Feature: Explorar landing informativa

Scenario: Acceder a la landing page
Given Que soy un visitante nuevo
When Ingreso a la URL de AlquilaFácil
Then Visualizo una landing page clara y atractiva
And Veo información sobre los beneficios principales

Scenario: Navegación fluida en la landing
Given Que estoy en la landing page
When Me desplazo por las secciones
Then La navegación es fluida y responsive
And Las animaciones cargan correctamente

US24:

Feature: Navegar por secciones clave desde el header

Scenario: Usar menú de navegación
Given Que estoy en la landing page
When Hago clic en una opción del menú (Espacios, Publicar, Monitoreo, Beneficios)
Then Soy desplazado suavemente a la sección correspondiente
And El indicador del menú resalta la sección actual

Scenario: Navegación responsive en móvil
Given Que accedo desde un dispositivo móvil
When Presiono el botón del menú hamburguesa
Then Se despliega el menú de navegación completo
And Puedo navegar a cualquier sección

US25:

Feature: Conocer los beneficios diferenciados

Scenario: Visualizar beneficios para arrendadores

Given Que estoy en la sección de beneficios

When Visualizo las tarjetas informativas

Then Veo claramente los beneficios para arrendadores

And Cada beneficio tiene ícono descriptivo y explicación

Scenario: Visualizar beneficios para arrendatarios

Given Que estoy explorando los beneficios de la plataforma

When Reviso las tarjetas de arrendatarios

Then Comprendo las ventajas de usar AlquilaFácil como arrendatario

And Los beneficios están presentados de forma atractiva

7.2.1.5. Execution Evidence for Sprint Review.

Durante el Sprint 1 se logró desarrollar e implementar exitosamente las funcionalidades core del sistema AlquilaFácil IoT, abarcando las 25 User Stories planificadas con un total de 103 story points completados. Los principales logros alcanzados incluyen:

Funcionalidades Implementadas:

- **Sistema de Autenticación (US01, US02):** Registro e inicio de sesión con validación de credenciales, integración con Firebase Auth y soporte para Google/Facebook Sign-In.
- **Gestión de Espacios (US03, US04, US05, US06, US12, US13):** Registro completo de espacios con formulario multipaso, búsqueda y filtrado avanzado por ubicación y categoría, visualización detallada de información de espacios, y administración de espacios publicados.
- **Sistema de Reservas (US07, US08):** Proceso completo de reserva, calendario interactivo de reservas con diferenciación visual por tipo de usuario (normal/premium), y gestión de conflictos de fechas.
- **Monitoreo IoT en Tiempo Real (US18-US22):** Control de aforo con alertas automáticas, monitoreo de niveles de ruido con umbrales configurables, detección de humo con alertas de emergencia, sistema de detección de movimiento en áreas restringidas, y registro completo de incidentes con historial y exportación.
- **Interacción Social y Seguridad (US09, US11, US15, US16):** Sistema de calificaciones y reseñas, gestión de espacios favoritos, reportes de espacios inseguros, y seguimiento de reportes realizados.
- **Gestión de Usuario (US10, US14, US17):** Notificaciones push en tiempo real sobre actividades, actualización de perfil y gestión de sesión, y sección de ayuda con FAQs y contacto.
- **Landing Page Informativa (US23, US24, US25):** Página de presentación responsive con navegación fluida y secciones de beneficios diferenciados.

Tecnologías Desplegadas:

- Landing Page en GitHub Pages con Tailwind CSS y animaciones AOS
- Web Application en Netlify con Vue.js 3, Vite, Pinia y SignalR para tiempo real
- Mobile Application APK con Flutter 3.4.3+
- Web Service en Render con ASP.NET Core 8.0, Docker y arquitectura DDD
- Edge Node en Render con Python 3.9, FastAPI y Uvicorn
- Base de Datos MySQL en Railway con esquema completo normalizado
- Simulación de dispositivos IoT en Wokwi (ESP32, sensores de humo, ruido, movimiento)

A continuación se presentan las principales vistas implementadas durante este Sprint, organizadas por plataforma (Web Application y Mobile Application), junto con el video demostrativo del sistema completo.

Video Demostrativo del Sprint 1

En el siguiente video se ilustra la navegación completa del sistema, mostrando las funcionalidades principales implementadas:

URL del Video: https://upcedupe-my.sharepoint.com/:v/g/personal/u202211399_upc_edu_pe/IQBd9dv_ptmoRKUyVtbrmlzLAXPnlihm8nq2lgAdEk21Pl0?e=frg4yR

Web Application - Principales Vistas Implementadas

1. Registro de Espacios (US03)

El proceso de registro de espacios permite a los arrendadores publicar sus locales mediante un formulario multipaso intuitivo que incluye información básica, detalles del espacio, servicios disponibles, y configuración de precios.

Empezar a usar AlquilaFácil es muy sencillo

Completa los siguientes pasos para registrar tu espacio en la aplicación.

**1. Describe tu espacio**

Comparte algunos datos básicos

**2. Haz que destaque**

Agrega una foto y un título a tu espacio, nosotros nos encargamos del resto

**3. Terminar y publicar**

Agrega las últimas configuraciones y publica tu espacio

Comencemos**2. Reserva de Espacios (US07)**

El sistema de reservas permite a los arrendatarios seleccionar fechas disponibles, completar información del evento, y proceder con el pago de forma segura.

Vista Principal - Formulario de Reserva:**Detalles del local:****Casa de playa Solsito**

Calle 202, Jesús María, Lima, Perú

Aforo: 10 personas

Propietario: Admin

Descripción:

Espacio ideal para evento con piscina

Características

- Piscina

Opciones:

Ver comentarios >

Reportar espacio >

Fecha y hora de inicio: Fecha y hora de fin: **Reservar****3. Monitoreo de Reservas y Eventos IoT (US08, US18-US22)**

El dashboard de monitoreo proporciona visualización en tiempo real del estado de los espacios, incluyendo datos de sensores IoT como aforo, niveles de ruido, alertas de humo y movimiento en áreas restringidas.

The dashboard header includes the AlquilaFácil logo, a search bar, and navigation links for Publicar, Calendario, Notificaciones, and Perfil.

Monitoreo e historial

Casa de playa Solsito

Monitoreo en tiempo real (selected) | **Historial de incidencias**

Índice	Medición	Valor	Última Actualización
1	Aforo del Espacio	0/10	El: 12/11/2025, 1:14:40
2	Ruido Máximo Registrado	0 dB	El: 12/11/2025, 1:14:40
3	Detección de Humo	Todo en orden	El: 12/11/2025, 1:14:40
4	Traspasos Inautorizados	Todo seguro	El: 12/11/2025, 1:14:40

Mobile Application - Principales Vistas Implementadas

1. Registro de Espacios (US03)

La aplicación móvil permite a los arrendadores registrar espacios desde cualquier lugar, con una interfaz optimizada para dispositivos móviles que facilita la captura de fotos y la geolocalización automática.

← Registrar espacio

Empezar a usar AlquilaFácil es muy sencillo.

Completa los siguientes pasos para
registrar tu espacio en la aplicación



1. Describe tu espacio

Comparte algunos
datos básicos



2. Haz que destaque

Agrega algunas



2. Reserva de Espacios (US07)

Los arrendatarios pueden explorar, filtrar y reservar espacios directamente desde la app móvil y notificaciones push instantáneas.

Seleccionar fecha

mié, 12 nov



noviembre de 2025 ▾ < >

L	M	X	J	V	S	D
				1	2	
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30

Cancelar

ACEPTAR



3. Calendario de Reservas (US08)

La app móvil ofrece gestión de reservas mediante un calendario con diferentes leyendas para entender el estado de una reserva.

← Calendario

Toca en una fecha para ver más detalles sobre la programación



7.2.1.6. Services Documentation Evidence for Sprint Review.

Durante el Sprint 1 se logró documentar exitosamente todos los endpoints del Web Service y Edge Node utilizando OpenAPI (Swagger), proporcionando una interfaz interactiva completa para la exploración y prueba de la API REST. Los principales logros en documentación de servicios incluyen:

Logros Alcanzados:

- **Documentación Completa de 15 Controllers** en el Web Service con ASP.NET Core, incluyendo todos los endpoints de autenticación, gestión de espacios, reservas, monitoreo IoT, notificaciones, y suscripciones.
- **Documentación de 4 Endpoints IoT** en el Edge Node con FastAPI para recepción de datos de sensores (humo, ruido, capacidad, y áreas restringidas).
- **Especificaciones Detalladas** para cada endpoint incluyendo verbos HTTP, parámetros, request bodies con schemas JSON, y responses esperados.
- **Integración con Swashbuckle** en el Web Service para generación automática de documentación OpenAPI 3.0.
- **Despliegue de Documentación** en entornos de producción (Render) con acceso público para testing.
- **Annotations y Descripciones** en todos los endpoints para facilitar el entendimiento y uso de la API.

Herramientas Utilizadas:

- **Swashbuckle.AspNetCore 6.5.0** para documentación automática en ASP.NET Core
- **FastAPI built-in docs** para documentación automática del Edge Node
- **OpenAPI 3.0 Specification** como estándar de documentación
- **Swagger UI** para interfaz interactiva de pruebas

Servicio	URL de Swagger/Docs	Descripción
Web Service API	https://alquilafacil-web-service-software.onrender.com/swagger/index.html	Documentación completa de 80+ endpoints del backend
Edge Node API	https://alquilafacil-edge.onrender.com/docs	Documentación de endpoints IoT FastAPI

Repositories:

- **Web Service:** <https://github.com/Emergentez/alquilafacil-web-service>
- **Edge Node:** <https://github.com/Emergentez/alquilafacil-edge-node>

Tabla de Endpoints Documentados

A continuación se presenta la tabla completa de endpoints documentados durante el Sprint 1, organizados por controller/módulo:

Web Service - RESTful API Endpoints

Controller	Endpoint	Verbo HTTP	Parámetros	Request Body Schema	Response	User Stories
Authentication	/api/v1/authentication/sign-in	POST	-	{ "email": "string", "password": "string" }	Token JWT del usuario autenticado	US01, US02
Authentication	/api/v1/authentication/sign-up	POST	-	{ "username": "string", "password": "string", "email": "string", "name": "string", "fatherName": "string", "motherName": "string", "dateOfBirth": "string", "documentNumber": "string", "phone": "string" }	Token JWT del usuario registrado	US01, US02
Comments	/api/v1/comments/local/{localId}	GET	localId	-	Lista de comentarios del espacio	US09
Comments	/api/v1/comments	POST	-	{ "userId": 0, "localId": 0, "text": "string", "rating": 0 }	Comentario publicado con calificación	US09
Invoices	/api/v1/invoices	POST	-	{ "subscriptionId": 0, "amount": 0, "date": "2025-06-20T21:00:13.031Z" }	Boleta generada	Subscriptions
Invoices	/api/v1/invoices	GET	-	-	Lista de todas las boletas	Subscriptions
Invoices	/api/v1/invoices/{invoiceId}	GET	invoiceId	-	Boleta específica	Subscriptions
LocalCategories	/api/v1/local-categories	GET	-	-	Lista de categorías de espacios	US05
LocalEdgeNodes	/api/v1/local-edge-nodes	POST	-	{ "localId": 0, "edgeNodeUrl": "string" }	Edge Node registrado para el espacio	US18-US22
LocalEdgeNodes	/api/v1/local-edge-nodes/local-id/{localId}	GET	localId	-	Edge Node del espacio	US18-US22

