

Universidad Peruana de Ciencias Aplicadas**Carrera de Ingeniería de Software - Ciclo VII****1ASI0572 - Desarrollo de Soluciones IOT****NRC 3355****Profesor: David Carlos Vera Olivera****INFORME DEL TRABAJO FINAL****Startup del Proyecto: Bibflip****Producto: Bib Flip**

Team Members:

Member	Code
Aranda Vallejos, Oscar Gabriel	U202218167
Bernaola Pérez, André Arturo	U202114192
Gutierrez Garcia, Jose Eduardo	U202221518
Oliveira Paucar, Mauricio	U201719831
Pedraza Maldonado, Joaquin	U202218514
Soriano Medrano, Diego	U202114793
Velarde Luyo, Piero Alberto	U20211A620

2025-20**Registro de Versiones del Informe**

En esta sección se resumen los avances y modificaciones realizadas durante el ciclo de vida de desarrollo del proyecto equivalente a un semestre.

Versión	Fecha	Autor	Descripción de modificación
1.0	09/09/2025	Aranda Vallejos, Oscar Gabriel	Creación y estructuración del informe
1.1	11/09/2025	Aranda Vallejos, Oscar Gabriel	Desarrollo del capítulo I
1.2	14/09/2025	Aranda Vallejos, Oscar Gabriel	Desarrollo del event storming y bounded context canvas
1.3	15/09/2025	Bernaola Pérez, André Arturo	Desarrollo del diagrama de clases y de base de datos
1.4	15/09/2025	Aranda Vallejos, Oscar Gabriel	Cierre del capítulo I y Competidores
1.5	15/09/2025	Soriano Medrano, Diego	Desarrollo de los Domain Message Flows Modeling y del Context Mapping
1.6	16/09/2025	Pedraza Maldonado, Joaquín	Desarrollo y cierre del capítulo III
1.7	16/09/2025	Gutierrez Garcia, Jose Eduardo	Desarrollo de Software Architecture
1.8	16/09/2025	Oliveira Paucar, Mauricio	Desarrollo de Tactical-Level Domain-Driven Design
1.9	17/09/2025	Velarde Luyo, Piero Alberto	Desarrollo del Needfinding
2.0	18/09/2025	Aranda Vallejos, Oscar Gabriel	Cierre de los capítulos II y IV

