



**Universidad Peruana de Ciencias Aplicadas  
Ingeniería de Software  
Ciclo 2025-10**

Course Project

**Profesor:** Marco Antonio Leon Baca  
**Curso:** Desarrollo de Soluciones IoT

FINAL REPORT

**Startup:** AlquilaEZ  
**Propuesta:** AlquilaFacil

**Team Members**

Ames Oviedo Mariano Jose  
Avalos Santos Anthony Piero  
Lecca Choccare Christopher Bryan  
Sosa Colca Angello Rodolfo  
Carlos Oswaldo Casimiro Fernandez

**Mayo 2025**

---

## Registro de Versiones

Versión	Fecha	Autor	Descripción de modificación
---------	-------	-------	-----------------------------

<b>Versión</b>	<b>Fecha</b>	<b>Autor</b>	<b>Descripción de modificación</b>
1era	22/04/2025	<ul style="list-style-type: none"> <li>- Ames</li> <li>Oviedo</li> <li>Mariano</li> <li>Jose</li> <li>- Avalos</li> <li>Santos</li> <li>Anthony</li> <li>Piero</li> <li>- Lecca</li> <li>Choccare</li> <li>Christopher</li> <li>Bryan</li> <li>- Sosa</li> <li>Colca</li> <li>Angello</li> <li>Rodolfo</li> <li>- Carlos</li> <li>Oswaldo</li> <li>Casimiro</li> <li>Fernandez</li> </ul>	<p>Se creo la ramificacion del repositorio. Se concluyo con los capitulo 1, 2, 3 y 4, realizando la presentacion respectiva de la documentacion. Tambien se realizo el video donde cada miembro del equipo expone ciertas partes del trabajo</p>
2da	10/05/2025	<ul style="list-style-type: none"> <li>- Ames</li> <li>Oviedo</li> <li>Mariano</li> <li>Jose</li> <li>- Avalos</li> <li>Santos</li> <li>Anthony</li> <li>Piero</li> <li>- Lecca</li> <li>Choccare</li> <li>Christopher</li> <li>Bryan</li> <li>- Sosa</li> <li>Colca</li> <li>Angello</li> <li>Rodolfo</li> <li>- Carlos</li> <li>Oswaldo</li> <li>Casimiro</li> <li>Fernandez</li> </ul>	<p>Se creo los repositorios para la Web Application frontend y Lading Page, asi como tambien el desarrollo de los capitulo 5 y 6 (Sprint Backlog 1). Tambien se continuo con el desarrollo de los productos de software mencionados y la presentacion de los mismos.</p>

## Project Report Collaboration Insights

URL del repositorio: <https://github.com/AlquilaEZ/alquilafacil-document-report>

## TB1

El equipo participó de forma activa en el repositorio de GitHub, registrando un total de 74 commits durante el desarrollo del primer entregable del informe.



El trabajo con ramas paralelas permitió que los integrantes del equipo avanzaran en distintas secciones del informe de forma simultánea, integrando sus cambios de manera ágil y minimizando conflictos. Esta estrategia colaborativa aseguró una participación equilibrada de todos los miembros en la construcción del entregable.

## Network graph

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



## TP

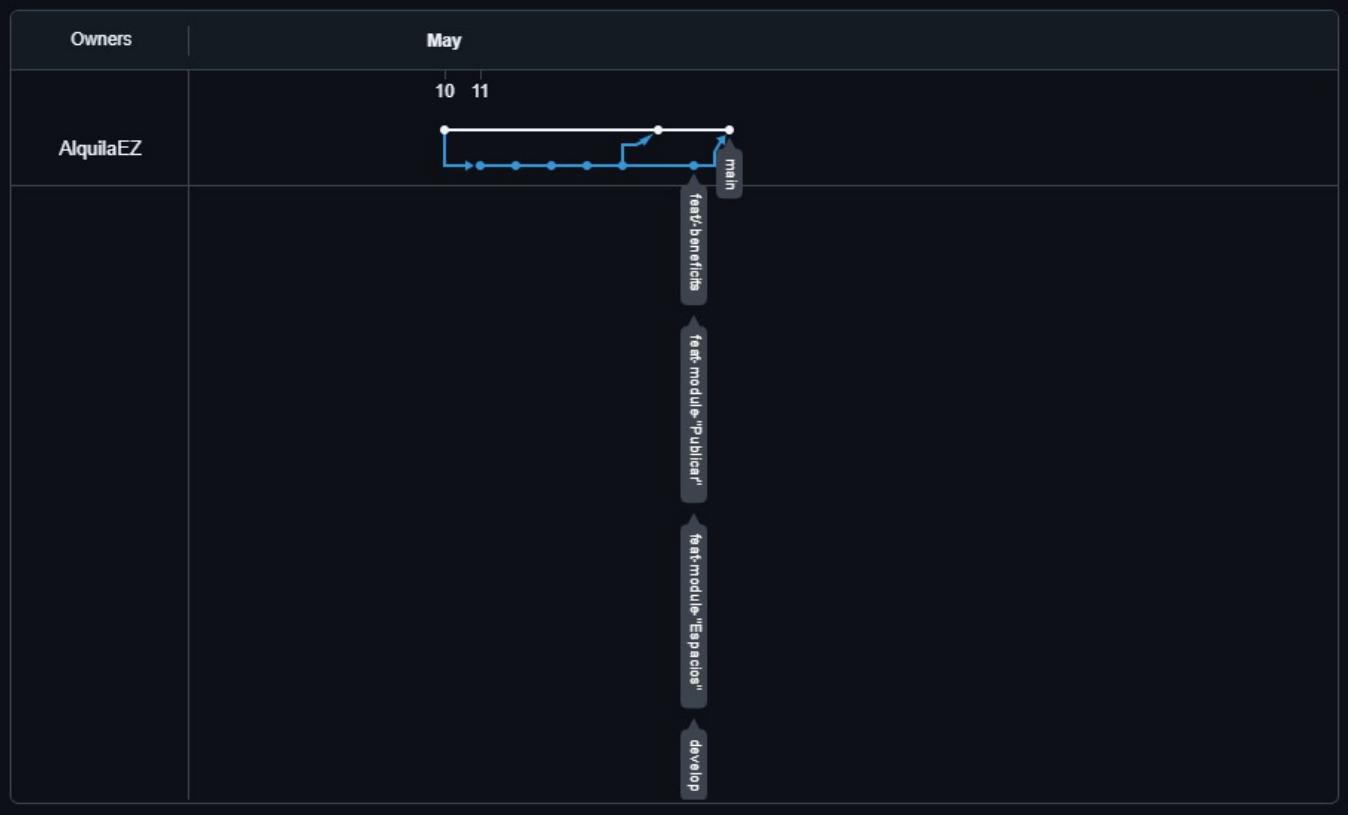
El uso de ramas paralelas permitió que los integrantes del equipo desarrollaran diferentes secciones de la Landing Page de forma simultánea, lo que facilitó la integración continua de avances sin generar conflictos significativos. Esta estrategia colaborativa promovió una distribución equitativa del trabajo y garantizó una participación activa de todos los miembros en la construcción del producto entregable del sprint.

- **Landing Page**



## Network graph

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



- \*\* Frontend Web Applications\*\*

## Network graph

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



## Contenido

### Tabla de contenidos

#### Tabla de contenidos

[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](#)

## Tabla de contenidos

---

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. User Journey Mapping
2.3.4. Empathy Mapping
2.3.5. 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: Solution Software Design
4.1. Strategic-Level Domain-Driven Design
4.1.1. Event Storming
4.1.1.1 Candidate Context Discovery
4.1.1.2 Domain Message Flows Modeling
4.1.1.3 Bounded Context Canvases
4.1.2. Context Mapping
4.1.3. Software Architecture
4.1.3.1. Software Architecture System Landscape Diagram
4.1.3.2. Software Architecture Context Level Diagrams
4.1.3.3. Software Architecture Container Level Diagrams
4.1.3.4. Software Architecture Deployment Diagrams
4.2. Tactical-Level Domain-Driven Design

## Tabla de contenidos

---

4.2.1. Bounded Context:
4.2.1.1. Domain Layer
4.2.1.2. Interface Layer
4.2.1.3. Application Layer
4.2.1.4. Infrastructure Layer
4.2.1.5. Bounded Context Software Architecture Component Level Diagrams
4.2.1.6. Bounded Context Software Architecture Code Level Diagrams
4.2.1.6.1. Bounded Context Domain Layer Class Diagrams
4.2.1.6.2. Bounded Context Database Design Diagram
Capítulo V: Solution UI/UX Design
5.1. Style Guidelines
5.1.1. General Style Guidelines
5.1.2. Web, Mobile and IoT Style Guidelines
5.2. Information Architecture
5.2.1. Organization Systems
5.2.2. Labeling Systems
5.2.3. SEO Tags and Meta Tags
5.2.4. Searching Systems
5.2.5. Navigation Systems
5.3. Landing Page UI Design
5.3.1. Landing Page Wireframe
5.3.2. Landing Page Mock-up
5.4. Applications UX/UI Design
5.4.1. Applications Wireframes
5.4.2. Applications Mock-ups
5.4.3. Applications User Flow Diagrams
5.5. Applications Prototyping
Capítulo VI: Product Implementation, Validation & Deployment
6.1. Software Configuration Management
6.1.1. Software Development Environment Configuration

## Tabla de contenidos

---

6.1.2. Source Code Management
6.1.3. Source Code Style Guide & Conventions
6.1.4. Software Deployment Configuration
6.2. Landing Page, Services & Applications Implementation
6.2.1. Sprint 1
6.2.1.1. Sprint Planning 1
6.2.1.2. Aspect Leaders and Collaborators
6.2.1.3. Sprint Backlog 1
6.2.1.4. Development Evidence for Sprint Review
6.2.1.5. Testing Suite Evidence for Sprint Review
6.2.1.6. Execution Evidence for Sprint Review
6.2.1.7. Services Documentation Evidence for Sprint Review
6.2.1.8. Software Deployment Evidence for Sprint Review
6.2.1.9. Team Collaboration Insights during Sprint

## Student Outcome

---

**ABET – EAC – Student Outcome 5** Criterio: La capacidad de funcionar efectivamente en un equipo cuyos miembros juntos proporcionan liderazgo, crean un entorno de colaboración e inclusivo, establecen objetivos, planifican tareas y cumplen objetivos

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 5.

Criterio específico	Acciones realizadas	Conclusiones
Trabaja en equipo para proporcionar liderazgo en forma conjunta.	<p><b>Christopher Bryan Lecca Choccare</b></p> <p><b>(TB1):</b> Me encargué de liderar el diseño estratégico, guiando al equipo en la identificación de bounded contexts y en la elaboración del Event Storming y Context Mapping. Coordiné la conexión lógica entre eventos y políticas, y aseguré la consistencia general en los flujos del sistema.</p> <p><b>(TP1):</b> Desarrollé la estructura</p>	<p><b>(TB1):</b> El equipo demostró una colaboración efectiva al asumir roles estratégicos que permitieron el avance fluido del proyecto. Cada miembro aportó liderazgo desde sus fortalezas, logrando construir una solución bien estructurada a través de trabajo conjunto y comunicación constante.</p> <p><b>(TP1):</b> El equipo logró organizarse estratégicamente durante el desarrollo del producto, asignando roles complementarios en el</p>

<b>Criterio específico</b>	<b>Acciones realizadas</b>	<b>Conclusiones</b>
	<p>completa de la Landing Page utilizando HTML y Tailwind CSS, asegurando un diseño limpio y adaptable. Participé en el desarrollo del Capítulo V, definiendo los lineamientos de estilo y wireframes. Además, fui responsable de redactar gran parte del Capítulo VI, incluyendo las evidencias de implementación y despliegue del producto.</p> <p><b>Angello Rodolfo Sosa Colca</b>  <b>(TB1):</b> Lideré el diseño táctico del contexto "Locals", definiendo sus capas y componentes. Me aseguré de que la estructura entre entidades, comandos, servicios y repositorios fuera coherente y alineada con los principios de DDD.  <b>(TP1):</b> Contribuí activamente en la codificación del frontend web desarrollado con Vue.js, implementando los componentes de interfaz para la gestión de espacios. También colaboré en la sección de prototipos del Capítulo V y en los aspectos técnicos del Capítulo VI relacionados a la arquitectura de software.</p> <p><b>Anthony Piero Avalos Santos</b>  <b>(TB1):</b> Participé activamente en la fase de descubrimiento del usuario. Me enfoqué en aplicar técnicas como Empathy Mapping, User Personas y User Journey para captar de manera precisa las necesidades reales y plasmarlas en el sistema.  <b>(TP1):</b> Diseñé la estructura de navegación de la Landing Page y desarrollé varias secciones como "Beneficios" y "Monitoreo",</p>	<p>diseño de interfaces, construcción de componentes técnicos y elaboración del entregable. Cada integrante asumió el liderazgo en secciones específicas del sistema, permitiendo el desarrollo paralelo de la Landing Page, la aplicación web y los capítulos del informe. Esta distribución equilibrada potenció la autonomía individual sin comprometer la coherencia del trabajo conjunto.</p>

<b>Criterio específico</b>	<b>Acciones realizadas</b>	<b>Conclusiones</b>
	<p>asegurando su conexión visual y lógica. También redacté partes del Capítulo V enfocadas en diseño centrado en el usuario y usabilidad de la aplicación web.</p> <p><b>Mariano José Ames Oviedo</b>  <b>(TB1):</b>encargué de elaborar la introducción de la solución, definiendo el perfil de la startup, los desafíos que busca resolver y su aporte a la sostenibilidad urbana. También consolidé los antecedentes para contextualizar el problema.  <b>(TP1):</b> Colaboré en la validación del contenido del Capítulo VI, asegurándome de que se describieran de forma clara los resultados del sprint. Además, participé en la organización y revisión del índice general y los entregables de UI/UX documentados en el Capítulo V.</p> <p><b>Carlos Oswaldo Casimiro Fernández</b>  <b>(TB1):</b>Me responsabilicé por la implementación de la capa de infraestructura del contexto, construyendo los repositorios, asegurando persistencia adecuada y validando integraciones con bases de datos.  <b>(TP1):</b> Aporté en la integración técnica del frontend con los servicios backend simulados, validando los flujos de datos. Además, participé en la documentación técnica del Capítulo VI, especialmente en la sección de despliegue y configuración de servicios.</p>	

<b>Criterio específico</b>	<b>Acciones realizadas</b>	<b>(TB1): Se logró una comunicación escrita efectiva a lo largo del entregable, permitiendo que las ideas fueran comprendidas por distintas audiencias.</b> <b>Conclusiones</b>
Comunica en forma escrita ideas y/o resultados con objetividad a público de diferentes especialidades y niveles jerárquicos.	<p><b>Mariano José Ames Oviedo</b>  <b>(TB1):</b> Me encargué de redactar las secciones introductorias del entregable, comunicando los objetivos y fundamentos del sistema de forma clara para cualquier lector.</p> <p><b>(TP1):</b> Redacté el perfil de la startup, objetivos generales del sistema y antecedentes clave, facilitando la comprensión del lector sobre la necesidad del producto. También revisé la coherencia del Capítulo VI en su conjunto.</p> <p><b>Anthony Piero Avalos Santos</b>  <b>(TB1):</b> Redacté resultados del análisis de usuarios, presentando mapas y perfiles de forma comprensible para distintos perfiles técnicos y no técnicos.</p> <p><b>(TP1):</b> Elaboré textos explicativos dentro de la Landing Page y participé en la redacción del Capítulo V, especialmente en los apartados relacionados a usabilidad y experiencia de usuario.</p> <p><b>Christopher Bryan Lecca</b>  <b>Choccare</b>  <b>(TB1):</b> Elaboré la descripción del diseño estratégico, cuidando que el lenguaje técnico pudiera ser entendido también por miembros no especializados del equipo.</p> <p><b>(TP1):</b> Organicé y redacté los contenidos técnicos del Capítulo VI (implementación, evidencia y despliegue) con lenguaje claro, permitiendo que incluso lectores no técnicos comprendieran las decisiones del equipo.</p>	<p><b>(TB1):</b> Se logró una comunicación escrita efectiva a lo largo del entregable, permitiendo que las ideas fueran comprendidas por distintas audiencias. El uso claro del lenguaje, apoyado por gráficos y ejemplos, facilitó la transmisión de resultados y decisiones de diseño.</p> <p><b>(TP1):</b> A través de un trabajo colaborativo, el equipo elaboró un entregable claro y estructurado, enfocado en comunicar tanto aspectos técnicos como conceptuales del sistema. Se utilizó un lenguaje comprensible, adecuado para lectores técnicos y no técnicos, y se apoyó la redacción con diagramas, flujos y ejemplos. Esta estrategia permitió reflejar con objetividad los resultados obtenidos en el Sprint y la lógica detrás de cada decisión tomada durante el desarrollo.</p>

Criterio específico	Acciones realizadas	Conclusiones
	<p><b>Angello Rodolfo Sosa Colca</b>  <b>(TB1):</b> Documenté los diagramas tácticos y estructuras de clases, asegurando una redacción técnica precisa pero accesible para otros desarrolladores o evaluadores.</p> <p><b>(TP1):</b> Redacté explicaciones técnicas del diseño UI/UX en el Capítulo V y contribuí en el Capítulo VI con la documentación del proceso de integración del frontend con la lógica de negocio.</p> <p><b>Carlos Oswaldo Casimiro Fernández</b>  <b>(TB1):</b> Expliqué el funcionamiento de los repositorios en lenguaje claro y estructurado, facilitando la revisión y el entendimiento por parte de terceros.</p> <p><b>(TP1):</b> Redacté la documentación relacionada a la configuración del entorno, despliegue de servicios y validación técnica en el Capítulo VI.</p>	

## 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)
--------	--

---

Soy Mariano, tengo 19 años actualmente curso mi 6to ciclo de la carrera de Ingeniería de Software. Me considero una persona responsable y dispuesta a afrontar desafíos. Considero que estas cualidades beneficiaran a mi equipo para poder completar un trabajo de calidad.

---

Nombre	Avalos Santos, Anthony Piero (U20211C596)
--------	--

---

Soy Anthony Avalos, tengo 21 años, actualmente me encuentro en el 9no ciclo de mi carrera de Ingeniería de Software. Soy un apasionado por el desarrollo Fullstack, me considero una persona super proactiva y con ganas de seguir aprendiendo una nueva tecnología emergente. Me gusta el trabajo en equipo y colaborar con grande profesionales

---

Nombre	Lecca Choccare, Christopher Bryan (U202211399)
--------	--

---

Mi nombre es Christopher, tengo 20 años y actualmente me encuentro cursando el 7mo 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	Sosa Colca, Angello Rodolfo (U202212077)
--------	---

---

Tengo 19 años y curso el 7mo ciclo de la carrera de Ingeniería de Software en la Universidad Peruana de Ciencias Aplicadas. Soy una persona enfocada, perseverante y colaborativo. Estas cualidades me permiten ser una persona que ayudará de manera óptima el requisito que se me imponga. Además, soy una persona que apoya cuando los demás se encuentran en problemas.

---

Nombre	Carlos Oswaldo Casimiro Fernandez (U202115412)
--------	--

---

---

Tengo 21 años y soy estudiante de Ingeniería de Software de la Universidad Peruana de Ciencias Aplicadas. Me considero una persona proactiva, educada y sociable con los que me rodean. Además,uento con habilidades peculiares como la animación en 3D, edición de videos cinematográficos y conocimiento en lenguajes como C++, Python.

## 1.2. Solution Profile

En esta sección se detallan los aspectos claves de nuestra solución de software, incluyendo sus características distintivas y propuestas de valor.

**Product Name:** AlquilaFácil

**Producto Descriptivo:** AlquilaFácil es una plataforma diseñada para simplificar el proceso de búsqueda y reserva de espacios para eventos, conectando de manera eficiente a arrendadores y arrendatarios. Ofrece una experiencia fluida, permitiendo a los usuarios buscar, reservar y gestionar espacios para una amplia variedad de eventos. Además, la plataforma incluye herramientas de gestión y seguimiento en tiempo real, asegurando una experiencia optimizada tanto para arrendadores como para arrendatarios.

**Monetización:** AlquilaFácil generará ingresos a través de un modelo de suscripción para arrendadores que deseen destacar sus propiedades y acceder a herramientas avanzadas de gestión de reservas. Estos planes premium ofrecerán beneficios adicionales, como una mayor visibilidad en los resultados de búsqueda y acceso prioritario a herramientas analíticas avanzadas. Además, monetizaremos la plataforma mediante la implementación de sensores en los locales, permitiendo a los propietarios recibir reportes detallados sobre la actividad en sus espacios. En contraste, el acceso a la plataforma será completamente gratuito para los arrendatarios, incentivando su participación y aumentando la oferta de espacios disponibles.

### 1.2.1. Antecedentes y Problemáticas

Actualmente, las personas tienen fácil acceso a herramientas tecnológicas para reservar y alquilar locales de eventos, en páginas web o mediante un aplicativo móvil. Esto genera una mayor demanda en nuestro país sobre el creciente mercado de las reservas online. Además, con el incremento de usuarios que tienen acceso a internet, uso de dispositivos para facilitar el proceso de reserva espacios para eventos. Según el INEI, en los primeros tres meses del año 2022, 73 de cada 100 personas de 6 y más años accedieron a Internet en el país, cifra que muestra un crecimiento de 5,1 y 17,7 puntos porcentuales comparado con igual trimestre de 2021 (67,4%) y 2019 (54,8%), respectivamente (Jasin, 2023).

También es importante mencionar la creciente demanda por la elaboración de eventos por parte de las marcas. Según Mañas (2024), las marcas apuestan cada vez más por los eventos para darse a conocer e interactuar directamente con su audiencia, pues muchas consideran que estas experiencias ayudan a mostrar sus valores y a construir relaciones sólidas con sus clientes.

Ante ello, podemos demostrar que en el Perú el número de personas con acceso a internet a incrementado considerablemente en los últimos dos años, por otro lado, también justificamos la necesidad de una plataforma para facilitar la búsqueda en alquiler de espacios ante la creciente demanda de estas por parte de las empresas.

#### Problemáticas - Técnica de las 5 'W's y 2 'H's

**What?**

**¿Cuál es el problema?**

El problema es la falta de una plataforma centralizada y enfocada en el alquiler de espacios para eventos. Actualmente, arrendadores y arrendatarios enfrentan dificultades al buscar y reservar espacios adecuados para sus necesidades.

### **¿Cuál es la relación con la persona en cuestión?**

La persona en cuestión es que, si es arrendador de un espacio para eventos, puede enfrentar dificultades al encontrar y atraer a potenciales clientes para alquilar su espacio. Del mismo modo, si es arrendatario, puede enfrentar desafíos al buscar y reservar un espacio adecuado para sus eventos.

### **When?**

### **¿Cuánto sucede el problema?**

El problema ocurre cada vez que arrendadores desean alquilar sus instalaciones para eventos o arrendatarios buscan espacios para celebrar sus actividades. Esta necesidad surge de manera recurrente y en diferentes momentos, ya sea para eventos planificados con anticipación o para aquellos que requieran una búsqueda rápida de un espacio disponible.

### **¿Cuándo utiliza el cliente el producto?**

Cuando organizaciones o personas necesitan encontrar y reservar un espacio para un evento, ya sea con anticipación o de manera urgente. También, para aquellas personas que poseen instalaciones para realizar eventos y deseen promocionar su espacio, conectándose así con clientes potenciales.

### **Where? ¿Dónde está el cliente cuando usa el producto?**

El cliente utiliza AlquilaFácil a través de su aplicación móvil.

### **¿A dónde se dirige?**

Se dirige a una plataforma en línea o a una aplicación móvil, donde puede buscar, comparar y reservar espacios para eventos de manera conveniente y eficiente.

### **¿Dónde surge el problema?**

El problema surge en la dificultad para encontrar y reservar espacios adecuados para eventos, ya sea durante la planificación anticipada de un evento o en situaciones donde se requiere una búsqueda rápida.

### **Why?**

### **¿Cuál es la causa del problema?**

La falta de una plataforma centralizada y eficiente que conecte a arrendadores con personas que buscan alquilar un espacio para eventos. La ausencia de un sistema unificado dificulta la búsqueda y reserva de espacios adecuados, lo que puede generar frustración y pérdida de oportunidades tanto para arrendadores como para arrendatarios.

### **How?**

### **¿En qué condiciones los clientes usan nuestro producto?**

Los clientes utilizan AlquilaFácil para buscar y reservar espacios para una variedad de eventos, ya sea para eventos sociales o corporativos.

## **¿Cómo prefieren los clientes acceder a nuestro producto?**

Los clientes prefieren acceder a nuestro producto, a través de una plataforma web o móvil.

## **¿Qué llevó al cliente a llegar a esta situación?**

La falta de una plataforma para el alquiler de espacios para eventos llevó tanto a los clientes que buscan espacios como a los arrendadores a esta situación.

## **How much?**

Los costos de AlquilaFácil varían según el plan de suscripción seleccionada. Ofrecemos opciones gratuitas para los usuarios básicos, así como planes de suscripción mensual o anual con tarifas moderadas para aquellos que deseen optar por un plan premium y beneficios adicionales.

### **1.2.2. Lean UX Process**

En esta parte, llevaremos a cabo el proceso de Lean UX, que abarca la visión del modelo de negocio que respaldará nuestro producto de software. Este enfoque metodológico nos permite concentrarnos en el diseño de nuestra solución y en la resolución de los problemas identificados a través del pensamiento de diseño.

#### **1.2.2.1. Lean UX Problem Statement**

Nuestro contexto se enmarca en el mercado de alquiler de espacios para eventos en Perú, un sector que ha experimentado un crecimiento sostenido en los últimos años. Con el aumento de la demanda por espacios flexibles y adaptables para diferentes tipos de eventos, también surgen desafíos en la manera en que los usuarios, tanto arrendadores como arrendatarios, interactúan dentro de este ecosistema.

Hemos identificado que, en el mercado de alquiler de espacios, los usuarios enfrentan dificultades por la falta de integración en el proceso. La fragmentación en la búsqueda, gestión de reservas y organización genera ineficiencias y una experiencia insatisfactoria.

¿Cómo podemos mejorar la conexión entre arrendadores y arrendatarios, optimizando el proceso de alquiler de espacios para eventos y ofreciendo una experiencia más fluida y eficiente?

#### **1.2.2.2. Lean UX Assumptions**

##### **Business Outcomes:**

- Asegurar que la plataforma procese al menos 1,000 reservas de eventos en el primer año, optimizando la gestión y la comunicación entre arrendadores y arrendatarios.
- Lograr una tasa de satisfacción del usuario del 90% o más mediante la implementación de herramientas que permitan verificar clasificaciones e historial de espacios, ayudando a los arrendatarios a tomar decisiones informadas.
- Renovación de suscripciones del 80% al final del primer año, indicando la satisfacción y la retención de clientes.

##### **User Assumptions:**

- ¿Quién es el usuario?

- Arrendatarios
  - Arrendadores
- ¿Dónde entra nuestro producto en su trabajo o su vida?
    - **Arrendatarios:** AlquilaFácil simplifica la búsqueda y reserva de espacios, optimizando la planificación de eventos y ahorrando tiempo y esfuerzo.
    - **Arrendadores:** AlquilaFácil facilita la promoción y gestión eficiente de sus espacios, aumentando su visibilidad y el número de reservas.
- 
- ¿Cuál es el problema que nuestro producto soluciona?
    - Comunicación ineficiente entre arrendatarios y arrendadores.
    - Dificultad para encontrar y reservar espacios adecuados.
    - Falta de transparencia y confianza en la calidad y disponibilidad de los espacios.
    - Necesidad de una herramienta centralizada para gestionar reservas y promover espacios.
- 
- ¿Cómo y cuándo nuestro producto es usado?
    - **Arrendatarios:** Usan la plataforma para buscar, filtrar y reservar espacios para sus eventos, tanto durante la planificación como en la fase de ejecución.
    - **Arrendadores:** Usan la plataforma para promocionar sus espacios y gestionar reservas.
- 
- ¿Qué características son importantes?
    - Interfaz intuitiva y fácil de usar para búsqueda y reserva de espacios.
    - Sistema de promoción y visibilidad para los espacios.
    - Herramientas para gestionar reservas y comunicarse eficientemente.
    - Acceso a clasificaciones y reseñas para tomar decisiones informadas.
- 
- **User Outcomes:**
    - Los arrendatarios de eventos lograrán una reducción del 20% en el tiempo promedio dedicado a la búsqueda y reserva de espacios para eventos dentro de los primeros seis meses de uso de la plataforma.
    - Los arrendadores de espacios aumentarán en un 25% el número de reservas y consultas recibidas en sus espacios durante el primer año de implementación de la plataforma.

- Los arrendatarios de eventos experimentarán una mejora del 30% en la satisfacción con la calidad y adecuación de los espacios reservados, medida a través de reseñas y calificaciones en la plataforma 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 cantidad de clientes recurrentes.

Sabremos que estamos en lo correcto cuando observemos un incremento en la cantidad de reservas realizadas por clientes recurrentes, proyectando un aumento del 25% dentro de los primeros seis meses de la implementación de esta mejora.

#### **Hipótesis 2**

Creemos que, al mejorar la función de búsqueda avanzada en nuestra plataforma, permitiendo a los usuarios filtrar por características específicas de los espacios (como capacidad, ubicación y servicios incluidos), aumentaremos la satisfacción del usuario y mejoraremos la precisión en la selección de espacios.

Sabremos que estamos en lo correcto cuando observemos una disminución del 25% en las consultas de soporte relacionadas con la búsqueda de espacios y un aumento en el tiempo promedio que los usuarios pasan en la plataforma explorando opciones.

#### **Hipótesis 3**

Creemos que, al ofrecer descuentos exclusivos en tarifas de alquiler de espacios seleccionados para los usuarios con suscripción premium, lograremos aumentar la conversión de usuarios de planes básicos a planes premium.

Sabremos que estamos en lo correcto cuando observemos un aumento del 30% en la cantidad de usuarios que actualizan su suscripción a premium y un incremento en el volumen de reservas realizadas por estos usuarios, dentro de los primeros seis meses de la implementación de esta estrategia.

#### **Hipótesis 4**

Creemos que, al implementar un sistema de calificación y comentarios para los espacios disponibles en nuestra plataforma, aumentaremos la confianza de los usuarios y mejoraremos la percepción de calidad de los espacios, lo que impulsará un mayor número de reservas.

Sabremos que estamos en lo correcto cuando observemos un aumento del 20% en la cantidad de reservas completadas, junto con una mejora en las calificaciones y comentarios proporcionados por los usuarios, dentro de los primeros tres meses de la implementación.

## Hipótesis 5

Creemos que, al implementar un programa de referidos que recompense a los usuarios por traer nuevos arrendadores o arrendatarios a nuestra plataforma, aumentaremos la adquisición de usuarios y fomentaremos el crecimiento orgánico de nuestra base de clientes.

Sabremos que estamos en lo correcto cuando observemos un aumento del 35% en la cantidad de nuevos registros y un incremento en el número de referidos realizados por los usuarios existentes, dentro de los primeros tres meses de implementación del programa.

### 1.2.2.4. 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

Después de realizar una investigación de mercado, hemos identificado tres plataformas que ofrecen características similares a las de nuestra aplicación para conectar arrendadores con arrendatarios de eventos. Estas son:

### Competidores:

- **Airbnb:** Es una plataforma en línea fundada en 2008 que revolucionó la industria hotelera al ofrecer una alternativa a los hoteles tradicionales. Permite a los usuarios alquilar alojamientos a corto plazo, desde habitaciones individuales hasta casas completas, en más de 191 países. Facilita la conexión entre anfitriones que desean alquilar sus propiedades y huéspedes en busca de alojamiento temporal.



- **Vrbo:** Es una plataforma en línea fundada en 1995 que facilita el alquiler de alojamientos vacacionales directamente a través de los propietarios. El nombre significa "Vacation Rentals by Owner" (Alquileres Vacacionales por Propietario). Es uno de los principales sitios web para alquileres vacacionales en todo el mundo, donde los usuarios pueden buscar y reservar casas, apartamentos, cabañas y villas en una variedad de destinos. Esto les permite encontrar opciones que se adapten a sus necesidades y preferencias.



- **Booking:** Es una plataforma que permite a los usuarios reservar alojamiento en hoteles, hostales, apartamentos y otros tipos de alojamiento en todo el mundo. También ofrece la posibilidad de reservar vuelos, alquilar coches y reservar actividades turísticas. Booking.com es una de las mayores plataformas de reserva de alojamiento en línea y ofrece una amplia variedad de opciones para viajeros de todo tipo. Los usuarios pueden buscar alojamientos según sus preferencias y presupuesto, y realizar reservas de manera rápida y conveniente a través de la plataforma.



### 2.1.1. Análisis competitivo

#### **Competitive Analysis Landscape**

---

**Escriba en el recuadro la pregunta que busca responder o el objetivo de 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.**

---

AlquilaFácil

Airbnb

Vrbo

Booking.com

## Competitive Analysis Landscape

**Escriba en el recuadro la pregunta que busca responder o el objetivo de 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.**

	AlquilaFácil	Airbnb	Vrbo	Booking.com
Perfil	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.	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.
Overview	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.	Ofrece una amplia variedad de opciones de alojamiento, desde habitaciones individuales hasta casas completas.	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.

## Competitive Analysis Landscape

**Escriba en el recuadro la pregunta que busca responder o el objetivo de 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.**

	AlquilaFácil	Airbnb	Vrbo	Booking.com
Ventaja competitiva ¿Qué valor ofrece a los clientes?	<p>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.</p>	<p>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.</p>	<p>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.</p>	<p>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.</p>

## Competitive Analysis Landscape

**Escriba en el recuadro la pregunta que busca responder o el objetivo de 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.**

	AlquilaFácil	Airbnb	Vrbo	Booking.com
Perfil de Marketing	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.
Mercado Objetivo				

## Competitive Analysis Landscape

Escriba en el recuadro la pregunta que busca responder o el objetivo de 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.

	AlquilaFácil	Airbnb	Vrbo	Booking.com
Estrategias de Marketing	<p>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.</p> <p>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.</p> <p>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.</p> <p>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.</p>			

## Competitive Analysis Landscape

Escriba en el recuadro la pregunta que busca responder o el objetivo de 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.

	AlquilaFácil	Airbnb	Vrbo	Booking.com
Productos & servicios	<p>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.</p>	<p>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.</p>	<p>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.</p>	<p>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.</p>

## Competitive Analysis Landscape

Escriba en el recuadro la pregunta que busca responder o el objetivo de 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.

	AlquilaFácil	Airbnb	Vrbo	Booking.com
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	<p>Fortalezas</p> <p>Solución completa para la planificación de eventos.</p>	Gran comunidad de anfitriones y usuarios.	Variedad de alojamientos en todo el mundo.	Interfaz fácil de usar y experiencia intuitiva del usuario.
	<p>Debilidades</p> <p>Dependencia de la disponibilidad de espacios para eventos.</p>	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.

## Competitive Analysis Landscape

**Escriba en el recuadro la pregunta que busca responder o el objetivo de 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.**

	AlquilaFácil	Airbnb	Vrbo	Booking.com
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

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 herramientas innovadoras para la gestión de reservas y un sistema de promoción destacada para que sus espacios tengan mayor visibilidad. Además, contarán con opciones para integrar servicios adicionales como catering o entretenimiento, facilitando así una oferta más completa para los clientes.

**Para los arrendatarios:** Nuestra interfaz será fácil de usar y estará diseñada para simplificar todo el proceso de búsqueda, comparación y reserva de espacios. AlquilaFácil también permitirá la personalización de eventos y la contratación de servicios complementarios desde la misma plataforma, brindando una experiencia integral y diferenciada.

#### **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 compartan 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.
- **Modelos de referidos:** Se incentivará a los usuarios actuales a invitar a nuevos miembros mediante recompensas como descuentos, servicios premium o visibilidad destacada.

## 2.2. Entrevistas

### 2.2.1. Diseño de entrevistas

Se ha definido una cierta cantidad de preguntas para cada uno de nuestros segmentos objetivo, con la finalidad de obtener información cualitativa como opiniones o descripciones. Esta información nos será de gran ayuda en 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 propósito de las entrevistas con arrendadores es comprender sus necesidades, expectativas y preocupaciones respecto a la gestión de sus propiedades destinadas a eventos. Nos interesa conocer su disposición a utilizar una plataforma centralizada como AlquilaFácil para facilitar la administración de reservas, pagos, comunicación con usuarios y resolución de conflictos. También buscamos indagar sobre su interés en herramientas que les brinden mayor visibilidad y control durante el uso de sus espacios.

La información recolectada nos permitirá diseñar funcionalidades que se alineen con sus prioridades, como mecanismos de aprobación de usuarios, sistemas de monitoreo, métodos de pago integrados y soporte ante

imprevistos. De este modo, buscamos generar confianza en el uso de la plataforma y ofrecer un valor agregado que optimice 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 vos 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 propósito de las entrevistas con este segmento es explorar sus hábitos de búsqueda, reservas y uso de espacios para eventos, así como sus expectativas en cuanto a seguridad, transparencia y facilidad en los procesos. Nos interesa saber cuánto valoran las calificaciones, qué tan cómodo les resulta el proceso actual de alquiler y qué tan importante consideran aspectos como la posibilidad de realizar reclamos o recibir notificaciones durante el evento.

Con esta información, podremos construir una experiencia centrada en el usuario que permita encontrar espacios adecuados de forma sencilla y confiable, integrando herramientas que garanticen seguridad, reputación compartida entre usuarios y arrendadores, y comunicación eficiente. Todo esto con el objetivo de crear una plataforma amigable, transparente 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 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

Para el registro de entrevistas se realizará 3 entrevista por segmento, dando un total de 6 entrevistas. Además, el formato de las entrevistas es mp4, cada entrevista es independiente debido a las diferentes preguntas y respuestas dadas por los entrevistados de cada segmento.

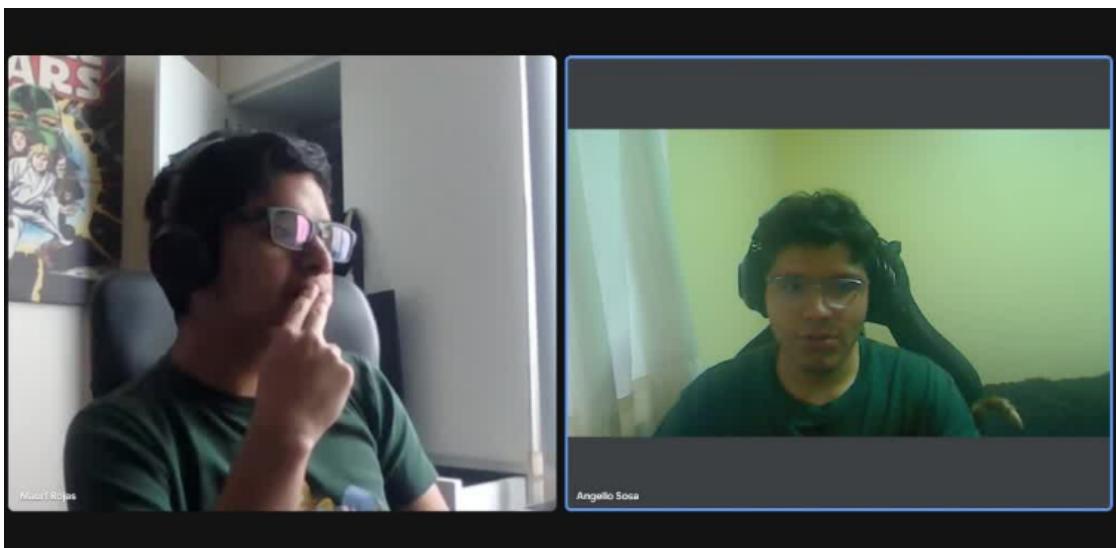
### **Segmento 1: Arrendadores de espacios para eventos**

#### **Entrevista #1**

Nombre	Mauricio Abraham
Apellidos	Rojas Ugarte
Edad	25
Distrito	Jesús María
Aplicaciones usadas	Facebook, WhatsApp, Excel.
Motivación	Proporcionar experiencias memorables a sus clientes, contribuyendo a la felicidad y la celebración de momentos importantes.
Frustración	Desafíos en la gestión de reservas y coordinación de eventos.
Tecnologías	Android, Windows
Browsers	Brave
Entrevistador	Angello Sosa

**Entrevista #1**

Evidencia

Enlace [Microsoft Stream](#)

Duración 00:00 min - 00:00 min

Resumen El entrevistado Mauricio Ugarte tiene 25 años y reside en Pueblo Libre, él es un arrendador con 3 años en el negocio. Durante la entrevista nos reveló diversos datos sobre cómo lleva a cabo el proceso de gestionar sus locales, dentro de esta información se encuentra que maneja el sistema de disponibilidad de sus locales de forma manual, mediante el uso de aplicativos como Excel. Asimismo, indicó que últimamente tuvo problemas con el alquiler de sus espacios debido a la falta de herramientas de gestión del local que le notifiquen lo que está sucediendo en el momento.

**Entrevista #2**

Nombre Lucero

Apellidos Smith Cardenas

Edad 40 años

Distrito San Isidro

Aplicaciones usadas Instagram, X.

Motivación Explorar nuevas oportunidades de negocio y expandir su presencia en el mercado de eventos sociales

Frustración Dificultades para promocionar eficazmente su local y atraer nuevos clientes.

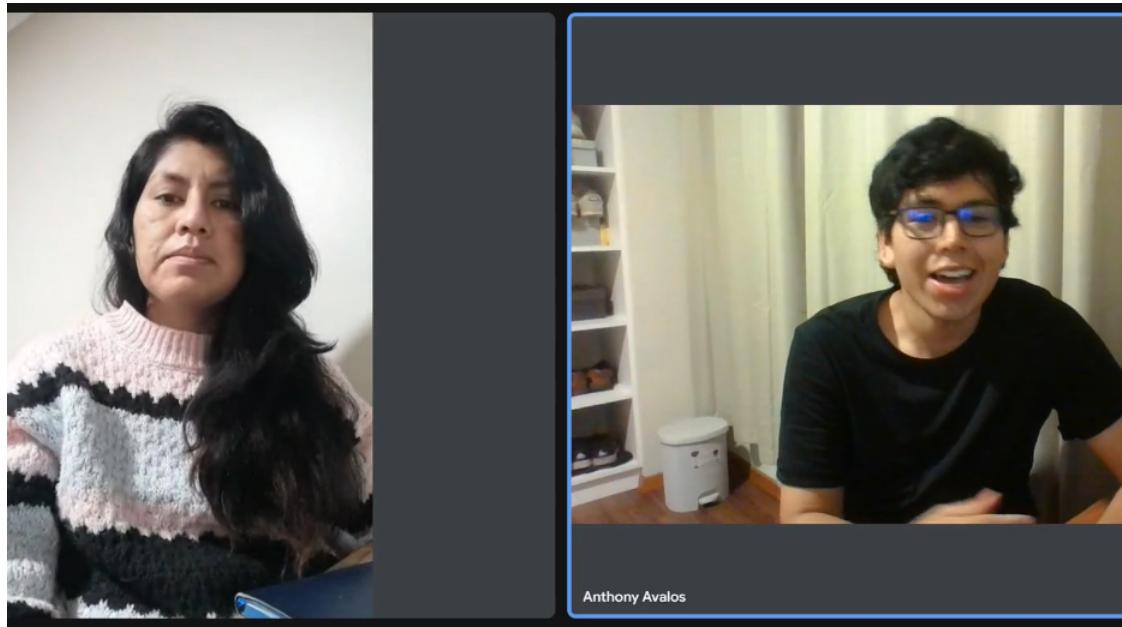
Tecnologías iOS, Android, Windows.