Controller	Endpoint	Verbo HTTP	Parámetros	Request Body Schema	Response	User Stories
Locals	/api/v1/locals	POST	-	{ "district": "string", "street": "string", "localName": "string", "country": "string", "city": "string", "price": 0, "photoUrl": "string", "descriptionMessage": "string", "localCategoryId": 0, "userId": 0, "features": "string", "capacity": 0 }	Espacio publicado	US03
Locals	/api/v1/locals	GET	-	-	Lista de todos los espacios publicados	US04
Locals	/api/v1/locals/{localId}	GET	localId	-	Detalles completos del espacio	US06
Locals	/api/v1/locals/{localId}	PUT	localId	{ "district": "string", "street": "string", "localName": "string", "country": "string", "city": "string", "price": 0, "photoUrl": "string", "descriptionMessage": "string", "localCategoryId": 0, "userId": 0, "features": "string", "capacity": 0 }	Espacio actualizado	US13
Locals	/api/v1/locals/get-all-districts	GET	-	-	Lista de distritos con espacios	US04, US05
Locals	/api/v1/locals/get-user-locals/{userId}	GET	userId	-	Espacios publicados por el usuario	US12
Notifications	/api/v1/notifications/{userId}	GET	userId	-	Notificaciones del usuario	US10
Notifications	/api/v1/notifications/{notificationId}	DELETE	notificationId	-	Notificación eliminada	US10
Plans	/api/v1/plans	GET	-	-	Planes de suscripción disponibles	Subscriptions
Profiles	/api/v1/profiles/user/{userId}	GET	userId	-	Perfil del usuario	US14
Profiles	/api/v1/profiles/subscription-status/{userId}	GET	userId	-	Estado de suscripción del usuario	Subscriptions
Profiles	/api/v1/profiles/bank-accounts/{userId}	GET	userId	-	Cuentas bancarias del usuario	Subscriptions

Controller	Endpoint	Verbo HTTP	Parámetros	Request Body Schema	Response	User Stories
Readings	/api/v1/readings	POST	-	{ "localId": 0, "sensorTypeId": 0, "message": "string", "timestamp": "2025-06-20T21:21:48.106Z" }	Lectura de sensor registrada	US18-US22
Readings	/api/v1/readings/local-id/{localId}	GET	localId	-	Lecturas de sensores del espacio	US22
Reports	/api/v1/reports	POST	-	{ "localId": 0, "title": "string", "userId": 0, "description": "string" }	Reporte publicado	US15
Reports	/api/v1/reports/get-by-user-id/{userId}	GET	userId	-	Reportes del usuario	US16
Reports	/api/v1/reports/get-by-local-id/{localId}	GET	localId	-	Reportes del espacio	US15
Reports	/api/v1/reports/{reportId}	DELETE	reportId	-	Reporte eliminado	US16
Reservations	/api/v1/reservations	POST	-	{ "startDate": "2025-06-20T21:26:54.049Z", "endDate": "2025-06-20T21:26:54.049Z", "userId": 0, "localId": 0, "price": 0, "voucherImageUrl": "string" }	Reserva creada	US07
Reservations	/api/v1/reservations/{reservationId}	PUT	reservationId	{ "startDate": "2025-06-20T21:26:54.049Z", "endDate": "2025-06-20T21:26:54.049Z", "userId": 0, "localId": 0, "price": 0, "voucherImageUrl": "string" }	Reserva actualizada	US08
Reservations	/api/v1/reservations/{reservationId}	DELETE	reservationId	-	Reserva cancelada	US08
Reservations	/api/v1/reservations/by-user-id/{userId}	GET	userId	-	Reservas del usuario	US08
Reservations	/api/v1/reservations/reservation-user-details/{userId}	GET	userId	-	Detalles de reservas combinadas	US08
Subscriptions	/api/v1/subscriptions	POST	-	{ "planId": 0, "userId": 0, "voucherImageUrl": "string" }	Suscripción creada	Subscriptions
Subscriptions	/api/v1/subscriptions	GET	-	-	Lista de suscripciones	Subscriptions
Subscriptions	/api/v1/subscriptions/{subscriptionId}	GET	subscriptionId	-	Suscripción específica	Subscriptions
Subscriptions	/api/v1/subscriptions/{subscriptionId}	PUT	subscriptionId	-	Suscripción activada (premium)	Subscriptions
Users	/api/v1/users/{userId}	GET	userId	-	Usuario específico	US01, US02, US14

Controller	Endpoint	Verbo HTTP	Parámetros	Request Body Schema	Response	User Stories
Users	/api/v1/users/{userId}	PUT	userId	{ "username": "string" }	Usuario actualizado	US14
Users	/api/v1/users	GET	-	-	Lista de usuarios (Admin only)	US14
Users	/api/v1/users/get-username/{userId}	GET	userId	-	Nombre de usuario	US14

Edge Node - IoT Sensor Endpoints

Endpoint	Verbo HTTP	Parámetros	Request Body Schema	Response	User Stories
/readings/smoke	POST	-	{ "device_id": 0, "message": "string" }	Lectura de sensor de humo procesada	US20
/readings/noise	POST	-	{ "device_id": 0, "message": "string" }	Lectura de sensor de ruido procesada	US19
/readings/capacity	POST	-	{ "device_id": 0, "message": "string" }	Lectura de sensor de aforo procesada	US18
/readings/restricted-area	POST	-	{ "device_id": 0, "message": "string" }	Lectura de sensor de movimiento procesada	US21
/readings/local-id/{local_id}	GET	local_id	-	Historial de lecturas del espacio	US22

Capturas de Documentación Swagger

A continuación se presentan capturas de la documentación interactiva generada con Swagger/OpenAPI:

Authentication Controller:

Se muestra un intento de Login al correo admin@gmail.com

POST /api/v1/authentication/sign-in

Parameters

No parameters

Request body application/json

```
{
  "email": "admin@gmail.com",
  "password": "Admin@123"
}
```

Responses

Curl

```
curl -X 'POST' \
'https://alquilarafacil-web-service-software.onrender.com/api/v1/authentication/sign-in' \
-H 'accept: */*' \
-H 'Content-Type: application/json' \
-d '{
  "email": "admin@gmail.com",
  "password": "Admin@123"
}'
```

Request URL

<https://alquilarafacil-web-service-software.onrender.com/api/v1/authentication/sign-in>

Server response

Code	Details
200	Response body <pre>{ "id": 1, "username": "Admin", "token": "eyJhbGciOiIodHRwOj8vd3d3LnczLm9yZy8yMDAxLzA0L3htbGRzaWctbWYzSNobWFjLXNoYTII1L1tsInR5cCI6IkpxVCJ9.eyJleHAiOiE3NjMzM1MDAsImh0dHA6LyZ2h1bkFzLnhtbHNvYXAb3JnL3dzLzIwMDU0valWR1bnRpdkHkVzxhah1zl3NpZC161tELC1odHRwOj8vc2NoZk1hc54bhxxzb2FwLm9yZy93cbyWDAl1zA1121z2h5SaX5L2NsVl1tcy9uWm11fjo1QWtaM411C1odh1hw018vc2Nz02h1hcy5taWlyb3NvZnQuY29tL3dzLzIwMDgvMDYvalW1bPwHkVzch-h1-13vbGU01JBZG1pb1tsIm1hdCI6MTc2Mjkyc0cwCwibm3mIjoxNzYyOTI4NzAwfQ.panv5d1brJydgCewqVIYpr8RxAdrls3Kw#_rs1araE" }</pre> <p>Download</p>

Comments Controller:

Se muestra la petición de comentarios con respecto a un local 1.

GET /api/v1/comment/local/{localId}

Parameters

Name	Description
localId <small>* required</small> integer(\$int32) (path)	<input type="text" value="1"/>

Responses

Code	Description	Links
200	OK	No links

Execute

Cancel

POST /api/v1/comment

Parameters

No parameters	
---------------	--

Request body

```
{ "userId": 2, "localId": 1, "text": "string", "rating": 5 }
```

Execute

Cancel

Reset

Invoices Controller:

Se muestra la petición de boletas por ID.

GET /api/v1/invoice

Parameters

No parameters

Responses

Code	Description	Links
200	OK	No links

Execute

GET /api/v1/invoice/{invoiceId}

Parameters

Name	Description
invoiceId * required	integer(\$int32) (path)
<input type="text" value="InvoiceId"/>	

Responses

Code	Description	Links
200	OK	No links

Execute

LocalCategories Controller:

Se muestra la petición para obtener todas las categorías de locales del sistema.

LocalCategories

GET /api/v1/local-categories

Parameters

No parameters

Responses

Curl

```
curl -X 'GET' \
'https://alquillafacil-web-service-software.onrender.com/api/v1/local-categories' \
-H 'accept: */*' Copy
```

Request URL

```
https://alquillafacil-web-service-software.onrender.com/api/v1/local-categories Copy
```

Server response

Code	Details
200	Response body <pre>[{"id": 1, "name": "Casa de playa", "photoUrl": "https://tse4.mm.bing.net/th?id=OIP.N62R-B5j13QHIL90hcdJiwHaHa&pid=Api&p=0&h=180"}, {"id": 2, "name": "Casa de campo", "photoUrl": "https://cdn-icons-png.flaticon.com/512/74/74951.png"}, {"id": 3, "name": "Casa urbana", "photoUrl": "https://tse1.mm.bing.net/th?id=OIP.LnALBZ2Bu7Mw46vJKewWAhHa&pid=Api&p=0&h=180"}, {"id": 4, "name": "Salón elegante", "photoUrl": "https://cdn3.iconfinder.com/data/icons/beauty-cosmetics-1-line/128/beauty-salon_beauty_salon_barbershop_glamour-512.png"}] Download</pre> Response headers <pre>cf-cache-status: DYNAMIC cf-ray: 99d3f76e98866fb-LIM content-encoding: br content-type: application/json; charset=utf-8 date: Wed, 12 Nov 2025 06:27:58 GMT rndr-id: 12bf1c1fb-875b-4e64 server: Kestrel vary: Accept-Encoding x-render-origin-server: Kestrel</pre>

LocalEdgeNodes Controller:

Se muestra la petición para crear un nodo Edge a un espacio en específico.

The screenshot shows a POST request to /api/v1/local-edge-nodes. The request body contains the following JSON:

```
{
  "localId": 1,
  "edgeNodeUrl": "https://local-node.com"
}
```

Below the request, there is a large blue button labeled "Execute".

The screenshot shows a PUT request to /api/v1/local-edge-nodes/{localId}. The "localId" parameter is set to 1. The request body is empty and has a type of application/json.

Locals Controller:

Se muestra la petición para obtener todos los locales de la plataforma.

GET /api/v1/locals

Parameters

No parameters

Execute Clear

Responses

Curl

```
curl -X 'GET' \
'https://alquilarafacil-web-service-software.onrender.com/api/v1/locals' \
-H 'accept: */*' 
```

Request URL

<https://alquilarafacil-web-service-software.onrender.com/api/v1/locals>

Server response

Code	Details	Links
200	<p>Response body</p> <pre>[{ "id": 1, "localName": "Casa de playa Solsito", "descriptionMessage": "Espacio ideal para evento con piscina", "address": "Calle 202, Jesús María, Lima, Perú", "price": 10, "capacity": 10, "photoUrls": ["https://res.cloudinary.com/dd2zyf0ii/image/upload/v1762928832/j1dsm1aa7bz1vh9tg9p.webp", "https://res.cloudinary.com/dd2zyf0ii/image/upload/v1762928832/dle10sgkpz9lmvfprsk.jpg", "https://res.cloudinary.com/dd2zyf0ii/image/upload/v1762928832/arqkhkmyxf8ouu27jsh9.jpg"], "features": "Piscina", "localCategoryId": 1, "userId": 1 }]</pre> <p> </p> <p>Response headers</p> <pre>cf-cache-status: DYNAMIC cf-ray: 9bd3fa183f366df7-LIM content-encoding: br content-type: application/json; charset=utf-8 date: Wed, 12 Nov 2025 06:29:48 GMT rnr-id: a2032h34-1b52-4674 server: cloudflare vary: Accept-Encoding x-render-origin-server: Kestrel</pre>	No links
Responses	Code Description	Links
200	OK	No links

Notifications Controller:

Se muestra la petición para obtener todas las notificaciones de un usuario.

GET /api/v1/notification/{userId}

Parameters

Name	Description
userId * required integer(\$int32) (path)	<input type="text" value="1"/>

Responses

Code	Description	Links
200	OK	No links

Execute

DELETE /api/v1/notification/{notificationId}

Parameters

Name	Description
notificationId * required integer(\$int32) (path)	<input type="text" value="1"/>

Responses

Code	Description	Links
200	OK	No links

Execute

Plans Controller:

Se muestra la petición para obtener todos los planes de la plataforma.