Versión	Fecha	Autor	Descripción de modificación
2.1	20/09/2025	Oliveira Paucar, Mauricio	Desarrollo del Capítulo V: Solution UI/UX Design
2.2	21/09/2025	Velarde Luyo, Piero Alberto	Diseño de wireframes y prototipos de alta fidelidad
2.3	22/09/2025	Gutierrez Garcia, Jose Eduardo	Desarrollo del Capítulo VI: Product Implementation, Validation & Deployment
2.4	23/09/2025	Aranda Vallejos, Oscar Gabriel	Desarrollo de la sección 6.1: Software Configuration Management
2.5	23/09/2025	Aranda Vallejos, Oscar Gabriel	6.1.1. Software Development Environment Configuration
2.6	24/09/2025	Bernaola Pérez, André Arturo	6.1.2. Source Code Management
2.7	24/09/2025	Soriano Medrano, Diego	6.1.3. Source Code Style Guide & Conventions
2.8	25/09/2025	Pedraza Maldonado, Joaquín	6.1.4. Software Deployment Configuration
2.9	26/09/2025	Velarde Luyo, Piero Alberto	6.2. Landing Page, Services & Applications Implementation
3.0	27/09/2025	Gutierrez Garcia, Jose Eduardo	6.2.1. Sprint 1 - Sprint Planning 1
3.1	28/09/2025	Soriano Medrano, Diego	6.2.1.2. Aspect Leaders and Collaborators
3.2	28/09/2025	Oliveira Paucar, Mauricio	6.2.1.3. Sprint Backlog 1
3.3	29/09/2025	Pedraza Maldonado, Joaquín	6.2.1.4. Development Evidence for Sprint Review
3.4	30/09/2025	Bernaola Pérez, André Arturo	6.2.1.5. Testing Suite Evidence for Sprint Review
3.5	01/10/2025	Aranda Vallejos, Oscar Gabriel	6.2.1.6. Execution Evidence for Sprint Review
3.6	02/10/2025	Velarde Luyo, Piero Alberto	6.2.1.7. Services Documentation Evidence for Sprint Review
3.7	03/10/2025	Gutierrez Garcia, Jose Eduardo	6.2.1.8. Software Deployment Evidence for Sprint Review
3.8	04/10/2025	Soriano Medrano, Diego	6.2.1.9. Team Collaboration Insights during Sprint
3.9	05/10/2025	Aranda Vallejos, Oscar Gabriel	Cierre de los capítulos V y VI
4.0	01/11/2025	Gutierrez Garcia, Jose Eduardo	6.2.2.1. Sprint Planning 2
4.1	02/11/2025	Aranda Vallejos, Oscar Gabriel	6.2.2.2. Aspect Leader and Collaborators
4.2	03/11/2025	Velarde Luyo, Piero Alberto	6.2.2.3. Sprint Backlog 2
4.3	04/11/2025	Pedraza Maldonado, Joaquín	6.2.2.4. Development Evidence for Sprint Review
4.4	05/11/2025	Bernaola Pérez, André Arturo	6.2.2.5. Testing Suite Evidence for Sprint Review
4.5	06/11/2025	Soriano Medrano, Diego	6.2.2.6. Execution Evidence for Sprint Review
4.6	07/11/2025	Aranda Vallejos, Oscar Gabriel	6.2.2.7. Services Documentation Evidence for Sprint Review
4.7	08/11/2025	Oliveira Paucar, Mauricio	6.2.2.8. Software Deployment Evidence for Sprint Review
4.8	09/11/2025	Soriano Medrano, Diego	6.2.2.9. Team Collaboration Insights during Sprint
4.9	10/11/2025	Todo el equipo de Bibflip	Validation Interviews y Video About-the-Product.
5.0	20/11/2025	Aranda Vallejos, Oscar Gabriel	6.2.3.1. Sprint Planning 3
5.1	21/11/2025	Gutiérrez García, José Eduardo	6.2.3.2. Aspect Leaders and Collaborators – Sprint 3
5.2	22/11/2025	Velarde Luyo, Piero Alberto	6.2.3.3. Sprint Backlog 3
5.3	23/11/2025	Pedraza Maldonado, Joaquín	6.2.3.4. Development Evidence for Sprint 3 Review
5.4	24/11/2025	Bernaola Pérez, André Arturo	6.2.3.5. Testing Suite Evidence for Sprint 3 Review
5.5	25/11/2025	Soriano Medrano, Diego	6.2.3.6. Execution Evidence for Sprint 3 Review
5.6	26/11/2025	Oliveira Paucar, Mauricio	6.2.3.7. Services Documentation Evidence for Sprint 3 Review
5.7	27/11/2025	Gutiérrez García, José Eduardo	6.2.3.8. Software Deployment Evidence for Sprint 3 Review
5.8	28/11/2025	Todo el equipo de Bibflip	6.2.3.9. Team Collaboration Insights during Sprint 3
5.9	30/11/2025	Todo el equipo de Bibflip	Integración del Trabajo Final (TF) y cierre del Capítulo VI del informe

Project Report Collaboration Insights

Se utilizaron Git y Github como herramientas fundamentales para el versionamiento y gestión del proyecto, lo cual permite una mejor colaboración y visualización de cambios en tiempo real. A continuación, se proporcionan los siguientes enlaces del proyecto:

Descripción	Enlace
Repositorio de la documentación	https://github.com/upc-pre-202510-1asi0572-3355-BibFlip/Report
Organización del proyecto	https://github.com/upc-pre-202510-1asi0572-3355-BibFlip

Contenido

Tabla de Contenidos

[Registro de Versiones del Informe](#)

[Project Report Collaboration Insights](#)

[Contenido](#)

[Student Outcomes](#)

[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](#)
 - **Business Outcomes:**
 - **Users:**
 - **User Outcomes & Benefits:**
 - **Feature Assumptions:**
 - **Business Assumptions:**
 - **User Assumptions:**
 - [1.2.2.3. Lean UX Hypothesis Statements](#)
 - [1.2.2.4. Lean UX Canvas](#)
 - [1.3. Segmentos objetivo](#)

[Capítulo II: Requirements Elicitation & Analysis](#)