Browsers Google Chrome

Entrevistador Anthony Avalos

## Entrevista #2

Evidencia



Enlace [Microsoft Stream](#)

Duración 00:00 min - 00:00 min

Resumen

Lucero es una profesional la cual dispone de muchas propiedades, donde frecuentemente los podes a disposición de amigos o conocidos para que puedan hacer sus reuniones y/o eventos. Ella nos comenta que AlquilaFacil sería una excelente opción para agestión reservas de sus espacios, mantener una comunicación con los usuarios, y gestionar adecuadamente el cuidado o reglas que este tenga durante el evento. Le gustaría que se maneje un sistema de calificaciones para ayudar a otros usuarios y también para ella misma tener conocimiento de que clase de usuario esta por alquilar su espacio. Nos comenta también que le parece una super idea el tener control total de lo que esté ocurriendo en el evento, como por ejemplo ver que el arrendatario este cumpliendo con las reglas que se le establecieron. También le parecería interesante el poder recibir alertas cuando se incumpla ciertos acuerdos establecidos. Ella cree que sensores de diversos tipos serían los adecuados para llevar un control total de lo que esté ocurriendo en el espacio que está dando a disposición de los usuarios. Le gustaría que todo se haga desde una interfaz sencilla y adaptable a todo tipo de dispositivos.

## Entrevista #3

Nombre Luis David

Apellidos Garcia Días

Edad 21 años

Distrito Magdalena

Aplicaciones usadas Instagram, Excel.

**Entrevista #3**


---

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

---




---

Enlace	<a href="#">Microsoft Stream</a>
Duración	00:00 min - 00:00 min
Resumen	<p>El entrevistado mencionó que, si se encuentra interesado en la aplicación, lo que mejoraría su negocio al poder gestionar de manera más sencillas sus locales y los arrendatarios. Mencionó que anteriormente ha tenido problemas a la hora de organizar reservas mientras no contaba con una herramienta especializada para llevar un recuento de sus reservaciones. También se mencionó que tuvo problemas con diferentes clientes, que de haber tenido un perfil de ellos de antemano se podrían haber evitado. Por último, se mostró interesado en poder integrar todo su proceso desde la promoción del local, la reserva y el pago, hasta la gestión y manejo del mismo desde una misma aplicación.</p>

---

**Segmento 2: Arrendatarios sociales frecuentes de espacios para eventos****Entrevista #1**


---

Nombre	Manuel Jesús
Apellidos	Chávez Cuba
Edad	21
Distrito	San Martín de Porres

---

**Entrevista #1**

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	<a href="#">Microsoft Stream</a>
Duración	00:00 min - 00:00 min
Resumen	<p>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.</p>

**Entrevista #2**

Nombre	Fabio Ernesto
Apellidos	Horna Silva
Edad	23
Distrito	Surco
Aplicaciones usadas	Instagram
Motivación	Establecerse como un organizador de eventos reconocido en su área.

### Entrevista #2

---

Frustración	Dificultades para encontrar espacios con características específicas.
Tecnologías	Android, Windows.
Browsers	Google Chrome, Brave
Entrevistador	Carlos Casimiro

---

Evidencia




---

Enlace	<a href="#">Microsoft Stream</a>
Duración	00:00 min - 00:00 min
Resumen	<p>Fabio Orna Silva, consultor de ingeniería, respondió sobre su experiencia como usuario y proveedor de espacios para eventos, resaltando los desafíos que enfrenta al alquilar y ofrecer inmuebles. Considera muy útil una app como Alquilar Fácil, ya que facilitaría la búsqueda, comparación de precios y reservas en un solo lugar. Destacó la importancia de contar con calificaciones y opiniones de otros usuarios, procesos rápidos y seguros, así como la posibilidad de hacer reclamos directamente desde la plataforma. Además, valoró la existencia de un sistema de reputación para arrendadores y arrendatarios, y sugirió integrar herramientas tecnológicas como cerraduras electrónicas y sistemas de gestión de reservas para mejorar la experiencia y confianza en el servicio.</p>

### Entrevista #3

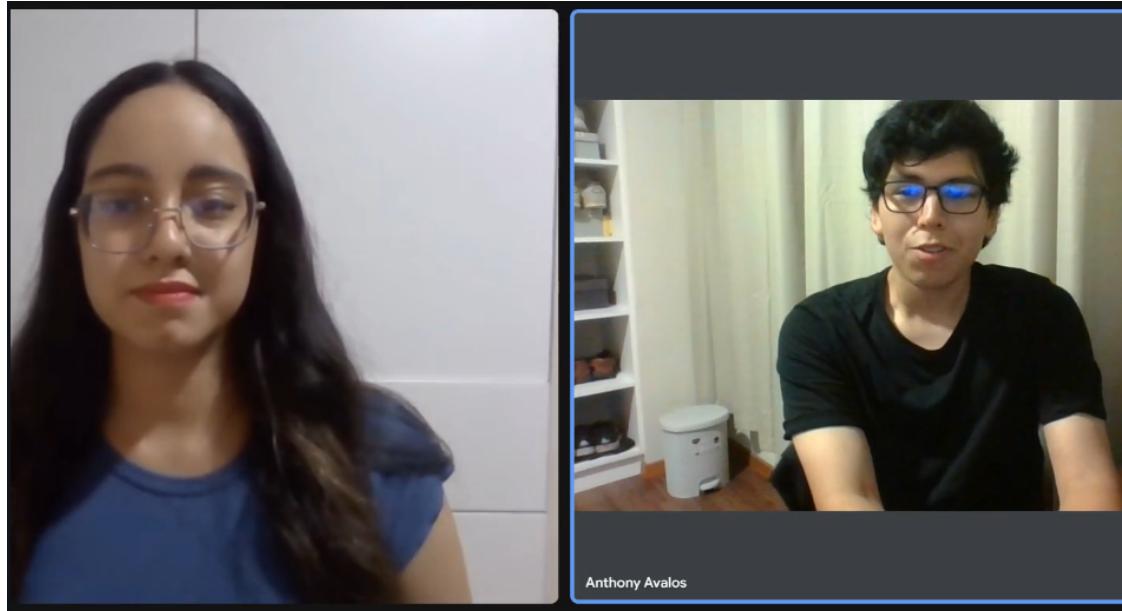
---

Nombre	Flavia
Apellidos	Cáceres Bustamente
Edad	23 años
Distrito	San Isidro

---

**Entrevista #3**

Aplicaciones usadas	Instagram, TikTok.
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	Anthony Avalos



Enlace	<a href="#">Microsoft Stream</a>
Duración	00:00 min - 00:00 min
Resumen	<p>Flavia es una estudiante de los últimos ciclos de su carrera. Su familia suele realizar eventos masivos frecuentemente, y le encargan a ella poder encontrar un lugar adecuado para llevarlo a cabo. Nos comenta que esta tarea se ve simplificada con una app como AlquilaFacil. Le gustaría que dentro de esta app se pueda ver reseñas de los lugares que posiblemente alquilara. Le parecería genial una plataforma super intuitiva y con una experiencia de usuario fluida. Nos comenta que muchas veces no puede llevar el control del cumplimiento de las reglas que se le puso al alquilar ciertos locales. Le gustaría que el local tenga tecnología como sensores de humo, si es que no se puede fumar en el local, o también nel control de aforo, que es uno de los problemas más frecuentes que tiene con sus arrendadores. Y más allá de eso, le gustaría que estas herramientas velen por su seguridad y la de sus invitados durante el evento. Le gustaría una aplicación móvil donde pueda recibir alertas si es que alguno de estos límites establecidos dentro del local se está saliendo de control. Una aplicación que te facilite todo este trabajo y te genere tranquilidad, le parecería de mucha utilidad.</p>

**2.2.3. Análisis de entrevistas**

Las entrevistas realizadas proporcionan una visión detallada y matizada sobre las expectativas y requisitos de los distintos actores involucrados en el ámbito de las plataformas de alquiler de espacios para eventos. En general, tanto los arrendadores como los arrendatarios y los usuarios finales destacan la necesidad imperiosa de contar con sistemas que aseguren la integridad de las transacciones financieras, la claridad en los acuerdos contractuales y una comunicación fluida y eficaz.

### **Arrendadores de espacios para eventos**

Para los arrendadores, la seguridad en las transacciones financieras y la integridad de los contratos son aspectos fundamentales. También valoran una promoción efectiva de sus propiedades, buscando una plataforma que les permita llegar a un público amplio y relevante. Además, aprecian una comunicación clara y eficiente con los arrendatarios para evitar malentendidos y conflictos. Algunos arrendadores están dispuestos a invertir en servicios premium que les brinden ventajas adicionales, como una mayor visibilidad de sus espacios en la plataforma. En resumen, los arrendadores buscan una plataforma que les ofrezca seguridad, visibilidad, comunicación eficiente y opciones para maximizar el rendimiento de sus espacios.

<b>Característica objetiva</b>	<b>Frecuencia</b>	<b>Porcentaje</b>	<b>Sustento desde entrevistas</b>
<b>Utiliza redes sociales para captar clientes</b>	3 de 3.	100%	Menciona Instagram y Facebook como fuentes principales de clientes
<b>Gestiona reservas manualmente</b>	3 de 3.	100%	Gestiona la disponibilidad con agenda física y Excel
<b>Recibe consultas por WhatsApp</b>	3 de 3.	100%	Coordina todo el proceso por WhatsApp
<b>Necesita un sistema de agenda para evitar dobles reservas</b>	3 de 3.	100%	Tuvo una doble reserva por falta de sistema automatizado

### **Arrendatarios para eventos**

Por otro lado, los arrendatarios tienen necesidades específicas al utilizar plataformas de alquiler de locales. Para ellos, la facilidad de encontrar y reservar espacios adecuados es crucial, especialmente en situaciones de emergencia o cambios de último momento. Valoran la transparencia en los contratos y una comunicación eficiente con los arrendadores para evitar malentendidos y asegurar una experiencia sin contratiempos. Además, buscan plataformas que ofrezcan una amplia variedad de opciones de espacios que se ajusten a sus necesidades específicas y horarios. La posibilidad de utilizar herramientas de búsqueda avanzada y filtrado también es importante para encontrar el lugar perfecto. En resumen, los arrendatarios de eventos buscan plataformas que les brinden facilidad de uso, transparencia en los procesos, comunicación eficiente y una amplia gama de opciones de espacios para eventos.

<b>Característica objetiva</b>	<b>Frecuencia</b>	<b>Porcentaje</b>	<b>Sustento desde entrevistas</b>
<b>Uso de redes sociales o sitios web para buscar espacios</b>	3 de 3.	100%	Todos los entrevistados mencionan usar redes o sitios web
<b>Realiza reservas con regularidad</b>	3 de 3.	100%	Todos mencionan alquilar espacios de forma habitual

Característica objetiva	Frecuencia	Porcentaje	Sustento desde entrevistas
<b>Busca espacios para eventos familiares o recreativos</b>	3 de 3.	100%	Todos organizan o buscan espacios para actividades recreativas o familiares
<b>Valora herramientas que faciliten la organización del tiempo (agenda)</b>	3 de 3.	100%	Mencionan la utilidad de una agenda o calendario para organizarse mejor

## Determinación de Personas

Basado en el análisis de las entrevistas, se ha determinado lo siguiente:

### Persona Principal: Arrendatarios

**Razón:** Los arrendatarios son los usuarios finales que interactúan directamente con la plataforma para encontrar y reservar espacios. Sus necesidades de facilidad de uso, transparencia, comunicación eficiente y una amplia gama de opciones son cruciales para el éxito del producto. Además, la facilidad de encontrar y reservar espacios adecuados es fundamental para ellos, lo que hace que sus requerimientos sean el enfoque principal del desarrollo.

### Persona Secundaria: Arrendadores

**Razón:** Aunque los arrendadores también son esenciales para el funcionamiento de la plataforma, su papel es secundario en comparación con los arrendatarios. Los arrendadores buscan seguridad, visibilidad y herramientas para la promoción y gestión de sus espacios. Estos aspectos son importantes, pero el enfoque primario debe estar en las necesidades y experiencias de los arrendatarios, que son quienes utilizan la plataforma de manera más activa y recurrente.

## 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

PERSONA: Sergio Aguirre

NAME	MARKET SIZE	TYPE
<b>Sergio Aguirre</b>	 60 %	Arrendatario
	<b>Background</b>	
	<p>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.</p>	
<b>Demographic</b>	<b>Goals</b>	
<input checked="" type="radio"/> Male      19 years  San Isidro, Lima Estudiante de Ingeniería de Software	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.	
<b>Skills</b>	<b>Motivators</b>	<b>Challenges</b>
Planificación de eventos  Creatividad en diseño de experiencias  Gestión eficiente de presupuestos y recursos 	<ul style="list-style-type: none"> <li>Ofrecer experiencias únicas y memorables para sus clientes.</li> <li>Expandir su negocio y establecerse como un organizador de eventos reconocido en su área.</li> </ul>	<ul style="list-style-type: none"> <li>Limitaciones presupuestarias que pueden afectar la calidad de los eventos.</li> <li>Dificultades para encontrar espacios que cuenten con características específicas.</li> <li>Falta de herramientas de gestión durante el desarrollo de sus eventos.</li> </ul>
<b>Preferred channels</b>	<b>Technology</b>	
 PC  Web app  Smartphone	       	

**UXPRESSIA**  
This persona was built in [uxpressia.com](https://uxpressia.com)

## Segmento 2: Arrendador

PERSONA: Claudia Camareño

NAME <b>Claudia Camareño</b>	MARKET SIZE 40 %	TYPE <b>Arrendador</b>
	<b>Background</b>  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.	
<b>Demographic</b>  Female 46 years Surco, Lima Podóloga	<b>Goals</b>  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.	
<b>Skills</b>  Organización  Comunicación y negociación  Adaptabilidad 	<b>Motivators</b> <ul style="list-style-type: none"><li>Ver crecer su negocio y convertir su local en un lugar de referencia para eventos en su comunidad.</li><li>Proporcionar experiencias memorables y satisfactorias a sus clientes, contribuyendo a la felicidad y la celebración de momentos importantes.</li><li>Explorar nuevas oportunidades de negocio y expandir su presencia en el mercado de eventos sociales.</li></ul>	<b>Challenges</b> <ul style="list-style-type: none"><li>Dificultades para promocionar eficazmente su local y atraer clientes nuevos y recurrentes.</li><li>Desafíos en la gestión de reservas y la coordinación de eventos, especialmente en períodos de alta demanda.</li><li>Preocupación la competencia de otros locales para eventos en la zona y la posibilidad de perder clientes ante ofertas más atractivas.</li><li>Posibles accidentes dentro de su local por falta de herramientas para gestionarlo.</li></ul>
	<b>Preferred channels</b>  PC  Smartphone	<b>Technology</b>        

**UXPRESSIA**  
This persona was built in [uxpressia.com](https://uxpressia.com)

### 2.3.2. User Task Matrix

## 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

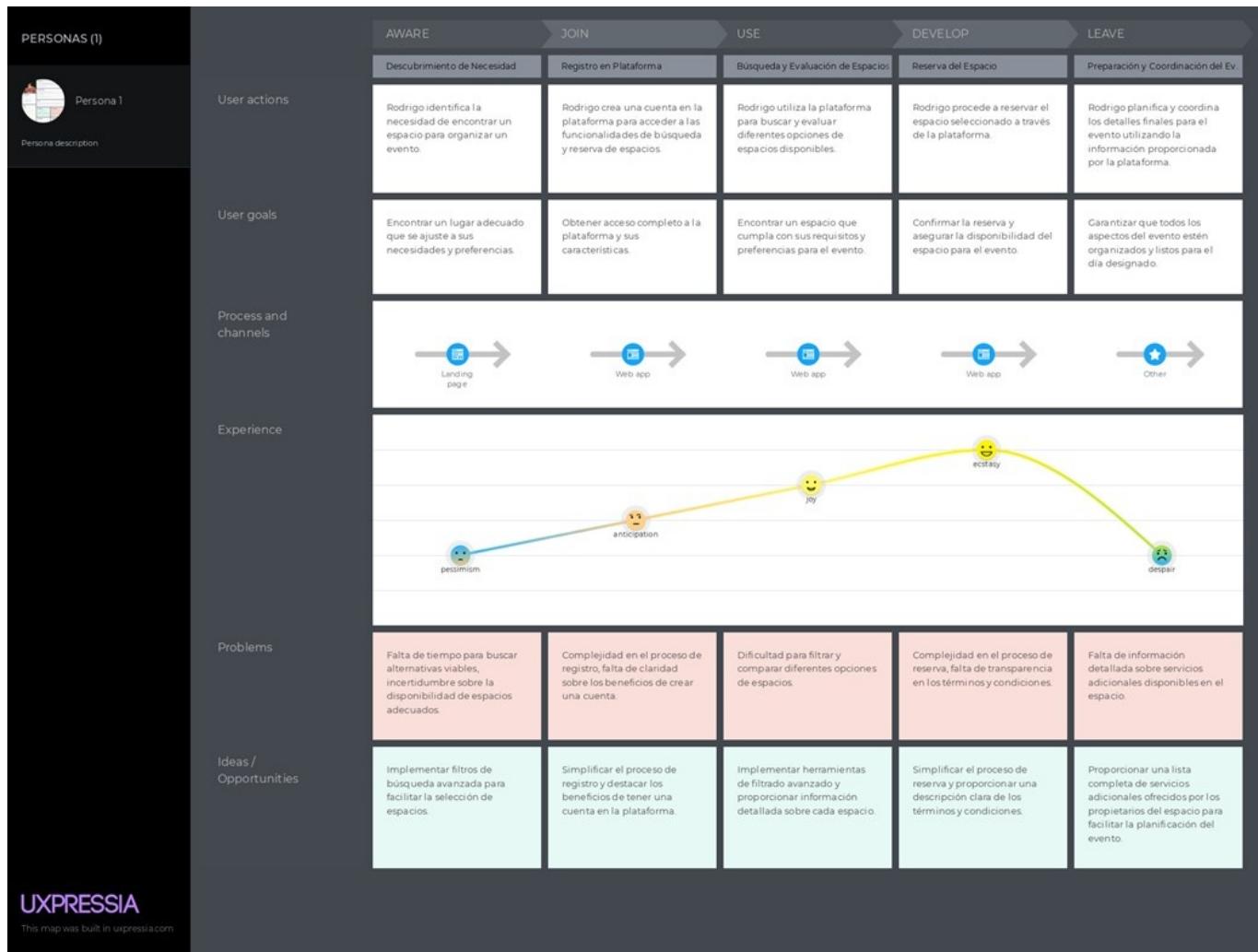
**Nombre:** Claudia Cañamero

**Rol:** Arrendadora

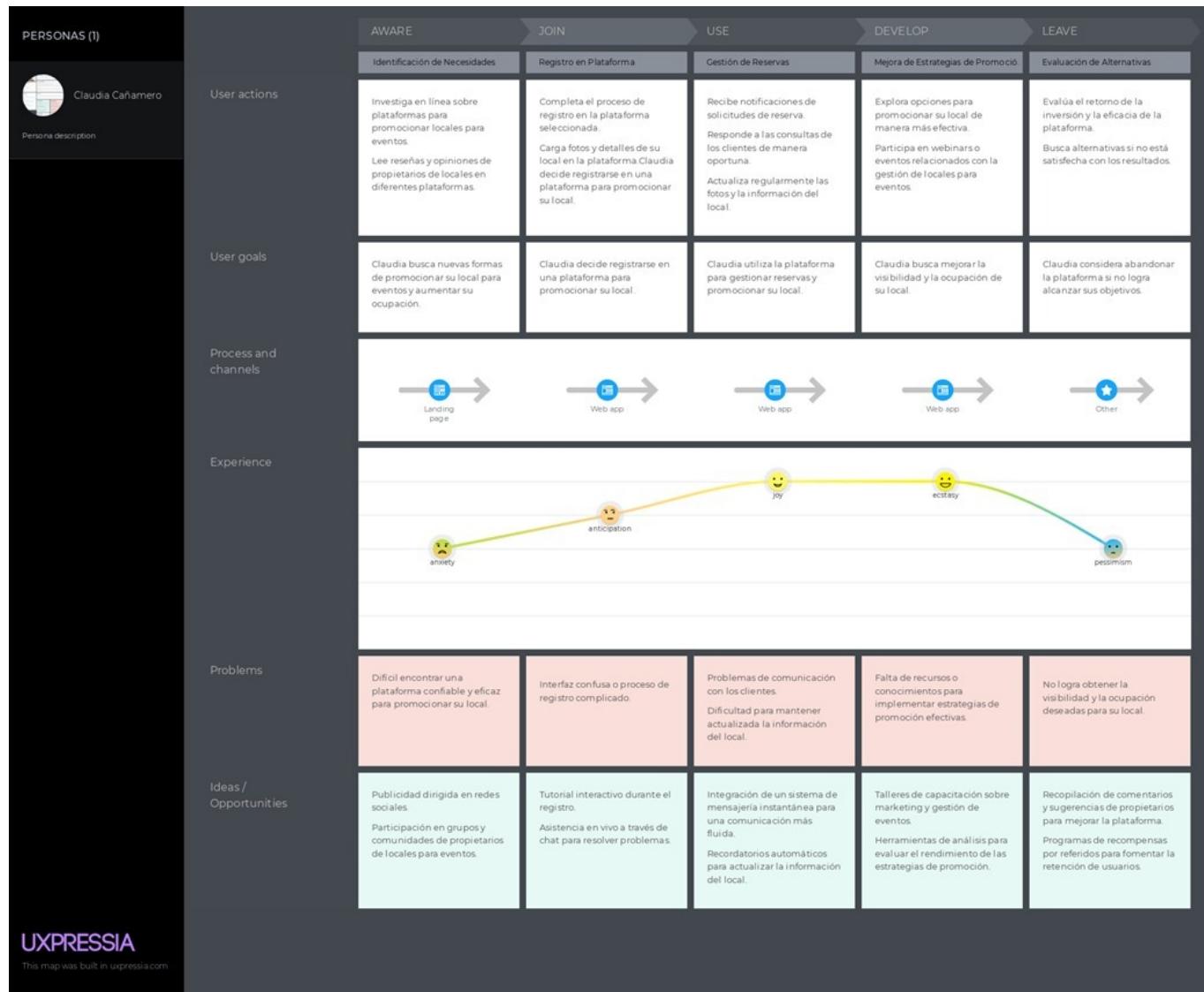
Task	Frequency	Importance
Gestionar reservas y coordinar eventos	High	High
Promocionar el local en redes sociales y otros canales de marketing	High	High
Mantener el local en óptimas condiciones de limpieza y mantenimiento	High	High
Establecer comunicación efectiva con clientes y proveedores	High	High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

### 2.3.3. User Journey Mapping

#### Persona Principal: Arrendatarios de eventos sociales



## Persona Secundaria: Arrendadores de espacios para eventos



### 2.3.4. Empathy Mapping

#### Persona Principal: Arrendatarios de eventos sociales

PERSONA: Sergio Aguirre

### 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 [uxpressia.com](https://uxpressia.com)

**Persona Secundaria: Arrendadores de espacios para eventos**

PERSONA: Claudia Cañamero

### 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.*

**UXPRESSIA**

This persona was built in [uxpressia.com](http://uxpressia.com)

### 2.3.5. As-is Scenario Mapping

#### Persona Principal: Arrendatarios

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

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	Publicar información 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 cliente 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	Pensar en posibles circunstancias donde el local resulte dañado		
Feeling	Ansiedad por encontrar un arrendatario que desee 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	Cansancio por el extenso proceso de búsqueda de cliente realizado	Preocupación por no olvidarse del acuerdo realizado	Preocupación por recibir de vuelta el 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

To-be Scenario: Arrendadores						
Steps	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
Doing	Publica información de su local en la aplicación	Espera a que el arrendatario seleccione su espacio en la aplicación	El arrendador coloca un precio adecuado por hora de reserva	La notificación del alquiler llega a la cuenta del arrendador	La reserva es agendada automáticamente en la sección de calendario	Accede a la sección de monitoreo para revisar el estado del espacio
Thinking	Evaluación de la idoneidad de los espacios para diferentes tipos de eventos	Incertidumbre por encontrar un cliente dispuesto a alquilar	Evaluación de la confiabilidad y seguridad del espacio para ser alquilado	Evaluación de la información adicional del espacio publicada	Evaluación de la confiabilidad y conveniencia	Evaluación de las reservas confirmadas hasta el momento
Feeling	Emoción por el pronto alquiler de su espacio	Ansiedad por encontrar un arrendatario en poco tiempo	Seguridad de que encontrará un arrendatario	Tranquilidad por saber que saldrá favorecido en este proceso	Alegre por la confirmación de la reserva	Aliviado de no perder mucho tiempo en la búsqueda del arrendatario
						Preparativos para dejar el local preparado para el evento
						Planificación de nuevos beneficios para próximas reservas
						Tranquilidad por saber que será notificado de cualquier circunstancia

## 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
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

### User Stories

<b>ID de la user story</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de aceptación</b>
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.	<p>Escenario 01: Registro exitoso</p> <p><b>Dado</b> que un usuario desea registrar su espacio en AlquilaFácil. <b>Cuando</b> el usuario completa el formulario de registro con la información requerida. <b>Entonces</b> el usuario recibe una confirmación de registro y puede acceder a su cuenta.</p> <p>Escenario 02: Validación de datos</p> <p><b>Dado</b> que un usuario completa el formulario de registro en AlquilaFácil. <b>Cuando</b> el usuario envía el formulario. <b>Entonces</b> los datos proporcionados se validan para garantizar la precisión y la autenticidad.</p>
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.	<p>Escenario 01: Inicio de sesión exitoso</p> <p><b>Dado</b> que un usuario registrado desea acceder a su cuenta en AlquilaFácil. <b>Cuando</b> el usuario ingresa su correo electrónico y contraseña correctos en el formulario de inicio de sesión. <b>Entonces</b> el usuario es autenticado exitosamente y se le otorga acceso a su cuenta.</p> <p>Escenario 02: Error en el inicio de sesión por credenciales incorrectas</p> <p><b>Dado</b> que un usuario registrado intenta acceder a su cuenta en AlquilaFácil. <b>Cuando</b> el usuario ingresa una combinación incorrecta de correo electrónico o contraseña. <b>Entonces</b> se le muestra un mensaje de error indicando que las credenciales son incorrectas y se le solicita que intente nuevamente.</p>

<b>ID de la user story</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de aceptación</b>
US03	Registrar espacios	<p>Como arrendador de un espacio para eventos en AlquilaFácil, quiero poder registrar mi espacio para comenzar a recibir solicitudes de reserva lo antes posible.</p>	<p>Escenario 01: Registro de datos básicos</p> <p><b>Dado</b> que un arrendador desea registrar su espacio en AlquilaFácil. <b>Cuando</b> completa distintos campos organizados en varios pasos detallados que incluyen información detallada sobre la propiedad. <b>Entonces</b> puede enviar el registro con éxito y recibir confirmación de su inclusión en la plataforma.</p> <p>Escenario 02: Validación de datos requeridos en cada paso del registro</p> <p><b>Dado</b> que un arrendador está completando el registro de su espacio en AlquilaFácil. <b>Cuando</b> rellena todos los campos correspondientes a un paso en específico. <b>Entonces</b> se activa el botón que le permite continuar al siguiente paso.</p>
US04	Buscar espacios disponibles	<p>Como arrendatario, quiero poder buscar fácilmente espacios disponibles en AlquilaFácil para encontrar el lugar perfecto para mi evento.</p>	<p>Escenario 01: Búsqueda principal por ubicación</p> <p><b>Dado</b> que un arrendatario busca un espacio para eventos en una ubicación específica. <b>Cuando</b> el arrendatario ingresa la ubicación deseada en el campo de búsqueda. <b>Entonces</b> se muestran los espacios disponibles en esa ubicación.</p> <p>Escenario 02: Búsqueda general de espacios</p> <p><b>Dado</b> que un arrendatario no ingresa algún texto en la barra de búsqueda. <b>Cuando</b> presiona sobre el ícono para buscar espacios. <b>Entonces</b> se muestran todos los espacios disponibles.</p>

<b>ID de la user story</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de aceptación</b>
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</p> <p><b>Dado</b> que un arrendatario desea un espacio con capacidad para un número específico de personas. <b>Cuando</b> el arrendatario aplica un filtro de capacidad en la búsqueda. <b>Entonces</b> se muestran solo los espacios que cumplen con ese criterio.</p> <p>Escenario 02: Filtrado por categoría</p> <p><b>Dado</b> que un arrendatario desea un espacio de una categoría en específico. <b>Cuando</b> el arrendatario aplica un filtro de categoría en la búsqueda. <b>Entonces</b> se muestran solo los espacios que cumplen con ese criterio.</p>
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</p> <p><b>Dado</b> que el arrendatario selecciona un espacio en AlquilaFácil. <b>Cuando</b> accede a la página de detalles del espacio. <b>Entonces</b> puede visualizar información detallada como el aforo máximo, descripción del espacio y servicios disponibles.</p> <p>Escenario 02: Visualización de reseñas del espacio</p> <p><b>Dado</b> que el arrendatario está revisando un espacio. <b>Cuando</b> accede a la página de comentarios. <b>Entonces</b> puede visualizar las reseñas y calificaciones dejadas por otros usuarios sobre ese espacio.</p>

ID de la user story	Título	Descripción	Criterios de aceptación
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 de para completar los detalles de reserva.</p> <p>Escenario 02: Pago de la reserva con PayPal <b>Dado</b> que el usuario está a punto de culminar el proceso de reserva de un espacio. <b>Cuando</b> el usuario presione el botón de realizar reserva. <b>Entonces</b> 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>

<b>ID de la user story</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de aceptación</b>
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</p> <p>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</p> <p>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> <p>Escenario 03: Existencia de reserva de espacio ajeno</p> <p>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</p> <p>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> <p>Escenario 02: Aporte de reseña</p> <p>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>

<b>ID de la user story</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de aceptación</b>
US10	Notificar sobre actividades de arrendatarios	Como arrendador de un espacio para eventos en AlquilaFácil, quiero recibir notificaciones <b>Cuando</b> 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</p> <p><b>Dado</b> que el arrendador ha publicado un espacio en AlquilaFácil. <b>Cuando</b> un arrendatario realiza una reserva para dicho espacio. <b>Entonces</b> 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> <p>Escenario 02: Notificación de reseña</p> <p><b>Dado</b> que el arrendador ha publicado un espacio en AlquilaFácil. <b>Cuando</b> un arrendatario realiza una reseña sobre dicho espacio. <b>Entonces</b> 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</p> <p><b>Dado</b> que el arrendatario está visualizando la página de detalles de un espacio. <b>Cuando</b> selecciona la opción de "Agregar a favoritos". <b>Entonces</b> el espacio se guarda en la lista de favoritos de forma local.</p> <p>Escenario 02: Eliminar de favoritos</p> <p><b>Dado</b> que el arrendatario ha guardado un espacio como favorito. <b>Cuando</b> selecciona la opción de "Eliminar de favoritos" en dicho espacio. <b>Entonces</b> el espacio se elimina de la lista de favoritos, y ya no se mostrará al realizar el filtrado.</p>

<b>ID de la user story</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de aceptación</b>
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><b>Dado</b> que el arrendador ha iniciado sesión en su cuenta de AlquilaFácil. <b>Cuando</b> accede a la sección "Mis espacios". <b>Entonces</b> puede ver una lista de todos los espacios que ha publicado, incluyendo información básica como nombre del espacio, ubicación.</p> <p>Escenario 02: Acceso a detalles y edición</p> <p><b>Dado</b> que el arrendador visualiza la lista de espacios publicados. <b>Cuando</b> selecciona un espacio específico. <b>Entonces</b> puede acceder a la página de detalles de ese espacio donde tiene la opción de editar la información.</p>
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><b>Dado</b> que el arrendador desea actualizar la información de su espacio. <b>Cuando</b> accede a la sección de edición de su espacio y realiza modificaciones en campos como descripción, aforo, servicios disponibles y tarifas. <b>Entonces</b> 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</p> <p><b>Temporal del espacio</b></p> <p><b>Dado</b> que el arrendador está realizando modificaciones en su espacio. <b>Cuando</b> activa la opción de "No disponible temporalmente" para realizar cambios. <b>Entonces</b> el espacio se muestra como no disponible para reservas.</p>

<b>ID de la user story</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de aceptación</b>
US14	Actualizar perfil y sesión de usuario	<p>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>	<p><b>Escenario 01: Modificación del nombre de usuario</b></p> <p><b>Dado</b> que el usuario ha iniciado sesión en su cuenta de AlquilaFácil. <b>Cuando</b> accede a la sección de configuración de perfil y edita el campo "Nombre de usuario". <b>Entonces</b> 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><b>Escenario 02: Cierre de sesión</b></p> <p><b>Dado</b> que el usuario desea salir de su cuenta en AlquilaFácil. <b>Cuando</b> selecciona la opción "Cerrar sesión" en la configuración de perfil. <b>Entonces</b> 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>
US15	Reportar espacios inseguros	<p>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>	<p><b>Escenario 01: Selección del espacio inseguro</b></p> <p><b>Dado</b> que el usuario ha iniciado sesión en su cuenta de AlquilaFácil. <b>Cuando</b> accede a la sección de "Reportar espacio" y selecciona el espacio desde la lista de reservas activas o buscando un espacio específico. <b>Entonces</b> puede ingresar un asunto y descripción del problema.</p> <p><b>Escenario 02: Confirmación del reporte</b></p> <p><b>Dado</b> que el usuario ha completado el formulario de reporte con todos los detalles requeridos. <b>Cuando</b> envía el reporte mediante el botón "Enviar". <b>Entonces</b> el reporte será registrado y revisado por el equipo de soporte de AlquilaFácil.</p>

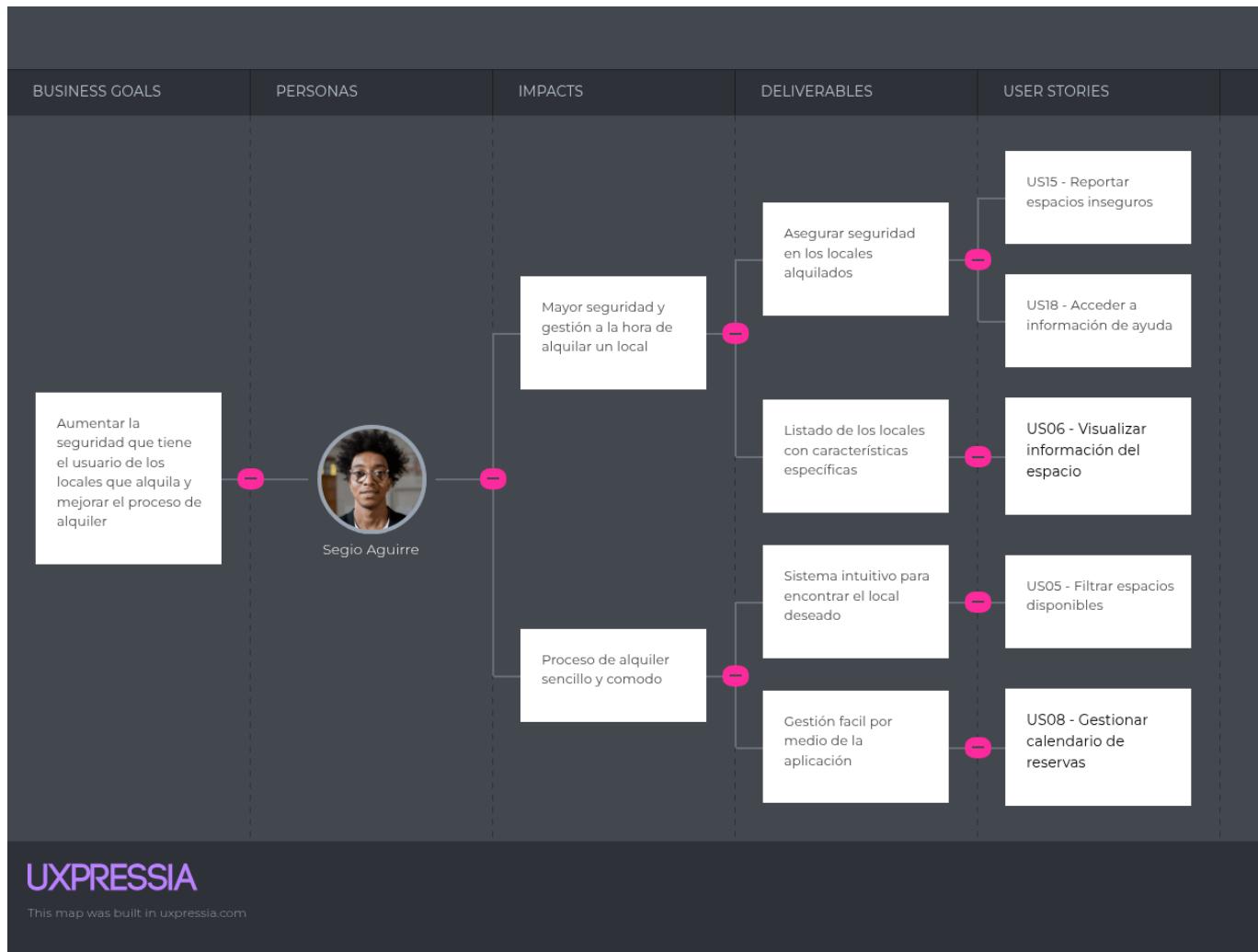
<b>ID de la user story</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de aceptación</b>
US16	Visualizar espacios reportados por el usuario	Como usuario de AlquilaFácil, quiero poder visualizar los espacios que he reportado, para realizar un seguimiento de mis reportes y gestionarlos.	<p>Escenario 01: Navegar a la sección de "Ver Espacios Reportados"</p> <p><b>Dado</b> que el usuario ha iniciado sesión en su cuenta de AlquilaFácil. <b>Cuando</b> accede a la sección de "Ver Espacios Reportados" desde el panel de control. <b>Entonces</b> 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> <p>Escenario 02: Eliminar un reporte deslizando hacia la izquierda</p> <p><b>Dado</b> que el usuario se encuentra en la lista de espacios reportados. <b>Cuando</b> desliza uno de los reportes hacia la izquierda. <b>Entonces</b> aparece una opción para confirmar la eliminación del reporte. Y si confirma, el reporte se elimina de la lista.</p>
US17	Iniciar sesión con servicios externos	Como usuario de AlquilaFácil, quiero poder iniciar sesión utilizando servicios externos como Google o Outlook, para ahorrar tiempo y evitar tener que crear una cuenta desde cero.	<p>Escenario 01: Inicio de sesión con Google</p> <p><b>Dado</b> que un usuario está en la pantalla de inicio de sesión de AlquilaFácil, <b>Cuando</b> hace clic en "Iniciar sesión con Google", <b>Entonces</b> el sistema redirige al usuario a la pantalla de autenticación de Google, donde ingresa sus credenciales y otorga permisos.</p> <p>Escenario 02: Inicio de sesión con Facebook</p> <p><b>Dado</b> que un usuario está en la pantalla de inicio de sesión de AlquilaFácil, <b>Cuando</b> hace clic en "Iniciar sesión con Facebook", <b>Entonces</b> el sistema redirige al usuario a la pantalla de autenticación de Google, donde ingresa sus credenciales y otorga permisos.</p>

<b>ID de la user story</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de aceptación</b>
US18	Acceder a información de ayuda	<p>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</p> <p><b>Cuando</b> lo necesite.</p>	<p>Escenario 01: Consultar preguntas frecuentes</p> <p><b>Dado</b> que un usuario tiene una duda común sobre el funcionamiento de la plataforma.</p> <p><b>Cuando</b> selecciona la opción de "Preguntas frecuentes" en el menú de ayuda. <b>Entonces</b> se muestra una lista de preguntas con sus respuestas para ayudar al usuario.</p> <p>Escenario 02: Acceder a información de contacto</p> <p><b>Dado</b> que un usuario necesita comunicarse con el soporte. <b>Cuando</b> selecciona la opción de "Información de contacto" en el menú de ayuda. <b>Entonces</b> se muestra la información de contacto, incluyendo el correo electrónico y número de teléfono del soporte.</p>
US19	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.	<ul style="list-style-type: none"> <li>- <b>Dado</b> que hay un evento activo, <b>cuando</b> se detecta que el número de personas se acerca al límite permitido, <b>entonces</b> se muestra una alerta en la app del arrendador.</li> <li>- <b>Dado</b> que se ha superado el aforo máximo permitido, <b>cuando</b> esto ocurre, <b>entonces</b> el sistema debe notificar al arrendador inmediatamente por notificación push <b>y</b> correo electrónico.</li> </ul>
US20	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.	<ul style="list-style-type: none"> <li>- <b>Dado</b> que se ha configurado un umbral de ruido, <b>cuando</b> los decibeles exceden ese límite, <b>entonces</b> el sistema envía una notificación al arrendador.</li> <li>- <b>Dado</b> que el evento sigue en curso, <b>cuando</b> los niveles de ruido vuelven a la normalidad, <b>entonces</b> el sistema registra automáticamente el tiempo total del exceso.</li> </ul>

<b>ID de la user story</b>	<b>Título</b>	<b>Descripción</b>	<b>Criterios de aceptación</b>
US21	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.	<ul style="list-style-type: none"> <li>- <b>Dado</b> que hay sensores de humo activos, <b>cuando</b> se detecta presencia de humo, <b>entonces</b> se envía una notificación crítica al arrendador.</li> <li>- <b>Dado</b> que se ha enviado una alerta de humo, <b>cuando</b> el usuario accede a la app, <b>entonces</b> debe visualizar el evento en un panel de incidentes con la hora exacta de detección.</li> </ul>
US22	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.	<ul style="list-style-type: none"> <li>- <b>Dado</b> que se ha definido una zona como restringida, <b>cuando</b> se detecta movimiento en esa área, <b>entonces</b> el arrendador recibe una alerta inmediata con la ubicación del incidente.</li> <li>- <b>Dado</b> que ocurre una intrusión, <b>cuando</b> el evento es registrado, <b>entonces</b> debe quedar un historial con fecha, hora y duración del movimiento.</li> </ul>
US23	Registro de Incidentes	Como arrendador, quiero acceder a un historial con todos los eventos registrados por sensores, para tener evidencia y mejorar la gestión.	<ul style="list-style-type: none"> <li>- <b>Dado</b> que han ocurrido incidentes durante un evento, <b>cuando</b> el arrendador accede al panel de historial, <b>entonces</b> puede ver un listado con fecha, tipo de incidente y duración.</li> <li>- <b>Dado</b> que se selecciona un evento del historial, <b>cuando</b> el usuario lo abre, <b>entonces</b> debe poder ver detalles como gráfico de actividad, sensores involucrados y ubicación dentro del local.</li> </ul>