GET /api/v1/plan

Parameters

No parameters

Responses

Curl

```
curl -X 'GET' \
'https://alquilafacil-web-service-software.onrender.com/api/v1/plan' \
-H 'accept: */*'
```

Request URL

<https://alquilafacil-web-service-software.onrender.com/api/v1/plan>

Server response

Code	Details	Links
200	<p>Response body</p> <pre>[{ "id": 1, "name": "Plan Premium", "service": "El plan premium te permitirá acceder a funcionalidades adicionales en la aplicación", "price": 20 }]</pre> <p>Response headers</p> <pre>cf-cache-status: DYNAMIC cf-ray: 99d3fb7dcf4b6df7-LIM content-encoding: br content-type: application/json; charset=utf-8 date: Wed, 12 Nov 2025 06:30:45 GMT rndr-id: 410bc8d0-5b85-434f server: cloudflare vary: Accept-Encoding x-render-origin-server: Kestrel</pre>	

Responses

Code	Description	Links
200	OK	No links

Profiles Controller:

Se muestra la petición para obtener el perfil de un usuario con ID 1.

GET /api/v1/profiles/user/{userId}

Parameters

Name	Description
userId * required integer(\$int32) (path)	1

Responses

Curl

```
curl -X 'GET' \
'https://alquilafacil-web-service-software.onrender.com/api/v1/profiles/user/1' \
-H 'accept: */*'
```

Request URL

<https://alquilafacil-web-service-software.onrender.com/api/v1/profiles/user/1>

Server response

Code	Details	Links
200	Response body <pre>{ "id": 1, "fullName": "Admin Alquila Facil", "phone": "000000000", "documentNumber": "000000000", "dateOfBirth": "01-01-2001" }</pre> <p>Copy Download</p> Response headers <pre>cf-cache-status: DYNAMIC cf-ray: 99468937f7614d-LIM content-encoding: br content-type: application/json; charset=utf-8 date: 12 Nov 2025 06:39:41 GMT x-request-id: fbd2932-24ce-4156 server: cloudflare vary: Accept-Encoding x-render-origin-server: Kestrel</pre>	No links

Responses

Code	Description	Links
200	OK	No links

Reservations Controller:

Se muestra la petición para obtener las reservaciones de un usuario con ID 2.

GET /api/v1/reservation/by-user-id/{userId}

Parameters

Name	Description
userId * required	integer(\$int32) (path)
2	

Responses

Curl

```
curl -X 'GET' \
'https://alquilafacil-web-service-software.onrender.com/api/v1/reservation/by-user-id/2' \
-H "accept: */*"
```

Request URL

<https://alquilafacil-web-service-software.onrender.com/api/v1/reservation/by-user-id/2>

Server response

Code	Details	Links
200	Response body <pre>[{ "id": 1, "startDate": "2025-11-13T01:37:00", "endDate": "2025-11-13T05:37:00", "userId": 2, "localId": 1 }]</pre>	Copy Download
	Response headers <pre>cf-cache-status: DYNAMIC cf-ray: 99d407667f6c6df9-LIM content-encoding: br content-type: application/json; charset=utf-8 date: Mon, 13 Nov 2025 06:38:52 GMT vndr-id: 39e70b34-ef88-4c6b server: cloudflare vary: Accept-Encoding x-render-origin-server: Kestrel</pre>	

Responses

Code	Description	Links
200	OK	No links

Edge Node API Documentation:

Se muestran los endpoints para obtener lecturas de diferentes sensores IoT.

Sensor Readings

POST	/api/v1/edge-node/edge/readings/smoke	Create Smoke Reading	▼
POST	/api/v1/edge-node/edge/readings/noise	Create Noise Reading	▼
POST	/api/v1/edge-node/edge/readings/capacity	Create Capacity Reading	▼
POST	/api/v1/edge-node/edge/readings/restricted-area	Create Restricted Area Reading	▼
GET	/api/v1/edge-node/edge/readings/local-id/{local_id}	Get Readings By Local Id	▼

7.2.1.7. Software Deployment Evidence for Sprint Review.

Durante el Sprint 1 se completó exitosamente el despliegue de todos los componentes digitales del sistema AlquilaFácil IoT en entornos de producción. Este proceso incluyó la configuración de cuentas en múltiples cloud providers, la configuración de recursos y servicios en la nube, la implementación de pipelines de CI/CD, y la configuración de bases de datos y servicios de integración. A continuación se detallan todas las actividades de deployment realizadas durante este sprint.

Resumen de Actividades de Deployment

Componentes Desplegados:

1. **Landing Page** → GitHub Pages
2. **Web Application** → Netlify
3. **Web Service (Backend API)** → Render con Docker
4. **Edge Node (IoT Gateway)** → Render con Docker
5. **Mobile Application** → APK Flutter distribuido vía GitHub Releases
6. **Base de Datos** → Railway MySQL

Cuentas y Servicios Configurados:

- Cuenta de GitHub para repositorios y GitHub Pages
- Cuenta de Netlify para deployment de frontend
- Cuenta de Render para servicios backend containerizados
- Cuenta de Railway para base de datos MySQL
- Cuenta de Firebase para autenticación y notificaciones push
- Configuración de dominios y SSL automático en todas las plataformas

1. Landing Page - Deployment en GitHub Pages

Plataforma: GitHub Pages

URL de Despliegue: <https://emergentez.github.io/alquilafacil-landing-page/>

Proceso de Deployment:

1. Configuración del Repositorio:

- Creación del repositorio **alquilafacil-landing-page** en GitHub
- Configuración de permisos de acceso para el equipo
- Estructura de carpetas con **src/** conteniendo HTML, CSS compilado y assets

2. Configuración de GitHub Pages:

- Acceso a Settings > Pages en el repositorio
- Selección de rama **main** como fuente de deployment
- Configuración de ruta **raíz /** para servir la landing
- GitHub Pages detecta automáticamente el **index.html**

3. Build Process:

- Ejecución local de **npm run build** para compilar Tailwind CSS
- Generación de **output.css** minificado desde **input.css**
- Commit y push de archivos compilados a rama **main**

4. Automatización:

- GitHub Actions configurado para rebuild automático en cada push
- SSL/HTTPS habilitado automáticamente por GitHub
- CDN de GitHub para servir contenido estático globalmente

Evidencia de Deployment:

The screenshot shows the GitHub Actions interface for the repository 'alquilafacil-landing-page'. A workflow named 'pages build and deployment' is displayed, showing a single job named 'deploy' that succeeded on October 7 at 9:51 AM. The job details show the deployment process to GitHub Pages, including the command 'Run actions/deploy-pages@v4' and the creation of a deployment artifact with ID '2f61f12b6a7ca1ad22b5a85ea36513b60b005a11a'. The workflow also includes steps for setting up the environment and evaluating the deployment URL.

Estado: Desplegado exitosamente y accesible públicamente

2. Web Application - Deployment en Netlify

Plataforma: Netlify

URL de Despliegue: <https://alquila-facil-app-iot.netlify.app/>

Proceso de Deployment:

1. Configuración de Cuenta Netlify:

- Creación de cuenta Netlify con autenticación GitHub
- Vinculación del repositorio [alquilafacil-web-app](#)
- Configuración de permisos de build y deploy

2. Configuración del Proyecto:

- **Build Command:** `npm run build`
- **Publish Directory:** `dist`
- **Node Version:** 18.x
- Detección automática de framework Vue.js + Vite

3. Variables de Entorno:

```
VITE_API_URL=[URL_del_Web_Service_en_Render]  
VITE_FIREBASE_API_KEY=[Firebase_API_Key]  
VITE_FIREBASE_AUTH_DOMAIN=[Firebase_Auth_Domain]  
VITE_FIREBASE_PROJECT_ID=[Firebase_Project_ID]
```

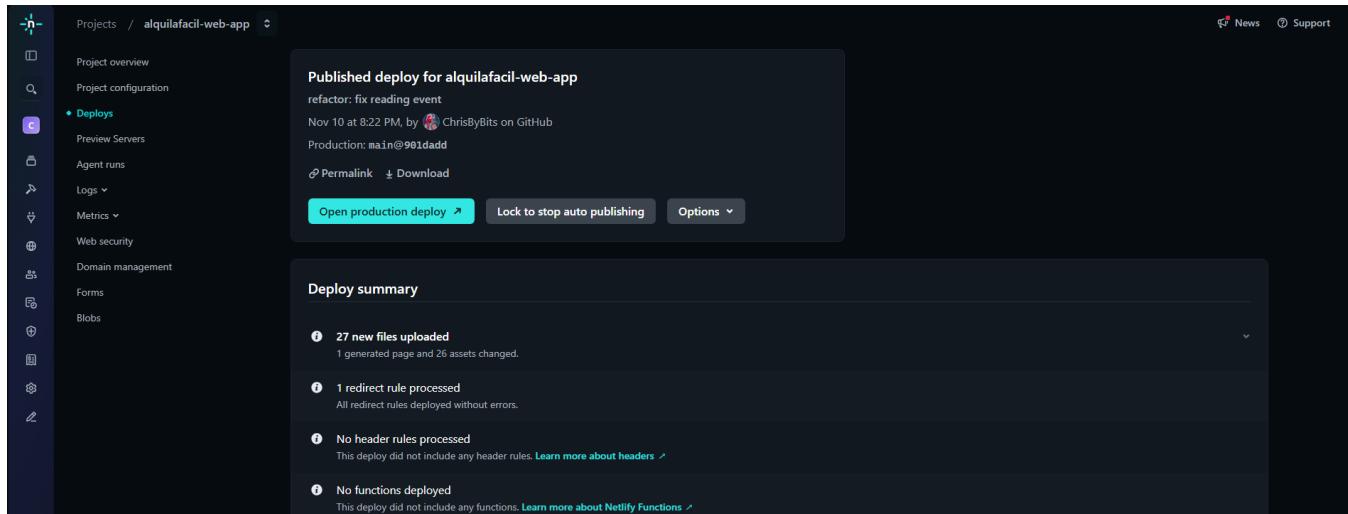
4. Configuración de Despliegue:

- Continuous Deployment habilitado desde rama `main`
- Branch deploys configurados para preview de PRs
- Redirects configurados en `netlify.toml` para SPA routing
- Configuración de headers de seguridad (CORS, CSP)

5. Optimizaciones:

- Asset optimization automático (minificación JS/CSS)
- Image optimization mediante Netlify Image CDN
- Compresión Gzip/Brotli habilitada
- CDN global con edge nodes en múltiples regiones

Evidencia de Deployment:



Estado: Desplegado exitosamente con SSL y CDN global

3. Web Service (Backend API) - Deployment en Render

Plataforma: Render (Docker Container)

URL de Despliegue: [\[URL_Render_Web_Service\]](#)

Swagger Docs: [\[URL_Render_Web_Service\]/swagger](#)

Proceso de Deployment:

1. Configuración de Cuenta Render:

- Creación de cuenta Render con autenticación GitHub
- Configuración de billing (free tier para desarrollo)
- Vinculación de repositorio [alquilafacil-web-service](#)

2. Creación del Web Service:

- Tipo: Web Service
- Environment: Docker
- Region: Oregon (US West)
- Instance Type: Free tier (512MB RAM)

3. Configuración de Docker:

- Render detecta automáticamente el [Dockerfile](#) en la raíz
- Build del contenedor multi-stage con .NET 8.0
- Exposición de puertos 8080 (HTTP) y 8081 (HTTPS)
- Health check endpoint configurado

4. Variables de Entorno:

```
ASPNETCORE_ENVIRONMENT=Production
ConnectionStrings__DefaultConnectionString=[Railway_MySQL_Connection_String]
JWT__Secret=[JWT_Secret_Key]
JWT__Issuer=AlquilaFacilAPI
JWT__Audience=AlquilaFacilClients
JWT__ExpirationMinutes=1440
CORS__AllowedOrigins=[Netlify_URL],[Mobile_App_Domain]
```

5. Configuración de Base de Datos:

- Conexión a Railway MySQL mediante connection string seguro
- Entity Framework Core migrations aplicadas en startup
- Connection pooling configurado para optimizar conexiones

6. Deployment Automático:

- Auto-deploy habilitado desde rama [main](#)
- Build time: ~5-7 minutos
- Zero-downtime deployments con health checks
- Logs persistentes en dashboard de Render

Evidencia de Deployment:

The screenshot shows the Render dashboard for the service 'alquilafacil-web-service-software'. The service is currently running on a free instance. The logs section displays deployment logs for commit 876441f, which updated the production database connection string. The logs also show the application starting and the hosting environment being set to Production. A purple banner at the top of the logs section states: 'Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.' There is a link to 'Upgrade now'.