- [2.1. Competidores](#)
 - [2.1.1. Análisis competitivo](#)
 - [2.1.2. Estrategias y tácticas frente a competidores](#)
- [2.2. Entrevistas](#)
 - [2.2.1. Diseño de entrevistas](#)
 - [2.2.2. Registro de entrevistas](#)
 - [2.2.3. Análisis de entrevistas](#)
- [2.3. Needfinding](#)
 - [2.3.1. User Personas](#)
 - [2.3.2. User Task Matrix](#)
 - [2.3.3. 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 Deployment Diagrams
- 4.2. Tactical-Level Domain-Driven Design
 - 4.2.1. Bounded Context: IAM
 - 4.2.1.1. Domain Layer
 - 4.2.1.2. Interface Layer
 - 4.2.1.3. Application Layer
 - 4.2.1.4. Infrastructure Layer
 - 4.2.1.6. Bounded Context Software Architecture Component Level Diagrams
 - 4.2.1.7. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.1.7.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.1.7.2. Bounded Context Database Design Diagram
 - 4.2.2. Bounded Context: Table Management
 - 4.2.2.1. Domain Layer
 - 4.2.2.2. Interface Layer
 - 4.2.2.3. Application Layer
 - 4.2.2.4. Infrastructure Layer
 - 4.2.2.6. Bounded Context Software Architecture Component Level Diagrams
 - 4.2.2.7. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.2.7.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.2.7.2. Bounded Context Database Design Diagram
 - 4.2.3. Bounded Context: Branching
 - 4.2.3.1. Domain Layer
 - 4.2.3.2. Interface Layer
 - 4.2.3.3. Application Layer
 - 4.2.3.4. Infrastructure Layer
 - 4.2.3.6. Bounded Context Software Architecture Component Level Diagrams
 - 4.2.3.7. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.3.7.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.3.7.2. Bounded Context Database Design Diagram
 - 4.2.4. Bounded Context: Booking
 - 4.2.4.1. Domain Layer
 - 4.2.4.2. Interface Layer
 - 4.2.4.3. Application Layer
 - 4.2.4.4. Infrastructure Layer
 - 4.2.4.6. Bounded Context Software Architecture Component Level Diagrams
 - 4.2.4.7. Bounded Context Software Architecture Code Level Diagrams
 - 4.2.4.7.1. Bounded Context Domain Layer Class Diagrams
 - 4.2.4.7.2. Bounded Context Database Design Diagram
 - 4.2.5. Bounded Context: IoT Device Monitoring

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 Wireflow Diagrams
 - 5.4.3. Applications Mock-ups
 - 5.4.4. 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
 - 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
 - 6.2.2. Sprint 2
 - 6.2.2.1. Sprint Planning 2
 - 6.2.2.2. Aspect Leaders and Collaborators
 - 6.2.2.3. Sprint Backlog 2
 - 6.2.2.4. Development Evidence for Sprint Review
 - 6.2.2.5. Testing Suite Evidence for Sprint Review
 - 6.2.2.6. Execution Evidence for Sprint Review
 - 6.2.2.7. Services Documentation Evidence for Sprint Review
 - 6.2.2.8. Software Deployment Evidence for Sprint Review
 - 6.2.2.9. Team Collaboration Insights during Sprint
 - 6.2.3. Sprint 3
 - 6.2.3.1. Sprint Planning 3
 - 6.2.3.2. Aspect Leaders and Collaborators
 - 6.2.3.3. Sprint Backlog 3
 - 6.2.3.4. Development Evidence for Sprint Review
 - 6.2.3.5. Testing Suite Evidence for Sprint Review
 - 6.2.3.6. Execution Evidence for Sprint Review
 - 6.2.3.7. Services Documentation Evidence for Sprint Review
 - 6.2.3.8. Software Deployment Evidence for Sprint Review
 - 6.2.3.9. Team Collaboration Insights during Sprint
- 6.3. Validation Interviews
 - 6.3.1. Diseño de Entrevistas
 - 6.3.2. Registro de Entrevistas
 - 6.3.3. Evaluaciones según heurísticas
- 6.4. Video About-the-Product

Conclusiones

Bibliografía

Anexos

Student Outcome

En Ingeniería de Software, el logro contribuye a alcanzar el ABET – EAC - Student Outcome 5: 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.