### 3.3 Impact Mapping

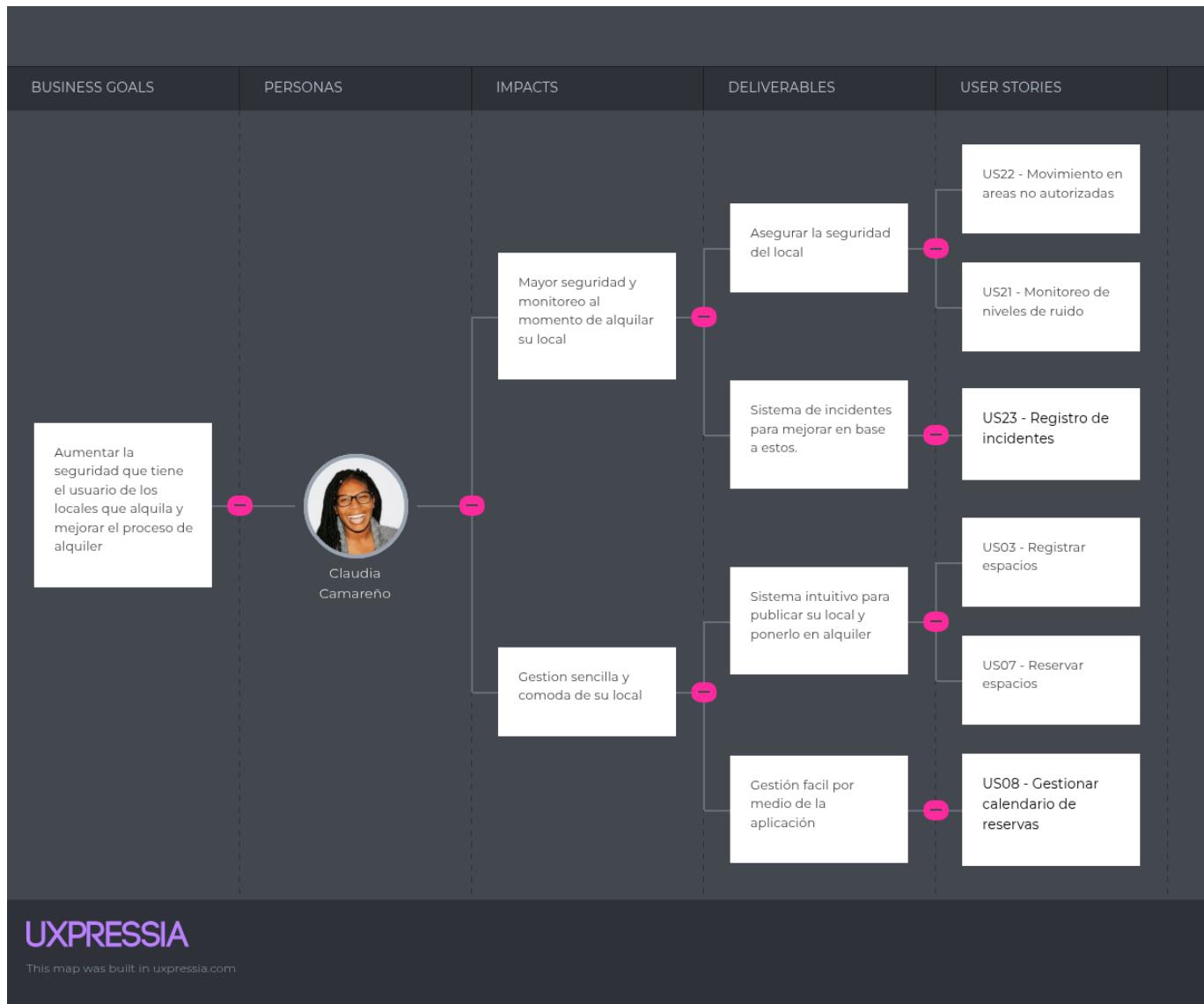
#### Arrendatarios:



**UXPRESSIA**

This map was built in [uxpressia.com](https://uxpressia.com)

## Arrendadores:



### 3.4 Product Backlog

User Stories	Título	Descripción	Story Points (1/2/3/5/8)
ID			
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

User Stories	Título	Descripción	Story Points (1/2/3/5/8)
ID			
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
US19	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
US20	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
US21	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
US22	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
US23	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

User Stories	Título	Descripción	Story Points
ID			(1/2/3/5/8)
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
US17	Permitir iniciar sesión con servicios externos	Como usuario de AlquilaFácil, quiero poder iniciar sesión utilizando servicios externos como Google o Outlook, para ahorrar tiempo y evitar tener que crear una cuenta desde cero.	2
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
US18	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
US24	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

User	Stories	Título	Descripción	Story Points
ID				(1/2/3/5/8)
US25	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
US26	Conocer los beneficios diferenciados		Como visitante, quiero ver tarjetas explicativas con los beneficios que ofrece AlquilaFácil tanto para arrendadores como arrendatarios.	5

Link de trello: <https://trello.com/b/63Zub4fa/product-backlog-iot>

## Capítulo IV: Solution Software Design

### 4.1. Strategic-Level Domain-Driven Design

#### 4.1.1. Event Storming

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](https://miro.com/app/board/uXjVIBJ9674=?share_link_id=539583810560)

#### 4.1.1.1 Candidate Context Discovery

##### Step 1: Unstructured Exploration:

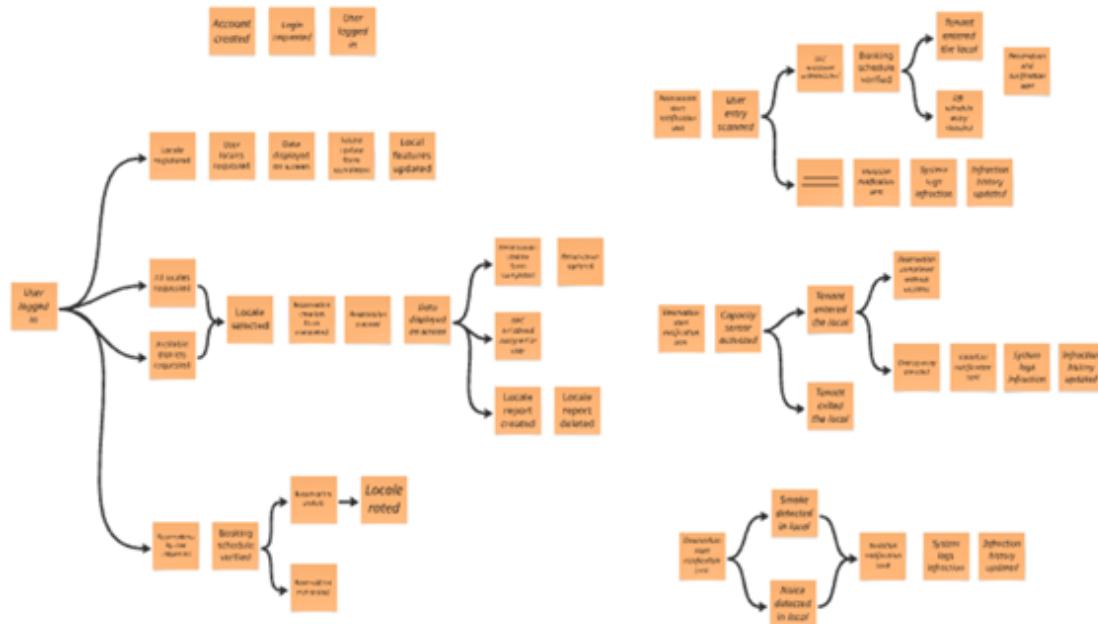
En la etapa inicial del EventStorming, se lleva a cabo una sesión de lluvia de ideas destinada a descubrir los eventos clave del dominio vinculados al negocio analizado. Es importante redactar estos eventos utilizando verbos en pasado, ya que deben reflejar acciones que ya han sucedido dentro del sistema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

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

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:

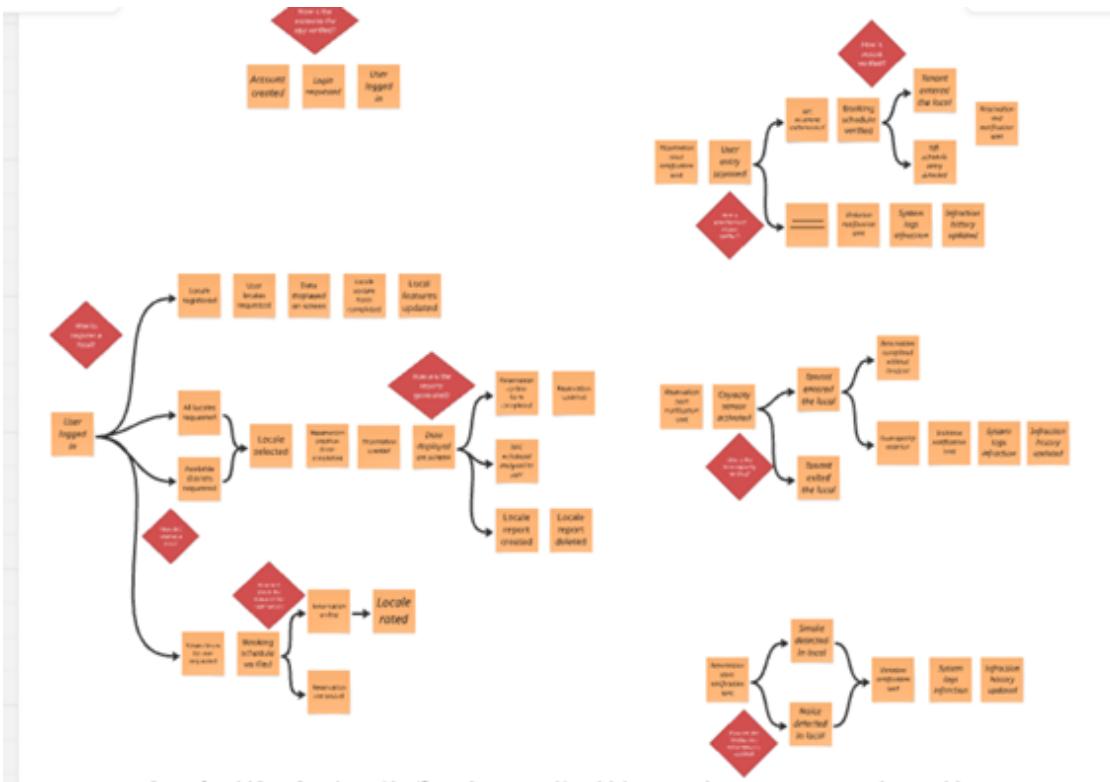
En esta segunda etapa, se toman los eventos de dominio identificados y se ordenan cronológicamente según su secuencia natural dentro del proceso. El primer enfoque consiste en construir un "camino feliz" (happy path), es decir, una secuencia ideal en la que todas las acciones se ejecutan correctamente 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:**

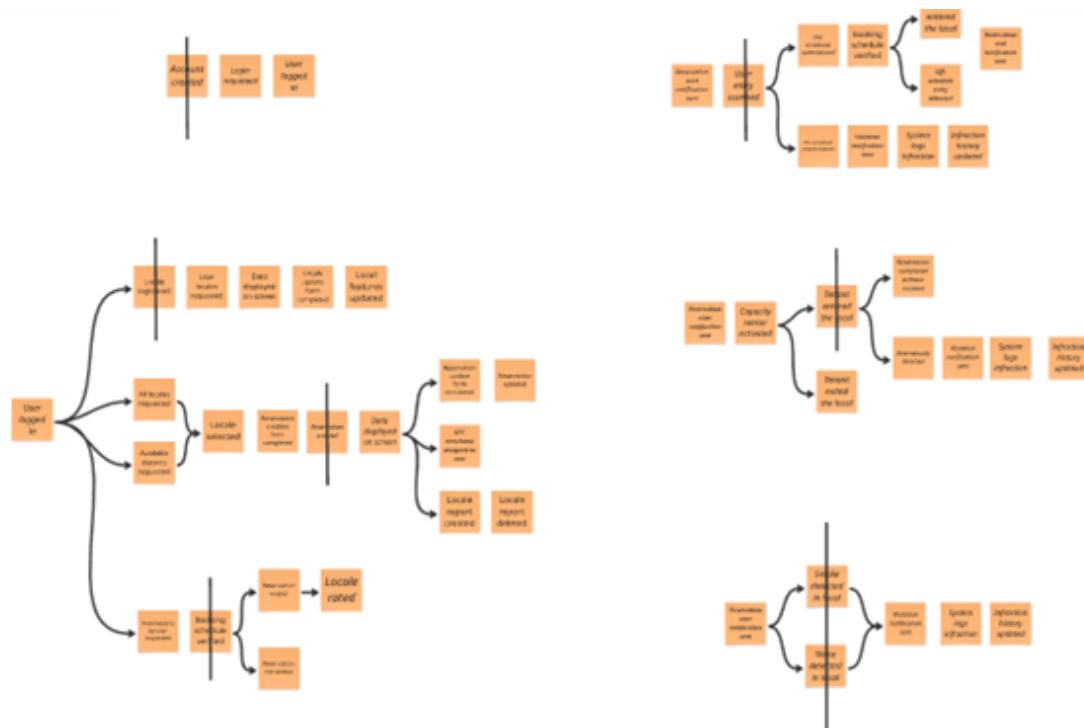
Una vez que los eventos fueron organizados en una secuencia temporal, utilizamos esta visión global del proceso para detectar áreas clave que requieren atención. Estos puntos críticos pueden manifestarse como cuellos de botella operativos, tareas manuales susceptibles de automatización, ausencia de documentación 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:

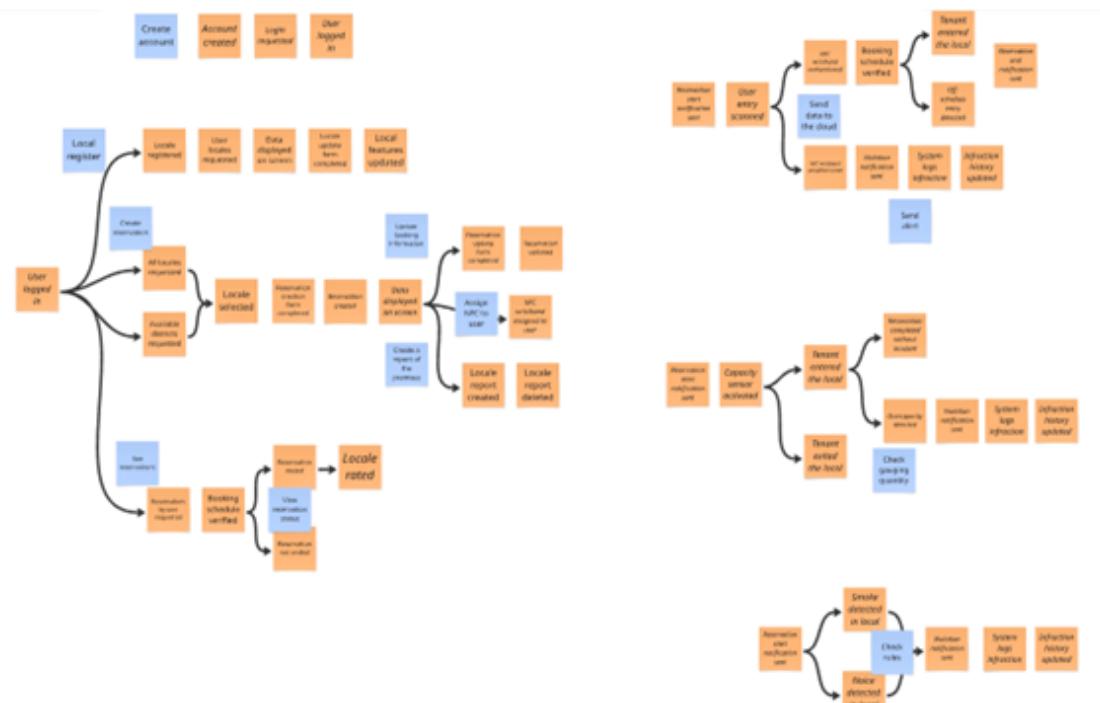
Una vez que se ha construido la línea de eventos completa, incluyendo los puntos de dolor, se procede a identificar aquellos eventos relevantes que representan un cambio significativo en el contexto o en la etapa del proceso. Estos se conocen como eventos principales, y se utilizan como puntos de corte 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.

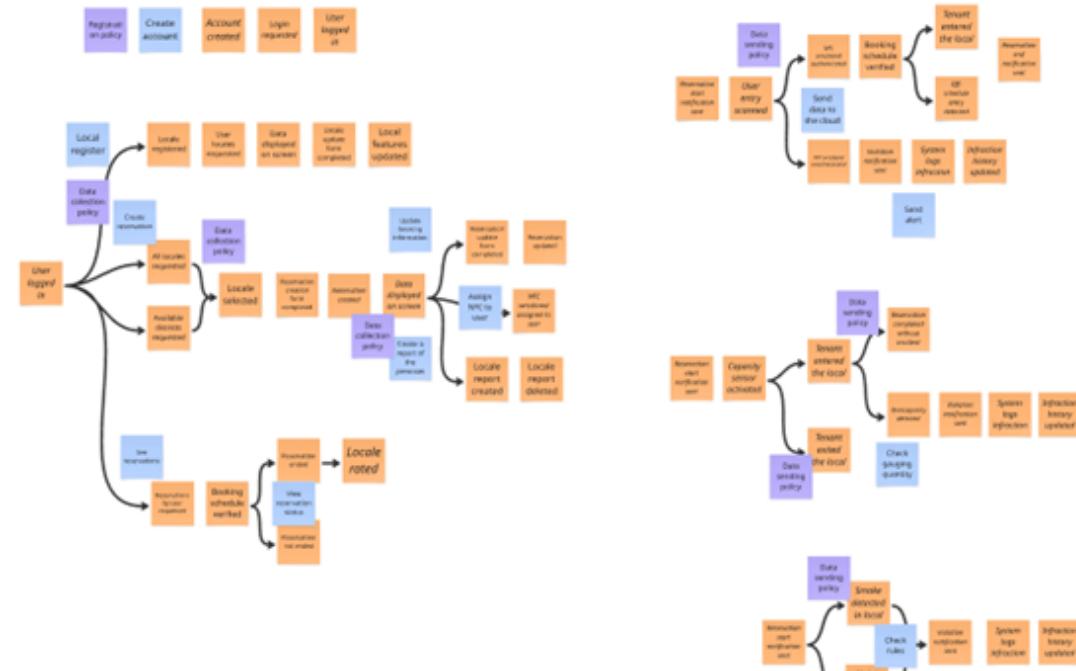
## **Step 5: Commands:**

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:**

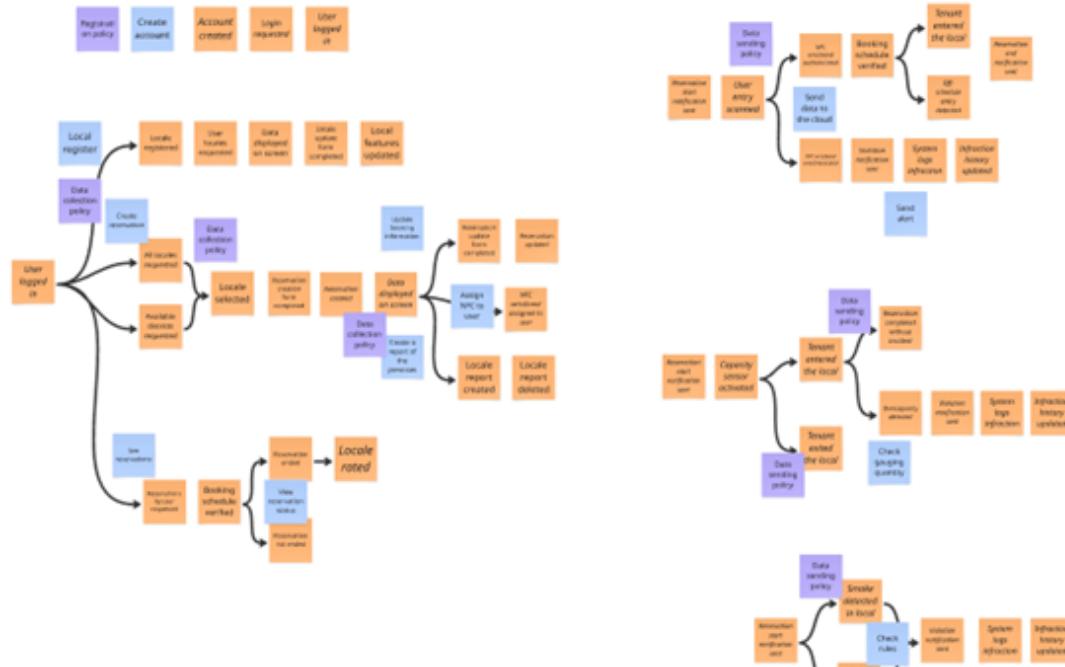
En esta etapa se identifican las políticas de automatización, es decir, reglas que permiten que ciertos comandos se ejecuten automáticamente en respuesta a eventos del dominio. Esto implica que, al producirse un evento específico dentro del sistema, se dispare de forma inmediata la acción correspondiente, 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.

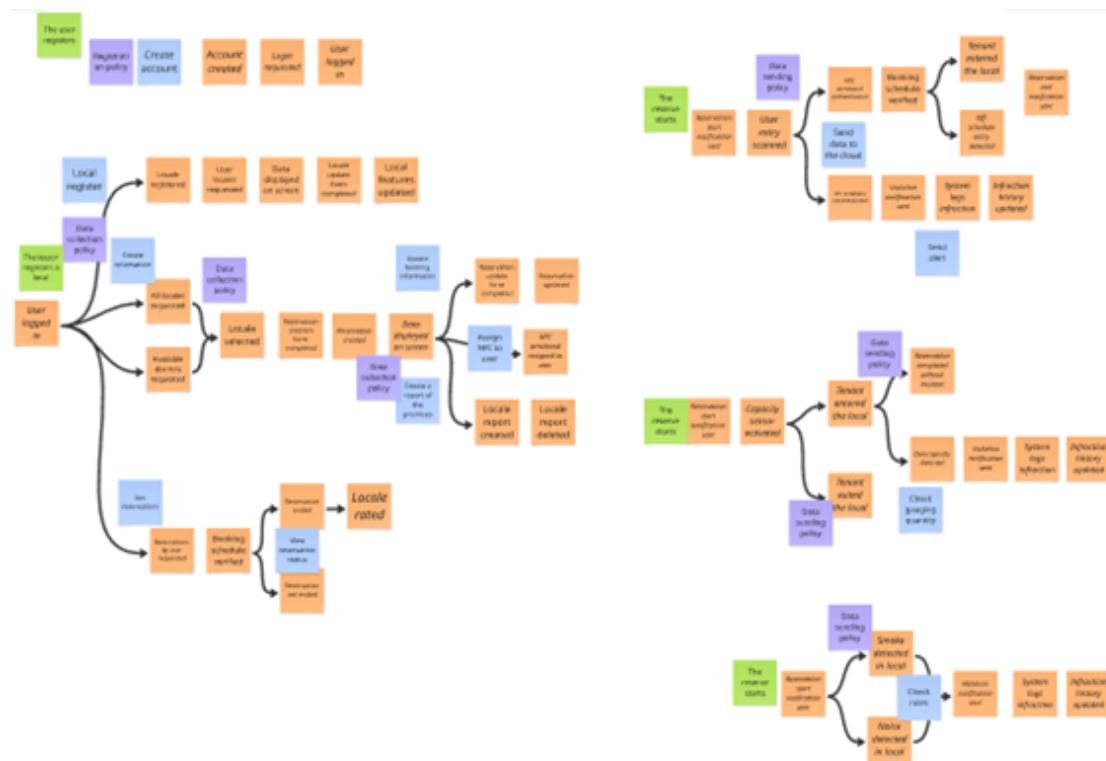
## Step 7: Read models:

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:

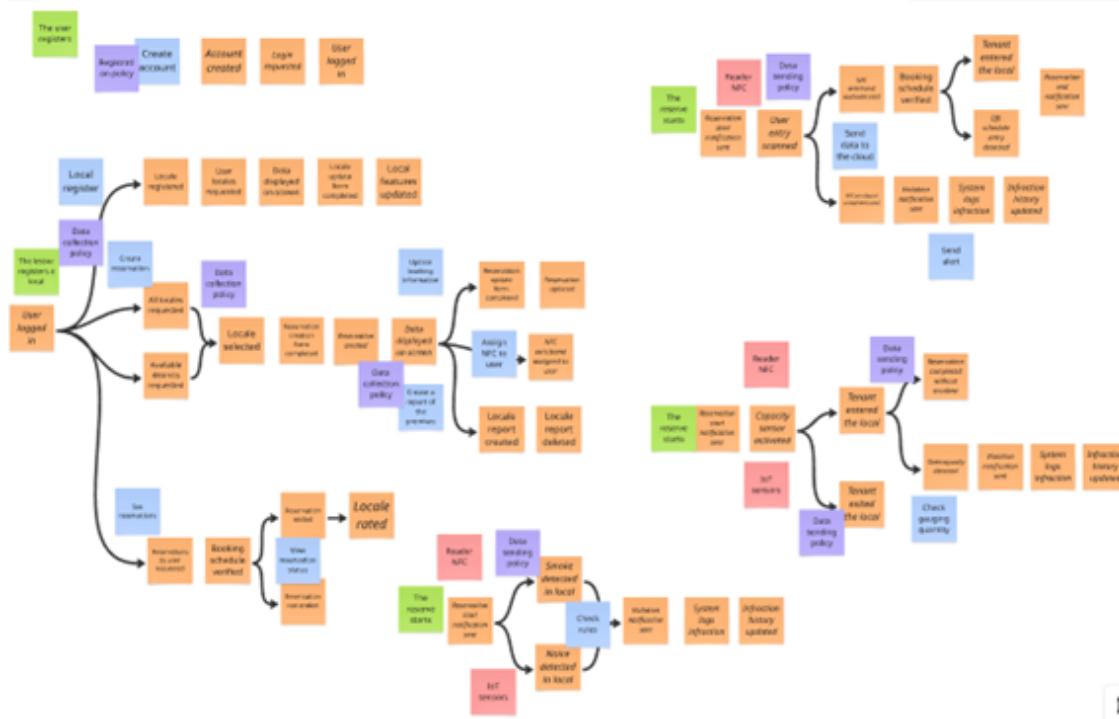
En este paso, integramos al modelo los sistemas externos, es decir, aquellos que están fuera del dominio principal del sistema que estamos diseñando. Estos sistemas pueden intervenir ejecutando comandos hacia nuestro sistema (como entradas) o bien recibir información generada por eventos del dominio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

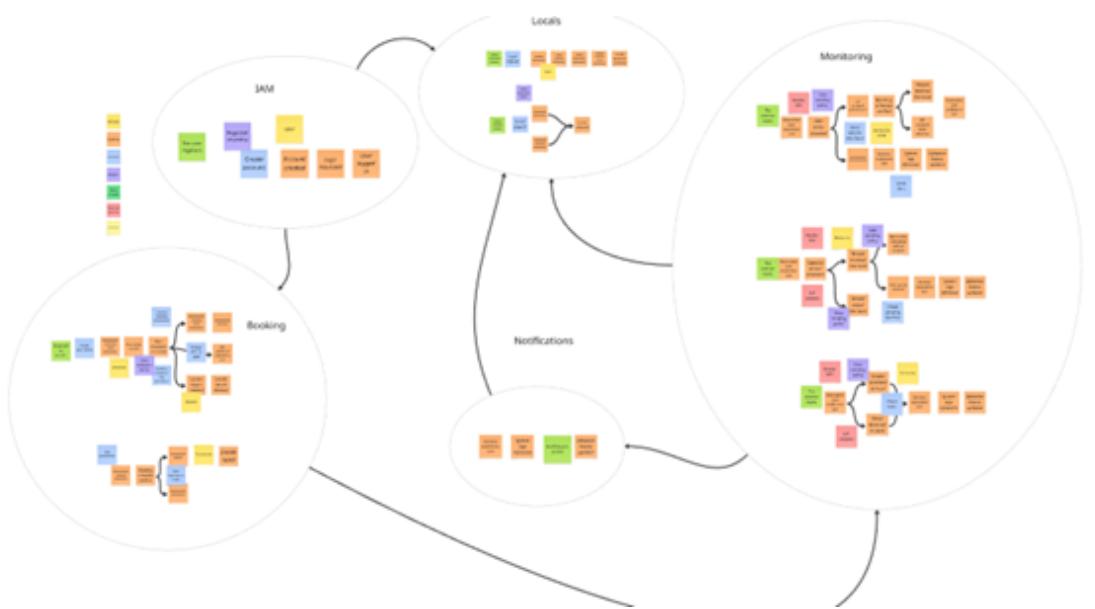
Después de haber definido todos los eventos y comandos del sistema, procedemos a agrupar aquellos conceptos que guardan una relación directa dentro de unidades llamadas agregados. Estos agregados representan componentes clave del dominio, ya que son los encargados de recibir comandos y, a partir de ellos,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

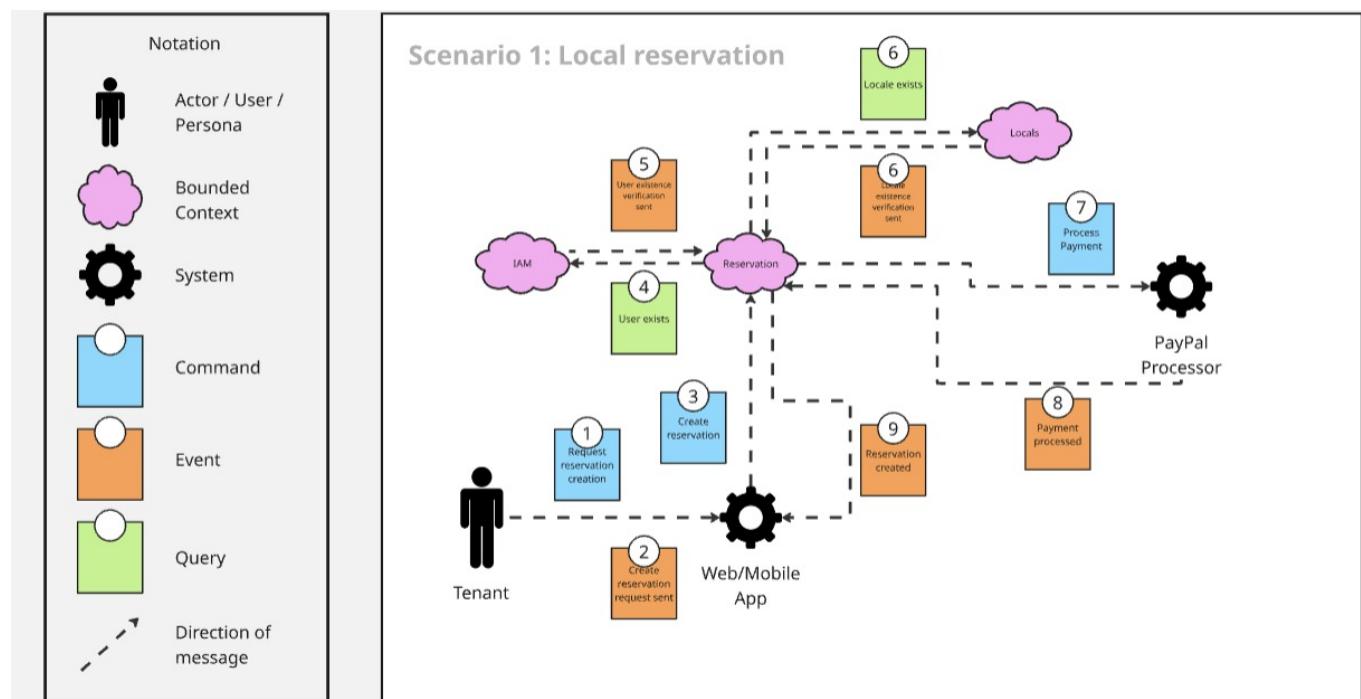
Finalmente, agrupamos los agregados que mantienen una relación directa entre sí, ya sea porque comparten funciones estrechamente relacionadas o porque están conectados a través de ciertas políticas de negocio. Estos conjuntos de agregados constituyen candidatos naturales para establecer los contextos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



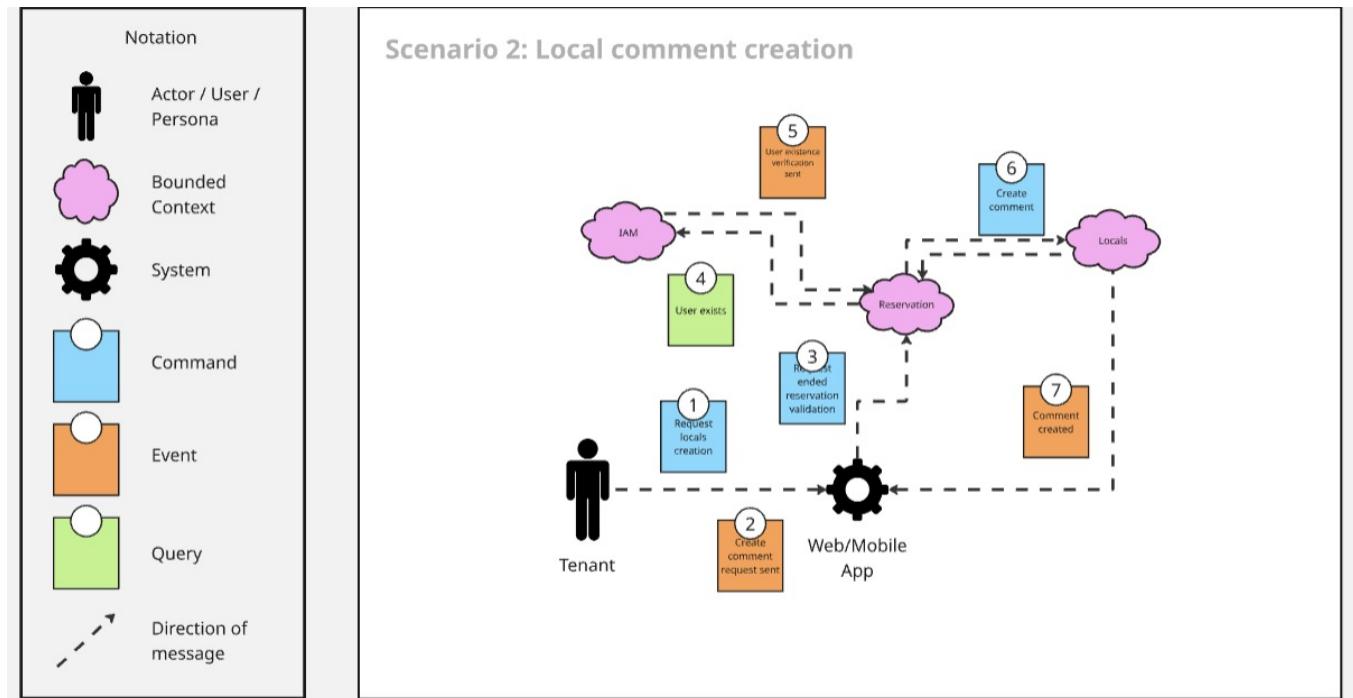
En esta etapa final, estructuramos el sistema en Bounded Contexts claramente definidos, agrupando los agregados que comparten responsabilidades, reglas de negocio o interacciones comunes. Esta segmentación permite organizar el dominio en módulos independientes, cada uno con un propósito específico y 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.

#### 4.1.1.2 Domain Message Flows Modeling

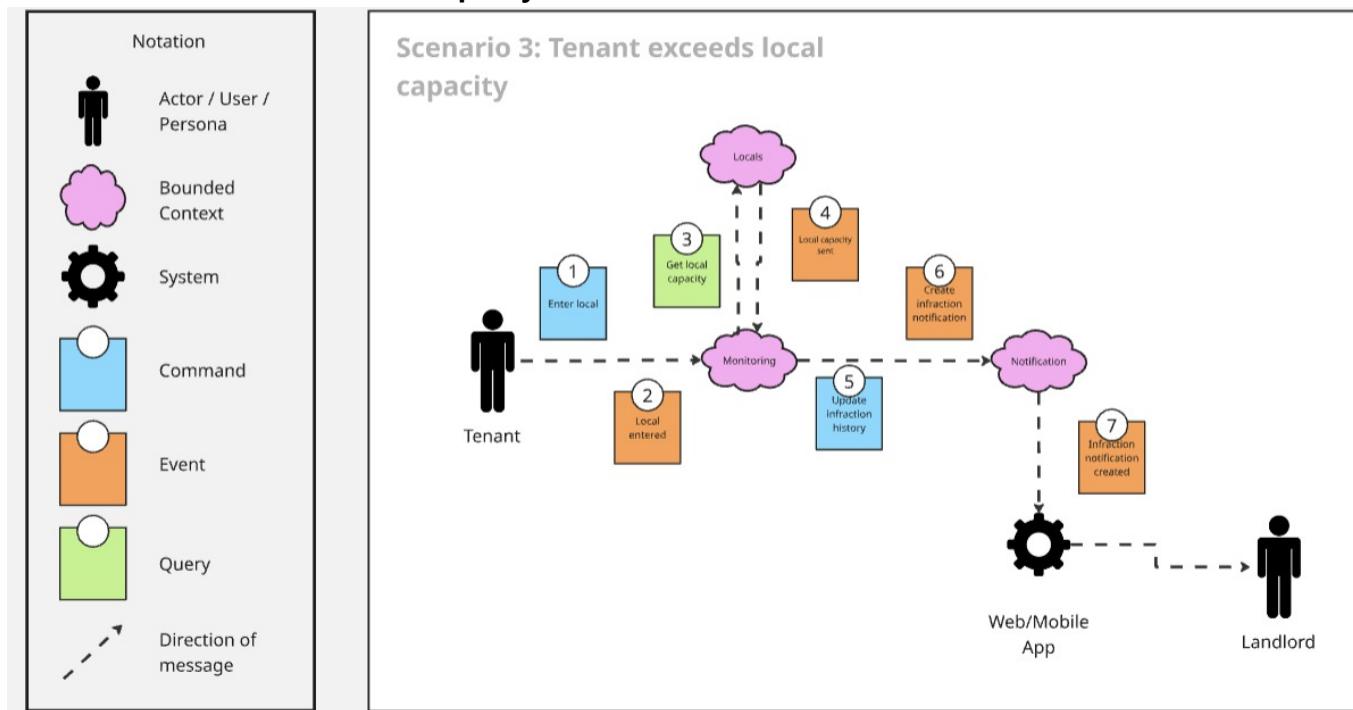
##### Scenario 1: Local reservation:



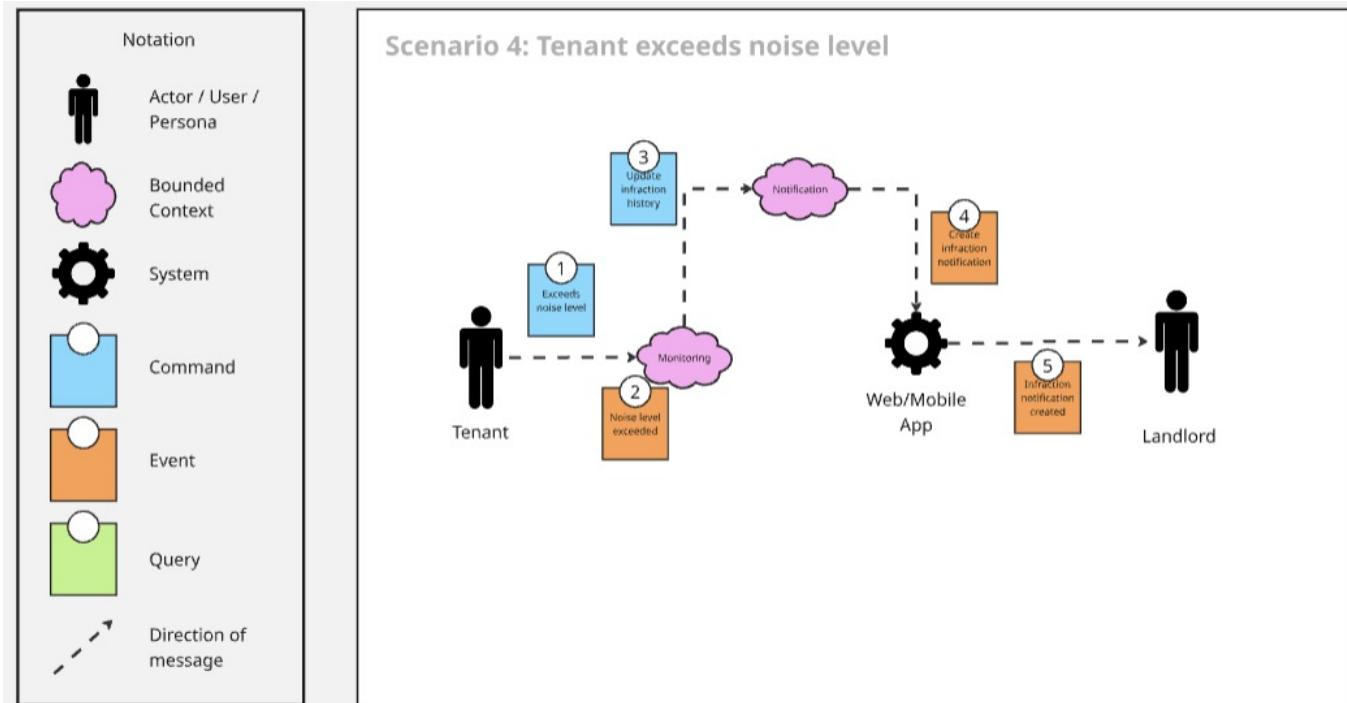
## Scenario 2: Local comment creation:



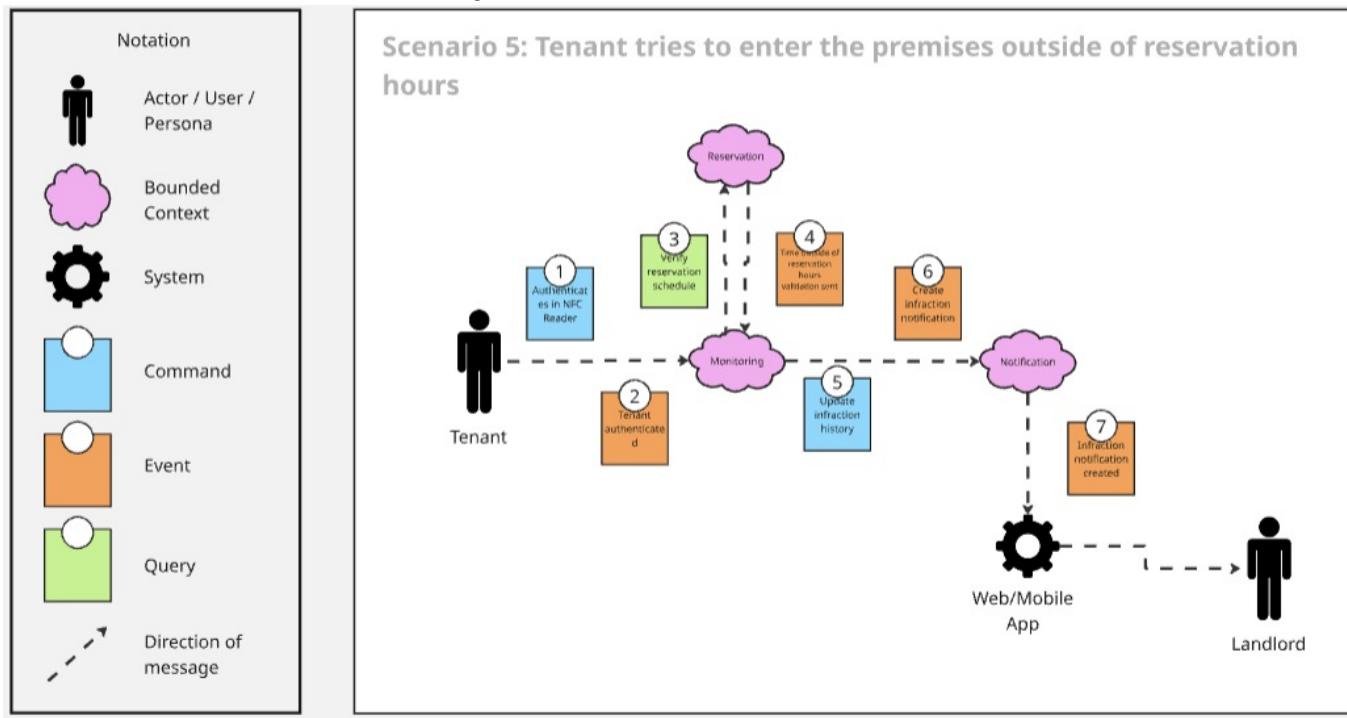
## Scenario 3: Tenant exceeds local capacity:



#### Scenario 4: Tenant exceeds noise level:

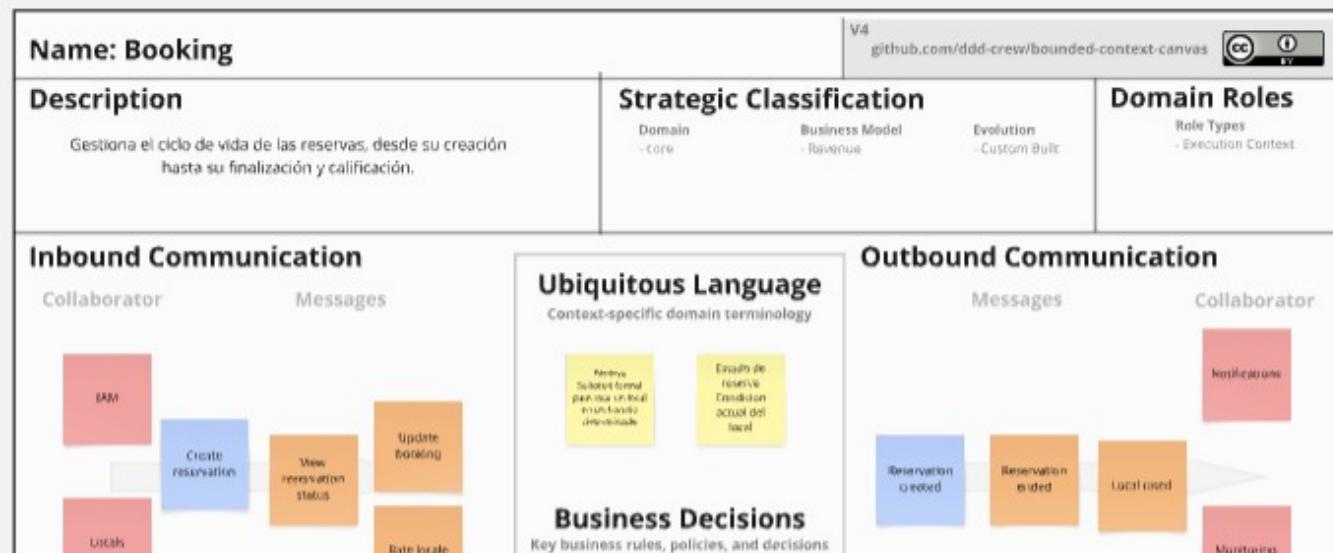
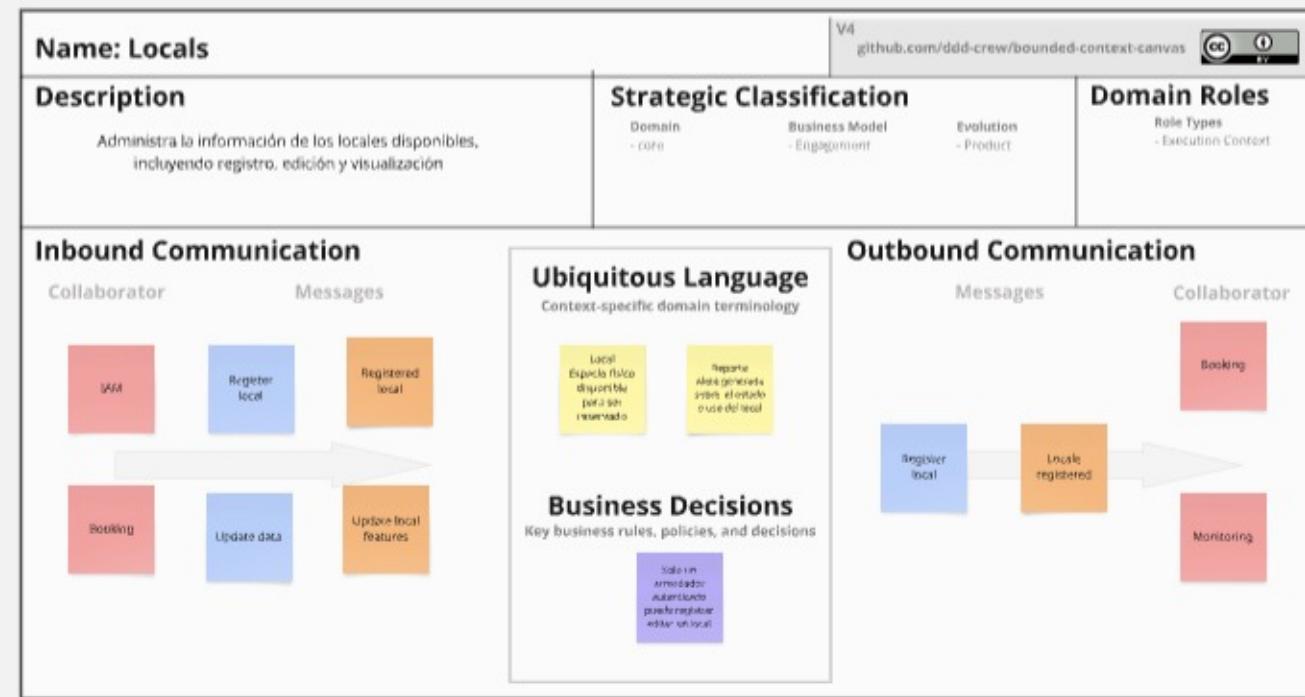
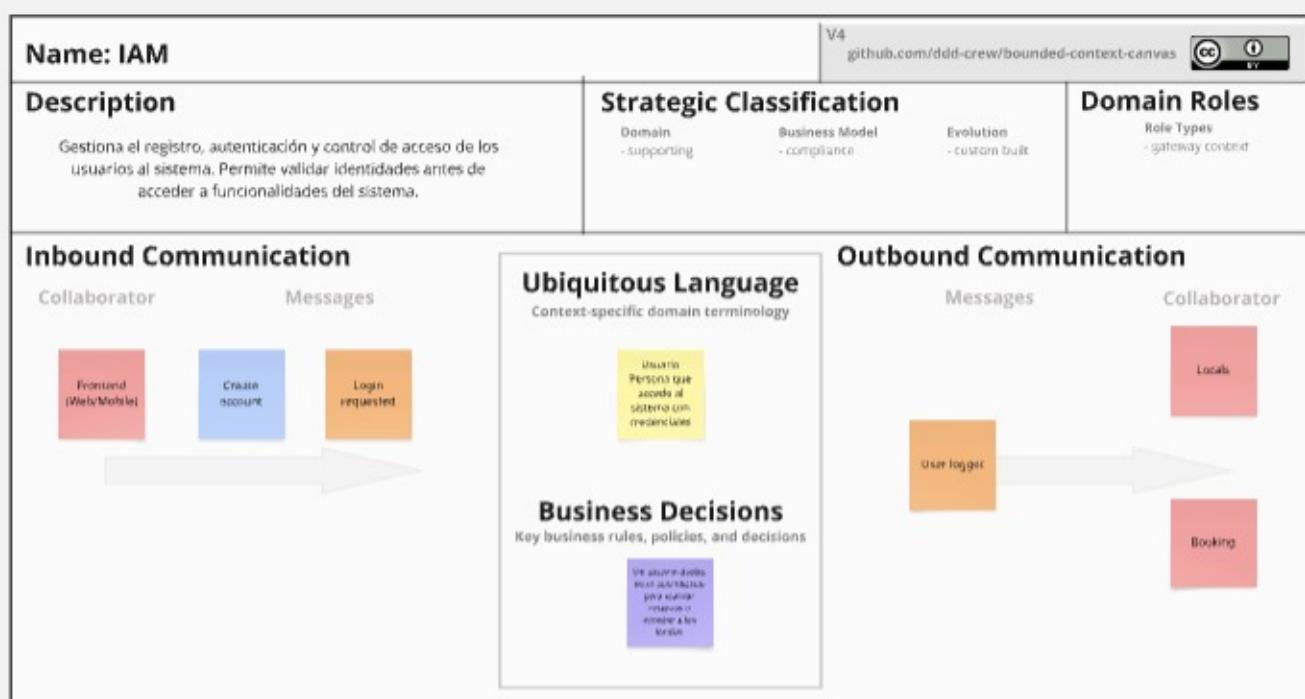


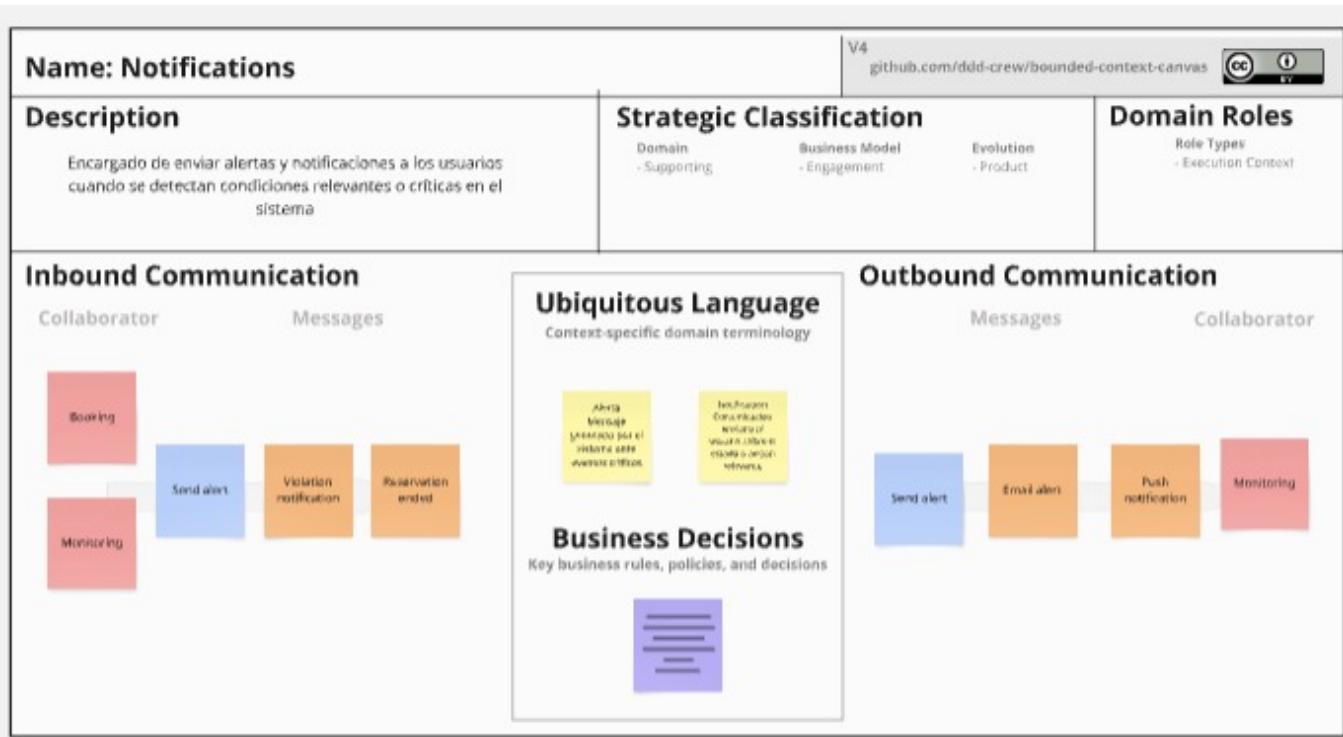
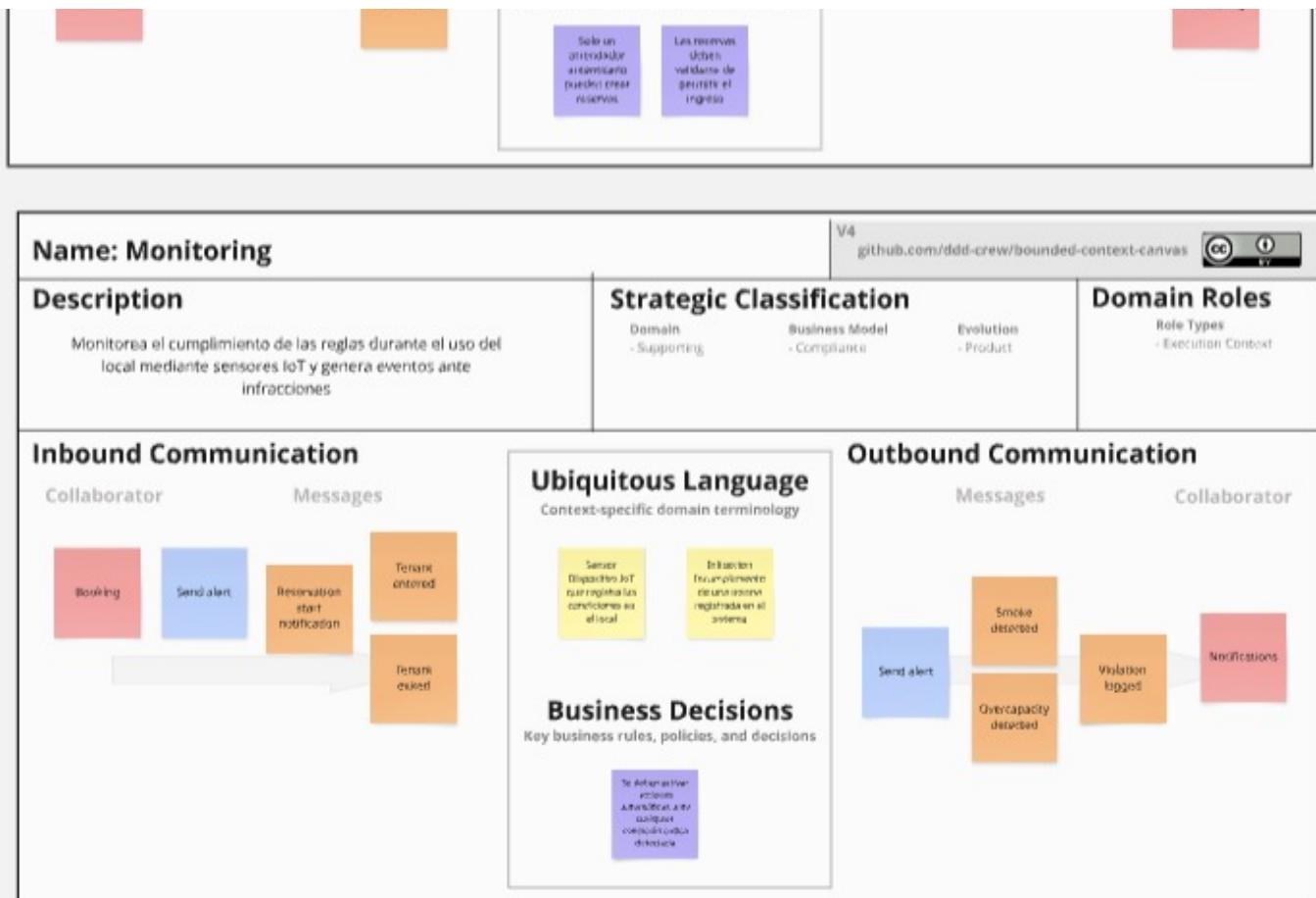
#### Scenario 5: Tenant tries to enter the premises outside of reservation hours :



#### 4.1.1.3 Bounded Context Canvases







## 4.1.2. 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?

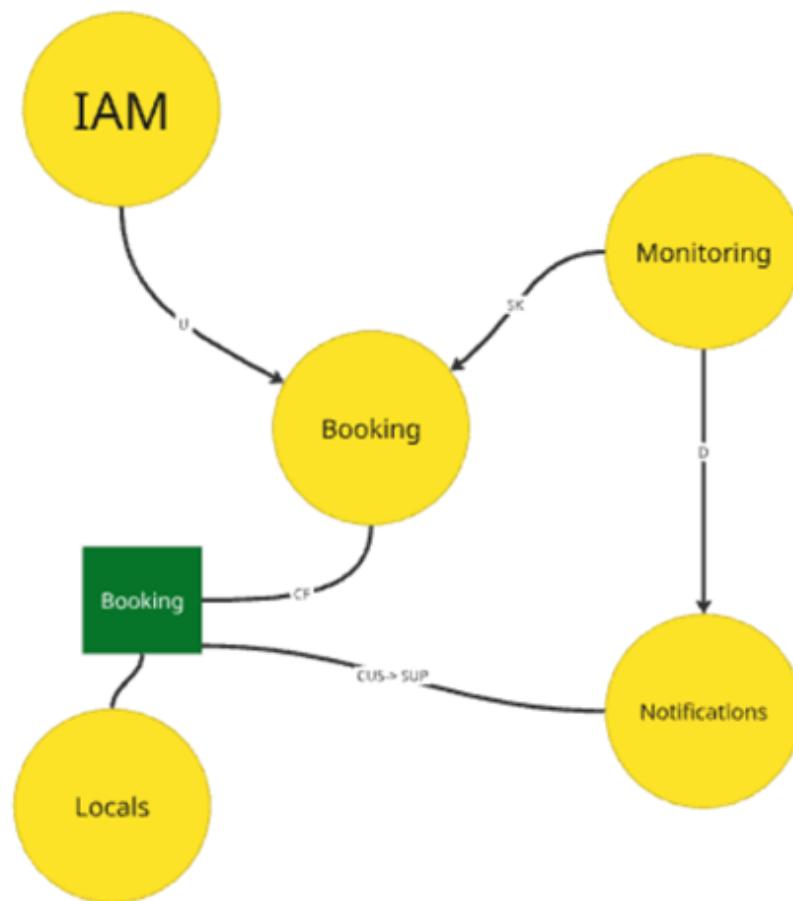
Impacto: IAM controlaría identidad + dispositivo asignado. Aumenta la carga de IAM, mezcla responsabilidades de autenticación con gestión de dispositivos. Discusión: Se recomienda mantener la asignación en Booking para no contaminar IAM con lógica operacional.

### 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.

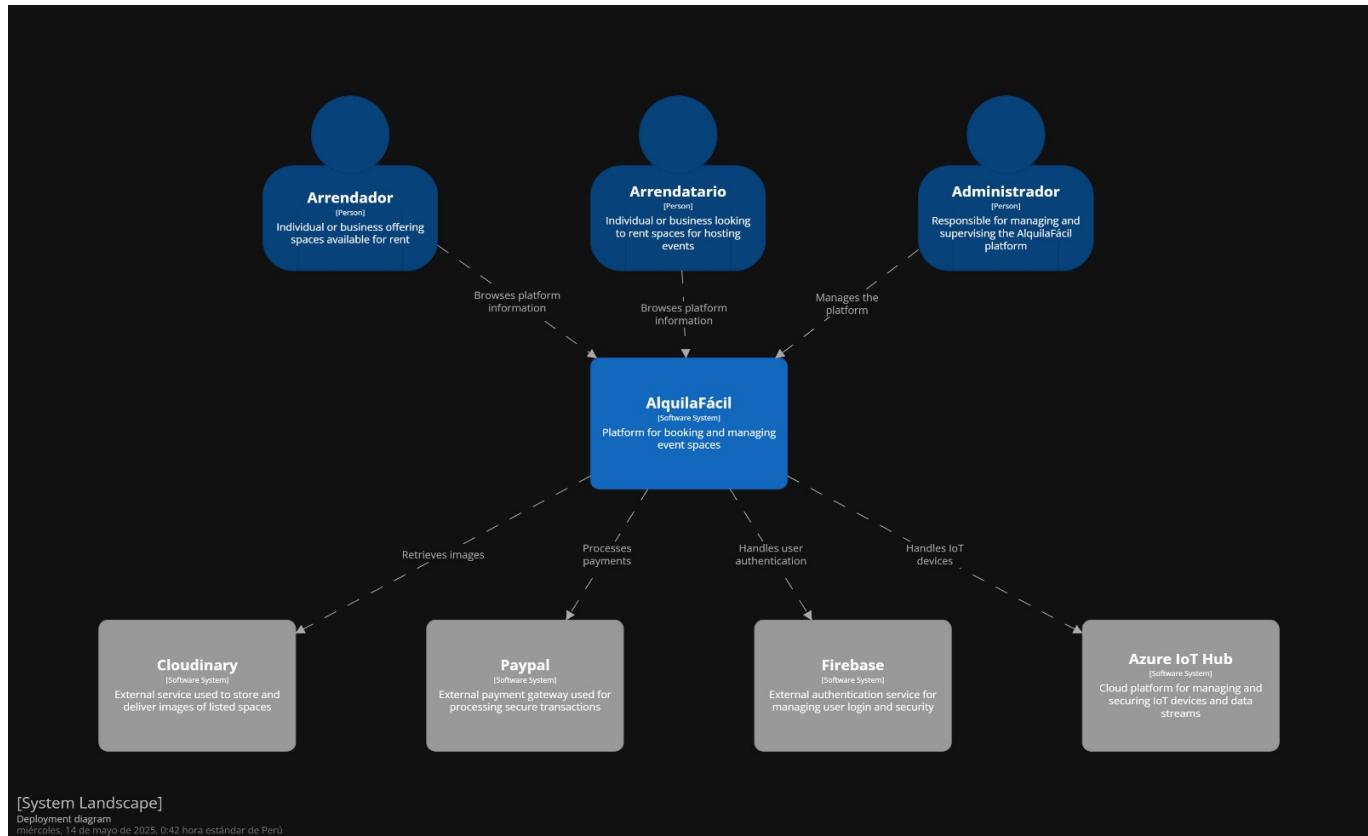
### 4. Patrones de Relaciones Sugeridos

- 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

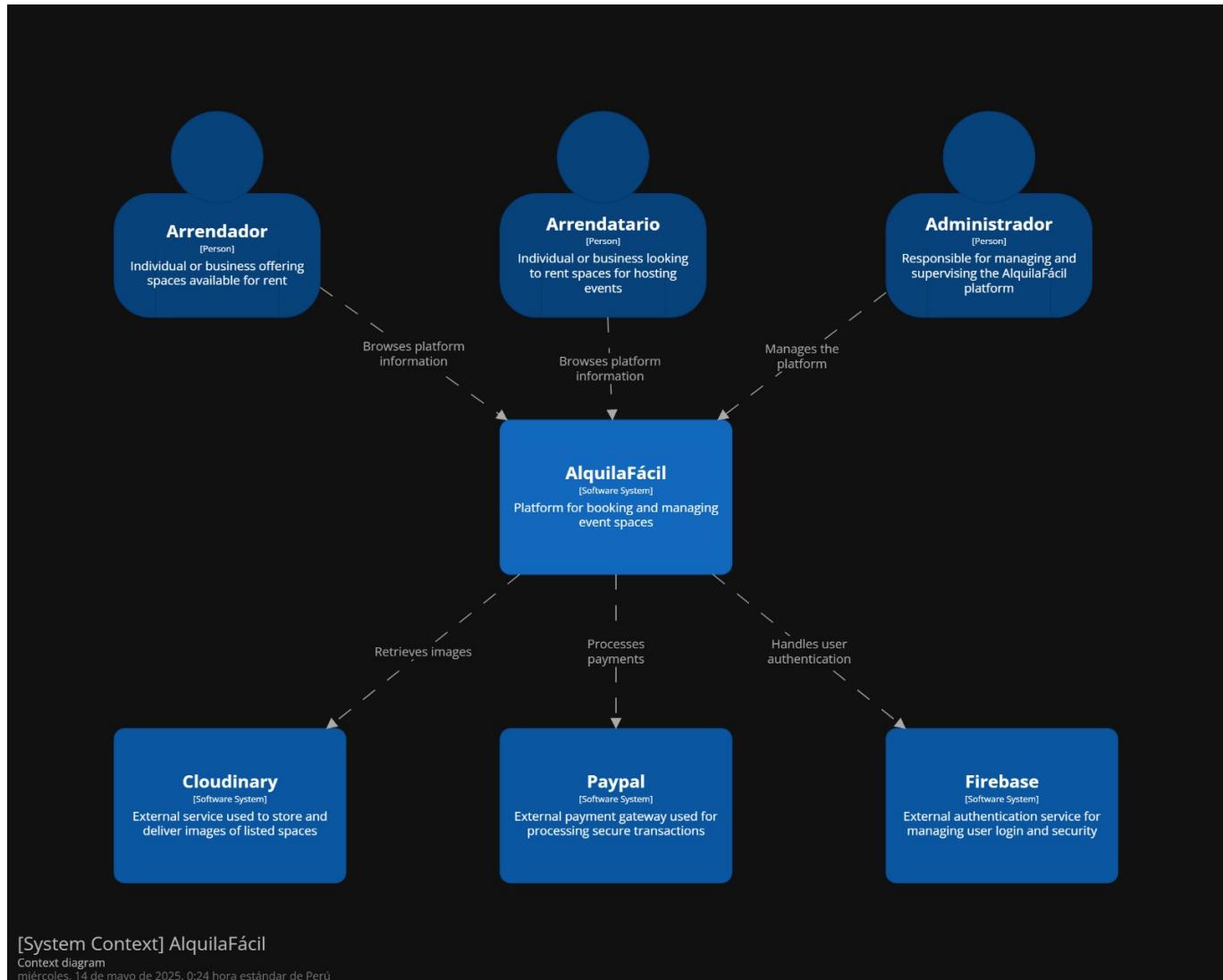


#### 4.1.3. Software Architecture

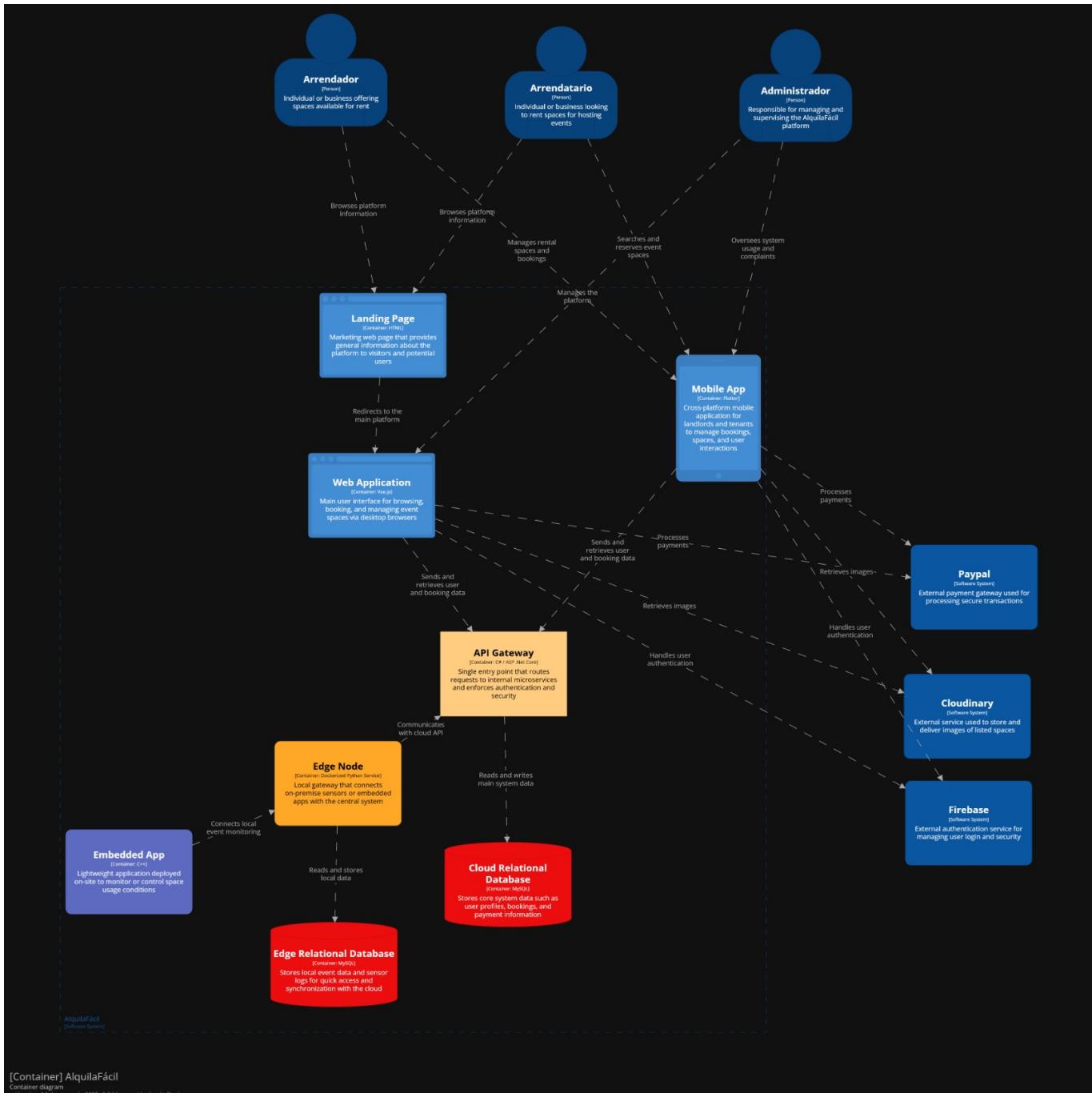
##### 4.1.3.1. Software Architecture System Landscape Diagram



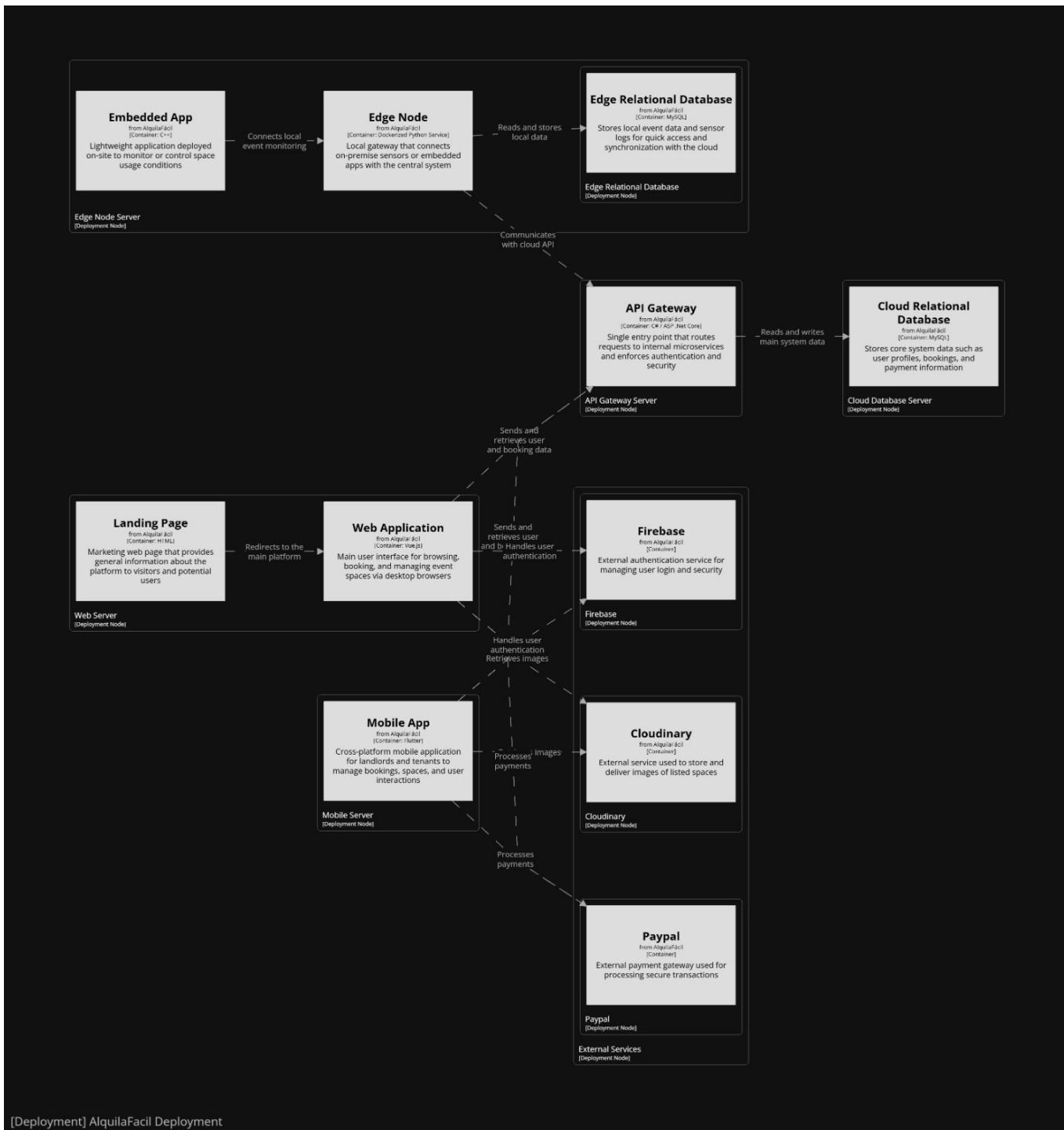
#### 4.1.3.2. Software Architecture Context Level Diagrams



#### 4.1.3.3. Software Architecture Container Level Diagrams



#### 4.1.3.4. Software Architecture Deployment Diagrams



## 4.2.1. Bounded Context: IAM Context

### 4.2.1.1. Domain Layer

#### Aggregates

##### 1. User

- **Descripción:** Representa un usuario en la aplicación.
- **Atributos:**
  - **Id:** Identificador único para el usuario.
  - **Username:** Nombre del usuario.
  - **Email:** Correo electrónico del usuario.
  - **RoleId:** El rol de acceso del usuario.

## Entities

### 1. UserRole

- **Descripción:** Representa el rol de acceso que puede tener un usuario (Tester, admin, usuario)
- **Atributos:**
  - **Id:** Identificador unico para el rol
  - **Role:** Nombre del rol

## Value Objects

### 1. EUserRoles

- **Descripción:** Enumerable que contiene todos los tipos de roles que puede tener un usuario.

## Commands

### 1. SeedUserRolesCommand:

- **Descripción:** Genera los roles de acceso para poder ser usados.

### 2. SignInCommand:

- **Descripción:** Inicio de sesión de un usuario ya registrado.

### 3. SignUpCommand:

- **Descripción:** Registro de un nuevo usuario en la aplicación.

### 4. UpdateUsernameCommand:

- **Descripción:** Permite el cambio del nombre de un usuario.

## Queries

### 1. GetAllUsersQuery:

- **Descripción:** Obtener todos los usuarios registrados.

### 2. GetUserByEmailQuery:

- **Descripción:** Obtener un usuario en base a su correo electronico.

### 3. GetUserByIdQuery:

- **Descripción:** Obtener un usuario en base a su Id.

### 4. GetUsernameByIdQuery:

- **Descripción:** Obtener el nombre de un usuario en base a su Id.

### 5. UserExistsQuery:

- **Descripción:** Verificar si el usuario ya existe.

## Repositories

## 1. UserRepository:

- **Descripción:** Interfaz que permite interactuar con la base de datos de usuarios.
- **Métodos:**
  - `FindByEmailAsync(string email)`: Devuelve un usuario en base al email proporcionado
  - `ExistByUsername(string username)`: Devuelve un valor positivo si el nombre de usuario se encuentra en otro usuario
  - `GetUsernameByIdAsync(int userId)`: Devuelve el nombre de usuario con en base a un id proporcionado
  - `ExistsById(int userId)`: Devuelve un valor positivo si hay un usuario con el id proporcionado

## 2. UserRoleRepository:

- **Descripción:** Interfaz que permite interactuar con la base de datos de roles de usuario
- **Métodos:**
  - `ExistsUserRole(EUserRoles role)`: Devuelve positivo si el rol de usuario ya existe.

### 4.2.1.2. Interface Layer

#### Facades

## 1. IamContextFacade:

- **Descripción:** Permite que otros bounded context accedan a este contexto a traves del patrón facade
- **Métodos:**
  - `CreateUser(string username, string password, string email)`: Permite crear un usuario desde fuera del bounded context.
  - `FetchUserIdByUsername(string username)`: Permite extraer el id de un usuario a traves de su nombre de usuario.
  - `FetchUsernameById(int userId)`: Devuelve un nombre de usuario en base a un id proporcionado

#### Controllers

## 1. AuthenticationController:

- **Descripción:** Expone endpoints para autenticación de usuarios.
- **Métodos:**
  - `SignIn(SignInResource signInResource)`: Endpoint para el inicio de sesión.
  - `SignUp(SignUpResource signUpResource)`: Endpoint para el registro de un nuevo usuario.

## 2. UsersController:

- **Descripción:** Expone endpoints para la gestión de usuarios
- **Métodos:**
  - `GetUserById(int userId)`: Endpoint que permite obtener un usuario a traves de su id
  - `GetAllUsers()`: Endpoint que permite obtener a todos los usuarios.

- `GetUserNameById`: Endpoint que permite obtener un nombre de usuario a través del id del usuario.
- `UpdateUser(int userId, UpdateUsernameResource updateUsernameResource)`: Endpoint que permite actualizar el nombre de usuario de un usuario

#### 4.2.1.3. Application Layer

##### Command Services

###### 1. SeedUserRoleCommandService:

- **Descripción:** Inicializa la tabla de roles de usuarios con los valores preestablecidos
- **Métodos:**
  - `Handle(SeedUserRolesCommand command)`: Valida y aplica el comando para generar los roles de usuario

###### 2. UserCommandService:

- **Descripción:** Maneja comandos para crear, actualizar o validar usuarios.
- **Métodos:**
  - `Handle(SignInCommand command)`: Valida y aplica el comando para iniciar la sesión del usuario.
  - `Handle(SignUpCommand command)`: Valida y aplica el comando para registrar un nuevo usuario.
  - `Handle(UpdateUsernameCommand command)`: Valida y aplica el comando para cambiar el nombre a un usuario existente.

##### Query Services

###### 1. UserQueryService:

- **Descripción:** Ofrece consultas para obtener información de los usuarios.
- **Métodos:**
  - `Handle(GetUserByIdQuery query)`: Valida y aplica la consulta para obtener un usuario dado su id.
  - `Handle(GetAllUsersQuery query)`: Valida y aplica la consulta para obtener todos los usuarios.
  - `Handle(GetUserByEmailQuery)`: Valida y aplica la consulta para obtener un usuario dado su email.
  - `Handle(GetUsernameByIdQuery)`: Valida y aplica la consulta para obtener el nombre de usuario dado el id de un usuario.
  - `Handle(UserExistsQuery)`: Valida y aplica la consulta para verificar que un usuario exista.

##### Outbound Services

###### 1. Hashing Service:

- **Descripción:** Maneja métodos para encriptar contraseñas y verificarlas.
- **Métodos:**
  - `HashPassword(string password)`: Permite hashear o encriptar una contraseña

- `VerifyPassword(string password, string passwordHash)`: Permite validar una contraseña hasheada o encriptada con una original

## 2. Token Service:

- **Descripción:** Maneja los tokens para poder manejar la sesión de los usuarios.
- **Métodos:**
  - `GenerateToken(User user)`: Permite generar un token único asociado a un usuario.
  - `ValidateToken(string token)`: Permite validar el token de un usuario.

### 4.2.1.4. Infrastructure Layer

#### Repositories (Implementación)

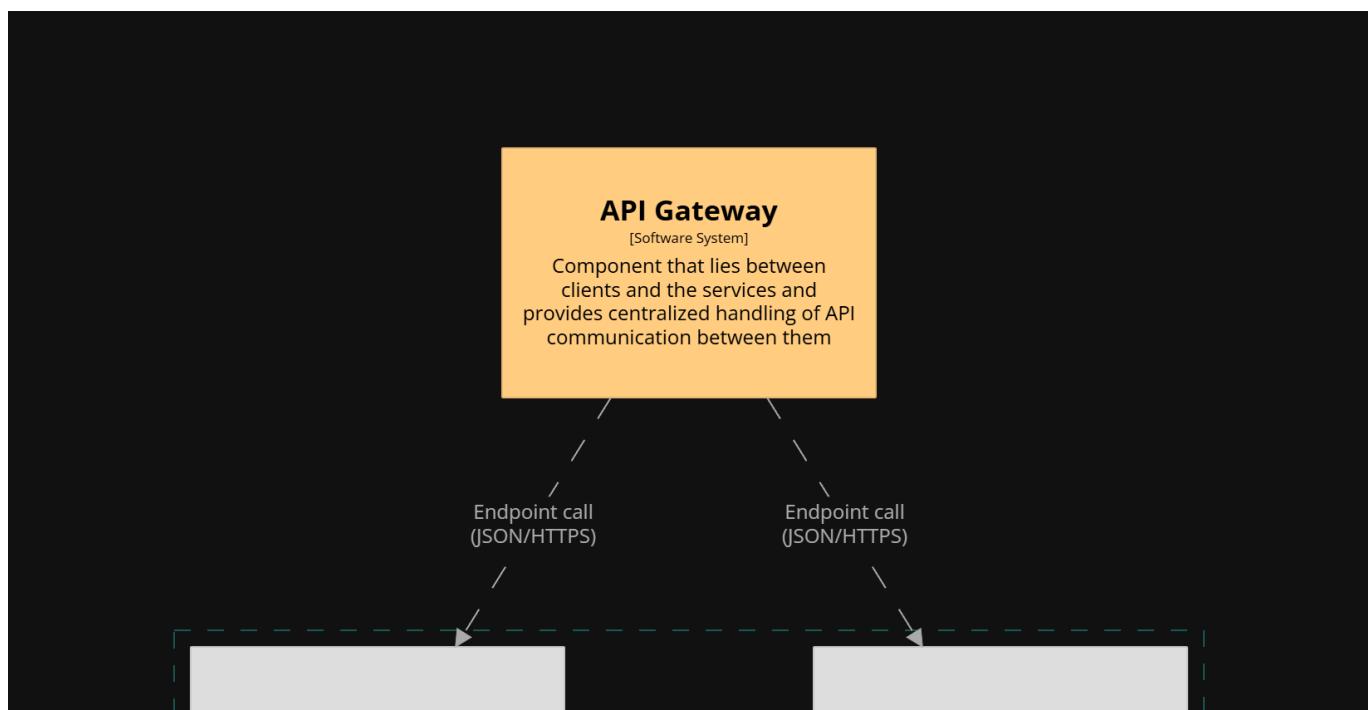
##### 1. UserRepository:

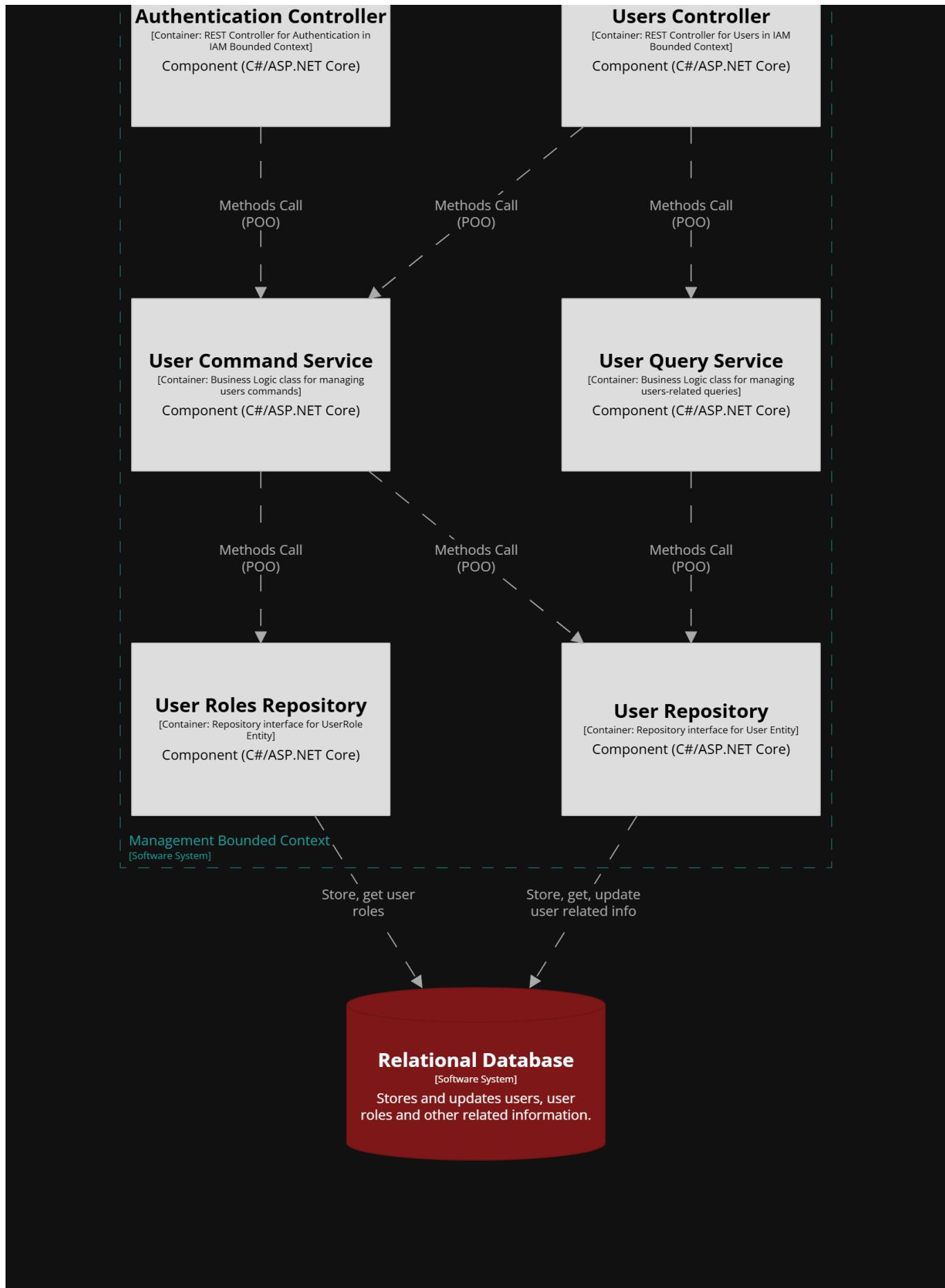
- **Descripción:** Implementación que permite interactuar con la base de datos de usuarios.
- **Métodos:**
  - `FindByEmailAsync(string email)`: Devuelve un usuario en base al email proporcionado
  - `ExistByUsername(string username)`: Devuelve un valor positivo si el nombre de usuario se encuentra en otro usuario
  - `GetUsernameByIdAsync(int userId)`: Devuelve el nombre de usuario con base a un id proporcionado
  - `ExistsById(int userId)`: Devuelve un valor positivo si hay un usuario con el id proporcionado

##### 2. UserRoleRepository:

- **Descripción:** Implementación que permite interactuar con la base de datos de roles de usuario
- **Métodos:**
  - `ExistsUserRole(EUserRoles role)`: Devuelve positivo si el rol de usuario ya existe.