Estado: Desplegado exitosamente y respondiendo a requests

4. Edge Node (IoT Gateway) - Deployment en Render

Plataforma: Render (Docker Container)

URL de Despliegue: <https://alquilafacil-edge-node-fsaa.onrender.com>

FastAPI Docs: <https://alquilafacil-edge-node-fsaa.onrender.com/docs>

Proceso de Deployment:

1. Creación del Web Service:

- Tipo: Web Service
- Environment: Docker
- Region: Oregon (US West)
- Instance Type: Free tier (512MB RAM)

2. Configuración de Docker:

- Dockerfile con Python 3.9.23-bookworm como base
- Instalación de dependencias con `pip install -r requirements.txt`
- FastAPI + Unicorn como servidor ASGI
- Exposición del puerto 80

3. Variables de Entorno:

```
BACKEND_API_URL=[URL_del_Web_Service]  
DATABASE_PATH=/app/alquilafacil.db  
WEBSOCKET_ENDPOINT=wss://wokwi.com/projects/433560726038801409  
LOG_LEVEL=INFO
```

4. Configuración de Integración:

- Conexión HTTP al Web Service para persistir lecturas
- Base de datos SQLite local para cache temporal
- WebSocket listener para datos de sensores Wokwi
- Validación de datos con Pydantic schemas

5. Deployment Automático:

- Auto-deploy desde rama `main`
- Build time: ~3-4 minutos
- Logs de lecturas de sensores en tiempo real
- Health check en endpoint raíz `/`

Evidencia de Deployment:

The screenshot shows the Render logs interface for the application 'alquilafacil-edge'. The left sidebar includes sections for Dashboard, Events, Settings, MONITOR (Logs selected), Metrics, MANAGE (Environment, Shell, Scaling, Previews, Disks, Jobs), and a footer with Changelog, Invite a friend, Contact support, and Render Status. The main area displays log entries from November 12, 2023, at 01:56:25 AM. The logs show the application starting up, watching for changes in '/app', and receiving sensor data requests via WebSocket and HTTP. A purple banner at the top right says 'Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.' and 'Upgrade now'.

```
Nov 12 01:56:25 AM  wjg2p INFO: Will watch for changes in these directories: ['/app']
Nov 12 01:56:25 AM  wjg2p INFO: Unicorn running on http://0.0.0.0:80 (Press CTRL-C to quit)
Nov 12 01:56:25 AM  wjg2p INFO: Started reloader process [1] using Watchfiles
Nov 12 01:56:29 AM  wjg2p INFO: Started serve process [8]
Nov 12 01:56:29 AM  wjg2p INFO: Waiting for application startup.
Nov 12 01:56:32 AM  wjg2p INFO: Application startup complete.
Nov 12 01:56:48 AM  wjg2p INFO: 10.17.194.196.0 - "WebSocket /api/v1/web-socket/ws/notifications/1" [accepted]
Nov 12 01:56:48 AM  wjg2p INFO: connection open
Nov 12 01:57:19 AM  wjg2p [green] Sign-in request sent to backend: 200
Nov 12 01:57:19 AM  wjg2p INFO: 10.17.136.4:0 - "GET /docs HTTP/1.1" 200 OK
Nov 12 01:57:21 AM  wjg2p INFO: 10.16.162.196.0 - "GET /openapi.json HTTP/1.1" 200 OK
```

Estado: Desplegado y recibiendo datos de sensores IoT

5. Base de Datos - Railway MySQL

Plataforma: Railway

Tipo: MySQL 8.0

Region: US West

Proceso de Configuración:

1. Creación de Proyecto Railway:

- Registro en Railway con GitHub authentication
- Creación de nuevo proyecto "AlquilaFacil"
- Adición de servicio MySQL desde template

2. Configuración de MySQL:

- Versión: MySQL 8.0
- Storage: 1GB (free tier)
- Generación automática de credenciales seguras
- Connection string público proporcionado

3. Configuración de Acceso:

```
Host: [Railway_MySQL_Host]
Port: [Railway_MySQL_Port]
Database: railway
Username: root
Password: [Generated_Secure_Password]
```

4. Inicialización del Esquema:

- Entity Framework Core migrations aplicadas desde Web Service
- Creación automática de tablas al iniciar el servicio
- Bounded contexts: IAM, Locals, Booking, Management, Notifications, Profiles, Subscriptions

5. Seguridad y Backups:

- Conexión encriptada SSL/TLS obligatoria
- Backups automáticos diarios por Railway
- Monitoring de uso de recursos en dashboard
- Logs de queries disponibles

Tablas Creadas:

- Users, Profiles, UserRoles (IAM)
- Locals, LocalCategories, Comments (Locals)
- Reservations (Booking)
- Readings, SensorTypes, LocalEdgeNodes (Management)
- Notifications (Notifications)
- Subscriptions, Plans, Invoices (Subscriptions)
- Reports (Management)

Estado: Base de datos operativa y conectada a servicios

6. Mobile Application - Distribution

Plataforma: GitHub Releases + APK Direct Download

URL de Descarga: [GitHub_Releases_URL_APK]

Proceso de Generación y Distribución:

1. Build del APK:

```
flutter clean
flutter pub get
flutter build apk --release
```

2. Configuración de Firma:

- Keystore generado con keytool
- Configuración en `android/key.properties`
- Firma digital del APK para distribución

3. Generación de APK:

- APK universal: ~45MB
- Min SDK: Android 5.0 (API 21)
- Target SDK: Android 13 (API 33)

4. Distribución:

- Upload del APK a GitHub Releases

- Tag de versión: v1.0.0-sprint1
- Release notes con changelog
- Link directo de descarga compartido

Estado: APK generado y disponible para descarga

7. Embedded Application - Wokwi Simulation

Plataforma: Wokwi Cloud Simulator

URL del Proyecto: <https://wokwi.com/projects/433560726038801409>

Configuración de Simulación:

1. Creación del Proyecto Wokwi:

- Microcontrolador: ESP32 DevKit v1
- Sensores configurados:
 - MQ-2 (Sensor de humo)
 - KY-037 (Sensor de ruido)
 - HC-SR501 (Sensor de movimiento PIR)
 - Contador de personas simulado

2. Configuración de Código:

- Código C/C++ con Arduino framework
- Bibliotecas: WiFi.h, HttpClient.h, ArduinoJson
- Lógica de lectura de sensores cada 5 segundos
- Envío de datos a Edge Node vía HTTP POST

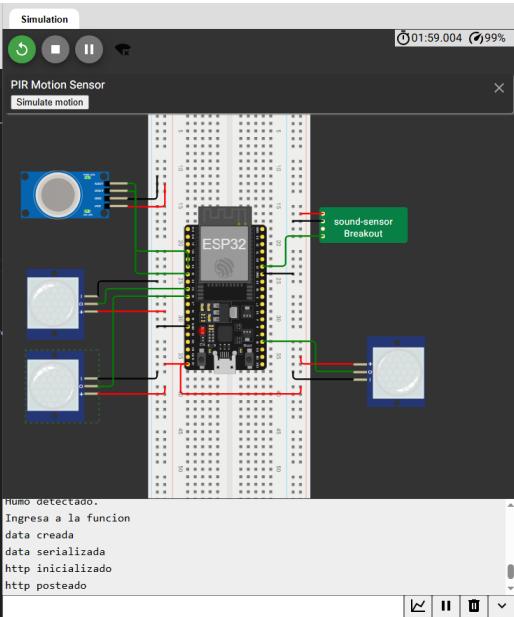
3. Configuración de Conectividad:

- WiFi credentials configurados para Wokwi Cloud
- Endpoint del Edge Node hardcodeado
- JSON payloads con formato { "device_id", "message" }

4. Simulación Continua:

- Proyecto compartido públicamente
- Simulación puede ejecutarse 24/7
- Logs de serial monitor visibles en tiempo real

Evidencia de Deployment:



The screenshot shows the Wokwi Cloud Simulator interface. On the left is the code editor with the sketch file open, displaying C/C++ code for an ESP32 device. The code includes definitions for various sensors and endpoints, such as PIR Motion Sensor, Smoke Sensor, and Sound Sensor. On the right is the simulation window, which shows a breadboard setup with an ESP32 DevKit v1 board at the center. Various sensors (PIR, Smoke, Sound) are connected to the pins, and a computer monitor is shown receiving data from the board. The simulation status bar at the top right indicates 01:59.004 and 99% battery. The serial monitor at the bottom shows text output related to sensor detections.

```

sketch.ino diagram.json sound-sensor.chip.json sound-sensor.chip.c Actuator.cpp Actuator.h CommandHandler.h Device.cpp Device.h EventHandler.h
ModestoTh Sensor.cpp Sensor.h RestrictedAreaMovementSensor.h RestrictedAreaMovementSensor.cpp MonitoringLocalDevice.h MonitoringLocalDevice.h
SmokeSensor.h SmokeSensor.cpp EntryMovementSensor.h EntryMovementSensor.cpp libraries.txt SoundSensor.h SoundSensor.cpp Library Manager

1 #include <ArduinoJson.h>
2 #include <Cmsis.h>
3 #include "MonitoringLocalDevice.h"
4 #include <Arduino.h>
5
6 #define DEVICE_ID 2234
7 //##define ENDPOINT_URL "https://<linkdeledge>/api/v1/edge-node/edge/"
8
9 #define ENDPOINT_SMOKE "https://alquillafacil-edge.onrender.com/api/v1/edge-node/edge/readings/smoke"
10 #define ENDPOINT_RESTRICTED "https://alquillafacil-edge.onrender.com/api/v1/edge-node/edge/readings/restricted-area"
11 #define ENDPOINT_ENTRY "https://alquillafacil-edge.onrender.com/api/v1/edge-node/edge/readings/capacity"
12 #define ENDPOINT_SOUND "https://alquillafacil-edge.onrender.com/api/v1/edge-node/edge/readings/noise"
13
14 char timeStringBuffer[20];
15
16 #define CONTENT_TYPE_HEADER "Content-Type"
17 #define APPLICATION_JSON "application/json"
18
19 MonitoringLocalDevice::MonitoringLocalDevice(int restrictedAreaPin, int smokeDigitalPin, int smokeAnalogPin, int innerMovementPin, int outerMovementPin, int soundPin) {
20   : smokeSensor(smokeDigitalPin, smokeAnalogPin, this),
21   : innerMovementSensor(restrictedAreaPin, this),
22   : outerMovementSensor(innerMovementPin, this),
23   : soundSensor(soundPin, this),
24   : pendingEvent(Event(-1))
25 }
26
27 void MonitoringLocalDevice::on(Event event) {
28   if (event == SmokeSensor::SMOKE_DETECTED_EVENT) {
29     Serial.println("Humo detectado.");
30     int analogValue = smokeSensor.readValue();
31
32     String mensaje = "Humo detectado en el area (valor: " + String(analogValue) + ")";
33     sendEventToServer("smoke", mensaje, ENDPOINT_SMOKE);
34   }
35
36   if (event == RestrictedAreaMovementSensor::RESTRICTED_AREA_ACCESS_DETECTED_EVENT) {
37     Serial.println("Movimiento no autorizado detectado.");
38
39     String mensaje = "Movimiento en el area restringida";
40   }
41 }

```

Estado: Simulación operativa y enviando datos

8. Servicios de Integración

Firebase Configuration

Servicios Habilitados:

- **Authentication:** Email/Password, Google OAuth, Facebook Login
- **Cloud Messaging:** Push notifications para mobile app
- **Hosting:** (Reservado para futuro deployment)

Configuración:

- Proyecto: [alquilafacil-iot](#)
- Web API Key configurado en Netlify
- Android package name registrado
- SHA-1 fingerprints configurados

Control de Versiones - Git Flow Workflow

Para todos los componentes se implementó la metodología **GitFlow Workflow**:

- **main:** Rama de producción estable, desplegada automáticamente
- **develop:** Rama de integración para features en desarrollo
- **feature/*:** Ramas individuales para cada funcionalidad
- **hotfix/*:** Ramas para correcciones urgentes en producción

Protección de Ramas:

- Pull requests obligatorios para merge a **main**
- Code reviews requeridos (mínimo 1 aprobación)
- Tests automáticos antes de merge
- No se permite push directo a **main**

Resumen de URLs de Deployment

Componente	URL/Acceso	Estado
Landing Page	https://emergentez.github.io/alquilafacil-landing-page/	<input checked="" type="checkbox"/> Activo
Web Application	https://alquila-facil-app-iot.netlify.app/	<input checked="" type="checkbox"/> Activo
Web Service API	[URL_Render_Web_Service]/swagger	<input checked="" type="checkbox"/> Activo
Edge Node	https://alquilafacil-egde-node-fsaa.onrender.com/docs	<input checked="" type="checkbox"/> Activo
Mobile APK	[GitHub_Releases_URL]	<input checked="" type="checkbox"/> Disponible
Wokwi Simulation	https://wokwi.com/projects/433560726038801409	<input checked="" type="checkbox"/> Activo
Railway MySQL	[Connection_String_Privado]	<input checked="" type="checkbox"/> Operativo

Nota: Todos los servicios están desplegados en entornos de producción con SSL/TLS, monitoreo activo, y configurados para auto-deployment desde sus respectivos repositorios.