Criterio específico	Acciones realizadas	Conclusiones
Trabaja en equipo para proporcionar liderazgo en forma conjunta	Aranda Vallejos, Oscar Gabriel TB1 Lideré la iniciativa de creación y estructuración del informe, estableciendo las bases para el trabajo colaborativo del equipo. Además, dirigí el desarrollo del event storming y el Domain Driven Design, facilitando la comprensión compartida del dominio entre todos los miembros. TP Durante el desarrollo del Capítulo V: Solution UI/UX Design , trabajé de manera conjunta con mi equipo para diseñar las interfaces y la experiencia del usuario de forma coherente, moderna y funcional. Lideré reuniones de diseño en las que cada integrante aportó ideas visuales y estructurales, y juntos tomamos decisiones sobre la organización de las vistas, la accesibilidad y los colores institucionales. Este proceso me permitió fortalecer mi capacidad para liderar desde la	TB1 La distribución efectiva del liderazgo técnico, donde cada miembro asume responsabilidades específicas según sus fortalezas, es fundamental para trabajar de manera más rápida y efectiva. Además, las revisiones periódicas del avance de cada integrante y la comunicación fluida permite garantizar una buena organización del proyecto. TP Como equipo, demostramos la capacidad de trabajar de manera colaborativa para liderar y ejecutar todas las etapas del desarrollo del producto, desde el diseño UI/UX hasta la validación y el despliegue final. A través de una comunicación efectiva, una planificación conjunta y la distribución equitativa de responsabilidades, ejercimos

Criterio específico	Acciones realizadas	Conclusiones
	<p>colaboración, escuchando las opiniones del grupo y guiando el diseño hacia un resultado unificado y centrado en el usuario.</p> <p>TB2</p> <p>Lideré la implementación de la primera versión del IoT Edge API y coordiné la actualización del Backend y el Web Frontend. Esto aseguró que la capa de comunicación con los sensores se integrara correctamente con la lógica de negocio, manteniendo la consistencia técnica del sistema.</p> <p>TF</p> <p>Durante el Trabajo Final asumí la facilitación de la Sprint Planning 3, ayudando al equipo a priorizar las historias relacionadas con la integración IoT completa (sensors + Edge API + Frontend). Guié la discusión de alcance para que el sprint se mantuviera realista y alineado con los objetivos del TF, y coordiné las decisiones clave durante las reuniones, compartiendo el liderazgo con mis compañeros para evitar que la planificación dependiera de una sola persona.</p> <p>Bernaola Pérez, André Arturo</p> <p>TB1</p> <p>Lideré el diseño de la base de datos y del diagrama de clases, proporcionando la base arquitectónica del software.</p> <p>TP</p> <p>Durante la implementación de la sección 6.1.3: Source Code Style Guide & Conventions, propuse un conjunto de reglas de codificación comunes para todo el equipo, lo que facilitó la lectura, el mantenimiento y la escalabilidad del proyecto. Lideré las revisiones de código y promoví el uso de convenciones uniformes en nombres de clases, métodos y estructuras. Esta experiencia me permitió desarrollar liderazgo técnico participativo, motivando al equipo a mantener la calidad del código y a trabajar de forma organizada y alineada con las buenas prácticas.</p> <p>TB2</p> <p>Participé activamente en el liderazgo colaborativo del equipo para la revisión de la evidencia de Testing Suite (6.2.2.5). Me aseguré de que los Acceptance Tests y Unit Tests para el Sprint 2 cumplieran con los criterios de aceptación de las User Stories, garantizando la calidad y robustez del Web Service.</p> <p>TF</p> <p>En el TF lideré el cierre técnico de la Testing Suite para Sprint 3, organizando los últimos refactors y correcciones necesarias antes del code freeze. Coordiné con el equipo la priorización de bugs críticos y la ejecución de pruebas de regresión sobre el Web Service y la integración IoT. Esto me permitió ejercer un liderazgo técnico colaborativo, orientado a asegurar que llegáramos al demo final con una base de código estable y validada.</p> <p>Soriano Medrano, Diego</p> <p>TB1</p> <p>Asumí el liderazgo en el desarrollo de los Domain Message Flows Modeling, facilitando la comprensión de las comunicaciones del sistema. Además, dirigí el desarrollo del Context Mapping, proporcionando claridad sobre las relaciones entre contextos.</p> <p>TP</p> <p>En la fase de 6.1.4: Software Deployment Configuration, trabajé con el equipo en la integración y despliegue del producto en un entorno controlado. Mi responsabilidad fue supervisar la correcta configuración del entorno de despliegue y apoyar la resolución de incidencias durante la ejecución. Mediante la comunicación constante con mis compañeros, logramos desplegar el software de manera estable y funcional. Este proceso fortaleció mis habilidades de liderazgo técnico compartido, ya que aprendí a guiar el trabajo colaborativo bajo presión y a valorar la importancia de la coordinación grupal en las etapas críticas del proyecto.</p> <p>TB2</p> <p>Asumí el liderazgo en la consolidación de la evidencia de Software Deployment (6.2.2.8) y la documentación de servicios (6.2.2.7). Esto</p>	<p>un liderazgo compartido orientado a resultados, garantizando la calidad técnica del proyecto y el cumplimiento de los objetivos establecidos. Esta experiencia fortaleció nuestras competencias colectivas en organización, resolución de problemas y toma de decisiones dentro de un entorno ágil y multidisciplinario.</p> <p>TB2</p> <p>El equipo demostró un liderazgo distribuido y efectivo, donde cada miembro asumió un rol técnico. Esta estructura de liderazgo conjunto facilitó la toma de decisiones técnicas rápidas y aseguró la cohesión del producto final. La supervisión de artefactos críticos (como el IoT Edge API y la documentación de servicios) y la aplicación de validaciones (como las Evaluaciones según Heurísticas) actuaron como actos de liderazgo centrados en la calidad y el cumplimiento de los objetivos del Sprint 2.</p> <p>TF</p> <p>En el Trabajo Final, el equipo consolidó un modelo de liderazgo distribuido, donde cada integrante asumió responsabilidades específicas en la planificación, prueba, despliegue, documentación y validación del sistema IoT. Este enfoque permitió que las decisiones clave (técnicas y de producto) se tomaran en forma conjunta, aprovechando las fortalezas individuales y garantizando que el resultado final representara el trabajo coordinado de todo el grupo.</p>