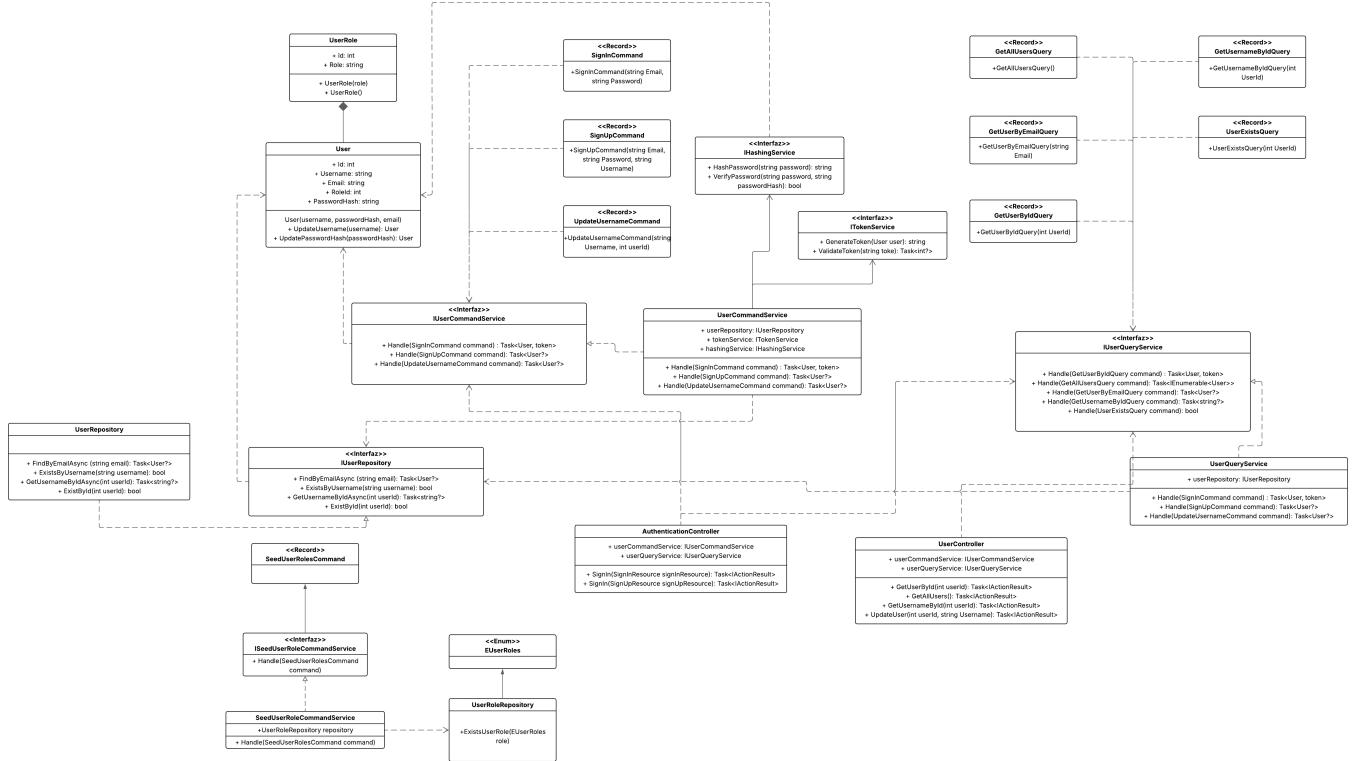
### 4.2.1.5. Bounded Context Software Architecture Component Level Diagrams



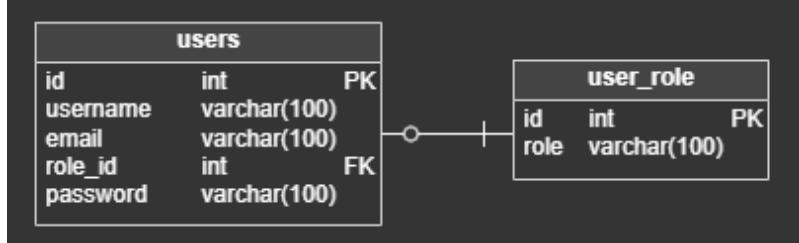


#### 4.2.1.6. Bounded Context Software Architecture Code Level Diagrams

##### 4.2.1.6.1. Bounded Context Domain Layer Class Diagrams



#### 4.2.1.6.2. Bounded Context Database Design Diagram



## 4.2.2. Bounded Context: Locals Context

### 4.2.X.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.

## 4.2.2.2. 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.

### 4.2.2.3. 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.

## 4.2.2.4. 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

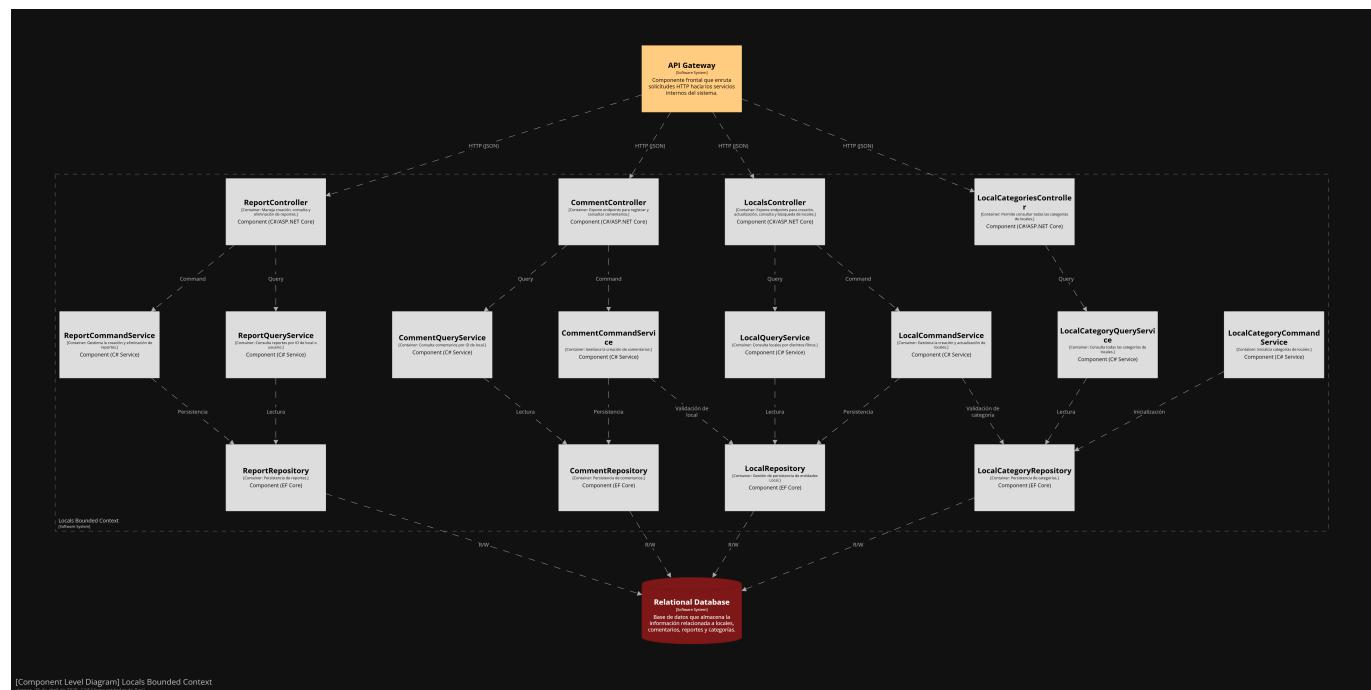
- **Descripción:** Implementación del repositorio para gestionar entidades `Local`.
- **Hereda de:** `BaseRepository<Local>`
- **Métodos implementados:**
  - `GetAllDistrictsAsync()`: Obtiene todos los distritos registrados donde hay locales, combinando ubicación con dirección.

- `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

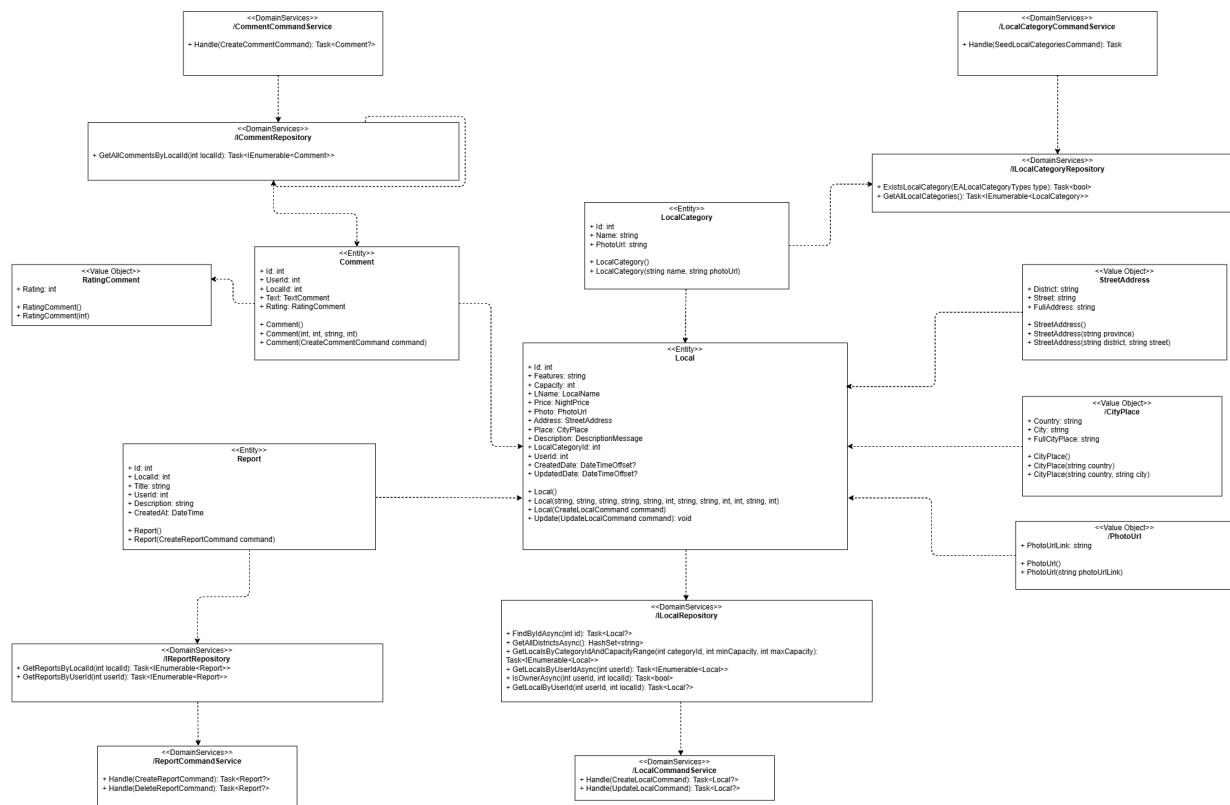
- **Descripción:** Implementación del repositorio para gestionar reportes relacionados a locales.
- **Hereda de:** `BaseRepository<Report>`
- **Métodos implementados:**
  - `GetReportsByLocalId(int localId)`: Devuelve todos los reportes asociados a un local.
  - `GetReportsByUserId(int userId)`: Devuelve todos los reportes realizados por un usuario específico.

#### 4.2.2.5. Bounded Context Software Architecture Component Level Diagrams

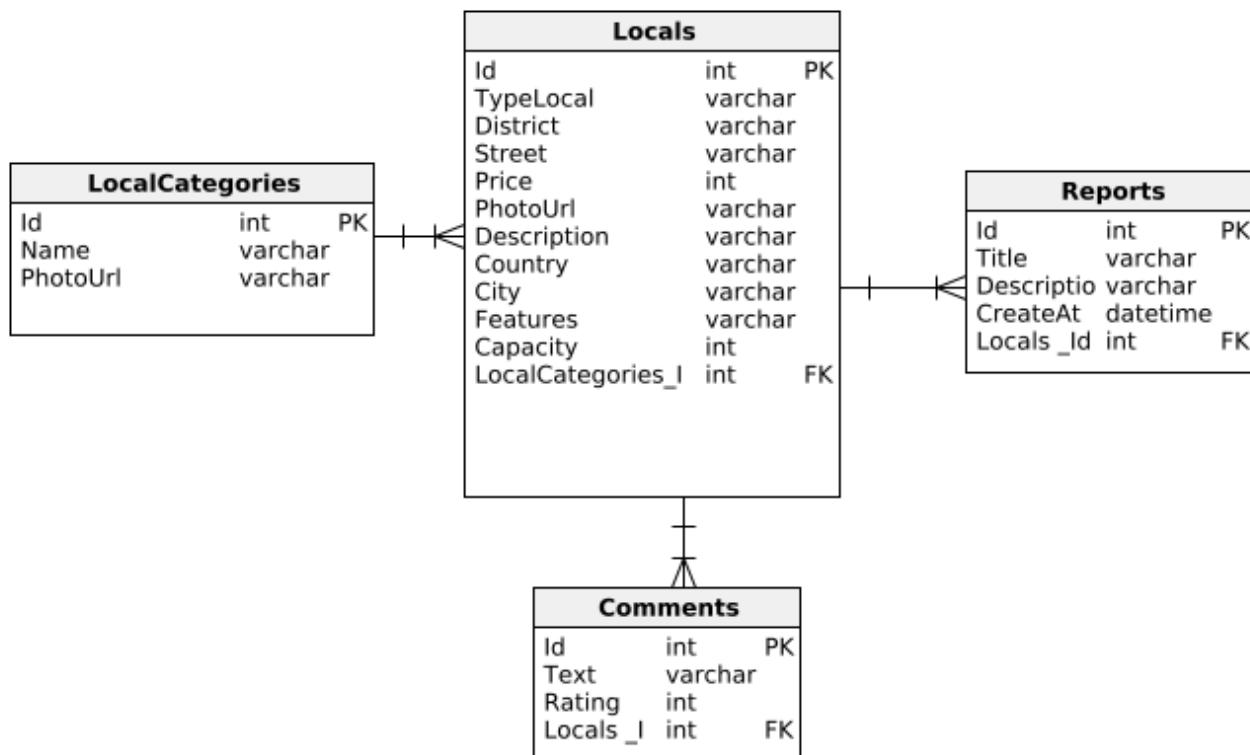


#### 4.2.2.6. Bounded Context Software Architecture Code Level Diagrams

##### 4.2.X.6.1. Bounded Context Domain Layer Class Diagrams



#### **4.2.2.6.2. Bounded Context Database Design Diagram**



#### 4.2.3. Bounded Context: Booking Context

#### 4.2.3.1. 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.

#### 4.2.3.2. 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.

#### 4.2.3.3. 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.

### 4.2.3.4. Application Layer

#### Repositories (implementaciones)

##### ReservationRepository

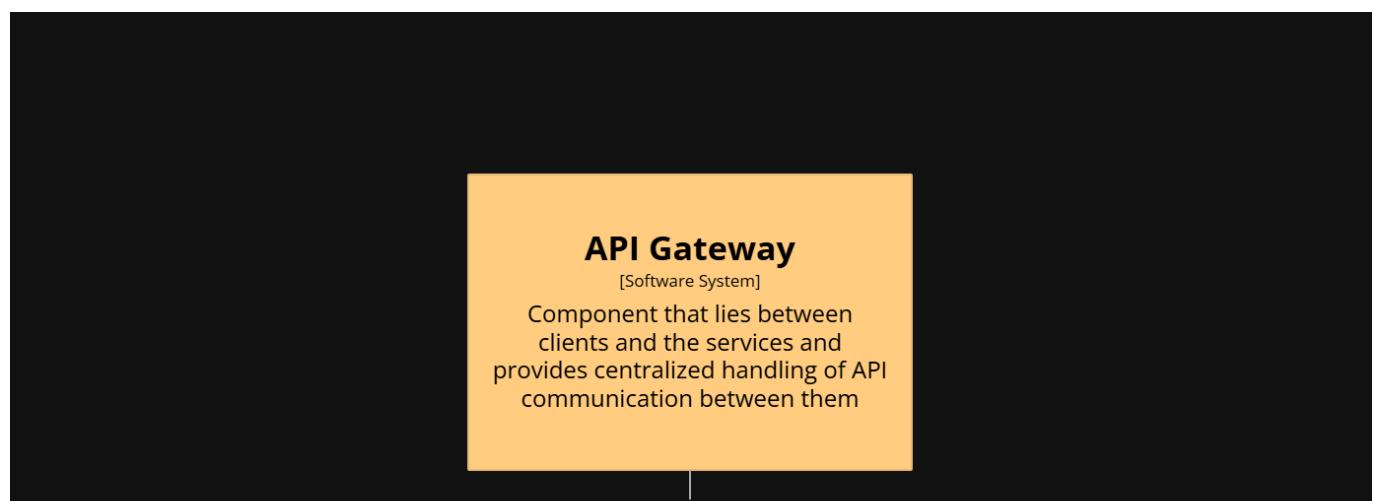
###### Descripción:

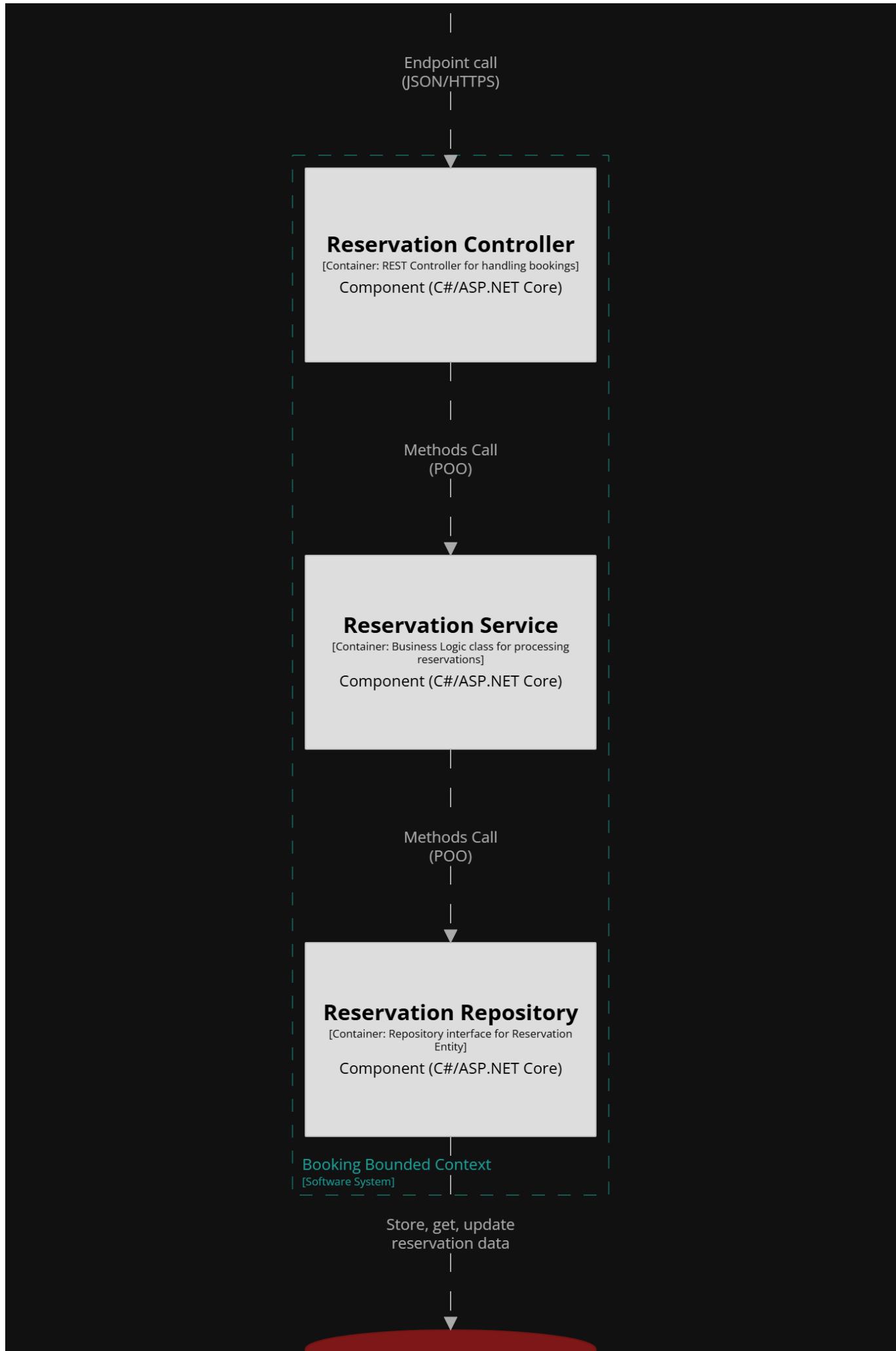
Implementación del repositorio para interactuar con la base de datos de reservas, utilizando Entity Framework Core.

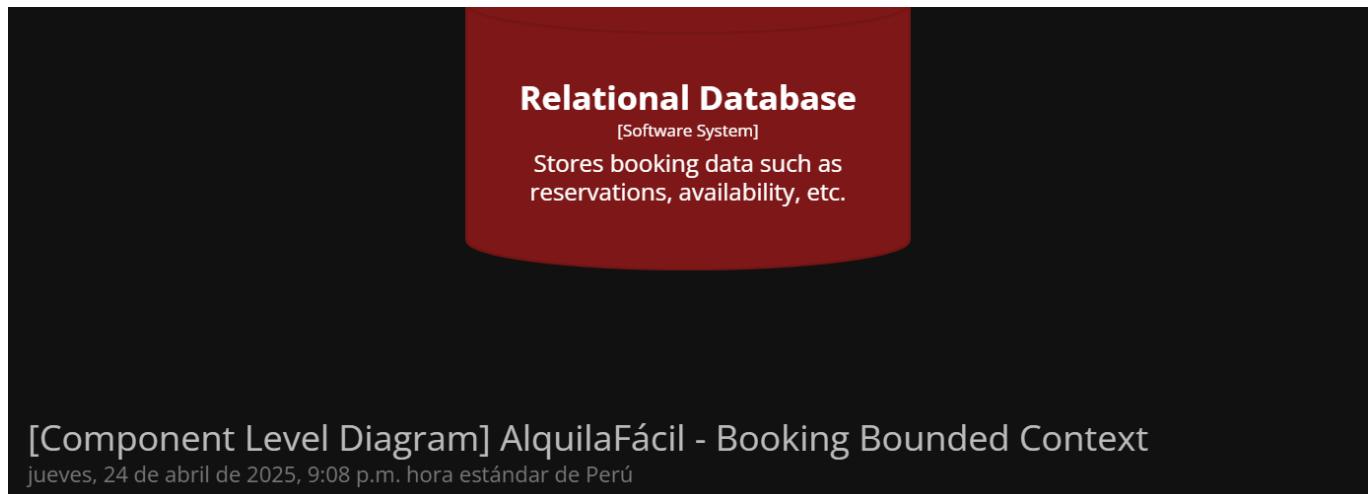
###### Métodos:

- `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.

### 4.2.3.5. Bounded Context Software Architecture Component Level Diagrams

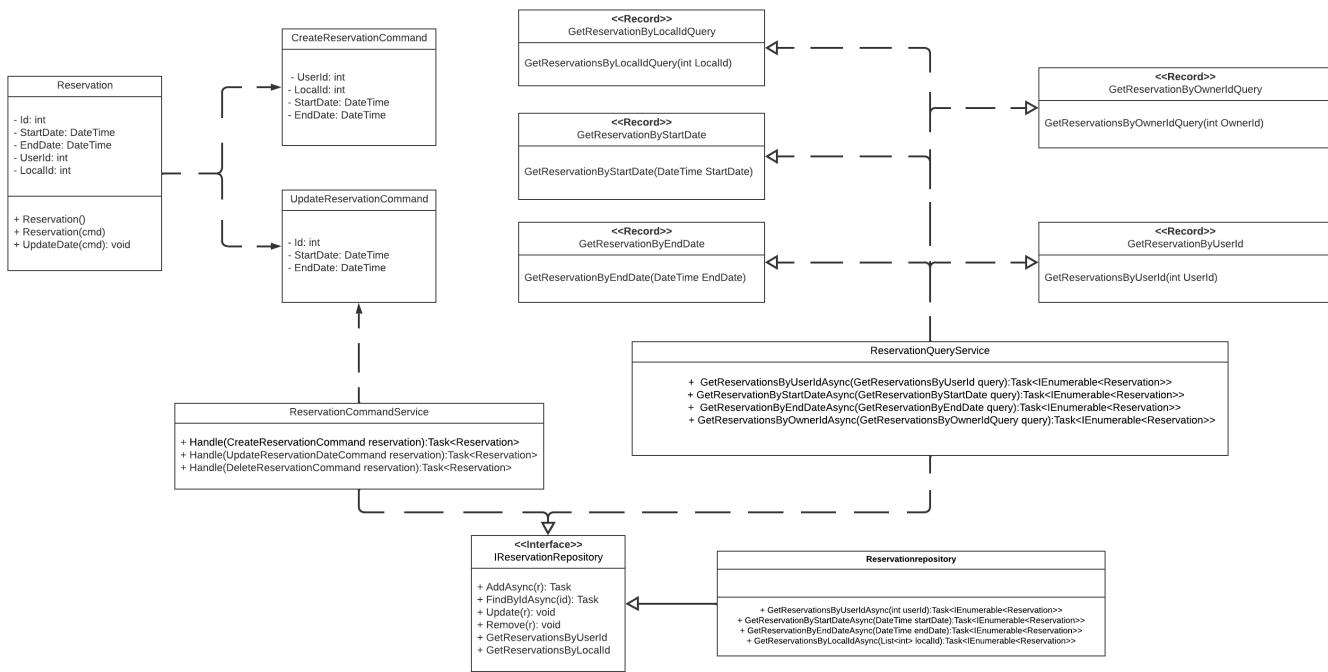




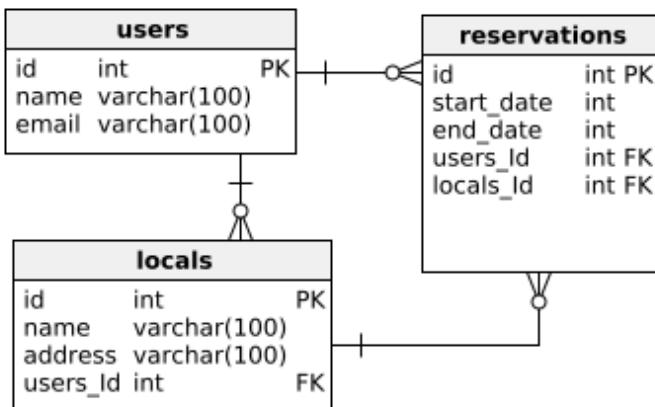


#### 4.2.3.6. Bounded Context Software Architecture Code Level Diagrams

##### 4.2.3.6.1. Bounded Context Domain Layer Class Diagrams



##### 4.2.3.6.2. Bounded Context Database Design Diagram



## 4.2.4. Bounded Context: Management Context

### 4.2.4.1. 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 de 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. NFC Tag

- **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.

## 4.2.4.2. 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.

#### 4.2.4.3. 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.

#### 4.2.4.4. 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.

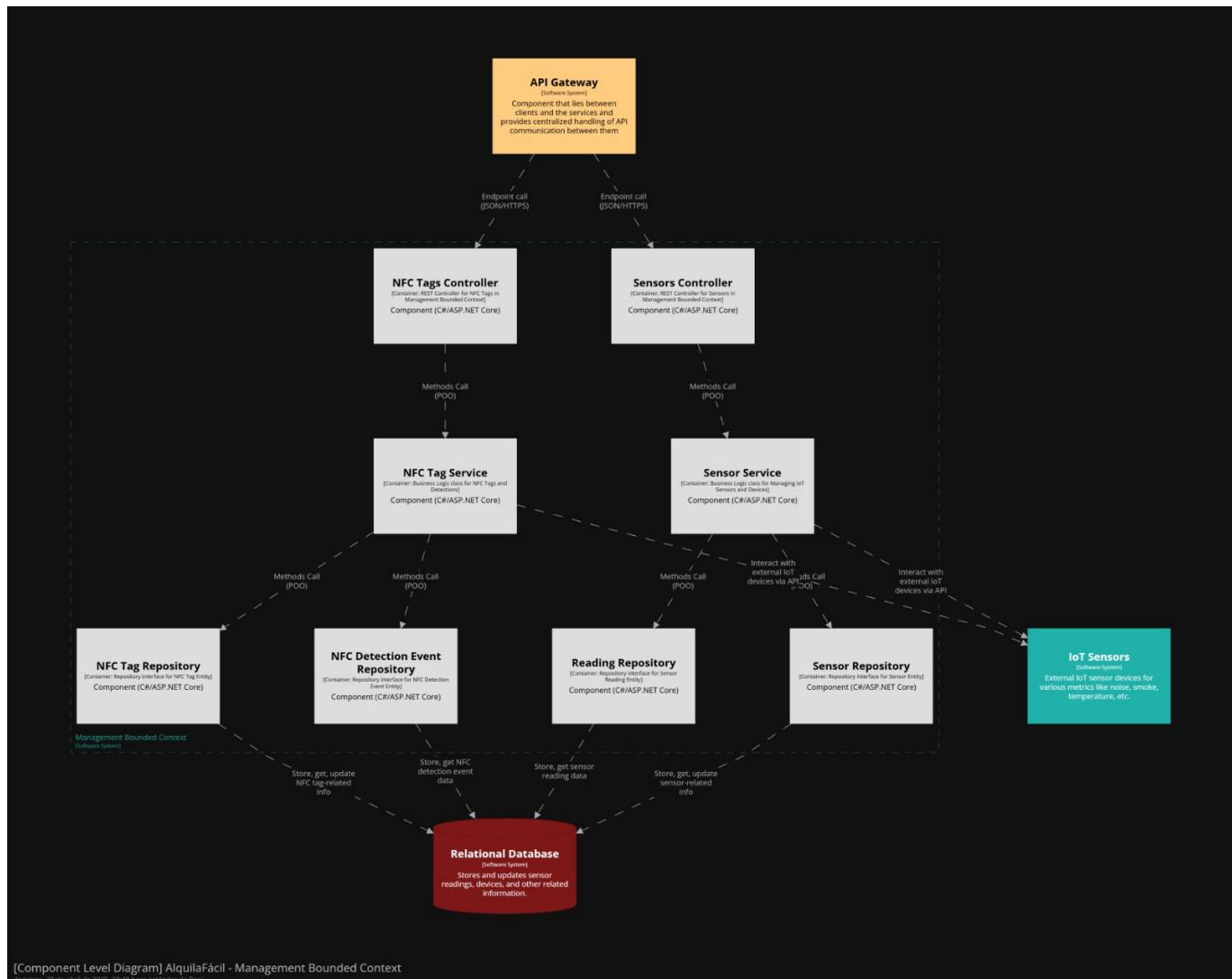
### 3. NFCTagRepository:

- **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:

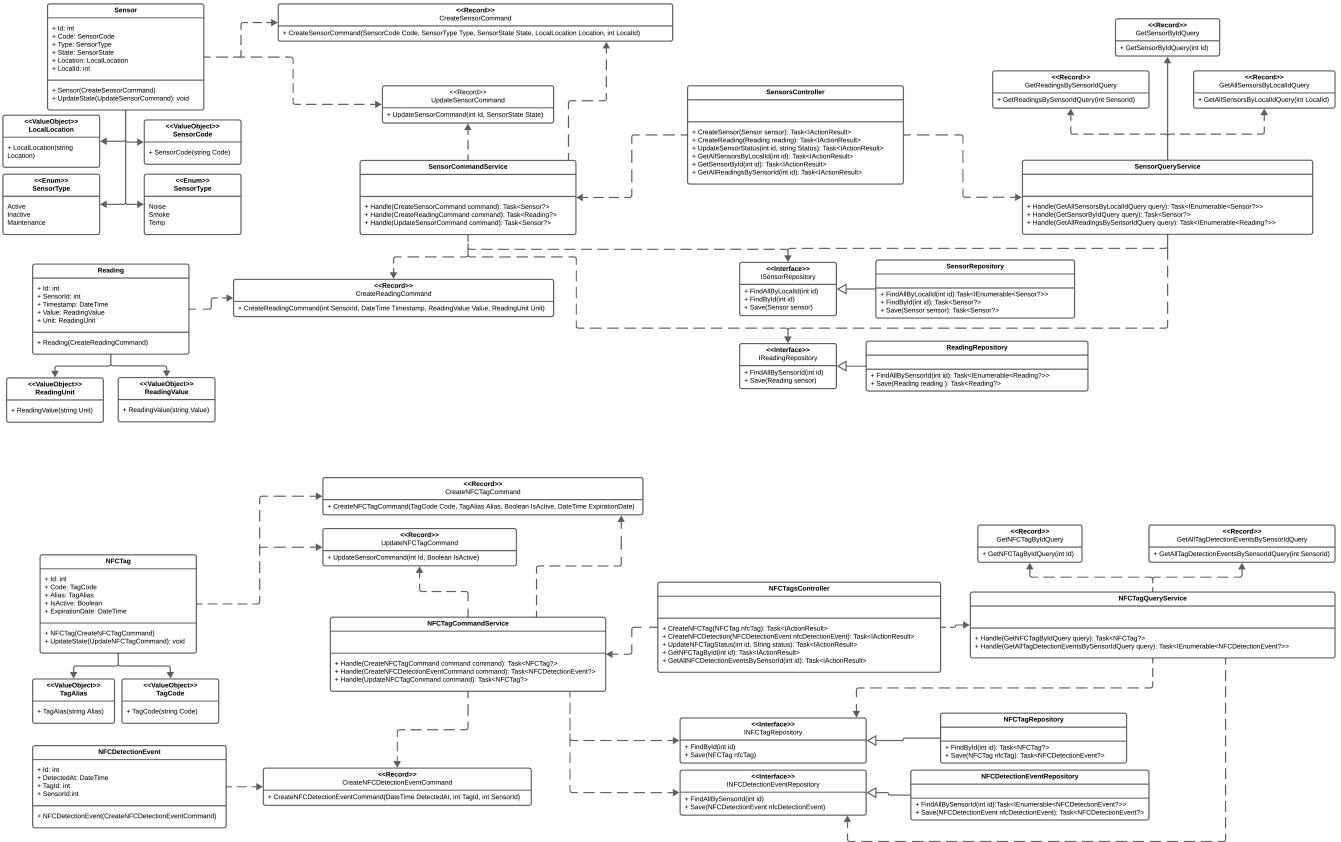
- **Descripción:** Implementación para interactuar con la base de datos de detecciones NFC.
- **Métodos:**
  - `FindAllBySensorId(int id)`: Busca todas las detecciones por ID de sensor.
  - `Save(NFCDetectionEvent nfcDetectionEvent)`: Guarda o actualiza una detección en la base de datos.

#### 4.2.4.5. Bounded Context Software Architecture Component Level Diagrams

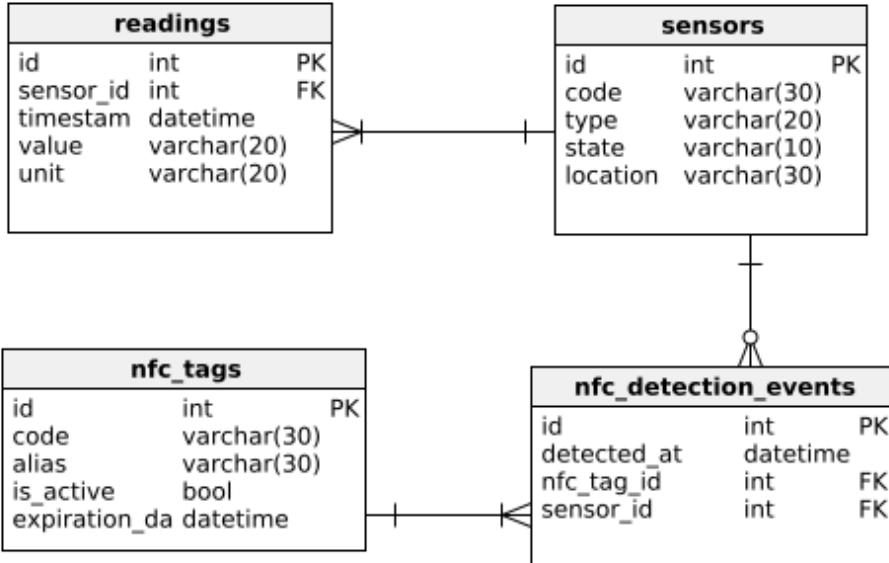


#### 4.2.4.6. Bounded Context Software Architecture Code Level Diagrams

##### 4.2.4.6.1. Bounded Context Domain Layer Class Diagrams



#### 4.2.4.6.2. Bounded Context Database Design Diagram



Vertabelo

#### 4.2.5. Bounded Context: Notifications Context

##### 4.2.5.1. 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 (sistema, alerta, información).
    - `CreationDate`: Fecha y hora de creación.
    - `State`: Estado de la notificación (`pendiente`, `enviada`, `leída`).
- 

#### 4.2.5.2. Interface Layer

##### Controllers

###### ◆ NotificationController

- **Descripción:** Expone endpoints para gestionar las notificaciones del sistema.
  - **Métodos:**
    - `SendNotification(SendNotificationRequestDTO dto)`: Envía una nueva notificación.
    - `GetAllNotificationsByUserId(int userId)`: Retorna todas las notificaciones asociadas a un usuario.
    - `GetUnreadNotificationsByUserId(int userId)`: Lista las notificaciones no leídas del usuario.
    - `MarkAsRead(int notificationId)`: Marca una notificación como leída.
- 

#### 4.2.5.3. 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.

##### Query Services

###### ◆ NotificationQueryService

- **Descripción:** Permite consultar notificaciones relacionadas a un usuario.
  - **Métodos:**
    - `Handle(GetAllNotificationsByUserIdQuery query)`: Devuelve todas las notificaciones del usuario.
    - `Handle(GetUnreadNotificationsByUserIdQuery query)`: Devuelve notificaciones no leídas.
- 

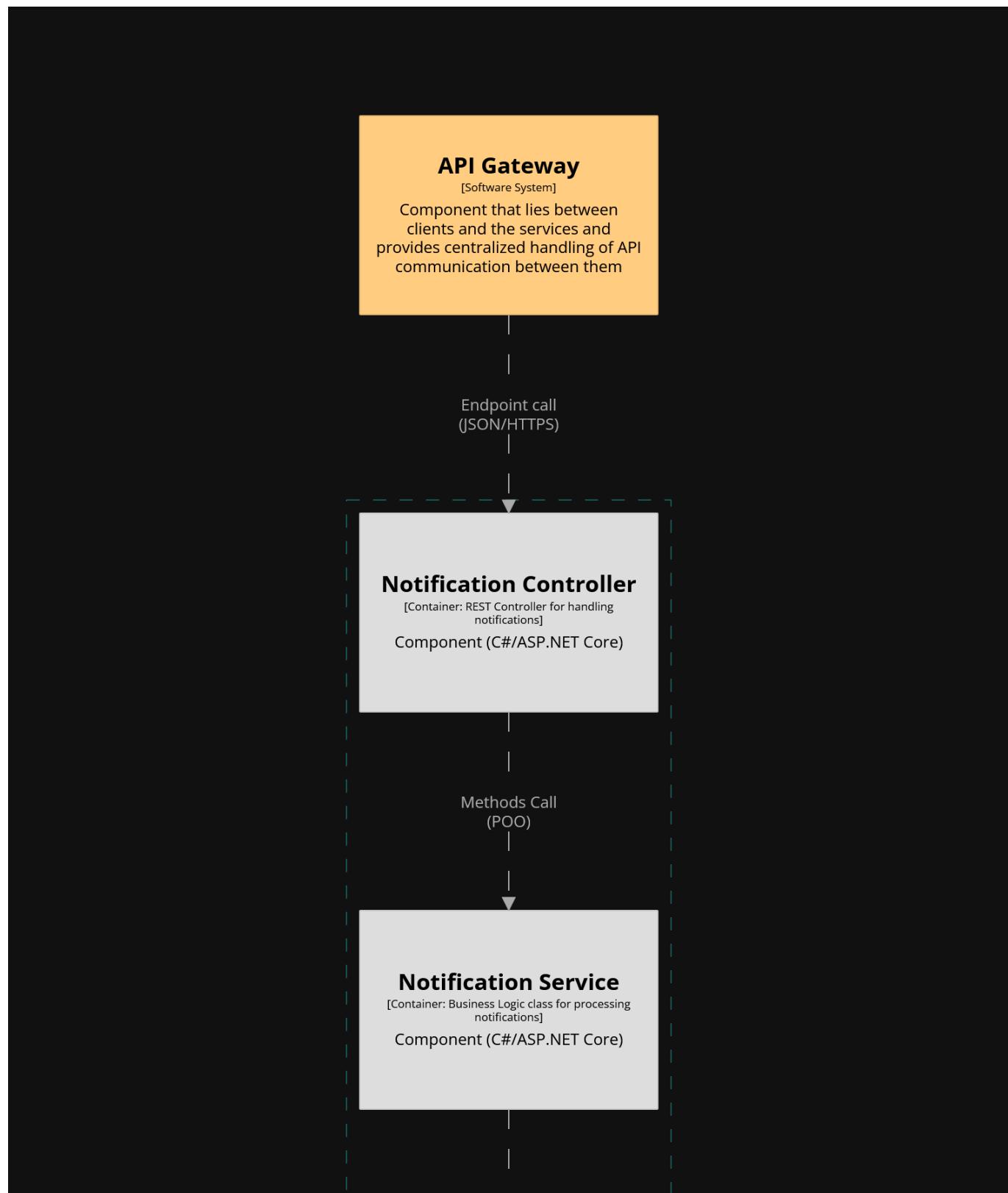
#### 4.2.5.4. Infrastructure Layer

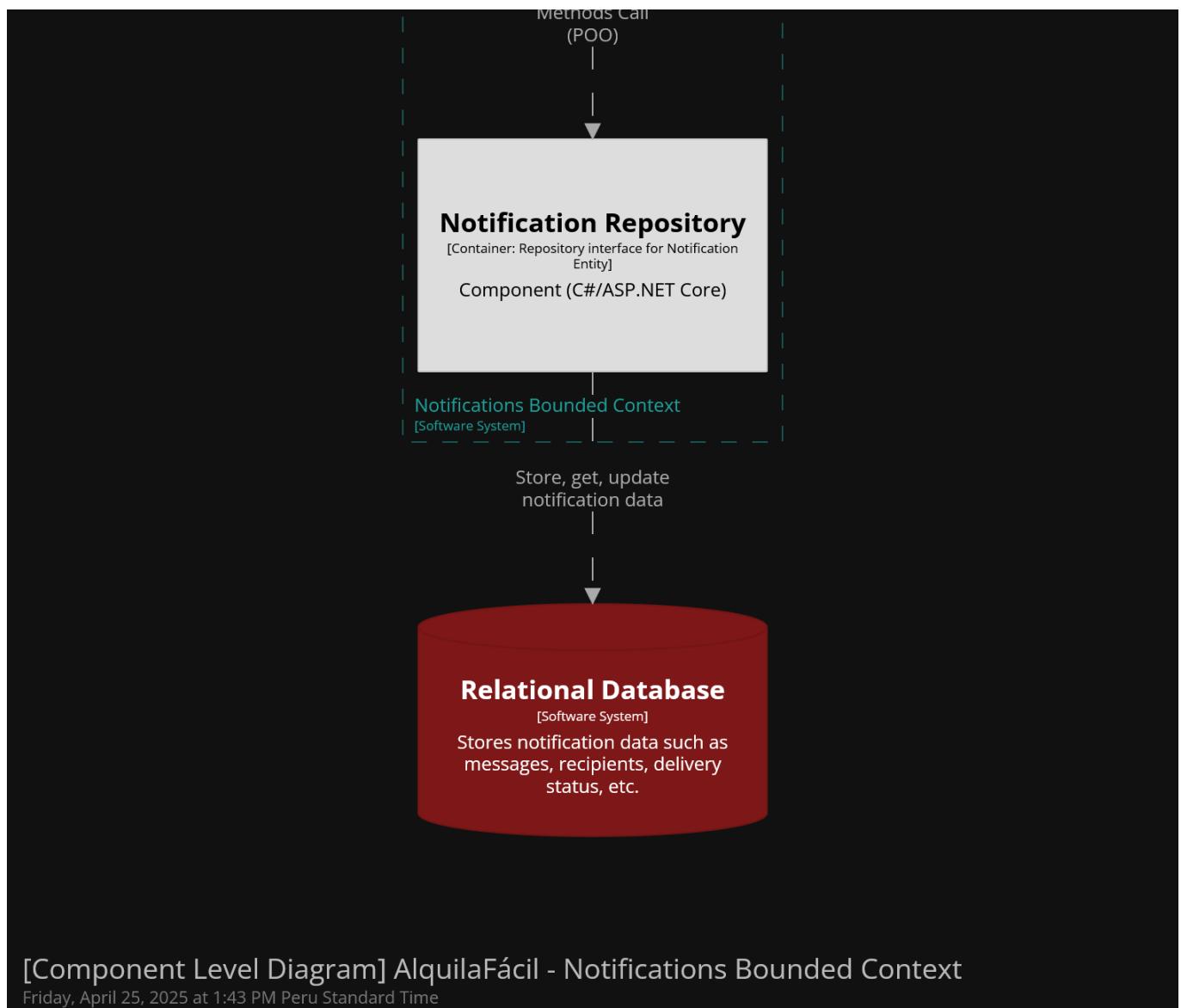
##### Repositories (Implementaciones)

## ❖ NotificationRepository

- **Descripción:** Implementa acceso a la base de datos de notificaciones.
- **Métodos:**
  - `findByUserId(userId)`: Lista de notificaciones por usuario.
  - `findUnreadByUserId(userId)`: Lista de notificaciones no leídas.
  - `markAsRead(notificationId)`: Marca una notificación como leída.
  - `save(notification)`: Guarda una nueva notificación.