7.2.1.8. Team Collaboration Insights during Sprint.

Durante el Sprint 1, el equipo trabajó de manera colaborativa aplicando buenas prácticas de control de versiones y comunicación efectiva para garantizar el cumplimiento de los objetivos establecidos.

• Canales de comunicación

La coordinación del equipo se realizó principalmente a través de **Discord**, donde se llevaron a cabo reuniones remotas para definir tareas, revisar avances y resolver bloqueos. Adicionalmente, se utilizaron **mensajes por WhatsApp** para recordatorios y organización rápida. Esta combinación permitió una comunicación fluida y en tiempo real.

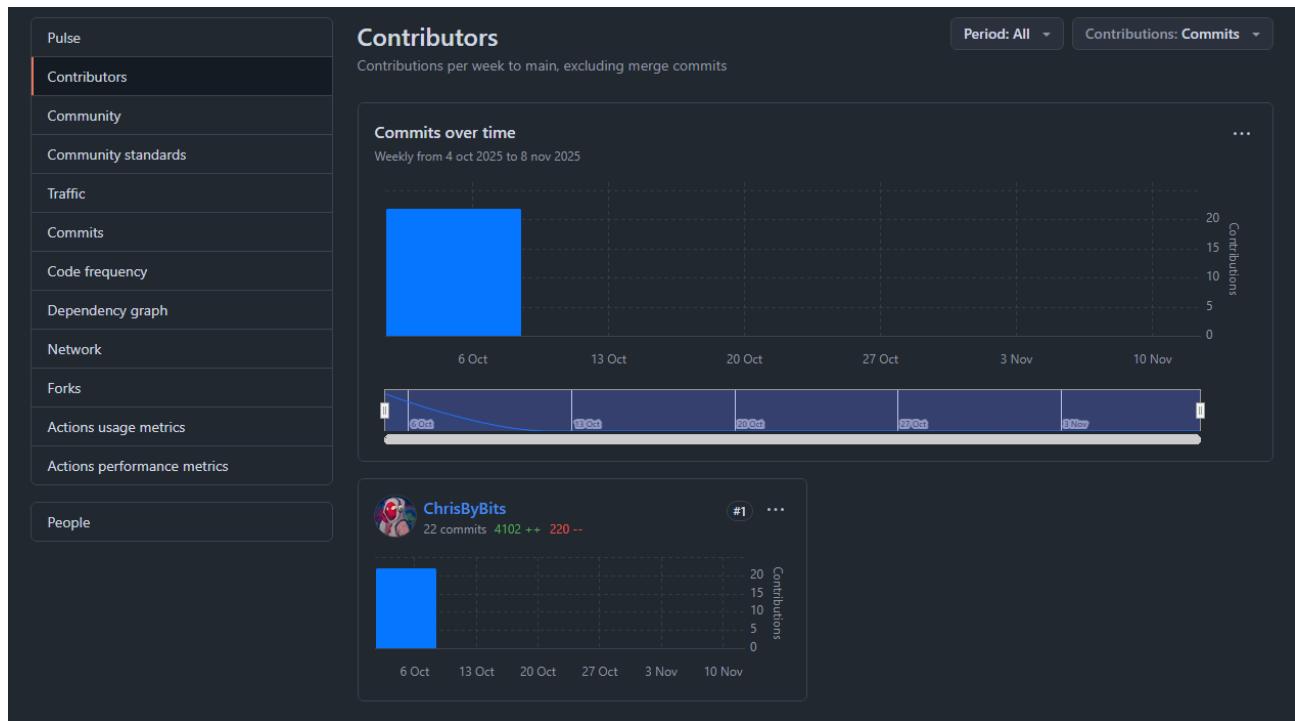
• Herramientas de gestión

GitHub Projects y commits verificados permitieron llevar un registro detallado de las tareas completadas y del responsable de cada cambio. Las tareas fueron distribuidas entre todos los miembros del equipo, y se registraron contribuciones en los distintos servicios: Landing Page, Frontend Web, y documentación técnica. El uso de etiquetas claras en los commits facilitó el rastreo de avances por funcionalidad y componente.

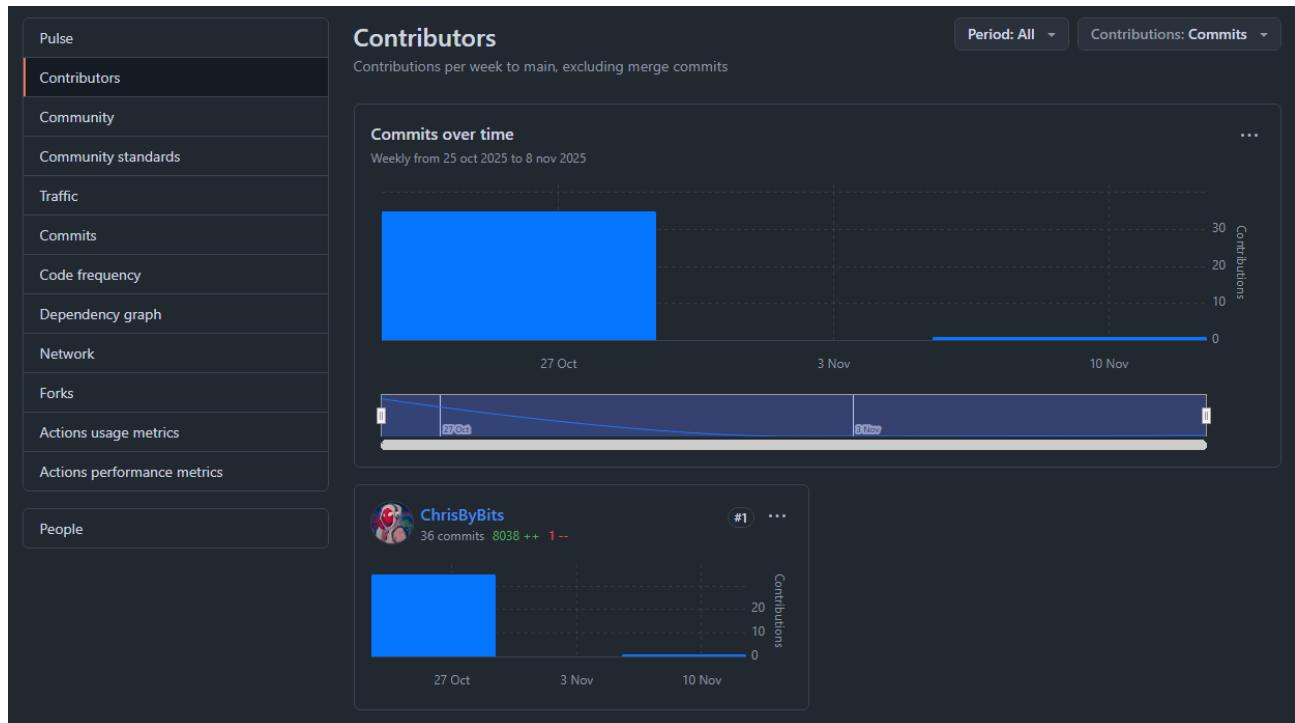
• Resultados obtenidos

Se logró mantener un flujo constante de integración y revisión de código sin conflictos mayores. Todos los miembros del equipo realizaron commits y participaron activamente en el proceso de desarrollo. La colaboración fue efectiva, permitiendo cumplir con los objetivos del sprint y completar las primeras versiones de las aplicaciones.