Criterio específico	Acciones realizadas	Conclusiones
	<p>aseguró que todos los endpoints de la Cloud API y Edge API estuvieran documentados con Swagger/OpenAPI y que las aplicaciones estuvieran correctamente desplegadas en Azure y Firebase.</p> <p><i>TF</i></p> <p>Durante el Trabajo Final tomé el liderazgo en las pruebas de integración extremo a extremo entre los sensores físicos, el IoT Edge API y los servicios en la nube. Organicé sesiones conjuntas para revisar logs, diagnosticar errores de comunicación y decidir en equipo qué ajustes eran prioritarios. Este rol me permitió guiar al grupo en la toma de decisiones técnicas bajo presión, siempre promoviendo que cada integrante aportara desde su experiencia.</p> <p>Pedraza Maldonado, Joaquín</p> <p><i>TB1</i></p> <p>Tomé la responsabilidad de liderar el desarrollo del capítulo III, coordinando con el equipo para mantener la consistencia técnica.</p> <p><i>TP</i></p> <p>En el desarrollo del Sprint 1, especialmente en los puntos 6.2.1.4 al 6.2.1.8, asumí un rol de coordinación durante las reuniones de revisión de sprint. Junto con mis compañeros, organizamos las evidencias de desarrollo, pruebas y despliegue, asegurando que el producto cumpliera los criterios de aceptación definidos. Participé activamente en las sesiones de testing y en la documentación de los resultados. Este proceso me ayudó a ejercer liderazgo en la validación del producto, fomentando la responsabilidad compartida y la mejora continua en cada iteración.</p> <p><i>TB2</i></p> <p>Lideré el desarrollo de la nueva versión del Mobile Frontend y completé la sección de Evaluaciones según Heurísticas (6.3.3.). Aseguré que la interfaz móvil se alineara con el diseño UI/UX final y que la usabilidad fuera validada con base en los estándares de Nielsen, dirigiendo la mejora continua del producto.</p> <p><i>TF</i></p> <p>En el TF lideré la consolidación del Mobile Frontend para el demo final, coordinando los últimos cambios de navegación y la alineación visual con la Landing Page y el Web Frontend. Preparé junto al equipo el flujo de demostración del sistema, definiendo qué pantallas y escenarios se mostrarían para evidenciar mejor el valor de la solución. De esta manera ejercí un liderazgo orientado a la experiencia del usuario y a la claridad del mensaje del producto.</p> <p>Gutiérrez García, José Eduardo</p> <p><i>TB1</i></p> <p>Lideré el desarrollo de los modelos C4 y comuniqué el diseño arquitectónico para el equipo.</p> <p><i>TP</i></p> <p>En la sección 6.1: Software Configuration Management, participé activamente en la configuración del entorno de desarrollo y la gestión del código fuente. Coordiné con mis compañeros la instalación de dependencias, la definición del flujo de trabajo en Git y la organización de ramas para evitar conflictos. Ejercí liderazgo técnico al guiar la integración continua y asegurar que todos los miembros del equipo pudieran ejecutar el proyecto sin inconvenientes. Este trabajo en conjunto me ayudó a comprender la importancia de la comunicación constante y la planificación compartida en entornos de desarrollo colaborativos.</p> <p><i>TB2</i></p> <p>Lideré la documentación de la Sprint Planning 2 (6.2.2.1-6.2.2.4), organizando el Sprint Backlog 2 y registrando las asignaciones de tareas técnicas de IoT. Esto proporcionó claridad sobre los objetivos del sprint, especialmente en la simulación de sensores y la implementación de firmware.</p> <p><i>TF</i></p> <p>Durante el Trabajo Final asumí el liderazgo en la gestión de versiones y despliegue final del sistema, coordinando los merges, releases y tags del repositorio para la versión del TF. Definí con el equipo la secuencia</p>	