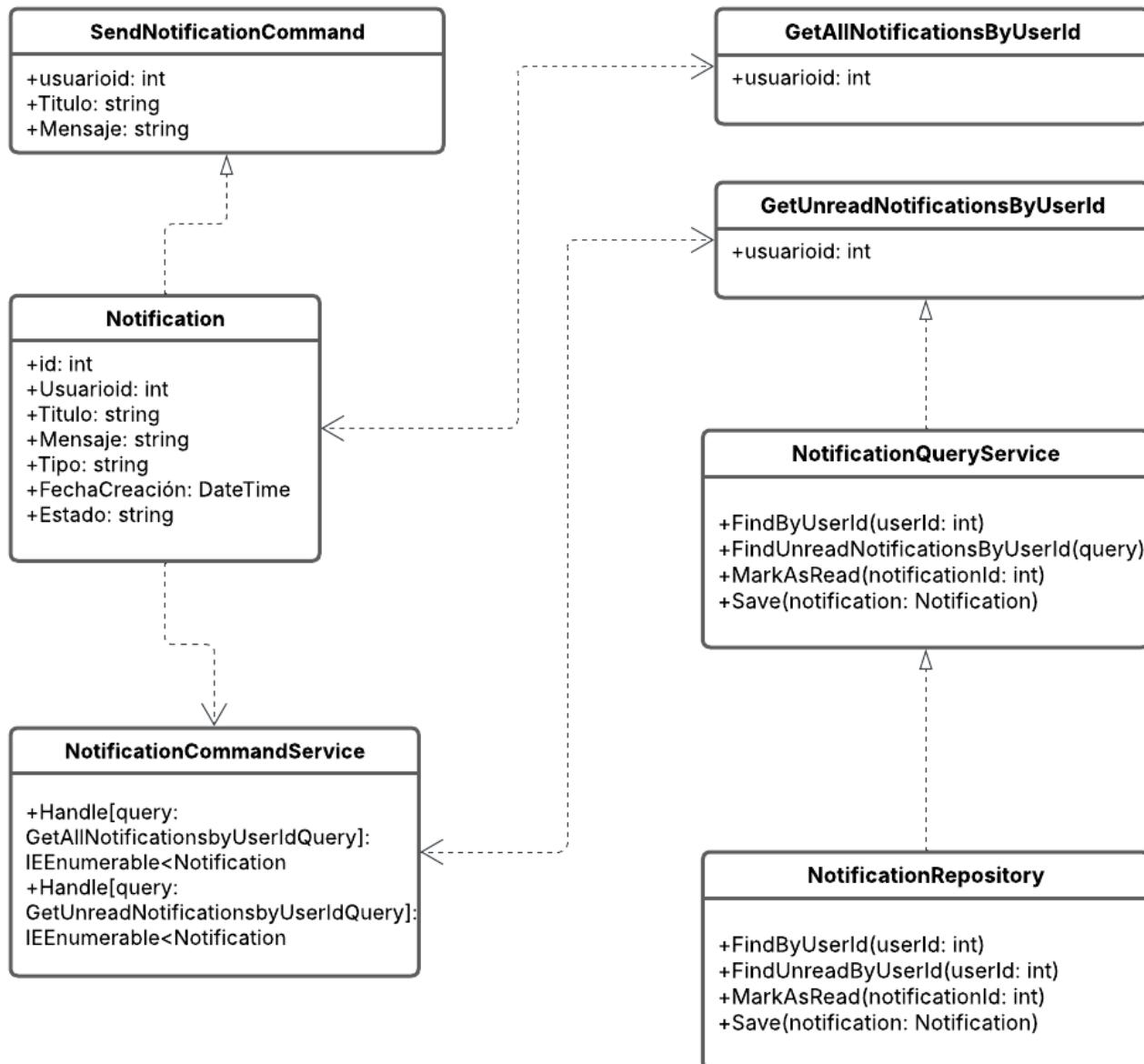
### 4.2.5.5. Bounded Context Software Architecture Component Level Diagrams



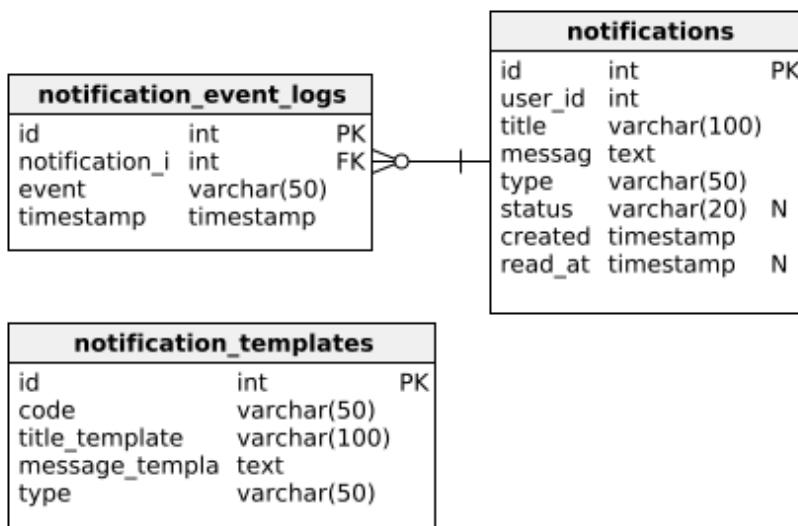


#### 4.2.5.6. Bounded Context Software Architecture Code Level Diagrams

##### 4.2.5.6.1. Bounded Context Domain Layer Class Diagrams



#### 4.2.5.6.2. Bounded Context Database Design Diagram



## 4.2.6. Bounded Context: Profile Context

### 4.2.6.1. Domain Layer

#### Aggregates

##### 1. Profile:

- **Descripción:** Representa el perfil de un usuario en la aplicación
- **Atributos:**
  - **Name:** Nombre del usuario
  - **Birth:** Fecha de nacimiento del usuario
  - **PhoneN:** Número de teléfono del usuario
  - **DocumentN:** Documento de identidad del usuario
  - **UserId:** Id asignado al usuario
  - **PhotoUrl:** Foto de perfil del usuario

#### Value Objects

##### 1. DateOfBirth:

- **Descripción:** Fecha de nacimiento de un usuario
- **Atributos**
  - **BirthDate:** Fecha de nacimiento

##### 2. DocumentNumber:

- **Descripción:** Documento de identidad del usuario
- **Atributos**
  - **NumberDocument:** Número del documento

##### 3. PersonName:

- **Descripción:** Nombre de una persona
- **Atributos**
  - **Name:** Primer nombre de la persona
  - **FatherName:** Apellido paterno de la persona
  - **MotherName:** Apellido materno de la persona
  - **FullName:** Nombre completo de la persona

#### 4. Phone:

- **Descripción:** Número de teléfono de una persona
- **Atributos**
  - **PhoneNumber:** Número de teléfono de la persona

### Commands

#### 1. CreateProfileCommand:

- **Descripción:** Comando que permite la creación de perfiles con el nombre, el telefono, el id del usuario y la foto del usuario
- 2. **UpdateProfileCommand:** Comando que permite la modificación de perfiles a partir del id del perfil y el atributo a modificar.

### Queries

#### 1. GetAllProfilesQuery:

- **Descripción:** Consulta que devuelve todas los perfiles almacenados.

#### 2. GetProfileByIdQuery:

- **Descripción:** Consulta que devuelve el perfil a partir de un id especificado

#### 3. GetProfileByUserIdQuery:

- **Descripción:** Consulta que permite obtener un perfil a partir de un id de usuario especificado

#### 4. IsUserSubscribeQuery:

- **Descripción:** Consulta que verifica si un usuario se encuentra suscrito a partir de su id

### Repositories (Interfaces)

#### 1. IProfileRepository:

- **Descripción:** Permite interactuar con la base de datos para la búsqueda o creación de perfiles.
- **Métodos:**
  - **FindByUserIdAsync(int userId):** Permite obtener un perfil a través de la id de un usuario

### Services (Interfaces)

#### 1. IProfileCommandService:

- **Descripción:** Maneja los comandos para la creación y actualización de perfiles
- **Métodos:**
  - `Handle(CreateProfileCommand command)`: Maneja el comando que permite la creación de perfiles.
  - `Handle(UpdateProfileCommand command)`: Maneja el comando que permite la actualización de perfiles.

## 2. IProfileQueryService:

- **Descripción:** Maneja las consultas de los perfiles desde la base de datos.
- **Métodos:**
  - `Handle(GetAllProfilesQuery)`: Maneja la consulta para obtener todos los perfiles.
  - `Handle(GetProfileByIdQuery)`: Maneja la consulta que devuelve un perfil a través de su id.
  - `Handle(GetProfileByUserIdQuery)`: Maneja la consulta que devuelve un perfil a través del id del usuario.
  - `Handle(IsUserSubscribeQuery)`: Maneja la consulta que verifica si el usuario está suscrito.

### 4.2.6.2. Interface Layer

#### Facades

##### 1. ProfilesContextFacade:

- **Descripción:** Permite que otros bounded context interactúen con el context de profiles.
- **Métodos:**
  - `CreateProfile(string name, string? fatherName, string? motherName, string dateOfBirth, string documentNumber, string phone, int userId, string photoUrl)`: Permite la creación de un perfil desde fuera del bounded context.

#### Controllers

##### 1. ProfilesController:

- **Descripción:** Expone endpoints para el manejo de los perfiles dentro de la aplicación
- **Métodos:**
  - `CreateProfile(CreateProfileResource createProfileResource)`: Endpoint para la creación de un perfil
  - `GetAllProfiles()`: Endpoint que muestra todos los perfiles registrados en la aplicación
  - `GetProfileById(int profileId)`: Endpoint que muestra un perfil en base a su id.
  - `GetProfileByUserId(int userId)`: Endpoint que muestra un perfil en base al id del usuario.
  - `IsUserSubscribed(int userId)`: Endpoint que verifica si un usuario está suscrito a través de su id
  - `UpdateFarm(UpdateProfileResource updateProfileResource)`: Endpoint que permite actualizar un perfil ya existente.

### 4.2.6.3. Application Layer

#### Command Services (Implementación)

##### 1. ProfileCommandService:

- **Descripción:** Maneja los comandos para la creación y actualización de perfiles
- **Métodos:**
  - `Handle(CreateProfileCommand command)`: Maneja el comando que permite la creación de perfiles.
  - `Handle(UpdateProfileCommand command)`: Maneja el comando que permite la actualización de perfiles.

## Query Services (Implementación)

### 1. ProfileQueryService:

- **Descripción:** Maneja las consultas de los perfiles desde la base de datos.
- **Métodos:**
  - `Handle(GetAllProfilesQuery)`: Maneja la consulta para obtener todos los perfiles.
  - `Handle(GetProfileByIdQuery)`: Maneja la consulta que devuelve un perfil a través de su id.
  - `Handle(GetProfileByUserIdQuery)`: Maneja la consulta que devuelve un perfil a través del id del usuario.
  - `Handle(IsUserSubscribeQuery)`: Maneja la consulta que verifica si el usuario está suscrito.

## Outbound Services

### 1. SubscriptionExternalService:

- **Descripción:** Permite el uso del bounded context de subscriptions.
- **Métodos:**
  - `IsUserSubscribeAsync(int userId)`: Verifica si un usuario cuenta con una subscripción.

### 2. UserExternalService:

- **Descripción:** Permite el uso del bounded context de users
- **Métodos:**
  - `UserExistsById(int userId)`: Verifica que un usuario existe en base a su id

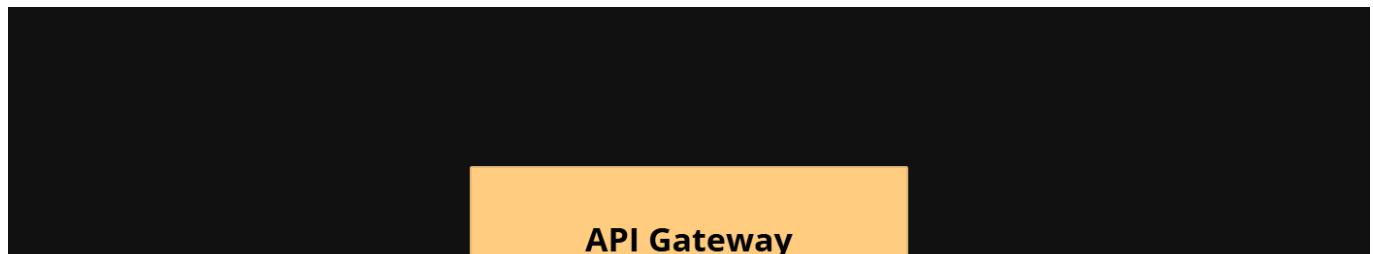
## 4.2.6.4. Infrastructure Layer

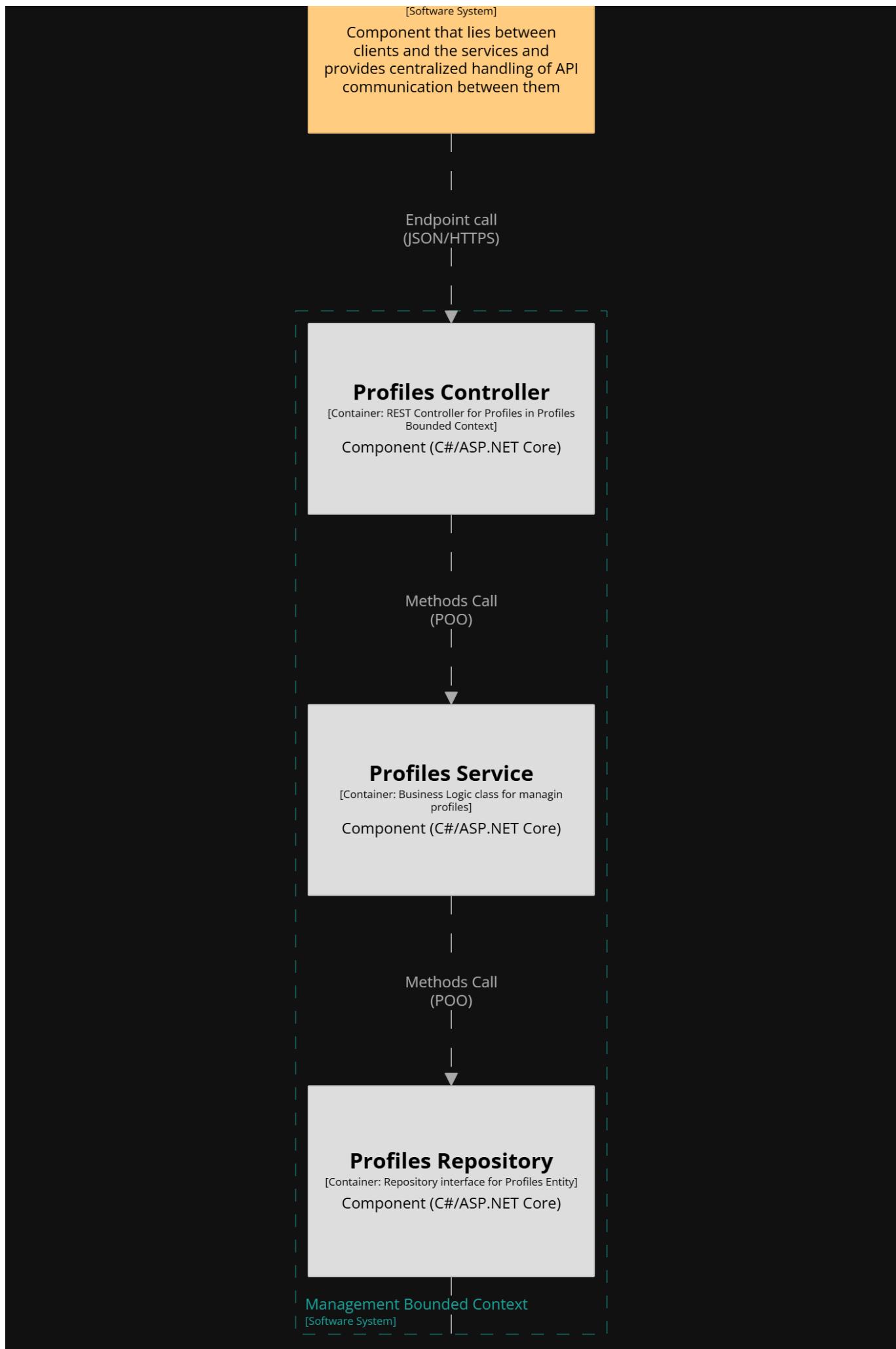
## Repositories (Implementación)

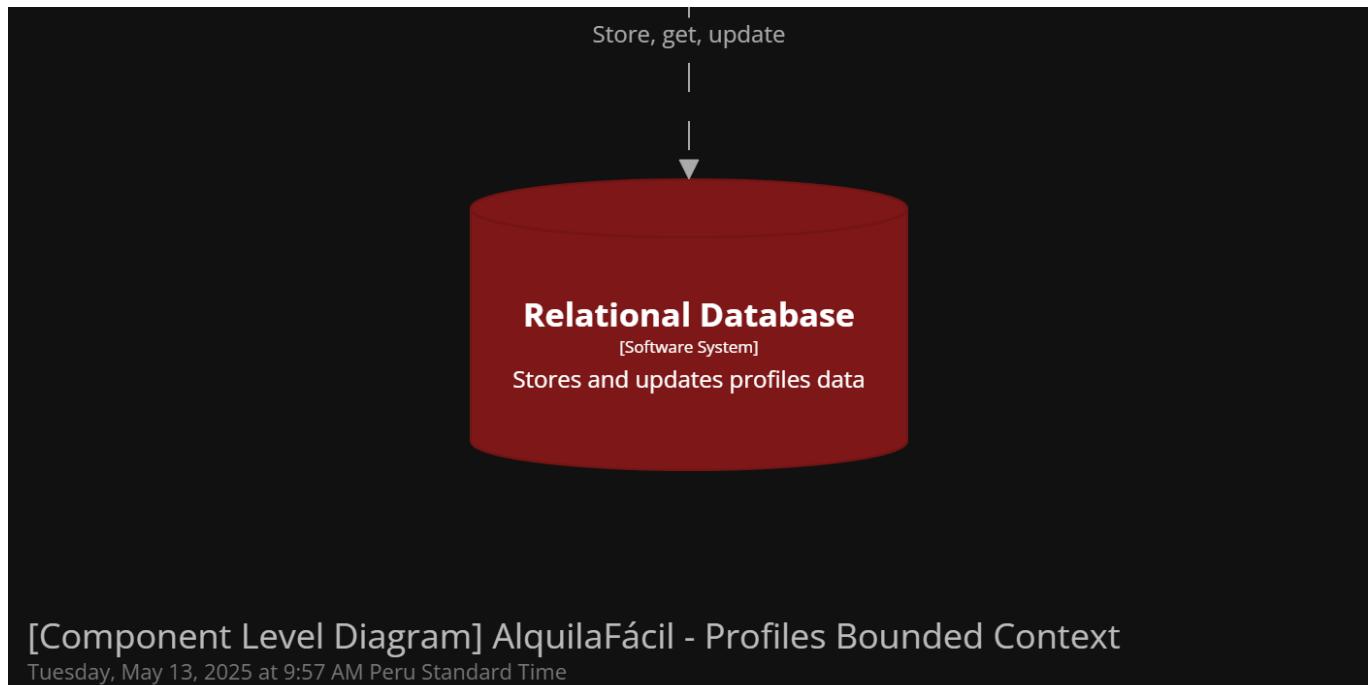
### 1. ProfileRepository:

- **Descripción:** Permite la conexión con la base de datos.
- **Métodos:**
  - `FindByUserIdAsync`: Permite la búsqueda de un perfil a través de una id de usuario especificada.

## 4.2.6.5. Bounded Context Software Architecture Component Level Diagrams





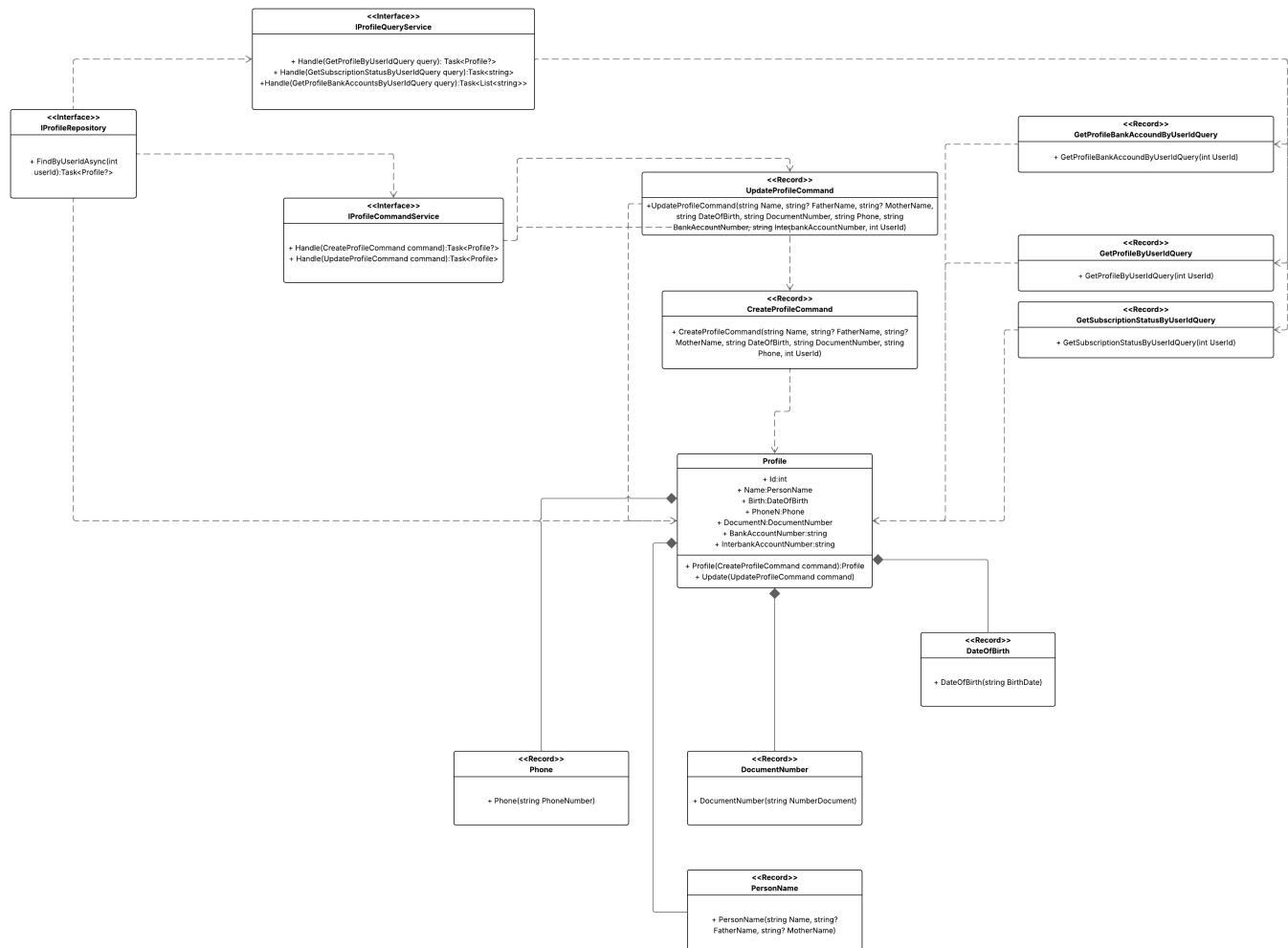


## [Component Level Diagram] AlquilaFácil - Profiles Bounded Context

Tuesday, May 13, 2025 at 9:57 AM Peru Standard Time

### 4.2.6.6. Bounded Context Software Architecture Code Level Diagrams

#### 4.2.6.6.1. Bounded Context Domain Layer Class Diagrams



#### 4.2.6.6.2. Bounded Context Database Design Diagram

profiles		
id	int	PK
birthDate	string	
phoneNumber	string	
numberDocume	string	
userId	int	



## 4.2.7. Bounded Context: Subscription Context

### 4.2.7.1. 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 unico 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 suscripción

### 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 de el 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 traves del id de su usuario
  - `FindByUserIdsListAsync`: Obtiene una lista de suscripciones a traves 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 subscripciones desde la base datos.
- **Métodos:**
  - `Handle(GetSubscriptionByIdQuery query)`: Maneja la consulta para obtener una subscripción en base a su id.
  - `Handle(GetAllSubscriptionsQuery query)`: Maneja la consulta para obtener todas las subscriptiones.
  - `Handle(GetSubscriptionByUserIdQuery query)`: Maneja la consulta para obtener una subscripción en base a la id de su usuario.
  - `Handle(GetSubscriptionsByUserIdQuery query)`: Maneja la consulta para obtener todas las subscripciones 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.

### 4.2.7.2. 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 subscripciones de varios usuarios a través de sus ids. `GetSubscriptionStatusByUserId(int userId)`: Obtiene los estados de la subscripción 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 subscripción dentro de la aplicación
- **Métodos:**
  - `GetAllPlans()`: Endpoint para obtener todos los planes de subscripción dentro de la aplicación

##### 3. **SubscriptionsController:**

- **Descripción:** Exponer endpoints para el manejo de las subscripciones 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

#### 4.2.7.3. 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.

#### 4.2.7.4. 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.

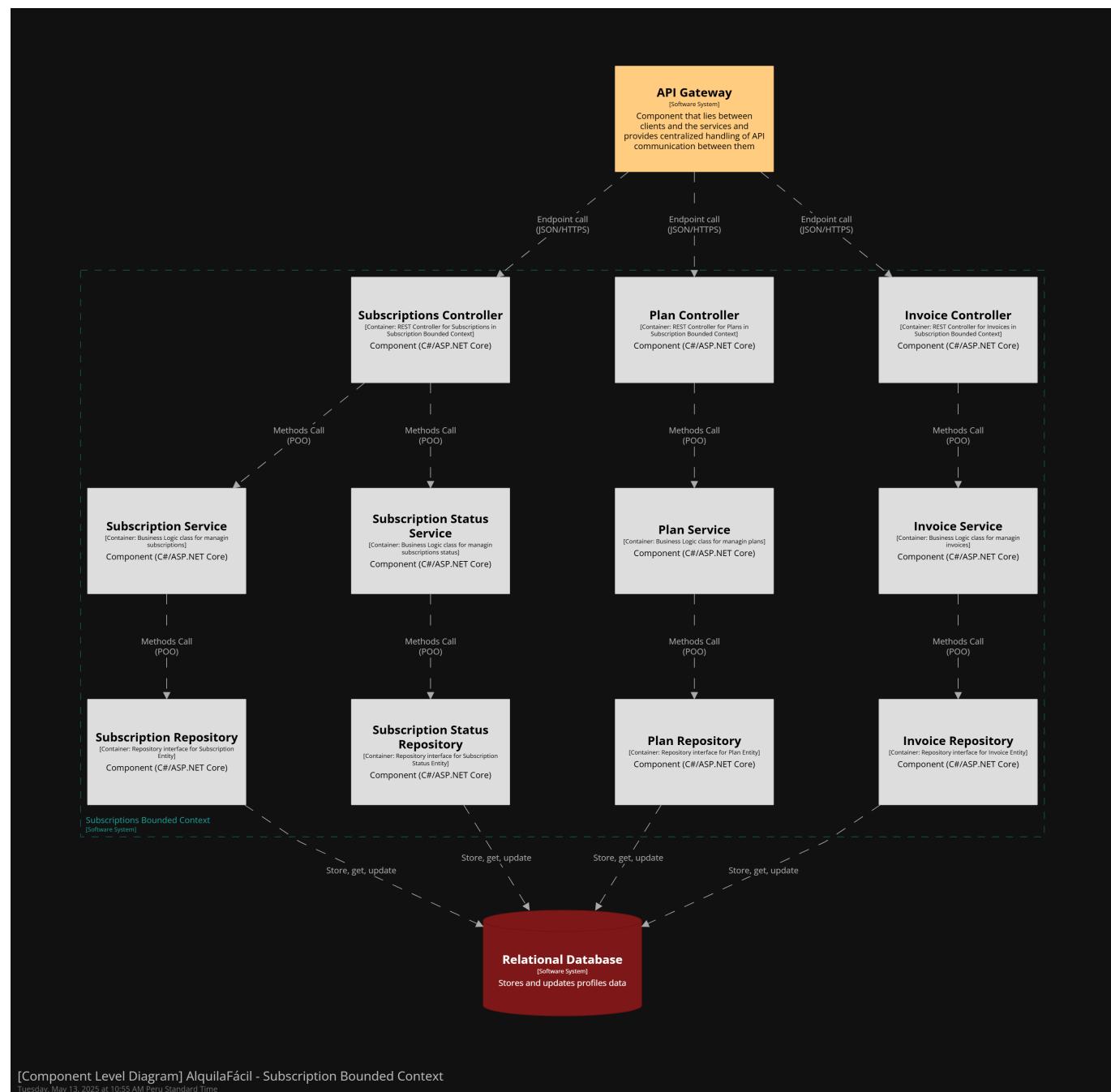
#### 3. ISubscriptionRepository:

- **Descripción:** Permite interactuar con la base de datos para almacenar u obtener datos de las subscripciones.
- **Métodos:**
  - `FindByUserIdAsync(int userId)`: Obtiene una subscripción a través del id de su usuario
  - `FindByUserIdsListAsync`: Obtiene una lista de subscripciones a través de una lista de ids de usuarios.

#### 4. ISubscriptionStatusRepository:

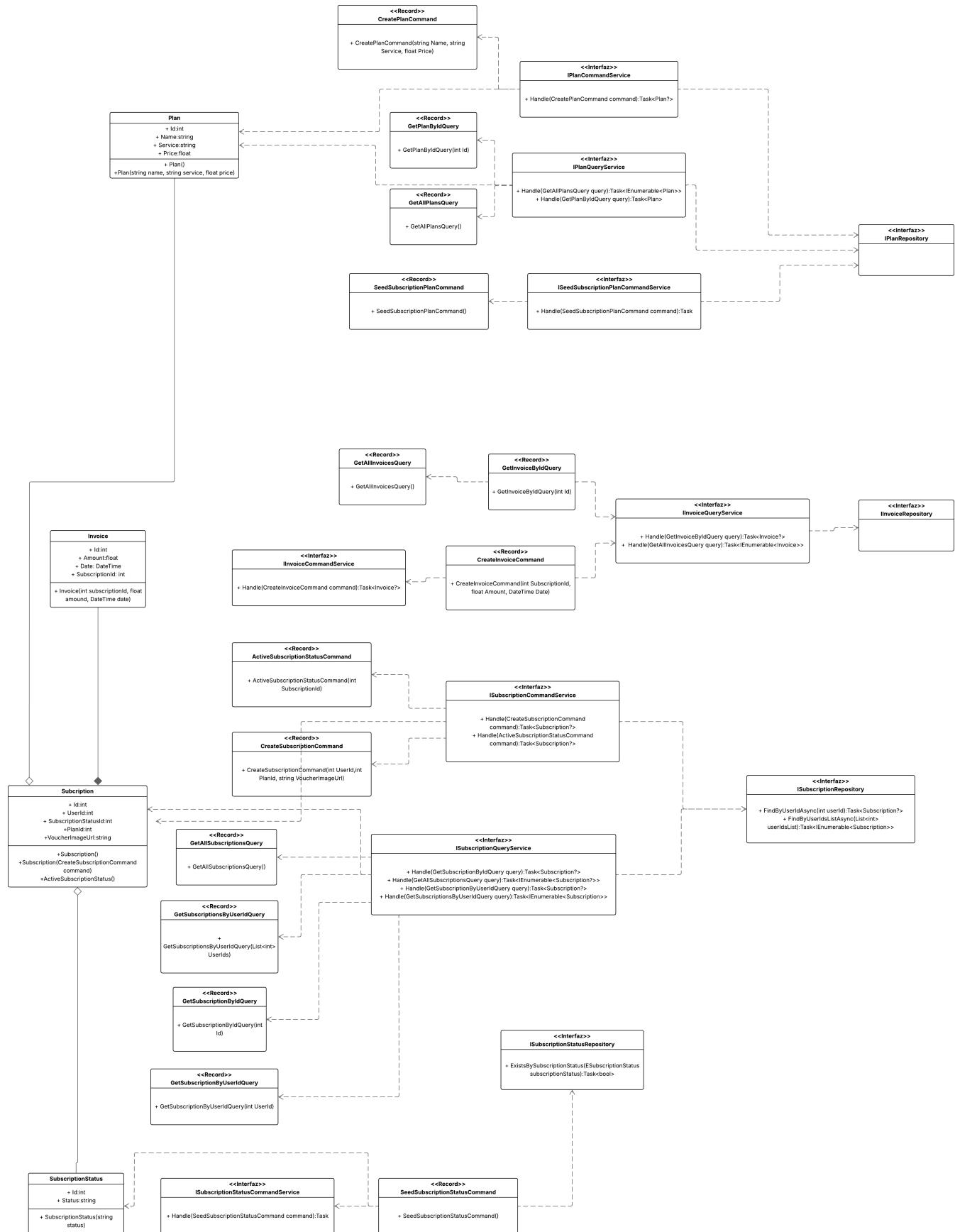
- **Descripción:** Permite interactuar con la base de datos para almacenar u obtener datos de los estados de subscripción.
- **Métodos:**
  - `ExistsBySubscriptionStatus(ESubscriptionStatus subscriptionStatus)`: Verifica si existe un estado.

#### 4.2.7.5. Bounded Context Software Architecture Component Level Diagrams

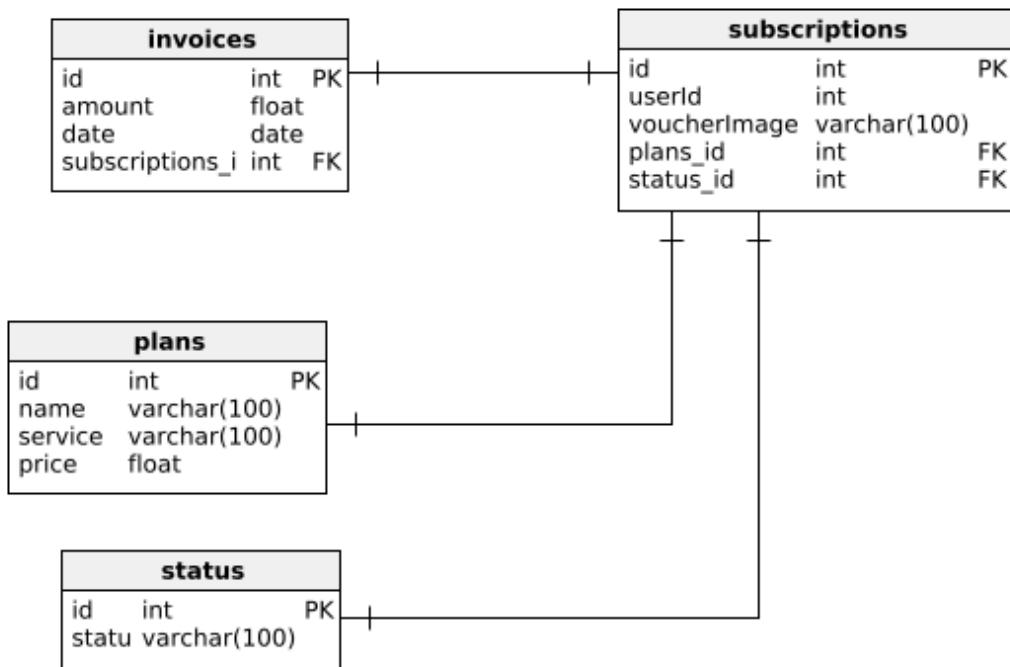


## 4.2.7.6. Bounded Context Software Architecture Code Level Diagrams

### 4.2.7.6.1. Bounded Context Domain Layer Class Diagrams



### 4.2.7.6.2. Bounded Context Database Design Diagram



## Capítulo V: Solution UI/UX Design

### 5.1. 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.

#### 5.1.1. General Style Guidelines.

##### **Overview:**

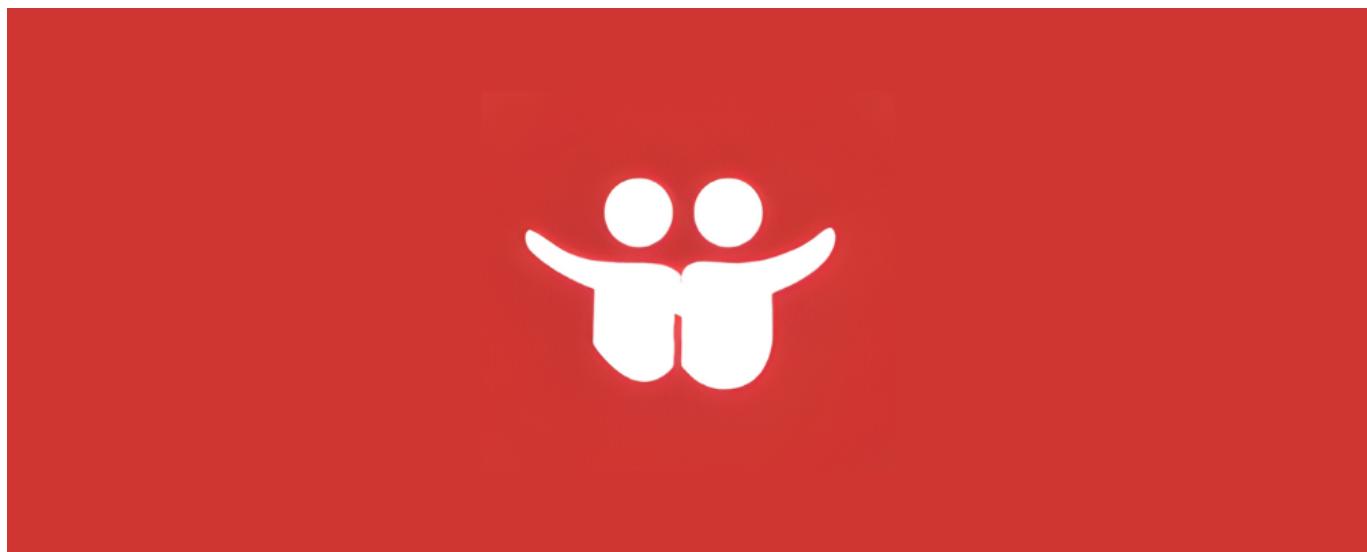
Deseamos capturar la atención del usuario desde el inicio mediante la creación y el diseño de una perspectiva del producto que establezca una conexión inmediata y reconocible.