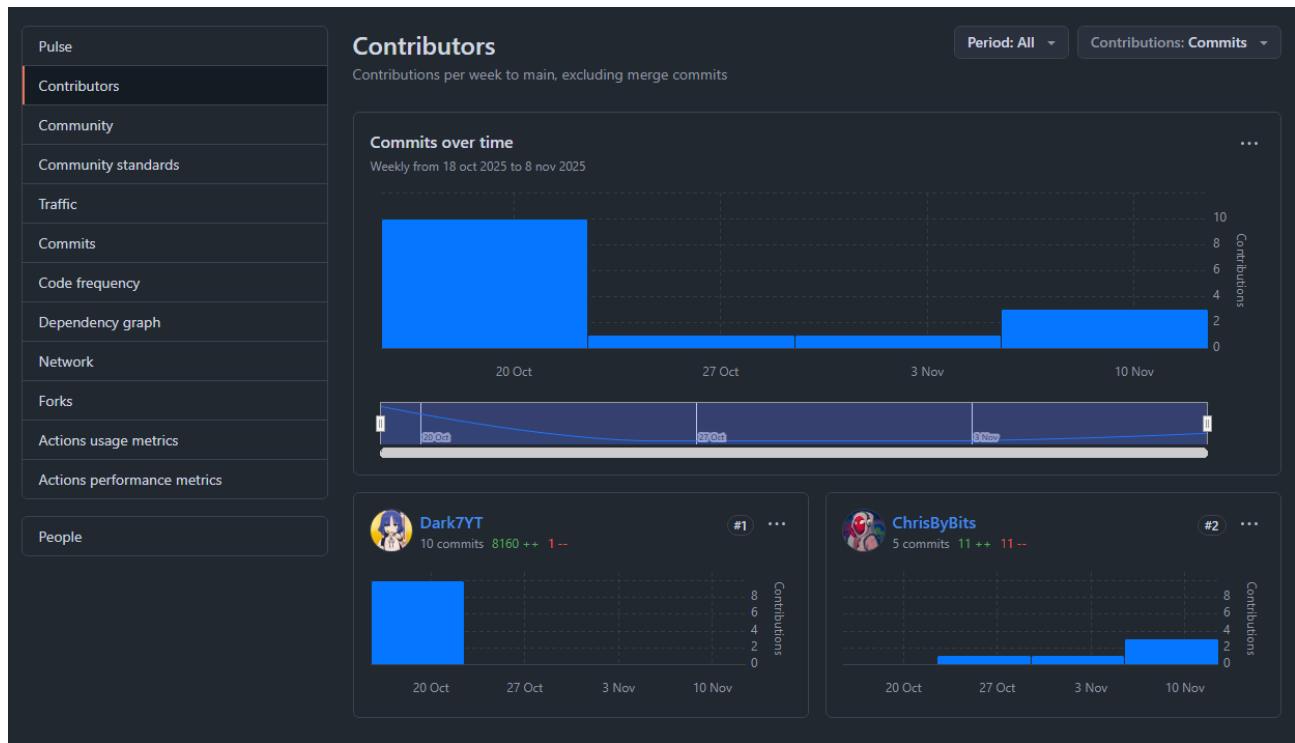
• Landing page



- **Web Application**



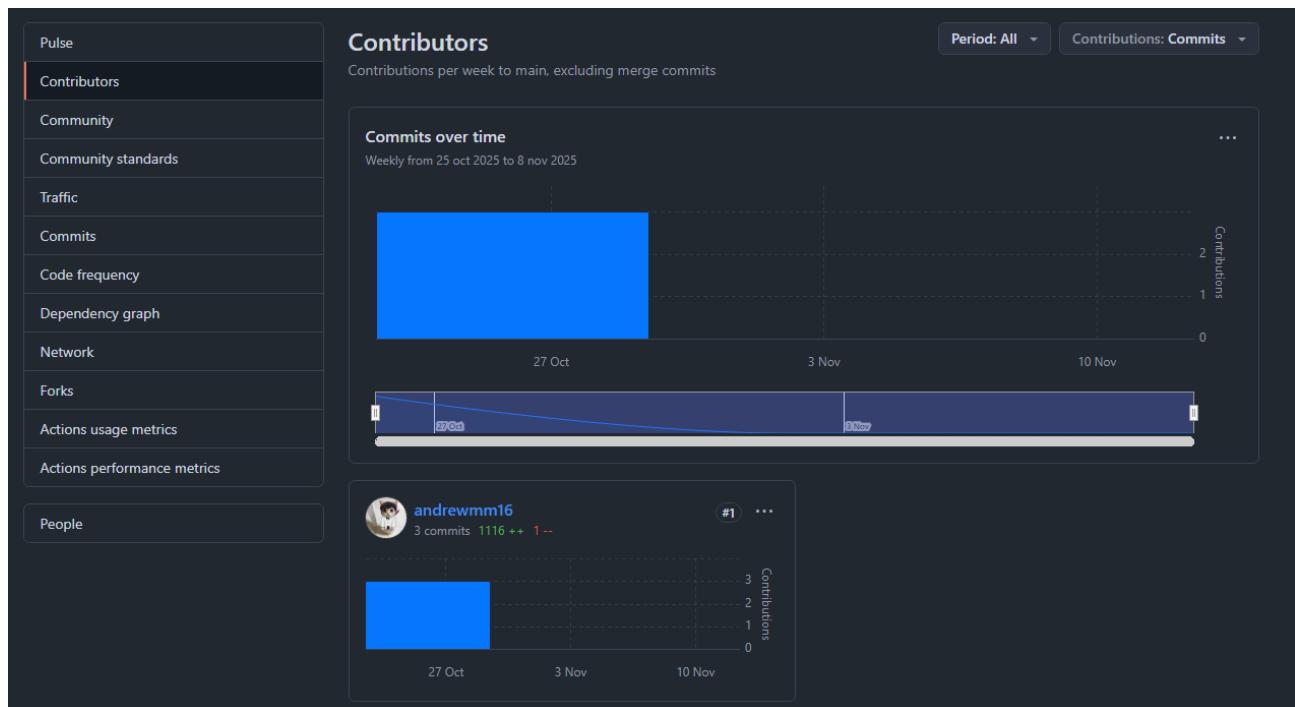
- **Web Service**



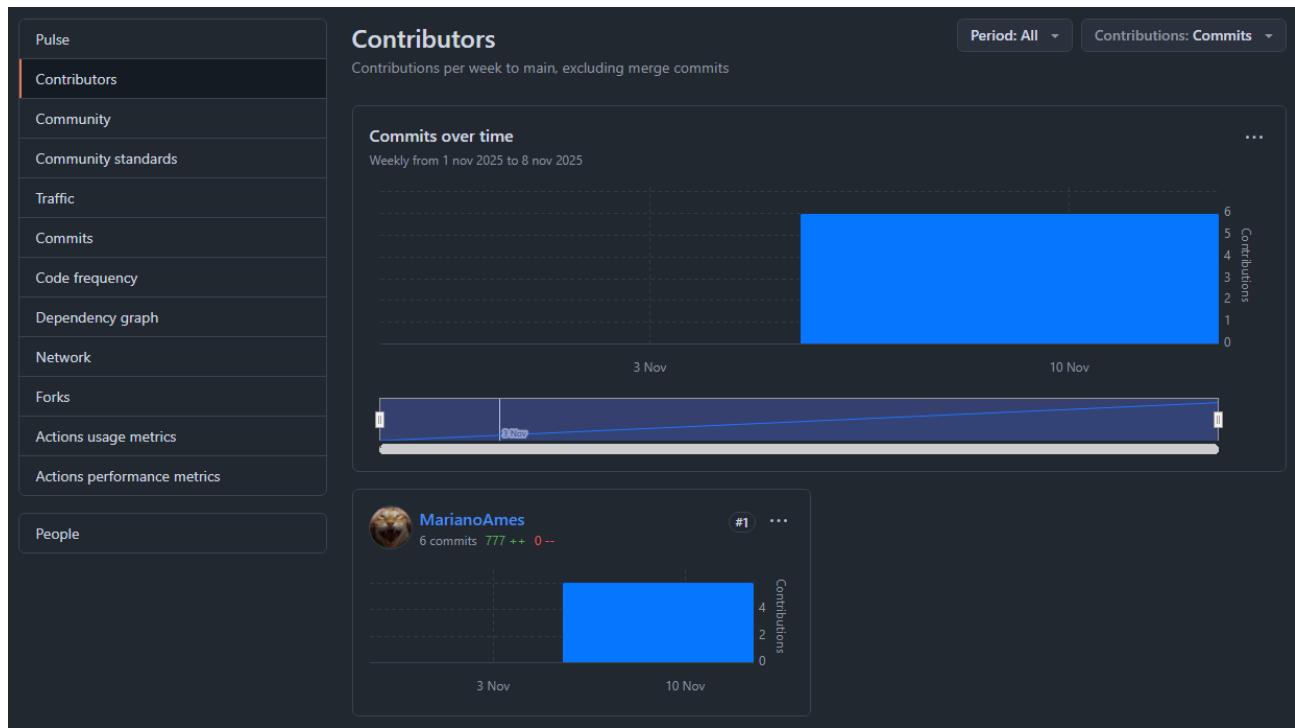
- **Mobile Application**

Insights([URL](#))

- **Edge Node**



- **Embedded App**



7.3. Validation Interviews

7.3.1. Diseño de Entrevistas

Segmento 1: Arrendadores de espacios para eventos

Objetivo: Validar la utilidad, facilidad de uso y valor percibido de la plataforma para propietarios de espacios que ofrecen alquiler.

1. Contexto del negocio
 - a. ¿Qué tipo de espacios ofreces para alquiler y con qué frecuencia los alquilas?
 - b. ¿Qué herramientas utilizas actualmente para gestionar reservas y clientes?
 - c. ¿Has tenido problemas relacionados a la gestión, pagos o comportamientos de los arrendatarios? ¿Cuáles?
2. Interacción con la Landing Page
 - a. ¿La información presentada en la landing page transmite claramente los beneficios de usar AlquilaFácil?
 - b. ¿Encontraste fácil el proceso para registrar o publicar tu espacio? ¿Qué parte te resultó más confusa?
 - c. ¿Qué aspecto visual o funcional mejorarías en la landing page?
3. Experiencia con la Aplicación Web
 - a. ¿Cómo calificarías la facilidad para gestionar tus espacios desde la aplicación (escala 1 a 5)?
 - b. ¿Pudiste configurar las reglas del espacio y monitorear las reservas sin dificultad?
 - c. ¿Qué funcionalidades consideras imprescindibles que aún no están disponibles?
4. Percepción general
 - a. ¿Qué te motivaría a usar esta plataforma en lugar de tus métodos actuales?
 - b. ¿Cómo percibes el valor de herramientas como alertas en tiempo real o reportes automáticos?
 - c. ¿Pagarías por planes premium con mayor visibilidad o alguna herramienta adicional?

Segmento 2: Arrendatarios sociales o corporativos

Objetivo: Validar la experiencia de búsqueda, reserva y confianza en la plataforma para personas o empresas que alquilan espacios para eventos.

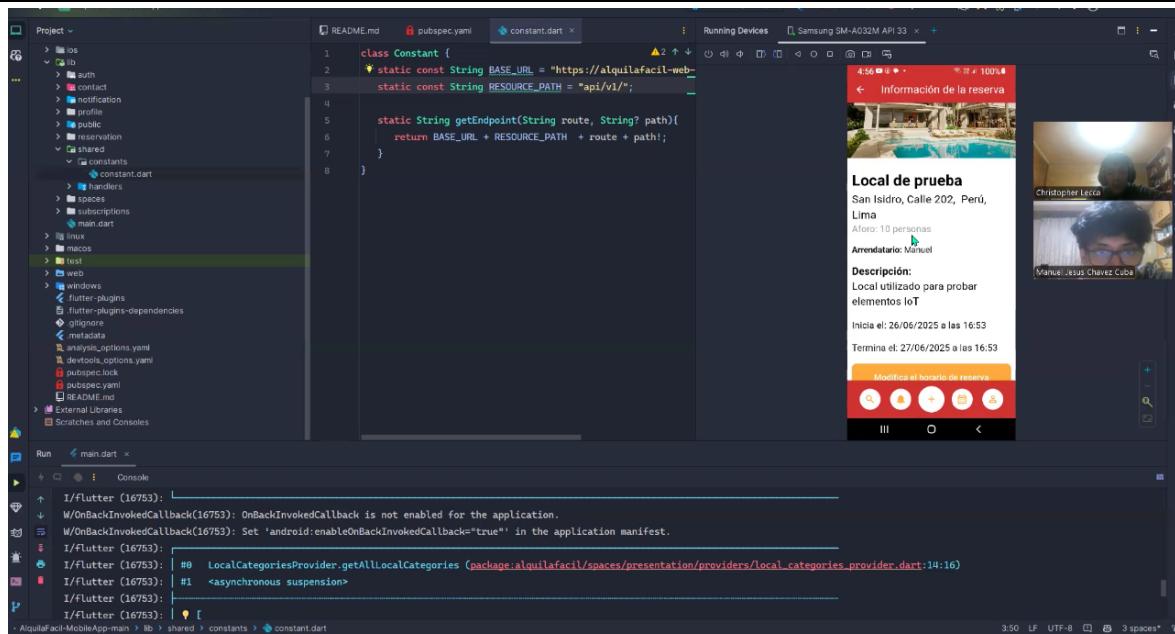
1. Contexto personal o profesional
 - a. ¿Con qué frecuencia necesitas alquilar espacios para eventos? ¿Para qué tipo de actividades?
 - b. ¿Qué plataformas o métodos usas actualmente para buscar y reservar espacios?
 - c. ¿Qué problemas has tenido en experiencias pasadas al alquilar espacios?
2. Interacción con la Landing Page
 - a. ¿La información de las secciones te animó a probar la aplicación?
 - b. ¿El flujo de navegación entre las secciones fue de tu agrado? ¿Qué mejorarías?
 - c. ¿Confías en la plataforma tras navegar por la landing page? ¿Qué faltó?
3. Experiencia con la Aplicación Web/Móvil
 - a. ¿Pudiste realizar todo el flujo de reserva sin dificultades? ¿Dónde te atascaste?
 - b. ¿Qué tan útil te parecieron los elementos como reseñas, calificaciones y verificación de espacios?
 - c. ¿Las herramientas presentadas para el monitoreo y seguridad durante tu reserva te agradaron?
4. Percepción general
 - a. ¿Qué valor le ves a AlquilaFácil frente a otras plataformas conocidas como Airbnb o Booking?
 - b. ¿Qué te motivaría a recomendar esta app a otras personas o empresas?
 - c. ¿El método de pago de la plataforma te resulta fácil y sencillo?
 - d. ¿Qué otra recomendación podrías brindarnos para mejorar el producto?

7.3.2. Registro de Entrevistas

Se llevarán a cabo tres entrevistas por segmento, sumando un total de seis. El registro será en formato mp4, y cada entrevista será independiente, ya que las preguntas y respuestas varían según los entrevistados de cada grupo.