Criterio específico	Acciones realizadas	Conclusiones
	<p>de despliegue de los componentes (firmware, Edge API, Backend y Frontends) y apoyé la resolución de conflictos de integración. Este rol de liderazgo técnico me permitió asegurar que todo el trabajo desarrollado durante los sprints convergiera en una versión final coherente y funcional.</p> <p>Oliveira Paucar, Mauricio</p> <p><i>TB1</i></p> <p>Asumí el liderazgo en Tactical-Level Domain-Driven Design, guiando al equipo en la documentación de la estructura de cada bounded context.</p> <p><i>TP</i></p> <p>Contribuí en los apartados 6.2.1.7 a 6.2.1.9, enfocándome en la documentación de los servicios, el despliegue y la recopilación de los aprendizajes del equipo durante el sprint. Coordiné la redacción de reportes técnicos y participé en la reflexión conjunta sobre los aciertos y desafíos del desarrollo. Este trabajo fortaleció mi capacidad para liderar la comunicación técnica del grupo, promoviendo la transparencia, la colaboración y el aprendizaje colectivo como pilares del trabajo en equipo.</p> <p><i>TB2</i></p> <p>Asumí el liderazgo en la actualización del Landing Page y la documentación del Project Report Collaboration Insights y el Student Outcome. Mantuve la coherencia visual del producto y coordiné la recopilación de los aprendizajes del equipo, promoviendo la transparencia y la mejora continua.</p> <p><i>TF</i></p> <p>En el TF lideré la documentación final del producto y la integración de los apartados de Student Outcome y conclusiones técnicas. Organicé la recolección de evidencias de todos los miembros y transformé esas contribuciones en una narrativa única y consistente para el informe y la presentación. Este liderazgo documental permitió que el trabajo del equipo se reflejara de forma clara, equilibrada y completa en el entregable final.</p>	
	<p>Velarde Luyo, Piero Alberto</p> <p><i>TB1</i></p> <p>Lideré el proceso de Needfinding, dirigiendo la investigación de usuarios y estableciendo las bases para las decisiones del equipo.</p> <p><i>TP</i></p> <p>Durante la etapa 6.2.1.1: Sprint Planning 1, participé activamente en la planificación inicial del sprint junto con mis compañeros, asumiendo un rol de liderazgo colaborativo para definir las prioridades y distribuir las tareas de forma equitativa. Propuse la organización del backlog en función de las capacidades y fortalezas de cada integrante, lo que nos permitió optimizar el tiempo y trabajar de manera más coordinada. Esta experiencia me enseñó la importancia de liderar desde la organización y la empatía, garantizando que todos los miembros del equipo comprendieran los objetivos del sprint y se sintieran parte del proceso de construcción del producto.</p> <p><i>TB2</i></p> <p>Lideré, junto con André, la revisión de la Testing Suite (6.2.2.5) y la implementación de los Acceptance Tests del Sprint 2, asegurando la trazabilidad de los requisitos de usuario a través de escenarios en Gherkin. Esto garantizó que las nuevas funcionalidades cumplieran con la lógica de negocio esperada.</p> <p><i>TF</i></p> <p>Durante el Trabajo Final lideré la preparación y síntesis de los resultados de validación con usuarios, organizando la información recogida en las entrevistas y pruebas de uso de la solución Bibflip. Guié la discusión en equipo sobre qué aprendizajes debían reflejarse como mejoras de producto y cuáles como hallazgos para trabajos futuros.</p> <p>Este rol me permitió ejercer un liderazgo centrado en la voz del usuario y en la toma de decisiones basada en evidencia.</p>	
Crea un entorno	Aranda Vallejos, Oscar Gabriel	TB1