##### **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

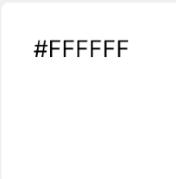
**Colors:**

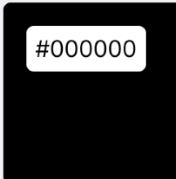
La gama de colores elegidos son:

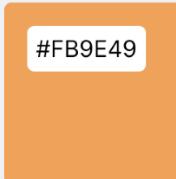
Auto Style Sheet

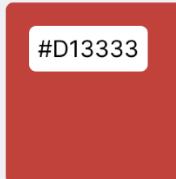
# Colors

## Used Background Colors

#FFFFFF

#000000

#FB9E49

#D13333

## Used Font Colors

Text

#000000

Text

#FB9E49

Text

#D13333

Text

#FFFFFF

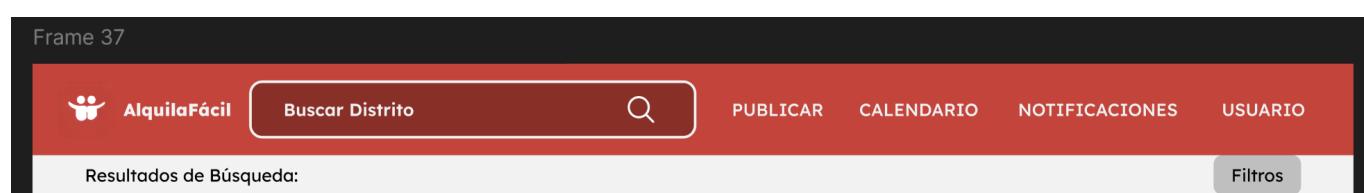
### 5.1.2. Web, Mobile and IoT Style Guidelines.

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.

Frame 37



AlquilaFácil

Buscar Distrito

PUBLICAR CALENDARIO NOTIFICACIONES USUARIO

Resultados de Búsqueda: Filtros

## 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 espaciado.

The image shows a mobile application interface with a light gray background. At the top, there is a large, empty white rectangular area. Below it, the word "Opciones" is displayed in a bold, black, sans-serif font. Underneath "Opciones", there are two red, bold, sans-serif links: "Ver comentarios >" and "Reportar Espacio >". Further down, there are two input fields. The first input field is labeled "Fecha y hora de inicio:" in bold black text. To its right is a large, rounded rectangular input box containing the placeholder text "dd/mm/aaaa --:--". In the top right corner of this input box is a small icon of a calendar with a date arrow. The second input field is labeled "Fecha y hora de fin:" in bold black text. To its right is a similar large input box with the placeholder text "dd/mm/aaaa --:--" and a calendar icon in the top right corner. Below these input fields is a large, orange, rounded rectangular button with the word "Reservar" in white, bold, sans-serif text.

## 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.

Frame 341

The screenshot shows a mobile application interface for 'AlquilaFácil'. At the top, there is a red header bar with the 'AlquilaFácil' logo, a search bar labeled 'Buscar Distrito' with a magnifying glass icon, and a 'PUBLICAR' button. A small red badge with the number '97' is visible in the top right corner. Below the header, the main content area has a white background. A blue rectangular box highlights the title 'Detalles del local:' and a large image of a decorated event space. To the right of the image is a blue button labeled 'Fill (1512) x 40 Hug'. On the left side of the main content area, there is a sidebar with the following information:

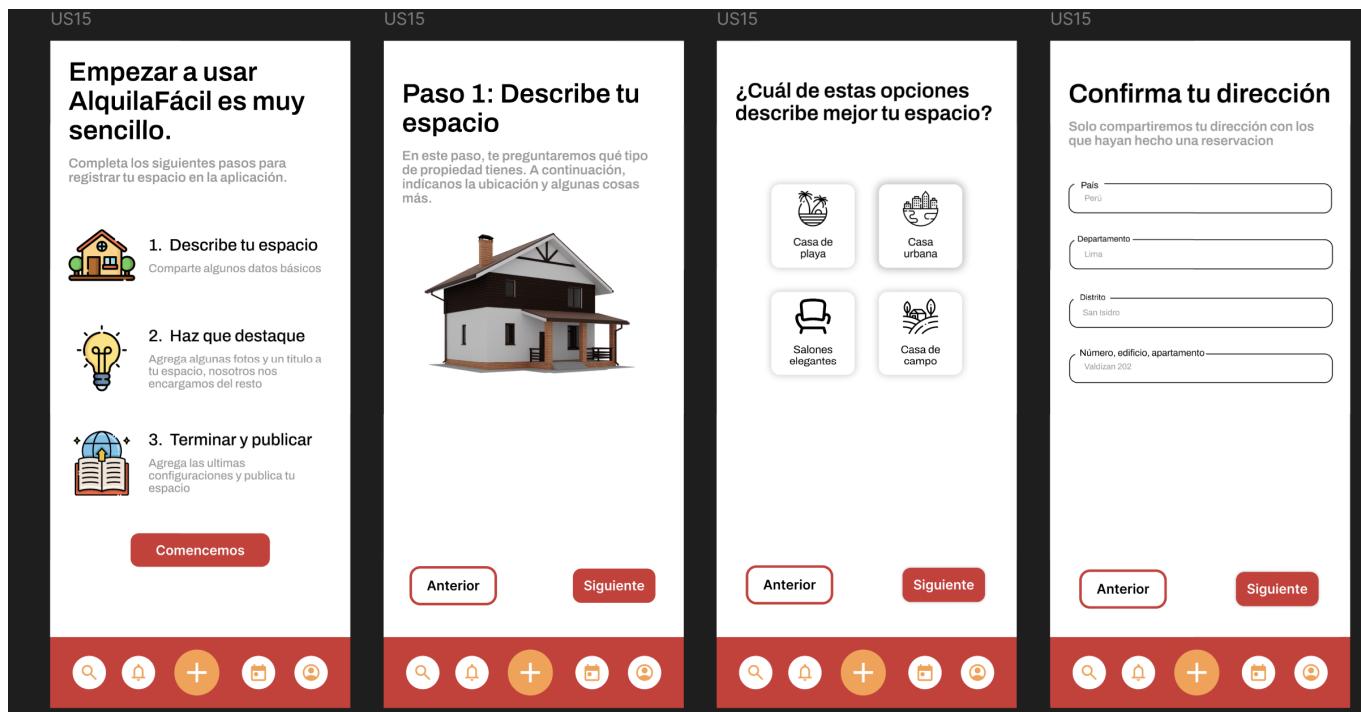
**Local 1**  
Perú, Lima  
Aforo: 10 personas  
**Propietario:** 10 personas  
**Descripción:**  
10 personas  
**Características**

## Mobile Guidelines

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

### Navbar:

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

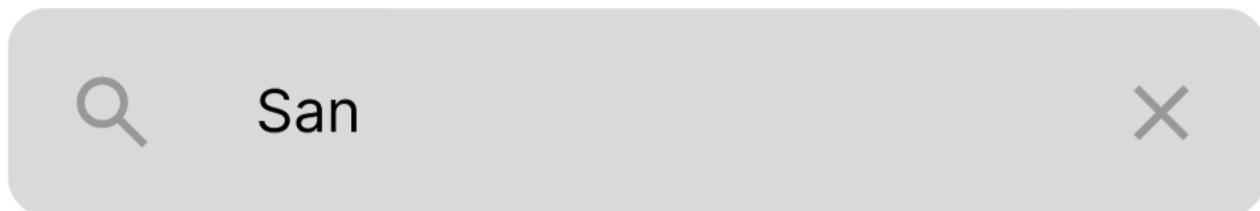


### Buttons:

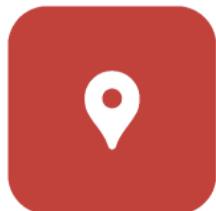
Ambas versiones tanto como para Android e iOS comparten similitudes en todos los aspectos, sin embargo hay ciertos botones por ejemplo que en la versión de iOS son enfocados más en colores Blancos o Negros, como por ejemplo:



## 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.

---

## 5.2. 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.

### 5.2.1. Organization Systems.

El sistema de organización de AlquilaFácil ha sido diseñado con el objetivo de optimizar la experiencia del usuario, permitiendo una gestión eficiente de la búsqueda y selección de espacios en alquiler. Este sistema se basa en los siguientes componentes clave:

- **Búsqueda y filtros personalizados:** La aplicación incorpora un motor de búsqueda avanzado acompañado de filtros que permiten a los usuarios refinar los resultados según criterios como ubicación, precio y características específicas del espacio.  
Esto asegura que el usuario solo vea opciones relevantes, facilitando la toma de decisiones.
- **Tarjetas de visualización de espacios:** Los espacios disponibles se muestran en formato de tarjetas visuales, donde se presenta la información más importante de manera clara y concisa.  
Cada tarjeta incluye:
  - Imagen del espacio
  - Ubicación
  - Precio
  - Opción de añadir a favoritos
- **Barra de navegación inferior:** En la parte inferior de la pantalla se encuentra una barra de navegación que agrupa las principales funciones de la aplicación, como:
  - Búsqueda
  - Notificaciones
  - Acceso al calendario o eventos
  - Gestión de reservas
- **Jerarquización de la información:** La información en la interfaz está organizada de manera que los elementos más relevantes, como el precio y la ubicación del espacio, sean inmediatamente visibles. Esto facilita al usuario la evaluación de cada opción sin necesidad de ingresar a detalles adicionales de cada anuncio.