Segmento 1: Arrendadores de espacios para eventos

Entrevista #1	
Nombre	Manuel Jesús
Apellidos	Chávez Cuba
Edad	21
Distrito	San Martín de Porres
Aplicaciones usadas	Facebook Marketplace
Motivación	Ofrecer experiencias únicas y memorables para sus clientes.
Frustración	Limitaciones presupuestarias que puedan afectar la calidad de los eventos.
Tecnologías	iOS, Android, Windows.
Browsers	Google Chrome, Brave
Entrevistador	Christopher Lecca



Enlace	Microsoft Stream
Duración	00:00 min - 09:52 min
Resumen	Manuel nos comentó que se encuentra aún se encuentra dentro del mercado de alquiler de espacios para organizar eventos. Durante la entrevista, se mostró satisfecho con el producto resultado AlquilaFácil, resaltó que la versión Web como Mobile son excelentes, confirmado que recomendaría su uso a familiares y personas cercanos. Como única recomendación, nos mencionó la idea de cambiar el orden en el que se muestran los locales a los usuarios, puesto que es complicado ir hasta la parte inferior del catálogo para encontrar la más reciente.

Entrevista #2	
Nombre	Angello Rodolfo
Apellidos	Sosa Colca
Edad	20
Distrito	San Borja
Aplicaciones usadas	Facebook Marketplace
Motivación	Maximizar el uso de sus espacios y generar ingresos adicionales mediante alquileres frecuentes.

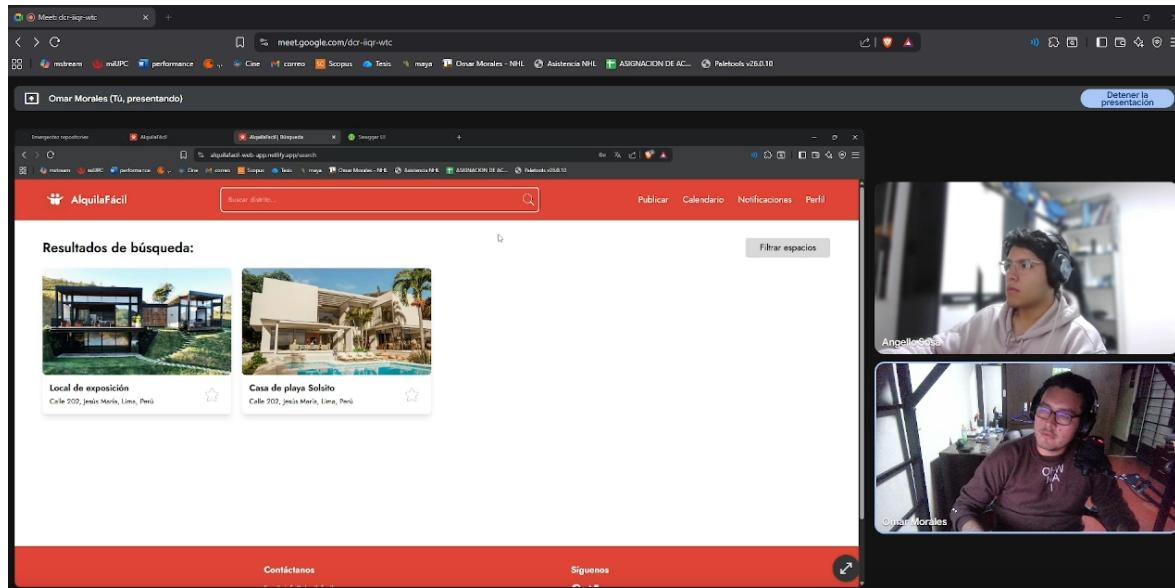
Entrevista #1

Frustración La falta de control y visibilidad sobre el comportamiento de los inquilinos y posibles daños en los espacios alquilados.

Tecnologías Android, Windows

Browsers Brave

Entrevistador [Nombre del entrevistador]



Evidencia

Enlace [Entrevista Video](#)

Duración 00:00 min - 12:45 min

Resumen Angello Sosa, arrendador de 20 años de San Borja, revisó la plataforma AlquilaFácil y mostró interés en las funcionalidades que ofrece para propietarios de espacios. Durante la entrevista, destacó la importancia del control y monitoreo de los espacios alquilados para evitar daños y generar confianza. Como recomendación principal, sugirió agregar una opción de modo oscuro (botón dark mode) a la landing page para mejorar la experiencia de usuario, especialmente durante el uso nocturno de la plataforma.

--- ---

Entrevista #3

Nombre Joaquín

Apellidos Pedraza Maldonado

Edad 21

Distrito San Miguel

Aplicaciones usadas Facebook Marketplace

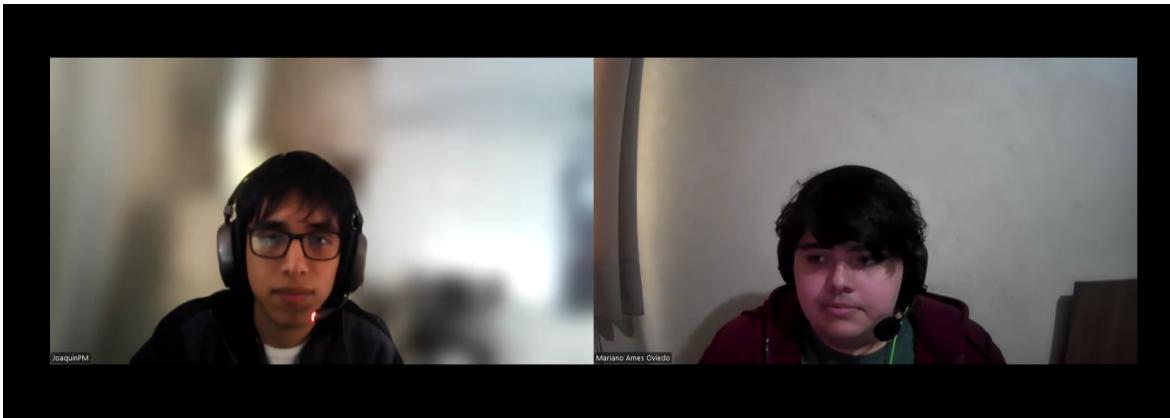
Motivación Facilitar la gestión de reservas en sus espacios y generar ingresos con mas alquileres

Frustración La falta de herramientas que le permitan una gestión sencilla de las reservas de sus espacios

Tecnologías Android, Windows

Browsers Google Chrome, Opera

Entrevistador Mariano Ames

Entrevista #1**Evidencia****Enlace** [Microsoft Stream](#)**Duración** 00:00 min - 10:35 min

Resumen Joaquin Pedraza, arrendador de 21 años de San Miguel, revisó la plataforma de AlquilaFacil y comentó que sentía interés sobre las funcionalidades que le permiten manejar de mejor manera las reservas, destacó el calendario y como recomendación principal sugirió añadir diferentes funcionalidades adicionales a este, como por ejemplo visualizar las reservas completadas o pendientes y poder añadir propios recordatorios.

Segmento 2: Arrendatarios sociales frecuentes de espacios para eventos**Entrevista #1**

Nombre Ricardo Rafael

Apellidos Rivas Carrillo

Edad 20

Distrito Callao

Aplicaciones usadas Facebook Marketplace

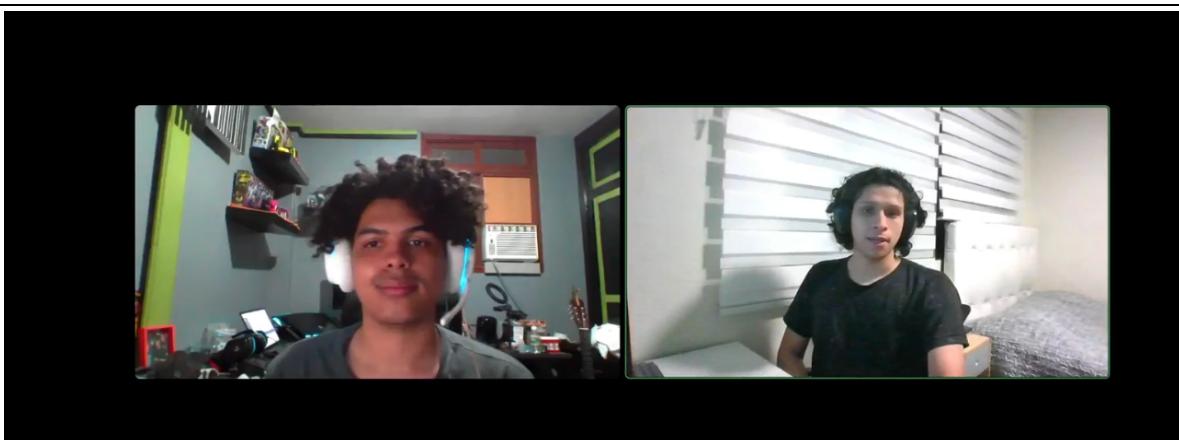
Motivación Realizar practicar en espacio adecuados para su grupo de baile

Frustración Limitaciones para encontrar espacios que cumplan las necesidades de su grupo de baile.

Tecnologías Android, Windows.

Browsers Opera, Edge

Entrevistador Sebastian Cachis

Evidencia**Enlace** [Microsoft Stream](#)**Duración** 00:00 min - 14:56 min

Entrevista #1

Resumen Ricardo Rivas, estudiante de 20 años de Ciencias de la Computación, comentó que tras ver la landing page, la app web y la app móvil de Alquila Fácil, le gustó mucho el resultado, destacando que el diseño es intuitivo, moderno y fácil de usar. Señaló que la plataforma refleja perfectamente lo que necesita para organizar eventos de baile o actuación, ya que permite comparar precios, ver fotos, leer opiniones y calificaciones, además de ofrecer un proceso de reserva rápido, seguro y con comunicación directa con el arrendador. También valoró el sistema de reputación entre usuarios, las notificaciones ante imprevistos y las medidas de seguridad como cámaras y control de acceso, afirmando que Alquila Fácil transmite confianza, transparencia y tecnología al facilitar el acceso a espacios para eventos.

Entrevista #2

Nombre	Walther
Apellidos	Cachay Villar
Edad	27
Distrito	Surco
Aplicaciones usadas	Facebook Marketplace
Motivación	Alquilar espacios casi semanalmente para shows y eventos (cumpleaños, quince años, fiestas de promoción, bodas, despedidas de solteros)
Frustración	El espacio propuesto a veces no cumple con lo que se promociona en el flyer
Tecnologías	Android, Windows, Contactos preestablecidos
Browsers	Opera, Edge
Entrevistador	Amner Llamo



Enlace [Microsoft Stream](#)

Duración 00:00 min - 14:22 min

Resumen Walther Kevin, arrendatario de 27 años, necesita alquilar semanalmente. Encontró la Landing Page de Alquila Fácil interesante y con un diseño "atractivo", destacando las secciones de Espacios, Publicar (pensando en futuro), y el Monitoreo con sensores. Sugirió añadir un enlace a Instagram y valoró la opción de idiomas. En la Aplicación Web, el flujo le pareció intuitivo y atractivo, y consideró las Reseñas como una función muy importante para generar confianza. Le encantó la idea de los sensores (humo, movimiento) e incluso sugirió uno para medir el aforo. Comparó Alquila Fácil positivamente con Booking, enfocada en la reserva de espacios para eventos. Su recomendación principal fue añadir una opción de mapas con enlace directo a Google Maps.

Entrevista #3

Nombre	Stefano Alessandro
Apellidos	Valenzuela Vallejos
Edad	23 años
Distrito	San Isidro
Aplicaciones usadas	Instagram, TikTok.

Entrevista #1

Motivación	Ofrecer experiencias únicas y satisfactorias a sus clientes.
Frustración	Falta de herramientas de gestión durante el desarrollo de sus eventos.
Tecnologías	iOS, macOS.
Browsers	Safari
Entrevistador	Christopher Lecca

The screenshot shows a web browser displaying the AlquilaFácil application at the URL <https://alquilafacil-app-iot.netlify.app/local/1>. The page features a large image of a modern, multi-story building with a pool and outdoor seating. Below the image, the text "Local de prueba" is displayed. To the right of the image, there is a sidebar with the following information:

- Ver comentarios >
- Reportar espacio >
- Fecha y hora de inicio: 28/06/2025
- Fecha y hora de fin: 29/06/2025
- Cuenta del propietario Alessandro Valenzuela
- Número de cuenta: 2451789632105478123
- Número de cuenta interbancaria: 01824517896321054789
- Cantidad a depositar: S/.240

Below the sidebar is a red camera icon with the text "Adjuntar imagen del voucher". At the bottom right of the page is a large orange button labeled "Reservar".