Entorno colaborativo e inclusivo, establece metas, planifica tareas y cumple objetivos	Acciones realizadas	Conclusiones
	<p>Establecí un ambiente colaborativo coordinando la gestión general del proyecto al definir objetivos claros y plazos específicos para cada entrega.</p>	<p>Establecer un entorno colaborativo mediante la división estratégica de tareas es vital para integrar cohesivamente todos los features en el producto final. Mantener la comunicación abierta y la planificación compartida fortalece la ejecución del proyecto.</p>
	<p>TP Durante la implementación del Capítulo VI, asumí la responsabilidad de coordinar los avances del equipo, estableciendo metas semanales claras y asegurando que cada integrante comprendiera sus objetivos. Promoví un entorno de apoyo mutuo y comunicación abierta, lo que permitió cumplir con las entregas planificadas y mantener la cohesión del grupo.</p>	<p>TP Como equipo, demostramos la capacidad de crear un entorno inclusivo y colaborativo, en el que la comunicación constante, la planificación compartida y el cumplimiento de metas comunes nos permitieron avanzar de forma organizada y eficiente. Cada integrante asumió responsabilidades claras, aportando desde sus fortalezas, lo que favoreció un ambiente de trabajo equilibrado, participativo y orientado al logro de los objetivos del proyecto.</p>
	<p>TB2 Mantuve el entorno de desarrollo unificado y establecí las metas de integración para el Edge API y el Backend. Definí los objetivos para la actualización del Web Frontend, asegurando que las tareas del equipo convergieran hacia un sistema final integrado y funcional, cumpliendo con la meta del Sprint.</p>	<p>TB2</p>
	<p>TF Durante el Trabajo Final me encargué de mantener un entorno de trabajo organizado e inclusivo al coordinar la agenda de reuniones para el cierre del proyecto, clarificando objetivos semanales y fechas límite. Me aseguré de que cada integrante tuviera espacio para expresar sus avances y bloqueos, y adaptamos el plan de tareas según la carga de cada uno, lo que permitió cumplir las metas del TF sin perder la cohesión del equipo.</p>	<p>El entorno de trabajo fue altamente colaborativo e inclusivo, fundamentado en una planificación ágil rigurosa. El equipo estableció metas claras y medibles a través del Sprint Planning 2 y el Sprint Backlog 2, asegurando la trazabilidad de tareas y responsabilidades. El cumplimiento de los objetivos se materializó con la integración exitosa de los componentes (Frontend, Backend, IoT), la evidencia de Software Deployment y la finalización de la fase de validación (Validation Interviews y Evaluaciones según Heurísticas), demostrando la capacidad del equipo para alcanzar objetivos complejos en el plazo establecido.</p>
	<p>TB2 Durante el Trabajo Final me encargué de mantener un entorno de trabajo organizado e inclusivo al coordinar la agenda de reuniones para el cierre del proyecto, clarificando objetivos semanales y fechas límite. Me aseguré de que cada integrante tuviera espacio para expresar sus avances y bloqueos, y adaptamos el plan de tareas según la carga de cada uno, lo que permitió cumplir las metas del TF sin perder la cohesión del equipo.</p>	<p>TF</p>
	<p>Bernaola Pérez, André Arturo TB1 Contribuí al entorno colaborativo considerando las necesidades de otros componentes del sistema y planificando el modelado de relaciones del sistema.</p>	<p>En el Trabajo Final, el equipo demostró la capacidad de diseñar y sostener un entorno colaborativo e inclusivo, en el que la planificación de tareas, la comunicación y el seguimiento de objetivos fueron compartidos. Las herramientas de gestión (tableros, repositorios, documentación) y las reuniones periódicas se utilizaron para que todos tuvieran claridad sobre el avance y pudieran aportar, lo que permitió cumplir los objetivos del TF y cerrar el proyecto con un alto nivel de coordinación y participación grupal.</p>
	<p>TP Durante la fase de 6.1.3: Source Code Style Guide & Conventions, fomenté un entorno inclusivo al proponer estándares de codificación consensuados. Planifiqué junto a mis compañeros revisiones de código semanales que garantizaron el cumplimiento de los objetivos técnicos y la mejora continua del producto.</p>	
	<p>TB2 Contribuí a un entorno inclusivo al participar en la planificación detallada del Sprint Backlog 2, asegurando que las tareas relacionadas con el testing (Acceptance y Unit Tests) se priorizaran y se distribuyeran equitativamente en la fase de validación (6.2.2.5).</p>	
	<p>TF En el TF fomenté un entorno colaborativo a través de revisiones de código abiertas, donde se invitó a todos los miembros a comentar y aprender sobre las decisiones técnicas tomadas en la Testing Suite y el Web Service. Expliqué las convenciones aplicadas y apoyé a mis compañeros en la corrección de issues de calidad de código, contribuyendo a que el cumplimiento de los objetivos técnicos fuera un esfuerzo compartido.</p>	
	<p>Soriano Medrano, Diego TB1 Promoví la colaboración analizando los bounded context que conectan el trabajo de todos los miembros del equipo. Además, logré facilitar la comprensión compartida al mapear las comunicaciones entre estos.</p>	
	<p>TP Durante la configuración del entorno de despliegue (6.1.4), mantuve una comunicación constante con mis compañeros, coordinando los tiempos y pasos del proceso. Esta planificación conjunta permitió cumplir los objetivos técnicos y garantizar un entorno colaborativo, ordenado y eficiente.</p>	
	<p>TB2 Contribuí al entorno de trabajo mediante la gestión y actualización del Trello y el registro de la colaboración en GitHub (6.2.2.9). Esta planificación y seguimiento transparente facilitó que todos los miembros conocieran el estado del proyecto y se enfocaran en cumplir</p>	