### 5.2.2. 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.).

### 5.2.3. 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"/>  
  
<meta name="copyright" content="Copyright AlquilaFacil team" />
```

:::

#### 5.2.4. 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.

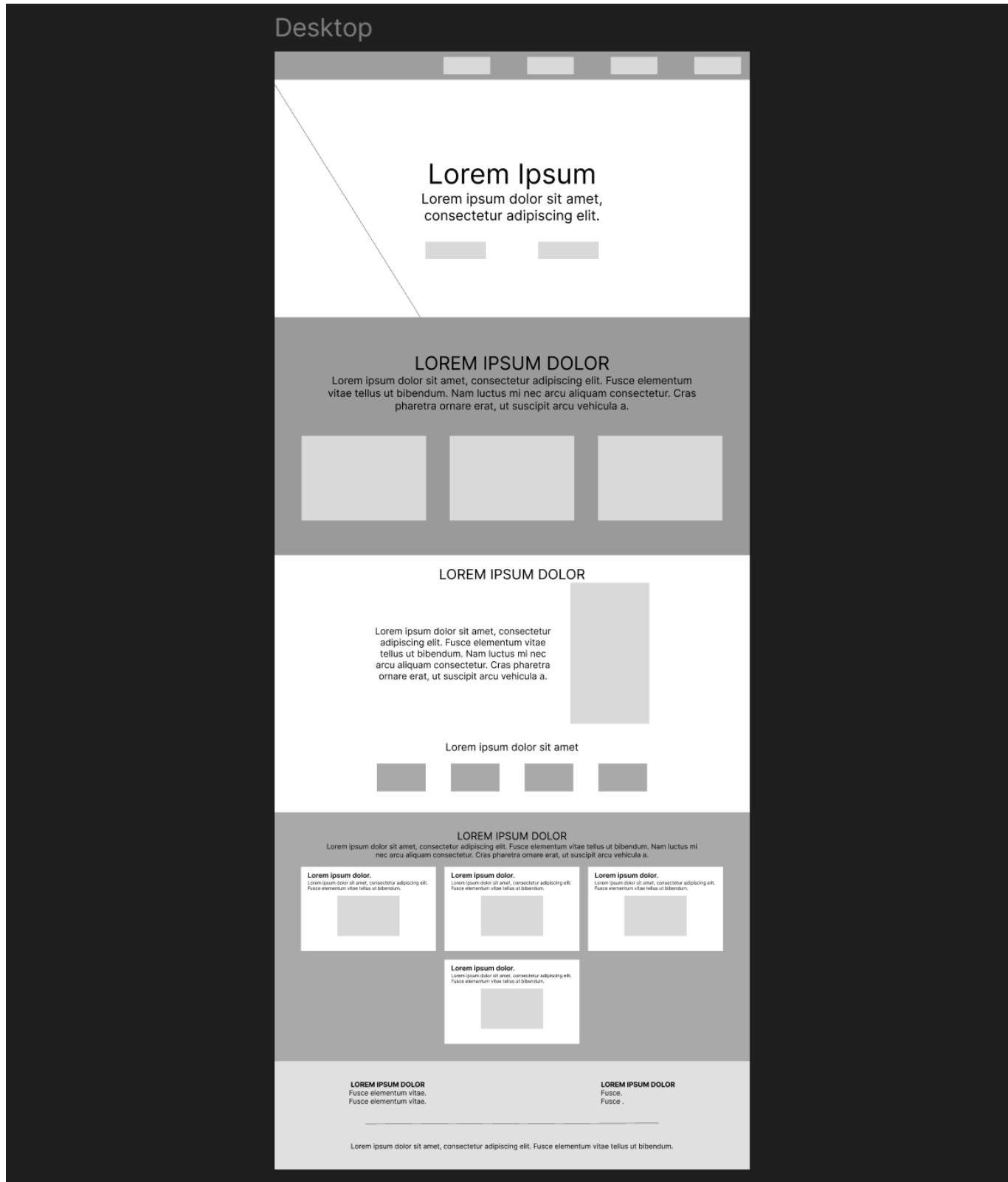
#### 5.2.5. Navigation Systems.

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:

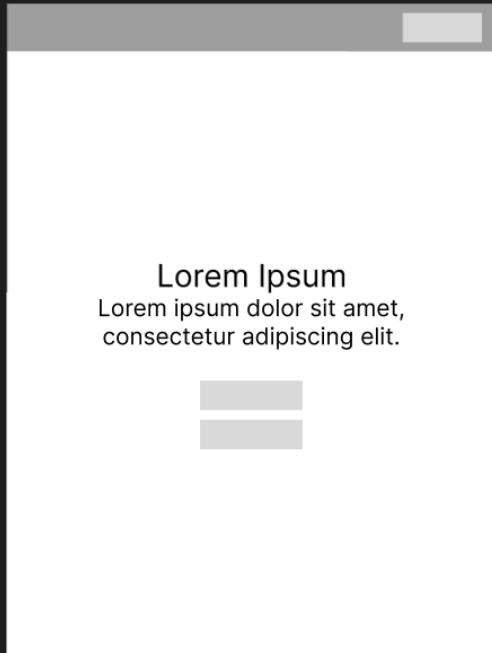
- **Barra de Navegación Inferior:** La aplicación cuenta con una barra de navegación fija en la parte inferior de la pantalla que agrupa las principales funcionalidades. Los iconos incluidos, como la lupa (búsqueda), la campana (notificaciones), el botón de "+" (añadir nuevo espacio), el calendario (gestión 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.

## 5.3. Landing Page UI Design.

### 5.3.1. Landing Page Wireframe.



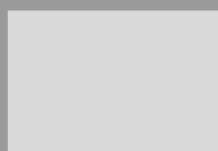
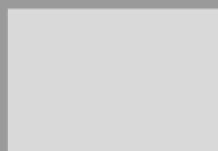
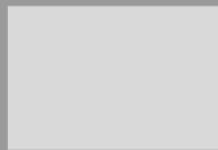
## Frame 22

**Lorem Ipsum**

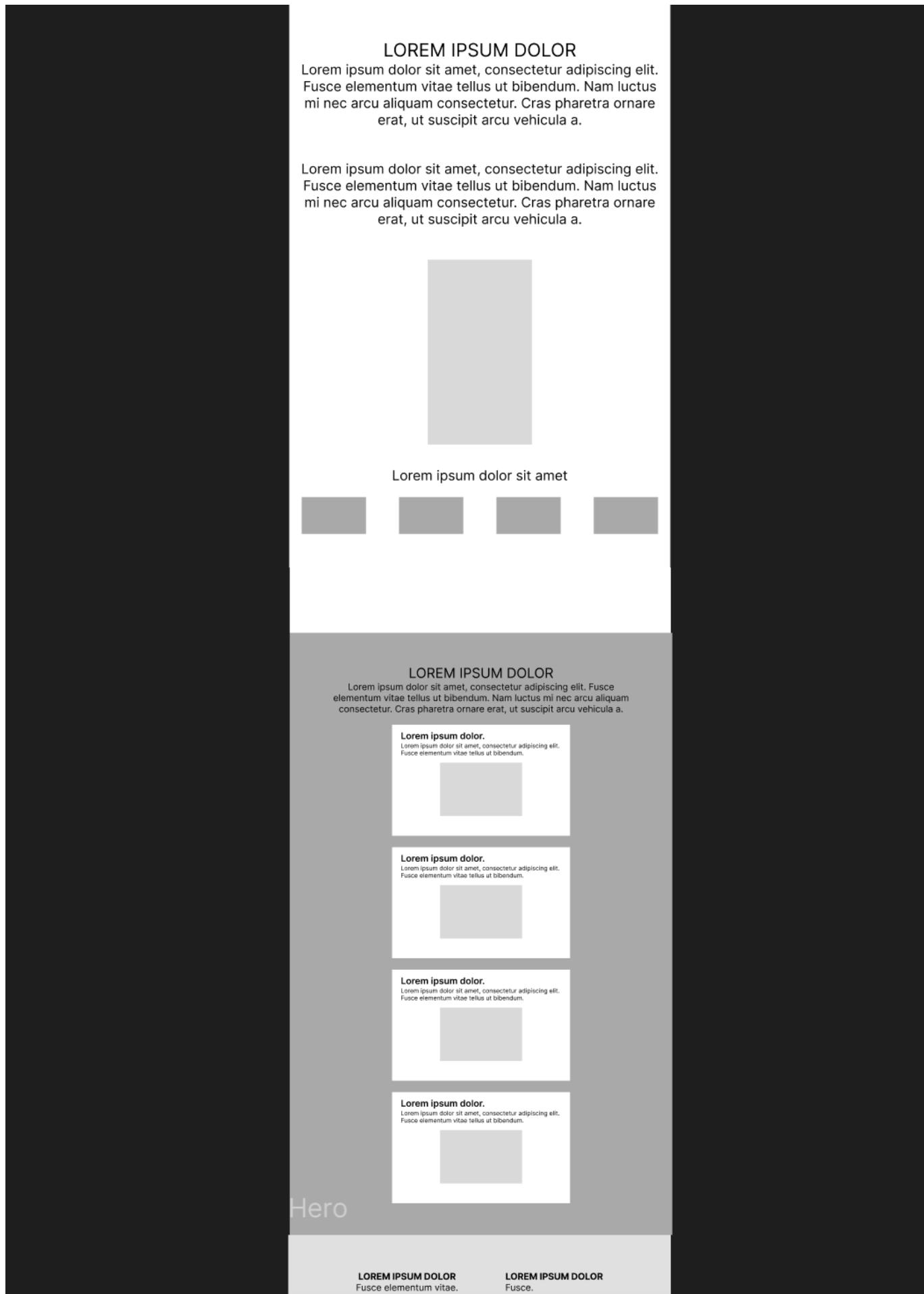
  Lorem ipsum dolor sit amet,  
  consectetur adipiscing elit.

**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.



## Frame 19



**Enlace:** <https://www.figma.com/design/iVWEVfK7bp5Ski017EYYZI/AlquilaFacil?node-id=2-259&p=f&t=KCXgjx4xgr76zKvo-0>

### 5.3.2. Landing Page Mock-up.

Desktop

The landing page features a dark header with the AlquilaFácil logo and navigation links: DESCUBRIR, PUBLICAR, BENEFICIOS, and EMPEZAR. A large smartphone on the left displays the mobile app's interface, showing event listings. The main background image shows a group of diverse people laughing and socializing over coffee in a cafe setting. The central headline reads "AlquilaFácil" in large red letters, followed by the tagline "El espacio ideal para tus eventos, al alcance de tus manos". Download links for the App Store and Google Play are present. Below the hero section, a red banner highlights "DESCUBRE ESPACIOS ÚNICOS" and describes exploring unique spaces for events. Three cards showcase specific venues: a modern event hall, a shared office space in the center, and an outdoor terrace.

**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  
100 personas  
Propietario: Hato Luna  
Descripción:  
Interior: decoración moderna perfecta  
Exterior: amplio espacio para eventos

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

Filtros	
Tipo de espacio	
<input checked="" type="checkbox"/>	Casa de campo
<input checked="" type="checkbox"/>	Casa urbana
<input checked="" type="checkbox"/>	Balnearios
<input checked="" type="checkbox"/>	Casa de fiesta

**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	
 Angelito 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 más detalles o agendar un nuevo evento

Lun	Mar	Miér	Jue	Viernes	Sáb	Domingo
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](mailto:info@alquilafacil.com)  
Teléfono: [+51 991 042 243](tel:+51991042243)

**Síguenos**  
[!\[\]\(22713a682310e863564d433f00fe1465\_img.jpg\)](#) [!\[\]\(66c74684cb02ad1b53a5264819a9d40a\_img.jpg\)](#)

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

# Mobile


AlquilaFácil
EMPEZAR

# Alquila Fácil

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](#)



Empezar a usar **AlquilaFácil** es muy sencillo.  
Completa los siguientes pasos para registrar tu espacio en la aplicación.

1. Describe tu espacio  
Completa algunos datos básicos.
2. Haz que destaque  
Agrega algunas fotos y un título atractivo, así como más información detallada.
3. Terminar y publicar  
Agrega las ofertas y condiciones y publica tu espacio.

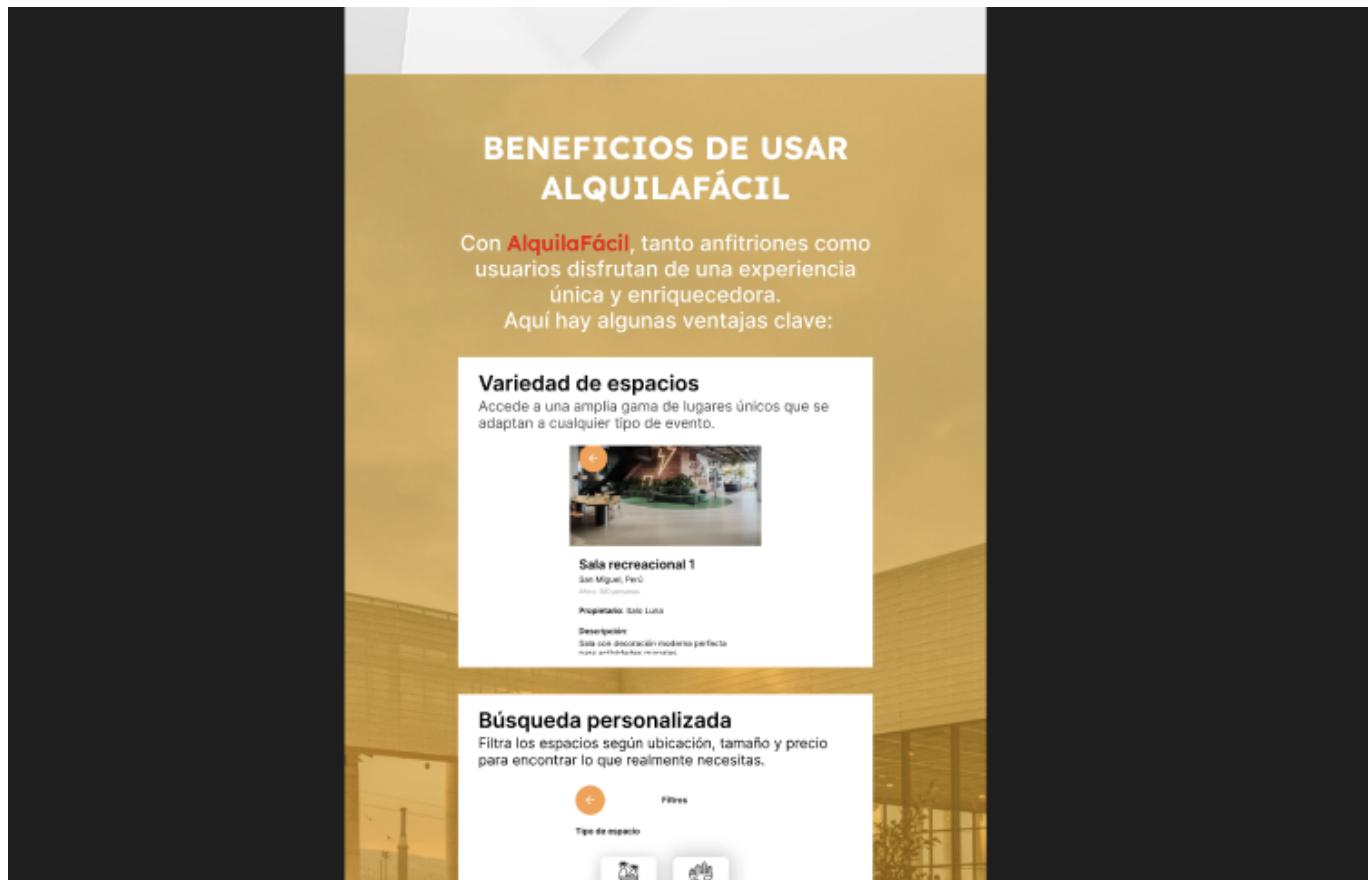
[Comenzar](#)

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

¿Qué esperas para comenzar a alquilar tus espacios?

150 / 176



**Enlace:** <https://www.figma.com/design/xgTMj9suBtUHH4bUyKpfq9/AlquilaFacil--IoT-?node-id=2-259>

## 5.4. Applications UX/UI Design.

### 5.4.1. Applications Wireframes.

The image displays five wireframe prototypes for the 'AlquilaFácil' application, each with a unique identifier and a brief description of its purpose:

- US01-1**: Shows a navigation bar with a search bar and links, followed by a 'Frame 37' wireframe which appears to be a placeholder or a loading screen.
- US02-2**: A registration form titled 'REGÍSTRATE' with fields for name, email, password, and a checkbox for terms and conditions.
- US03-AC1**: A wireframe for a search results page titled 'Resultados de búsqueda' showing a single item with a large 'X' over it.
- US03-AC2**: A wireframe for a detailed view of a local listing, showing a preview image and a form for editing details like location and price.
- US07**: A wireframe for creating a new listing, featuring sections for 'DETALLES DEL LOCAL', 'OPCIONES', and 'TIPO DE ESPACIO' with various input fields and dropdowns.

**Frame 42**

**US07**

**Frame 43**

**US08 - AC2**

**US15**

**US15**

**US09 - AC2**

**US15**

**US15**

**US15**

**US09 - AC2**

**US15**

**US15**

**US15**

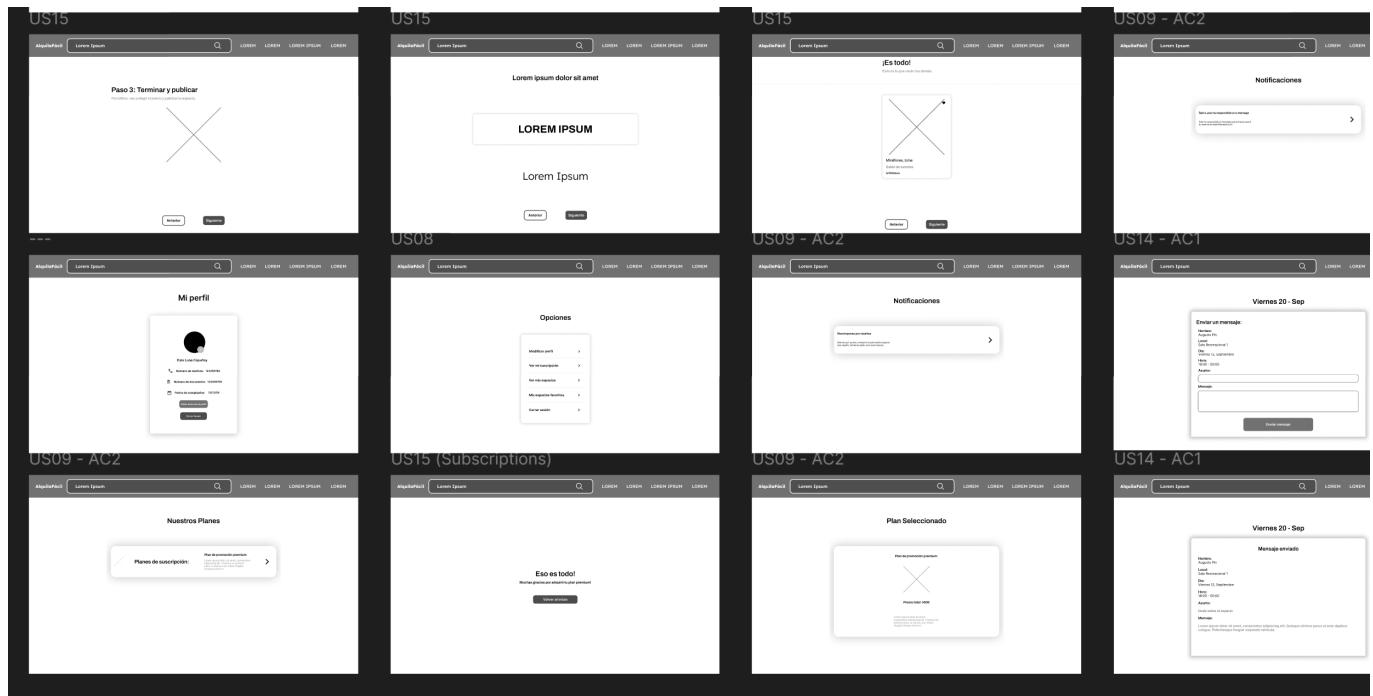
**US09 - AC2**

**US15**

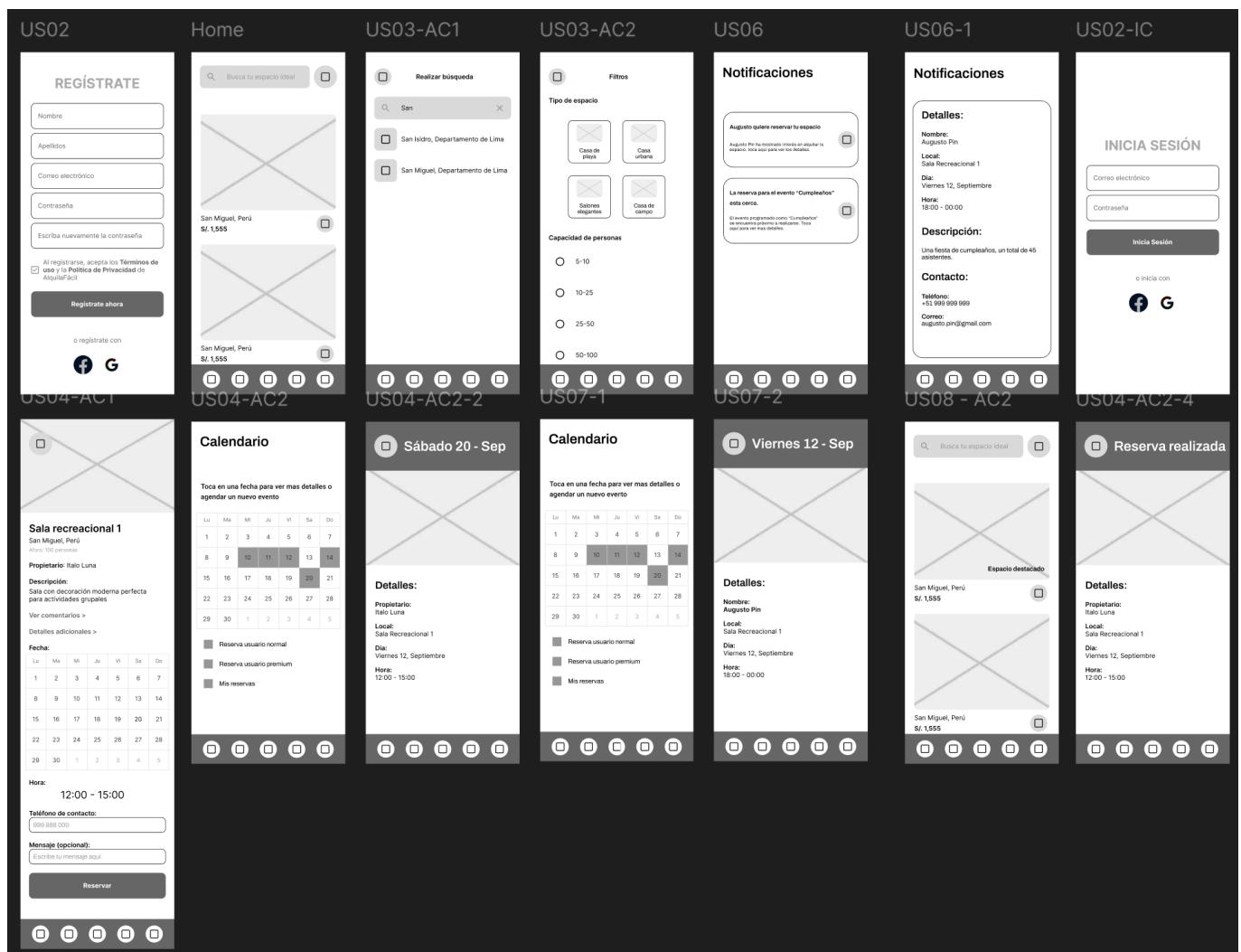
**US08**

**US09 - AC2**

**US14 - AC1**



**Enlace WebPage:** <https://www.figma.com/design/xgTMj9suBtUHH4bUyKpfq9/AlquilaFacil--IoT?-node-id=76-3769>

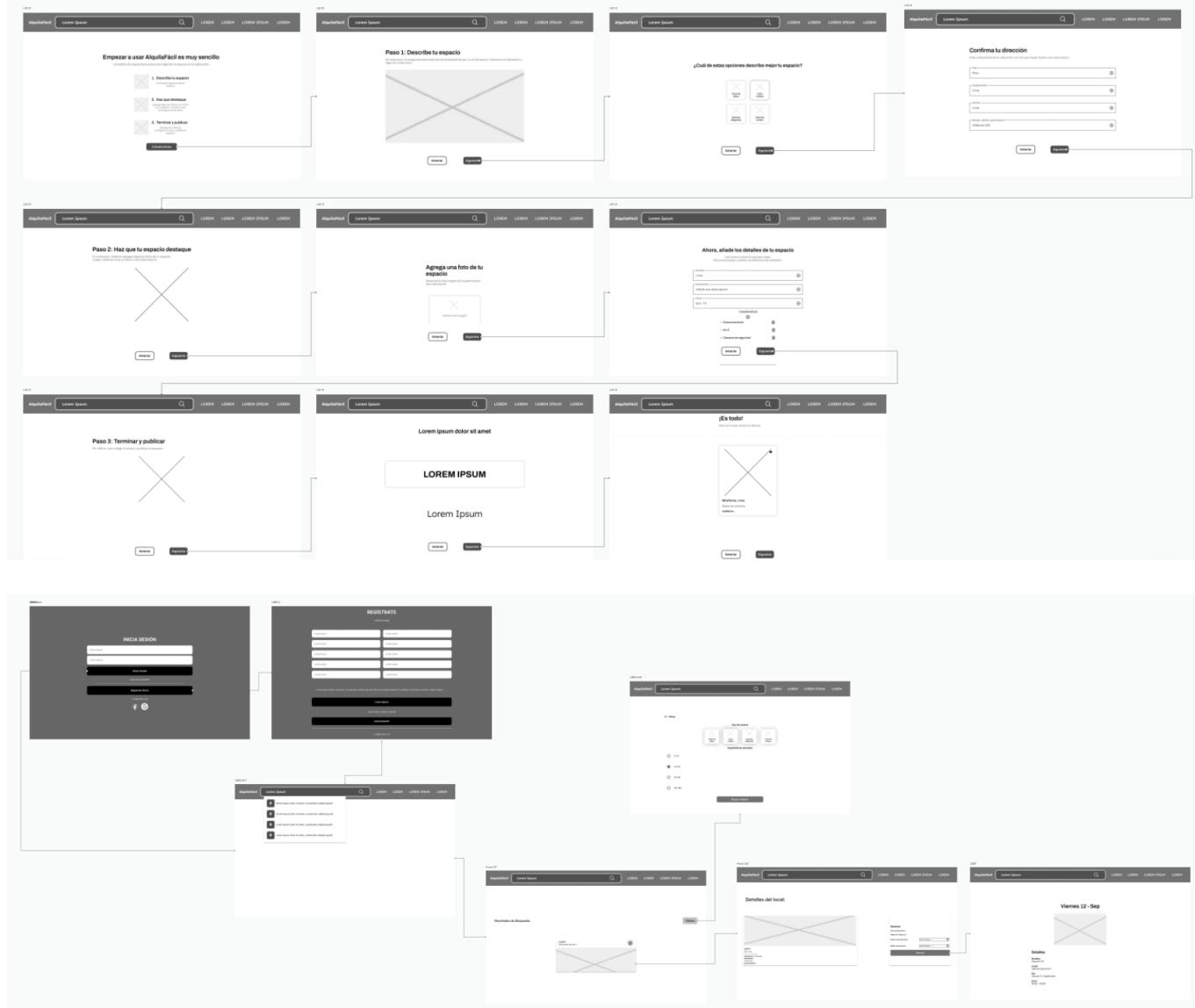




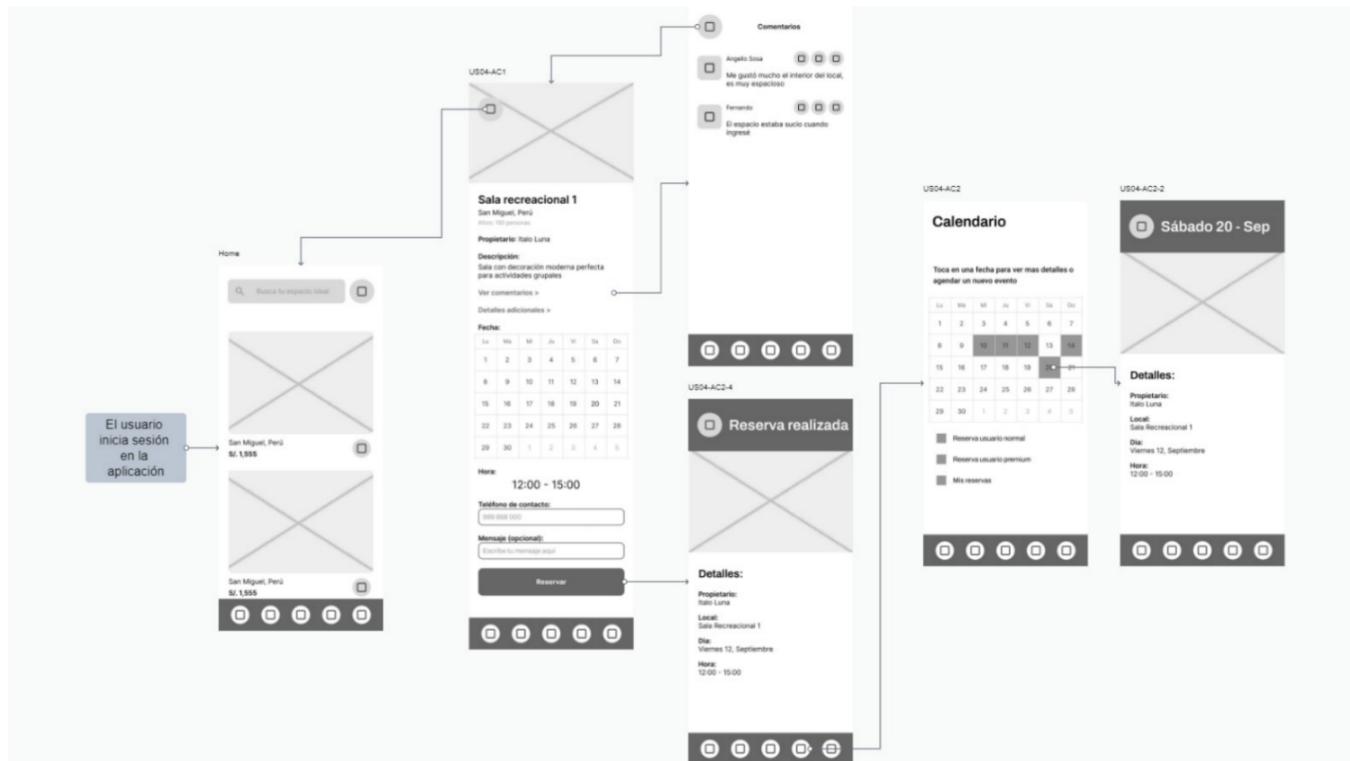
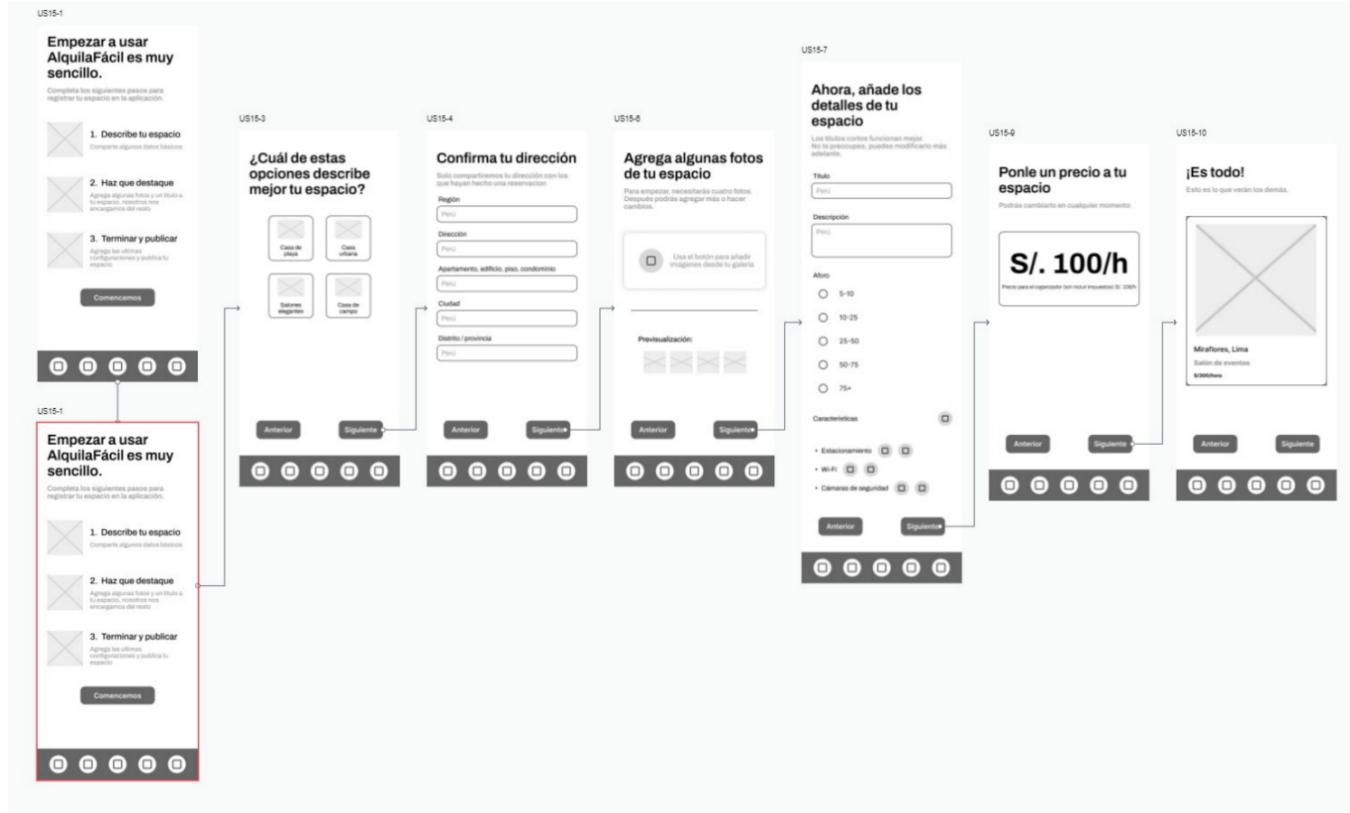
**Enlace Mobile:** <https://www.figma.com/design/xgTMj9suBtUHH4bUyKpfq9/AlquilaFacil--IoT?-node-id=76-3768>

#### 5.4.2. Applications Wireflow Diagrams.

#### WebPage



## Android



#### 5.4.2. Applications Mock-ups.

**USUZ-1**

**USUZ-2**

**Frame 37**

**US03-AC1**

**Frame 341**

**US05-AC2**

**Frame 42**

**US07**

**Frame 43**

**US08 - AC2**

**US15**

**US15**

**US9 - AC2**

**US15**

**US15**

**US15**

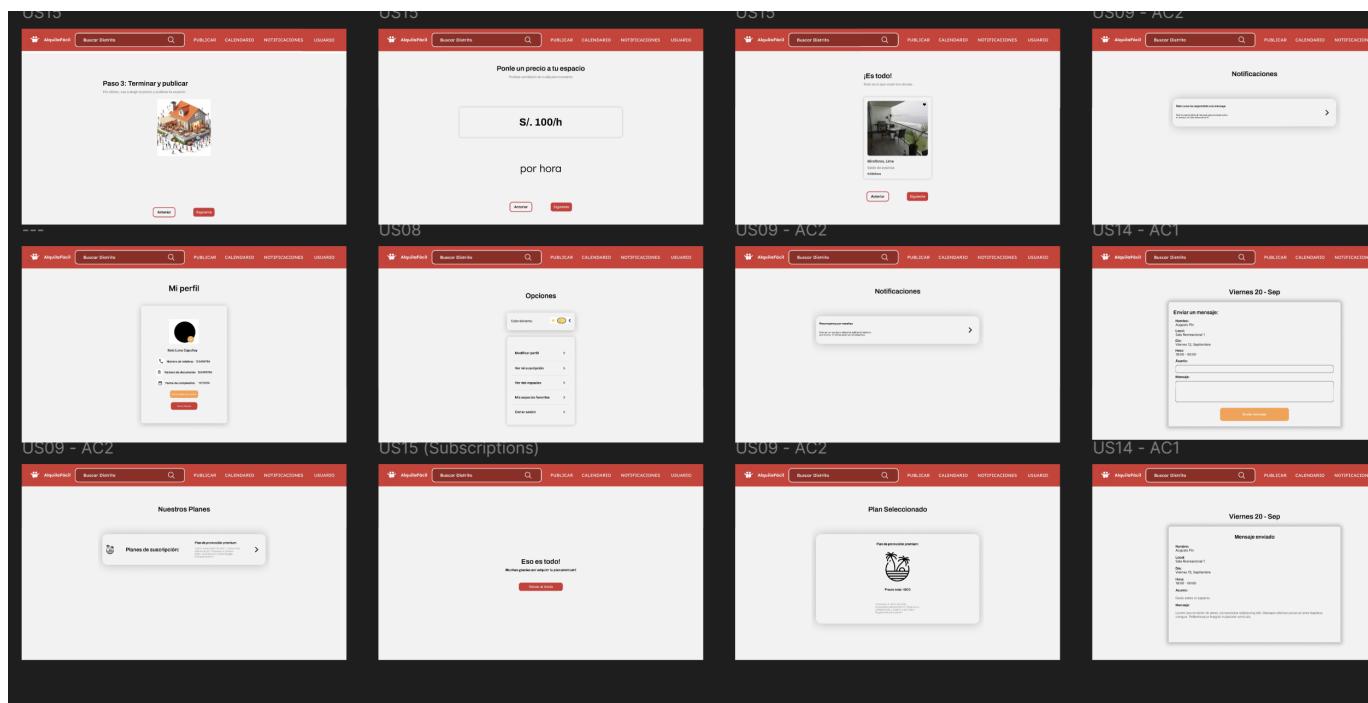
**US9 - AC2**

**US15**

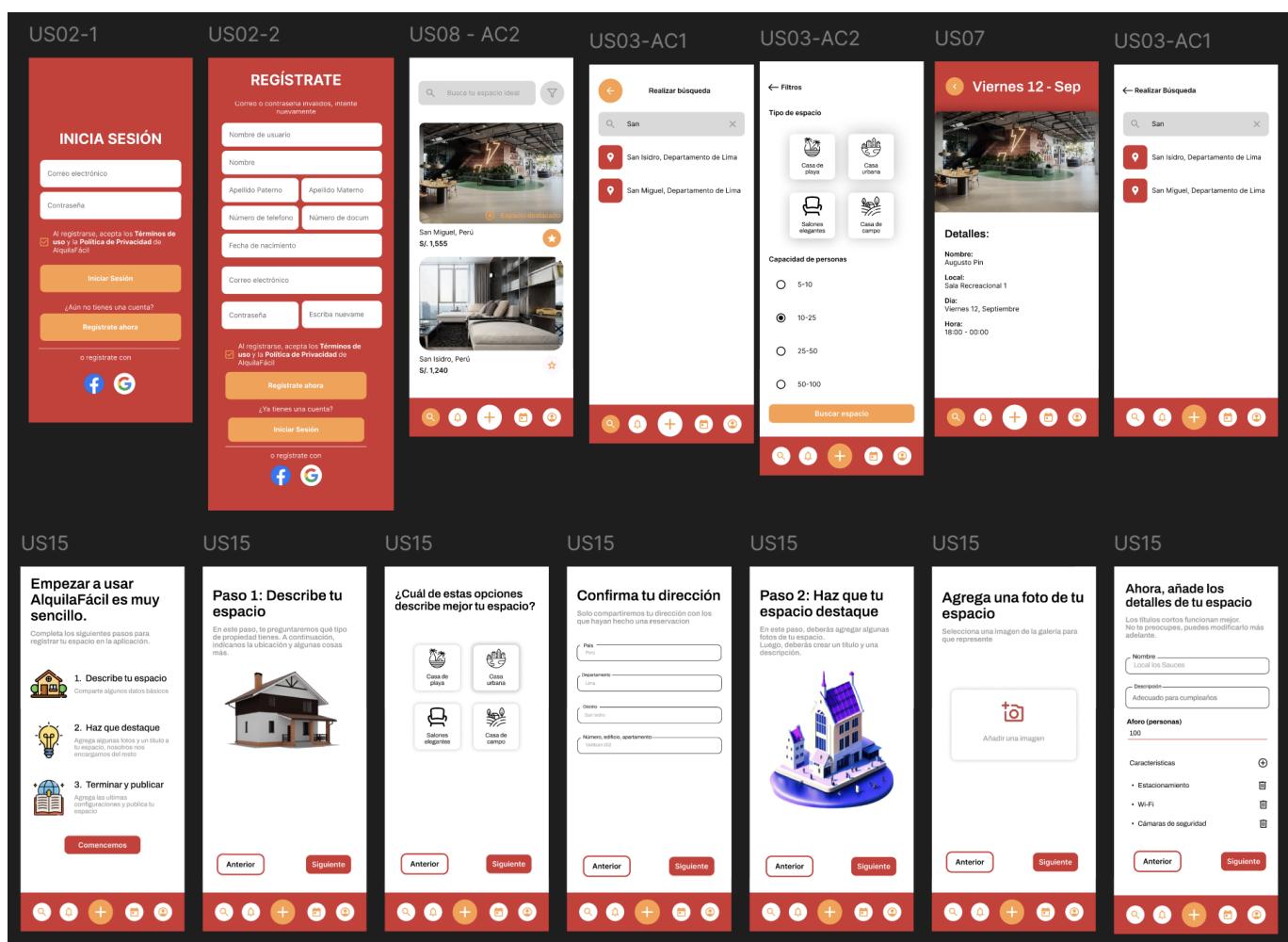
**US15**

**US15**

**US09 - AC2**



**Enlace WebPage:** <https://www.figma.com/design/xgTMj9suBtUHH4bUyKpfq9/AlquilaFacil--IoT?-node-id=76-3767>



**US15** **US15** **US15 B** **US15 B** **US15 B**

**US08** **US11-AC01** **US14** **US05-AC1** **US09 - AC2** **US14 - AC1**

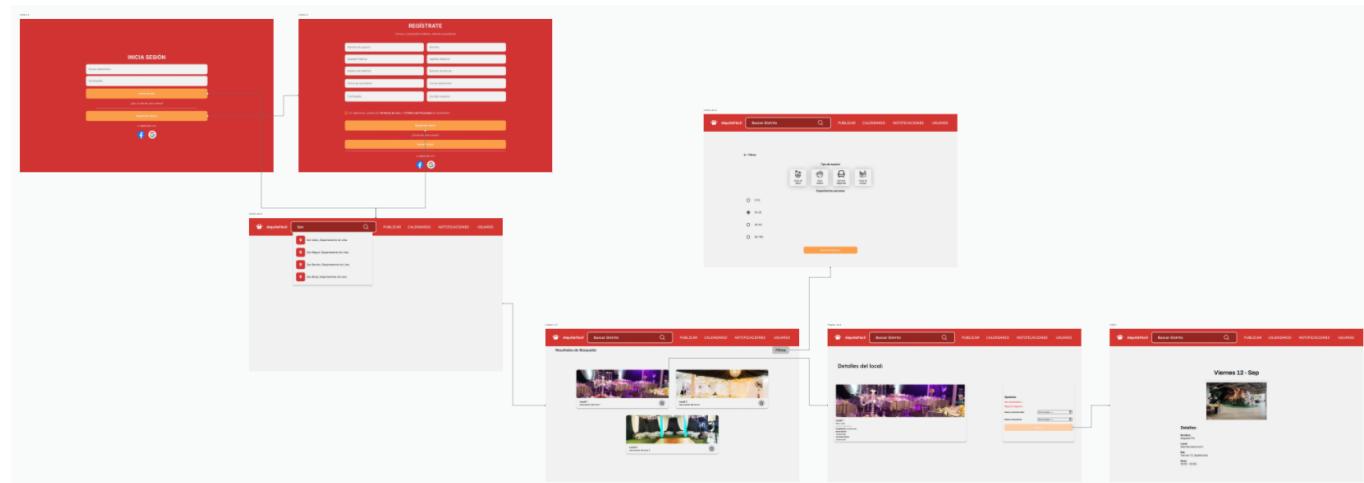
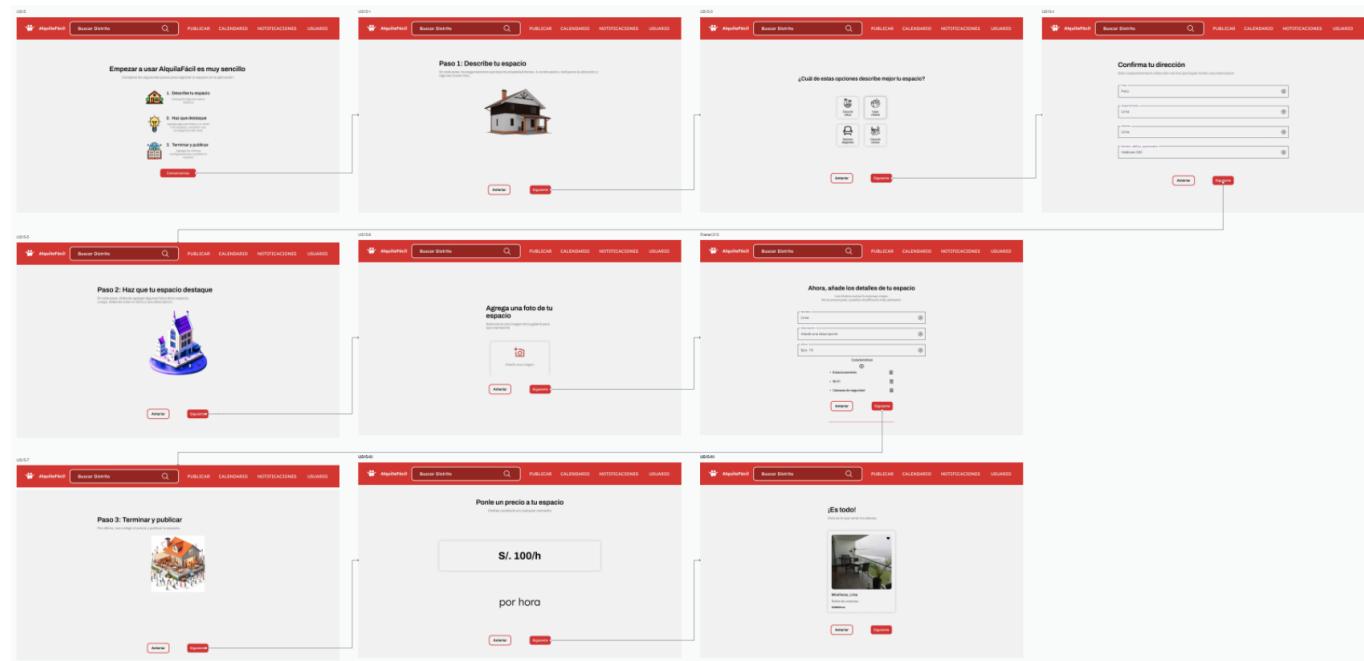
**US14 - AC1** **US14 - AC2** **US19** **US15 (Subscri...)** **US15 (Subscri...)** **US15 (Subscri...)**

**Home** **US06** **US04-AC1** **US04-AC2** **US03-AC2** **US14**

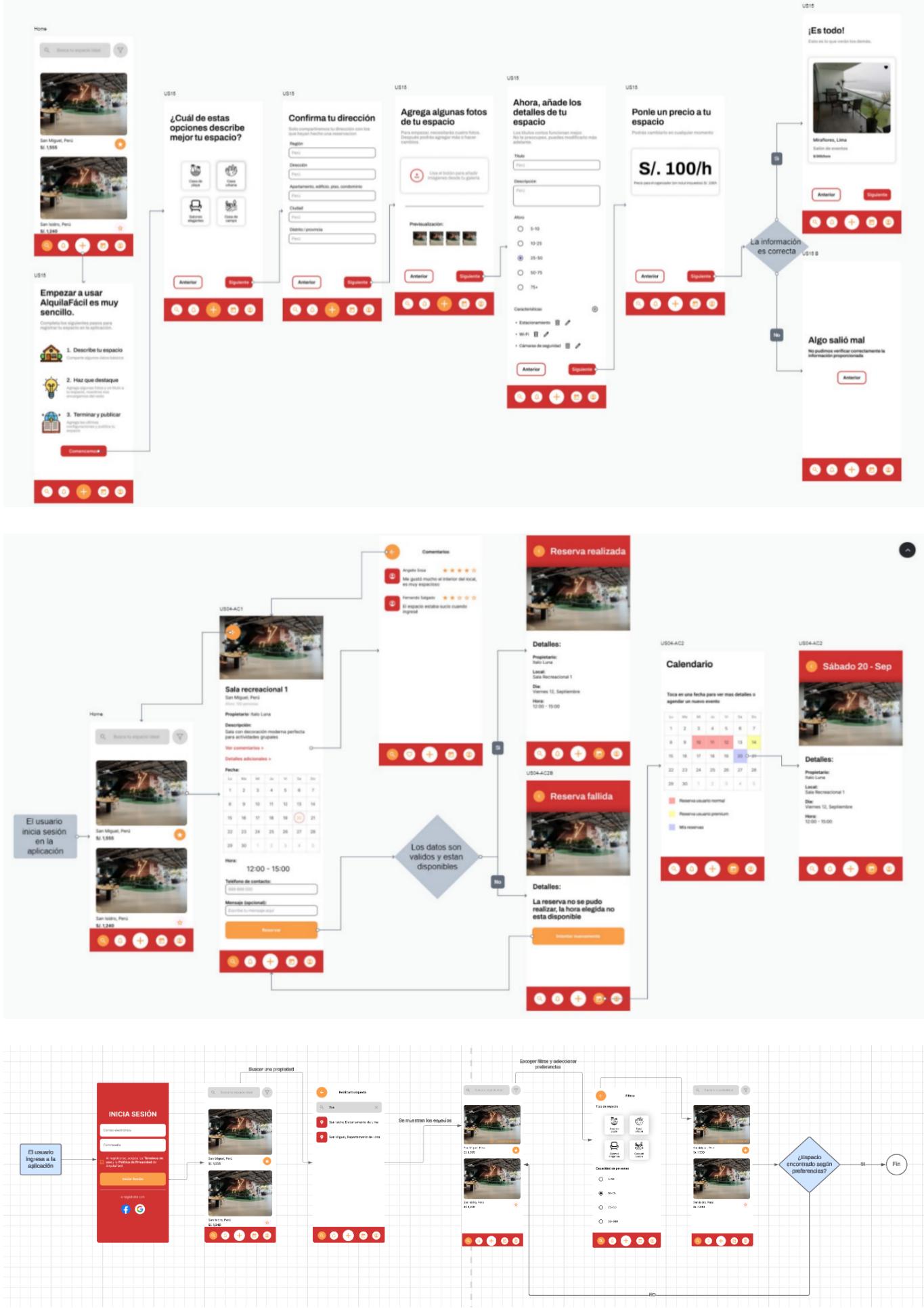
**Enlace Mobile:** <https://www.figma.com/design/xgTMj9suBtUHH4bUyKpfq9/AlquilaFacil--IoT-?node-id=72-5>

### 5.4.3. Applications User Flow Diagrams.

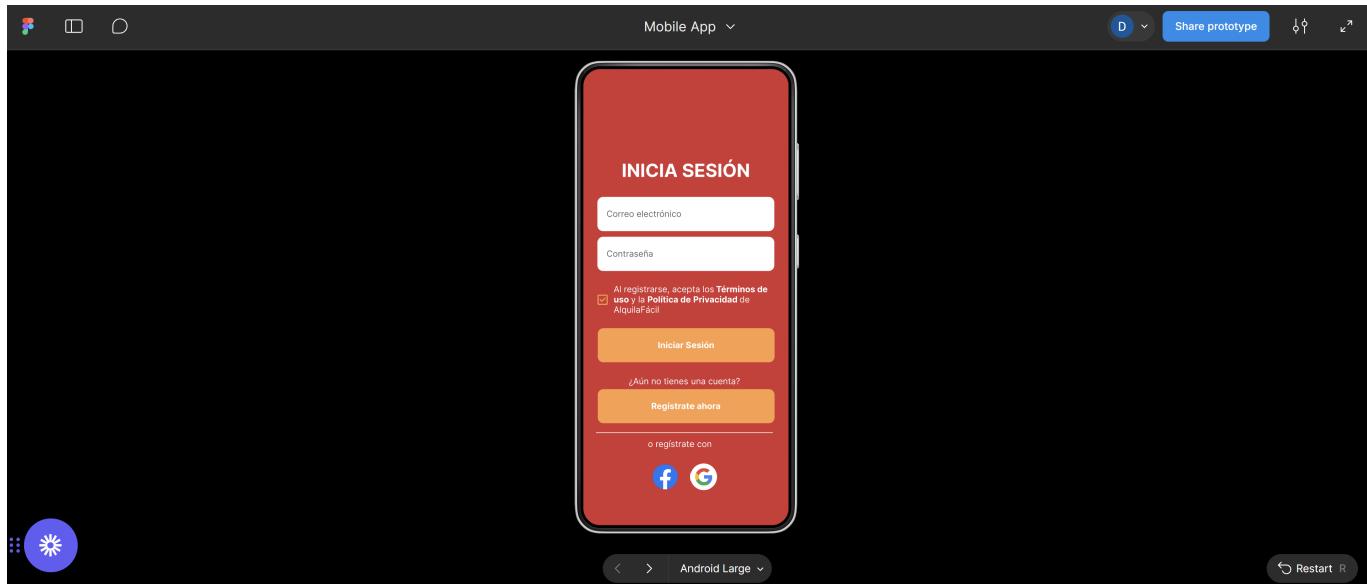
#### Web Page



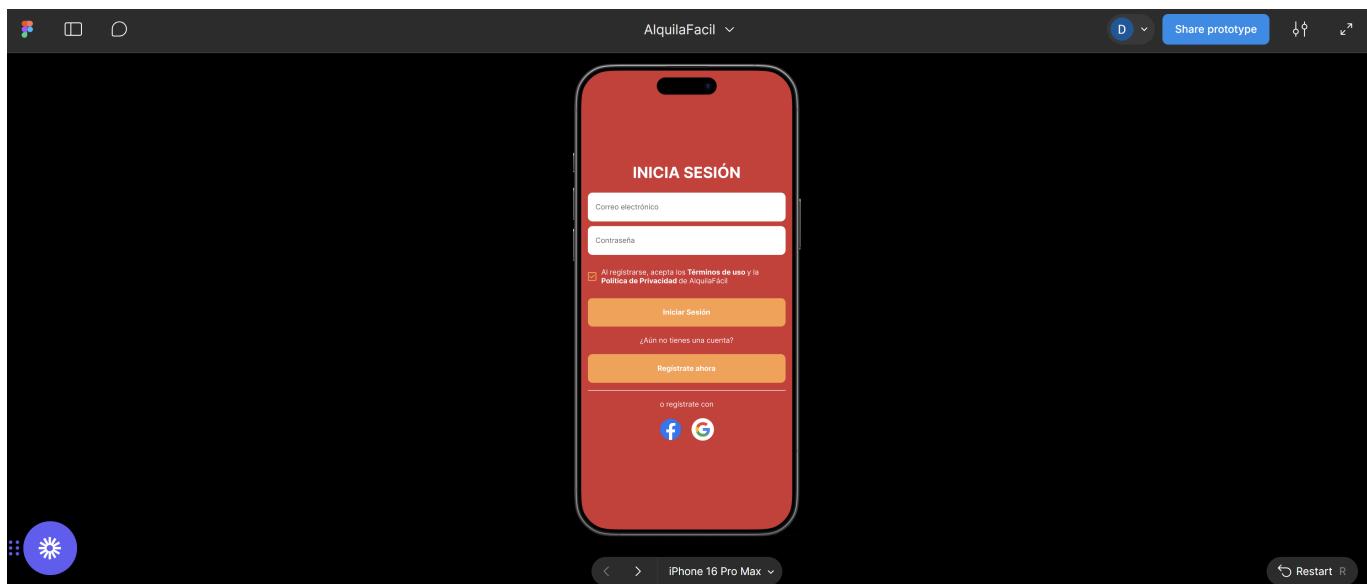
#### Android



## 5.5. Applications Prototyping.



**Enlace Mobile Android:** <https://www.figma.com/proto/xgTMj9suBtUHH4bUyKpfq9/AlquilaFacil--IoT-?node-id=105-4793&p=f&t=KcylhvbiCsSOhkGb-0&scaling=scale-down&content-scaling=fixed&page-id=72%3A3&starting-point-node-id=105%3A4793>



**Enlace Mobile IOS:** <https://www.figma.com/proto/xgTMj9suBtUHH4bUyKpfq9?page-id=76%3A517&node-id=76-517&p=f&t=wbbhV2RsUAI2DKQg-0&scaling=min-zoom&content-scaling=fixed&starting-point-node-id=76%3A1466>

**Enlace WebPage:** <https://www.figma.com/proto/xgTMj9suBtUHH4bUyKpfq9?page-id=76%3A3767&node-id=76-3767&p=f&t=wbbhV2RsUAI2DKQg-0&scaling=scale-down&content-scaling=fixed&starting-point-node-id=2002%3A8410>

---

## Capítulo VI: Product Implementation, Validation & Deployment.

### 6.1. Software Configuration Management.

### 6.1.1. 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.
	MySQL Workbench 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.
	Node.js Entorno de ejecución JavaScript utilizado para el desarrollo de funcionalidades del backend de forma escalable.
	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.
	VS Code Editor de código fuente ligero y extensible que facilitó el desarrollo del sistema con múltiples extensiones útiles.
	Vue.js Framework progresivo para construir interfaces de usuario dinámicas y componentes reutilizables en el frontend.
Software Deployment	Netlify Servicio en la nube que permitió desplegar la landing page del sistema de forma automática, continua y con URL pública.

### 6.1.2. Source Code Management.

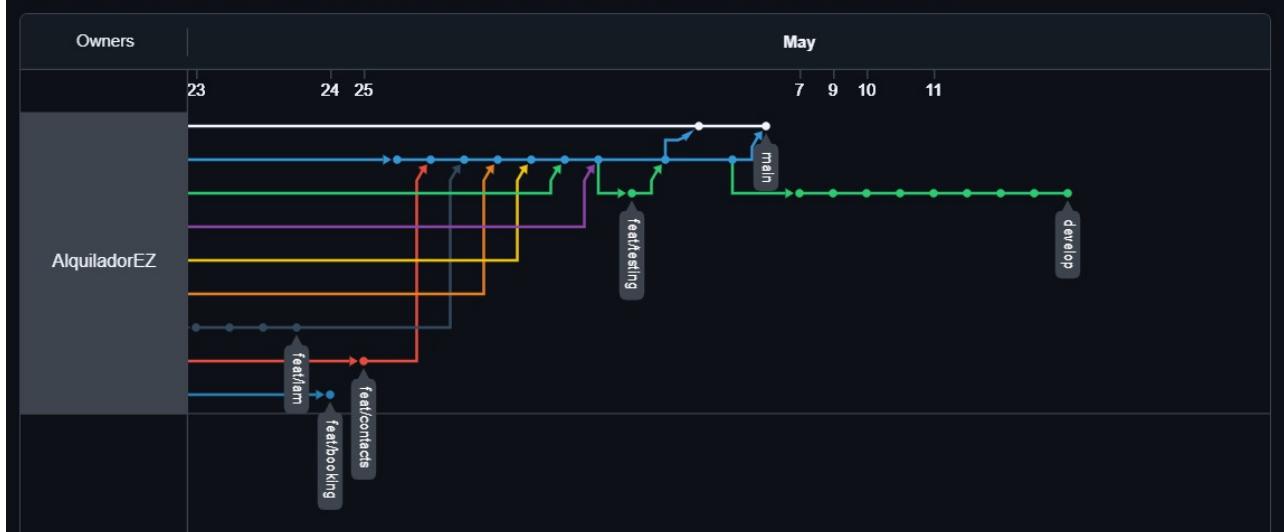
Se empleó la metodología Git Flow con el objetivo de mantener un mayor control sobre la gestión del proyecto y permitir el desarrollo simultáneo tanto de la implementación de la aplicación como de la elaboración del informe.

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.

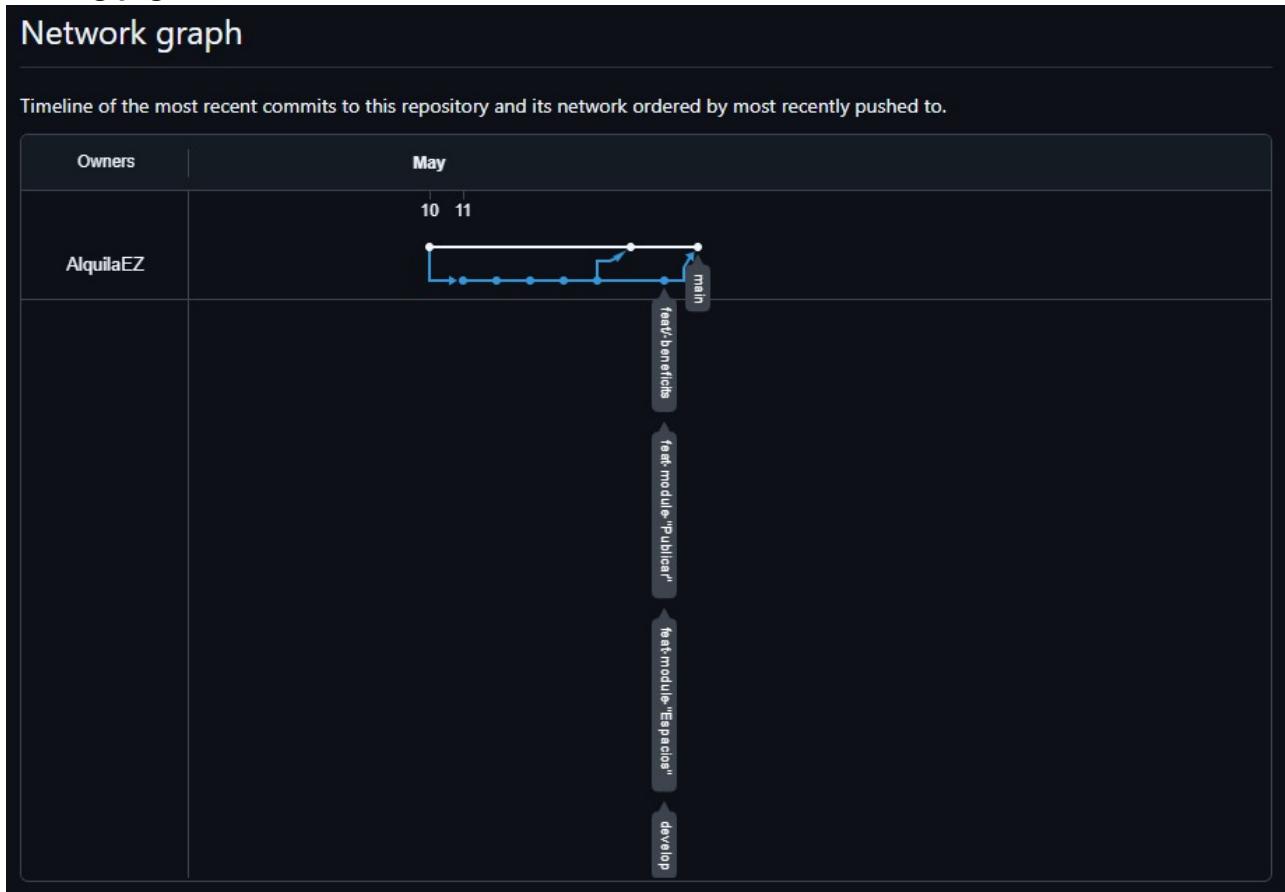
- **Web Application**

#### Network graph

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



- **Landing page**



URL del repositorio del Report en GitHub: <https://github.com/AlquilaEZ/alquilafacil-document-report>

URL del repositorio del Landing Page en GitHub: <https://github.com/AlquilaEZ/alquilafacil-landing-page>

URL del repositorio del Frontend en Github: <https://github.com/AlquiladorEZ/AlquilaFacil-FrontEnd>

### 6.1.3. Source Code Style Guide & Conventions.

HTML es un lenguaje de marcado utilizado para estructurar el contenido de una página web. A través de etiquetas, permite definir distintos tipos de elementos como textos, imágenes, videos, entre otros.

Convenciones comunes en HTML:

- La primera línea del archivo debe incluir la declaración del tipo de documento.
- Las etiquetas deben escribirse siempre en minúsculas, lo que favorece la claridad y uniformidad del código.
- Es recomendable usar comillas al asignar valores a los atributos para mejorar la legibilidad.

CSS es un lenguaje diseñado para aplicar estilos a los elementos HTML. Permite crear interfaces visuales más atractivas y usables para los usuarios.

Convenciones comunes en CSS:

- Utilizar guiones para separar palabras en nombres de clases compuestas (por ejemplo: .box-container) mejora la lectura.

- Preferir la unidad de medida rem favorece la adaptabilidad del diseño a diferentes tamaños de pantalla, promoviendo un diseño responsive.

#### 6.1.4. Software Deployment Configuration.

Para llevar a cabo el despliegue de la landing page, se utilizaron diversas herramientas enfocadas en la gestión de versiones y el control del flujo de trabajo del equipo.

- **Git:** Utilizado como sistema de control de versiones, permitió registrar los cambios realizados en el proyecto y facilitar la colaboración entre los integrantes del equipo.
- **GitHub:** Plataforma en la que se alojó el repositorio del proyecto. Allí se almacenaron todas las versiones del código, 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.

### 6.2. Landing Page, Services & Applications Implementation.

#### 6.2.1. Sprint 1

##### 6.2.1.1. Sprint Planning 1

Sprint #	Sprint 1
<b>Sprint Planning Background</b>	
Date	07/05/2025
Time	12:00
Location	Modalidad remota por Discord
Prepared By	Lecca Choccare Christopher Bryan
Attendees (to planning meeting)	<ul style="list-style-type: none"> <li>- Ames Oviedo Mariano Jose</li> <li>- Avalos Santos Anthony Piero</li> <li>- Lecca Choccare Christopher Bryan</li> <li>- Sosa Colca Angello Rodolfo</li> <li>- Carlos Oswaldo Casimiro Fernandez</li> </ul>
<b>Sprint Goal &amp; User Stories</b>	
Sprint 1 Velocity	68
Sum of Story Points	68

#### 6.2.1.2. 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	IAM Service	Locals Service	Booking Service	Frontend Web	Landing Page

Team Member (Last Name, First Name)	GitHub Username	IAM Service	Locals Service	Booking Service	Frontend Web	Landing Page
Ames Oviedo, Mariano	MarianoAmes	C	C	C	C	L
Avalos Santos, Anthony Piero	Anthony2000AA	C	C	C	C	C
Lecca Choccare, Christopher Bryan	ChrisByBits	L	L	L	L	C
Sosa Colca, Angello Rodolfo	AngelloSosa	C	C	C	C	C
Casimiro Fernández, Carlos Osvaldo	Sigma42O	C	C	C	C	C

### 6.2.1.3. 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	Tarea técnica para registrar usuario	Implementar funcionalidad: Registrar usuario	6	Mariano Ames Oviedo	Done
US02	Iniciar sesión	T-02-1	Tarea técnica para iniciar sesión	Implementar funcionalidad: Iniciar sesión	6	Anthony Avalos Santos	Done

User Story ID	US Title	Task ID	Task Title	Description	Estimation (Hours)	Assigned To	Status
US03	Registrar espacios	T-03-1	Tarea técnica para registrar espacios	Implementar funcionalidad: Registrar espacios	16	Angello Sosa Colca	Done
US04	Buscar espacios disponibles	T-04-1	Tarea técnica para buscar espacios disponibles	Implementar funcionalidad: Buscar espacios disponibles	10	Carlos Casimiro Fernández	Done
US05	Filtrar espacios disponibles	T-05-1	Tarea técnica para filtrar espacios disponibles	Implementar funcionalidad: Filtrar espacios disponibles	10	Christopher Lecca Choccare	Done
US06	Visualizar información del espacio	T-06-1	Tarea técnica para visualizar información	Implementar funcionalidad: Visualizar información del espacio	10	Mariano Ames Oviedo	Done
US07	Reservar espacios	T-07-1	Tarea técnica para reservar espacios	Implementar funcionalidad: Reservar espacios	16	Anthony Avalos Santos	Done
US12	Visualizar espacios propios publicados	T-12-1	Tarea técnica para ver espacios publicados	Implementar funcionalidad: Visualizar espacios propios	6	Mariano Ames Oviedo	Done
US13	Modificar espacios publicados	T-13-1	Tarea técnica para modificar espacios	Implementar funcionalidad: Modificar espacios publicados	6	Anthony Avalos Santos	Done
US24	Explorar landing informativa	T-24-1	Tarea técnica para landing informativa	Implementar funcionalidad: Landing clara e informativa	10	Christopher Lecca Choccare	Done
US25	Navegar por secciones desde el header	T-25-1	Tarea técnica para navegación	Implementar funcionalidad: Navegación por secciones	10	Anthony Avalos Santos	Done

User Story ID	US Title	Task ID	Task Title	Description	Estimation (Hours)	Assigned To	Status
US26	Ver beneficios diferenciados	T-26-1	Tarea técnica para beneficios	Implementar funcionalidad: Tarjetas de beneficios	10	Anthony Avalos Santos	Done

#### 6.2.1.4. 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)
AlquilaEZ/alquilafacil-landing-page	/main	83052f0	Merge pull request #2 from develop	Fusión final desde la rama de desarrollo	11/05/2025
	/main	090b1b7	add: configure tailwind deploy	Configuración para despliegue con TailwindCSS	11/05/2025
	/main	0c11e67	update: text from module benefics	Actualización de contenido textual en módulo de beneficios	11/05/2025
	/main	85ed349	update: modify images from all modules	Sustitución de imágenes en todos los módulos de la landing	11/05/2025
	/main	b0a8f6c	feat: new cards functions	Agregado de nuevas tarjetas funcionales para la landing	10/05/2025
	/main	13dc960	Created project first version	Creación del primer prototipo funcional del proyecto	10/05/2025
	/develop	53fba97	update: diseño from módulo principal	Ajustes de diseño en la sección principal	11/05/2025

<b>Repository</b>	<b>Branch</b>	<b>Commit ID</b>	<b>Commit Message</b>	<b>Commit Message Body</b>	<b>Committed on (Date)</b>
	/develop	1fa7c20	update: design module spaces	Actualización del diseño en módulo de espacios	11/05/2025

- **Web Application**

<b>Repository</b>	<b>Branch</b>	<b>Commit ID</b>	<b>Commit Message</b>	<b>Commit Message Body</b>	<b>Committed on (Date)</b>
AlquilaEZ/alquilafacil-web-app	/US01	14097e9	Merge pull request #2 from AlquilaEZ/US07	Integración del módulo de filtros con reservas	12/05/2025
	/US01	d7d9bc3	update: module reservation spaces design	Diseño actualizado del módulo de reservas	12/05/2025
	/US02	731d1d1	feat: module sign-in completed	Funcionalidad de inicio de sesión completada	12/05/2025
	/US05	000115a	feat: add filters search	Implementación de filtros de búsqueda	13/05/2025
	/US13	27974de	feat: add locals modification model	Modelo para modificar locales implementado	13/05/2025
	/US12	cad8238	feat: reservation of spaces init module	Inicio del módulo de reservas de espacios	12/05/2025
	/US07	d7d9bc3	update: module reservation spaces design	Mejoras en la presentación del módulo de reservas	12/05/2025
	/US06	cad8238	feat: reservation of spaces init module	Implementación del inicio del módulo de reservas	12/05/2025

Repository	Branch	Commit ID	Commit Message	Commit Message Body	Committed on (Date)
	/US04	2917700	Update ReservationDetails.page.vue	Actualización en la vista de detalles de reserva	14/05/2025
	/US04	7880321	Update ReservationDetails.page.vue	Corrección de errores en la vista de reserva	14/05/2025
	/main	b5544ea	Create first project	Estructura base del proyecto inicializada	10/05/2025

#### 6.2.1.5. Testing Suite Evidence for Sprint Review.

#### 6.2.1.6. Execution Evidence for Sprint Review.

Como resultado del primer Sprint, se presenta el despliegue de la Landing Page, así como también la primera versión de la Web Application:

**Landing Page:** <https://guileless-gaufre-89df8f.netlify.app/src/#hero>



**Web Application:** ACA VA LA IMAGEN DE LA WEB APP DESPLEGADA

#### 6.2.1.7. Services Documentation Evidence for Sprint Review.

A continuación se presentan los enlaces a los productos desplegados durante el Sprint 1, los cuales corresponden a los principales servicios desarrollados:

Servicio	Descripción	URL de despliegue
----------	-------------	-------------------

Servicio	Descripción	URL de despliegue
Landing Page	Sitio informativo de AlquilaFácil con navegación	<a href="https://guileless-gaufre-89df8f.netlify.app/src/#hero">https://guileless-gaufre-89df8f.netlify.app/src/#hero</a>
Aplicación Web	Plataforma operativa para arrendadores y arrendatarios	(pendiente)

#### 6.2.1.8. Software Deployment Evidence for Sprint Review.

Durante el Sprint 1 se desarrollaron y desplegaron dos componentes principales del sistema: la Landing Page informativa y la interfaz Frontend Web de la aplicación. A continuación, se describen las tecnologías utilizadas y los procedimientos aplicados para su control de versiones y despliegue.

#### Tecnologías utilizadas

- **HTML5:** Lenguaje de marcado utilizado en la construcción de la estructura de la Landing Page. Permite definir secciones como encabezados, párrafos, imágenes, botones, etc.
- **Tailwind CSS:** Framework de utilidad para aplicar estilos de manera rápida y responsive en la Landing Page, facilitando un diseño moderno sin necesidad de escribir CSS personalizado extenso.
- **Vue.js:** Framework progresivo de JavaScript utilizado para el desarrollo del Frontend Web de la aplicación. Permite crear componentes reutilizables y gestionar el estado de forma eficiente.
- **JavaScript:** Lenguaje de programación usado junto a Vue para controlar interacciones del usuario y lógica de interfaz.
- **Git:** Sistema de control de versiones que facilitó el trabajo colaborativo del equipo. Permitió mantener un historial ordenado de los cambios y trabajar simultáneamente en diferentes funcionalidades.
- **GitHub:** Plataforma de alojamiento de repositorios que permitió la gestión centralizada del código fuente, las ramas y los pull requests del equipo.
- **Netlify:** Plataforma usada para desplegar la Landing Page del proyecto, permitiendo una entrega continua y acceso público desde un dominio gratuito.

#### Control de versiones con Git Flow

Para mantener un flujo de trabajo estructurado, se implementó la metodología **GitFlow Workflow**. Esta estrategia permitió trabajar de manera ordenada mediante ramas específicas:

- **main:** Contiene la versión estable y lista para producción.
- **develop:** Rama donde se integran las funcionalidades en desarrollo antes de ser validadas y fusionadas.
- Ramas de características: Para el desarrollo individual de funcionalidades (por ejemplo, landing modules, navegación, módulos Vue, etc.).

Este enfoque facilitó la colaboración, evitó conflictos de código y aseguró un despliegue limpio al finalizar el sprint.

#### 6.2.1.9. 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.

- **Estrategia de colaboración técnica**

Se aplicó la metodología **GitFlow Workflow**, lo que permitió distribuir el trabajo en ramas específicas para cada funcionalidad tanto en el desarrollo de la Landing Page como en el Frontend Web desarrollado con Vue.js. La rama `develop` funcionó como entorno de integración, mientras que cada integrante trabajó en ramas funcionales individuales, realizando posteriormente *pull requests* para su revisión y fusión al repositorio principal. Finalmente la rama `develop` se fusionó a la rama `main`, asegurando que solo el código validado y revisado se integrara a la versión estable del proyecto.

- **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, completar la primera versión de la Landing Page y avanzar significativamente en la interfaz del sistema.

### Web Application



## Landing page

### Network graph

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



## Conclusiones

- 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.

- La Landing Page de AlquilaFácil se presenta como una herramienta clave para atraer y educar a los usuarios sobre la plataforma. Su diseño intuitivo y atractivo, junto con la información clara y concisa sobre los beneficios y características del servicio, contribuye a generar confianza y facilitar la adopción de la plataforma por parte de nuevos usuarios.
- Finalmente, AlquilaFácil no solo apunta a consolidarse en el mercado local, sino que tiene una visión ambiciosa de posicionarse como líder global en el sector de alquiler de espacios para eventos. Su compromiso con la innovación, la optimización de procesos y la excelencia en la experiencia de usuario sienta las bases para una expansión sostenida, diferenciándose como un referente en eficiencia, confiabilidad y servicio en la industria.

## 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 Widgits. (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/>
- Jasin, 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

---

- Video de exposición: [https://upcedupe-my.sharepoint.com/:f/g/personal/u202212077\\_upc\\_edu\\_pe/EsaJcJikO3pChqWobVLM7LsBUxbNd-ZJkHe5819nZ3CYpQ?e=xmGXjS](https://upcedupe-my.sharepoint.com/:f/g/personal/u202212077_upc_edu_pe/EsaJcJikO3pChqWobVLM7LsBUxbNd-ZJkHe5819nZ3CYpQ?e=xmGXjS)