Enlace	Microsoft Stream
Duración	00:00 min - 10:12 min
Resumen	Nuestro entrevistado, Stefano, nos comentó que el flujo de navegación de la Landing Page fue satisfactoria, al igual que con la Web Application y Mobile Application. En general, ambas versiones de la aplicación le agradaron, resultando en que su única recomendación fuese el expandir el catálogo de espacios disponibles, lo cual se logrará una vez la versión final del producto se encuentre disponible.

7.3.3. Evaluaciones según heurísticas

UX Heuristics & Principles Evaluation
Usability – Inclusive Design – Information Architecture

CARRERA: Ingeniería de Software

CURSO: Arquitecturas de Software Emergentes

SECCIÓN: 14653

PROFESORES : Todos

APP A EVALUAR: AlquilaFácil

TAREAS A EVALUAR

El alcance de esta evaluación incluye la revisión de la usabilidad de las siguientes tareas:

General:

- Registro de usuario
- Inicio de sesión de usuario

Segmento de arrendadores:

- Publicación de un espacio
- Vista de espacio publicado
- Gestión de espacios publicados
- Calendario de reserva
- Monitoreo del espacio

Segmento de arrendatarios:

- Navegación sobre espacios publicados

- Vista de un espacio específico
- Reserva de un espacio
- Calendario de reserva
- Adquisición de suscripción

ESCALA DE SEVERIDAD

Nivel	Descripción
1	Problema superficial: puede ser fácilmente superado por el usuario o ocurre con muy poca frecuencia. No necesita ser arreglado a no ser que exista disponibilidad de tiempo.
2	Problema menor: ocurre con mayor frecuencia o es un poco más difícil de superar para el usuario. Prioridad baja para su corrección.
3	Problema mayor: ocurre frecuentemente o los usuarios no son capaces de resolverlo. Prioridad alta para su corrección.
4	Problema muy grave: error de gran impacto que impide al usuario continuar. Debe ser corregido antes del lanzamiento.

TABLA RESUMEN

#	Problema	Escala de severidad	Heurística/Principio violado
1	No se puede adjuntar más de una imagen por local	3	Usability: Flexibilidad y eficiencia de uso
2	No hay opción de tema claro/oscuro	2	Usability: Preferencias del usuario
3	No hay forma de contactar al soporte desde la app	3	Usability: Ayuda y documentación
4	Los espacios más recientes se encuentran últimos en la lista	2	Usability: Flexibilidad y eficiencia de uso

DESCRIPCIÓN DE PROBLEMAS

PROBLEMA #1: No se puede adjuntar más de una imagen por local

- **Severidad:** 3
- **Heurística violada:** Usability – Flexibilidad y eficiencia de uso
- **Problema:**
El sistema permite subir únicamente una imagen por local, limitando la información visual que se puede proporcionar al usuario interesado. Esto afecta negativamente la experiencia de exploración de los espacios.

Detalles del local:



Local de prueba

San Isidro, Calle 202, Perú, Lima

Aforo: 10 personas

Propietario: Admin

Descripción:

Local utilizado para probar elementos IoT

- **Recomendación:**

Permitir adjuntar múltiples imágenes por local, con una interfaz que permita visualizarlas en carrusel o miniaturas.

PROBLEMA #2: No hay opción de tema claro/oscuro

- **Severidad:** 2
- **Heurística violada:** Usability – Preferencias del usuario
- **Problema:**
La aplicación no cuenta con opción de personalización visual, como el modo claro y oscuro, lo cual limita la comodidad visual según preferencias o condiciones de iluminación.

Panel de control

The screenshot shows a profile settings screen with the following fields:

- Nombre: Omar (with an 'Editar' button)
- Apellido paterno: Morales (with an 'Editar' button)
- Apellido materno: Montalvo (with an 'Editar' button)
- Teléfono: 987456123 (with an 'Editar' button)
- Número de documento: 99988877 (with an 'Editar' button)
- Fecha de nacimiento: 2025-01-20 (with an 'Editar' button)

On the left, there is a sidebar with navigation links: Modificar perfil, Mi suscripción, Espacios publicados, Espacios favoritos, Soporte, and Reportes realizados, each preceded by a right-pointing arrow.

- **Recomendación:**

Implementar un selector de tema para que el usuario elija entre modo claro, oscuro o automático según el sistema operativo.

PROBLEMA #3: No hay forma de contactar al soporte desde la app

- **Severidad:** 3
- **Heurística violada:** Usability – Ayuda y documentación
- **Problema:**
No existe una sección visible para que los usuarios puedan contactar al equipo de soporte o enviar reportes de errores desde la aplicación.

Panel de control

The screenshot shows a 'Soporte - Preguntas frecuentes' (Support - Frequently Asked Questions) section with the following questions:

- ¿Cómo reservo un espacio?
- ¿Qué sucede después de subir el comprobante de pago?
- ¿AlquilaFácil gestiona el dinero o reembolsa pagos?
- ¿Cómoactivo una suscripción premium?
- ¿Qué beneficios tengo como usuario premium?
- ¿Qué hago si sospecho de un fraude?
- ¿Puedo posponer una reserva?

On the left, there is a sidebar with navigation links: Modificar perfil, Mi suscripción, Espacios publicados, Espacios favoritos, Soporte, Reportes realizados, and Cerrar sesión, each preceded by a right-pointing arrow.

- **Recomendación:**

Añadir una sección de contacto o botón flotante que permita enviar correos o mensajes al equipo de soporte de forma directa.

PROBLEMA #4: Textos pequeños en dispositivos móviles

- **Severidad:** 2
- **Heurística violada:** Usability – Flexibilidad y eficiencia de uso
- **Problema:**
El sistema muestra los espacios publicados de forma que los más recientes se encuentran en la parte inferior y los más antiguos en la superior.

Resultados de búsqueda:



Local de prueba

Local utilizado para probar elementos IoT



- **Recomendación:**

Invertir el orden en el que se muestran los espacios recién publicados.

7.4. Video About-the-Product

Conclusiones y recomendaciones

- AlquilaFácil se consolida como una solución innovadora en el mercado de alquiler de espacios para eventos, ofreciendo una plataforma que simplifica y optimiza el proceso de búsqueda, reserva y gestión de lugares. Su enfoque integral no solo facilita las operaciones tradicionales, sino que también transforma la experiencia de arrendadores y arrendatarios, garantizando una interacción fluida, rápida y eficiente en cada etapa del proceso.
- La propuesta de valor de AlquilaFácil se centra en la eficiencia y en la experiencia del usuario, incorporando herramientas avanzadas de gestión y seguimiento en tiempo real. Estas características distintivas permiten que tanto arrendadores como arrendatarios gestionen sus eventos de manera más organizada y profesional, reduciendo tiempos de respuesta y mejorando significativamente la satisfacción general en la utilización de la plataforma.
- En cuanto a su estrategia de monetización, AlquilaFácil opta por un modelo inteligente y equilibrado: los arrendadores pueden acceder a planes de suscripción premium que les brindan ventajas competitivas, como mayor visibilidad y acceso a analíticas avanzadas, mientras que los arrendatarios disfrutan de un acceso gratuito a la plataforma. Esta estructura fomenta el crecimiento de la oferta de espacios disponibles y asegura un flujo constante de usuarios, fortaleciendo la comunidad dentro del ecosistema de AlquilaFácil.
- El uso de sensores IoT para el monitoreo de espacios y la gestión de reservas en tiempo real representa un avance significativo en la forma en que se gestionan los eventos. Esta tecnología no solo mejora la eficiencia operativa, sino que también proporciona datos valiosos para la toma de decisiones estratégicas, permitiendo a los arrendadores optimizar sus espacios y a los arrendatarios disfrutar de una experiencia más personalizada y adaptada a sus necesidades.

Bibliografía

- A Clean Approach to Flutter Development through the Flutter Clean Architecture Package. (2023). *IEEE Conference Publication*. IEEE Xplore. Recuperado de <https://ieeexplore.ieee.org/document/9071367>
- Airbnb. (2024). *Airbnb – Sitio Oficial*. Recuperado de <https://www.airbnb.com.pe/>
- Booking.com. (2024). *Booking.com: La mayor selección de hoteles, casas y alquileres*. Recuperado de <https://www.booking.com/index.es.html>
- Cvent. (2023). *Cvent's Fourth Annual Benchmark Report: The Future of Event Technology in a Post-Pandemic World*. Cvent. Recuperado de <https://marketing.conference-news.co.uk/en-gb/cvent-event-tech-report-2024>
- Hybrid Development in Flutter and its Widgets. (2022). *IEEE Conference Publication*. IEEE Xplore. Recuperado de <https://ieeexplore.ieee.org/document/9995973>
- Forrester Research, Inc. (2024). *The Global State of B2B Events, 2024: Marketers Continue to Ride a Wave of Transformation*. Forrester. Recuperado de <https://www.forrester.com/report/the-global-state-of-b2b-events-2024-marketers-continue-to-ride-a-wave-of-transformation/RES181128>
- Inmuebles de Lima. (2024). *Encuentra el inmueble de tus sueños en Lima*. Recuperado de <https://inmueblesdelima.com.pe/>

- Jasín, N. (2023, 11 de julio). *Acceso a internet en Perú: Los aspectos clave para reducir la brecha entre el mundo laboral y el educativo*. Business Empresarial. Recuperado de <https://www.businessempresarial.com.pe/acceso-a-internet-en-peru-los-aspectos-clave-para-reducir-la-brecha-entre-el-mundo-laboral-y-el-educativo/>
 - Mañas, L. (2024). *Agencia de eventos Olympia= Olympia events agency*. Recuperado de <https://repositori.uji.es/xmlui/handle/10234/208284>
 - Vrbo. (2024). *Sitio oficial de Vrbo®*. Recuperado de <https://www.vrbo.com/es-mx/>
-

Anexos

Repositorio general

- **Repositorio del proyecto (documentación completa):**
[GitHub – AlquilaFácil Report](#)
- **Repositorio de Landing Page:**
[GitHub – AlquilaFácil Landing Page](#)

Despliegues

- **Landing Page:**
[AlquilaFácil Landing Page](#)

Lean UX Canvas

- **Lean UX Canvas:**
[Ver en GitHub](#)

Mapas de Empatía

- **Empathy Map - Arrendatario:**
[Ver en GitHub](#)
- **Empathy Map - Arrendador:**
[Ver en GitHub](#)

Escenarios

- **As-Is Scenario Mapping - Arrendatarios:**
[Ver en Imgur](#)
- **As-Is Scenario Mapping - Arrendadores:**
[Ver en Imgur](#)
- **To-Be Scenario:**
[Ver en GitHub](#)
[Ver en GitHub](#)

Mapa de Impacto

- **Impact Map - Arrendador:**
[Ver en GitHub](#)
- **Impact Map - Arrendatario:**
[Ver en GitHub](#)

Diagramas de Arquitectura de Software

- **Diagramas (Lucidchart):**
[Abrir en Lucidchart](#)
- **Context Diagram:**
[Ver en GitHub](#)
- **Container Diagram:**
[Ver en GitHub](#)
- **Deployment Diagram:**
[Ver en GitHub](#)
- **Landscape Diagram:**
[Ver en GitHub](#)

Vídeos de Exposición (Anexo 1)

Contenido	URL del Vídeo
Vídeo de Exposición TB1	Link del Vídeo: https://upcedupe-my.sharepoint.com/:v/g/personal/u202211399_upc_edu_pe/ESgo7U5_auJlkKMfm2eQHo8w1pkgzwIq5RJI3YjKjch2w
Vídeo de Exposición TP1	Link del Vídeo: https://upcedupe-my.sharepoint.com/:v/g/personal/u202212383_upc_edu_pe/EU2iWmGhVO1Cv7Hil0N63tlBi6PW_YVQ7Vjz3A-WzD30f