Criterio específico	Acciones realizadas	Conclusiones
	<p>los objetivos de despliegue de los servicios.</p> <p><i>TF</i></p> <p>Durante el Trabajo Final mantuve actualizados los tableros de Trello/GitHub Projects con el estado real de las tareas de integración y despliegue, de forma transparente para todo el equipo. Esto permitió que todos conocieran en qué punto del proceso estábamos, qué dependencias existían y qué tareas eran críticas para llegar a tiempo al demo final, fortaleciendo así la planificación conjunta y el enfoque en los objetivos comunes.</p>	
	<p>Pedraza Maldonado, Joaquín</p> <p><i>TB1</i></p> <p>Logré definir las historias de usuario que forman parte de los requisitos que necesitamos cumplir y, en compañía del equipo, establecí los tiempos requeridos para implementar cada historia en el product backlog.</p> <p><i>TP</i></p> <p>En el desarrollo de la Landing Page y los Servicios (6.2), participé en la planificación del sprint y la distribución de tareas, asegurando que todos los integrantes pudieran aportar según sus fortalezas. Contribuí a mantener un entorno colaborativo donde la comunicación y la organización fueron claves para cumplir los objetivos de cada entrega.</p> <p><i>TB2</i></p> <p>Asumí la tarea de COMPLETAR las Evaluaciones según Heurísticas (6.3.3) y la nueva versión del Mobile Frontend. Establecí el objetivo de mejorar la usabilidad, planificando las correcciones de diseño basadas en la retroalimentación de las heurísticas de Nielsen para cumplir el objetivo de una mejor experiencia de usuario.</p> <p><i>TF</i></p> <p>En el TF promoví un entorno inclusivo organizando sesiones de trabajo conjunto para cerrar los detalles del Mobile Frontend y las evaluaciones de usabilidad. Durante estas sesiones, recogí sugerencias de diseño de todos los compañeros y las incorporé en el backlog de ajustes finales, asegurando que las decisiones sobre la interfaz no fueran solo individuales, sino resultado de la colaboración del grupo.</p>	
	<p>Gutiérrez García, José Eduardo</p> <p><i>TB1</i></p> <p>Establecí un entorno colaborativo diseñando los diagramas de componentes que integran el trabajo de todos los bounded context.</p> <p><i>TP</i></p> <p>En la sección 6.1.2: Source Code Management, organicé el flujo de trabajo colaborativo en Git, estableciendo ramas y políticas de integración que facilitaron el cumplimiento de los objetivos del sprint. Gracias a la planificación compartida, logramos mantener un entorno ordenado y productivo para todo el equipo.</p> <p><i>TB2</i></p> <p>Me centré en la planificación de tareas de la sección 6.2.2 (Sprint 2), organizando las actividades del equipo en el backlog y asegurando que las dependencias entre el firmware (TS004) y la integración de la plataforma embebida (TS005) estuvieran claras. Esto facilitó el cumplimiento de los objetivos de IoT.</p> <p><i>TF</i></p> <p>Durante el Trabajo Final contribuí al entorno colaborativo al preparar guías breves de configuración y despliegue para que todos pudieran correr el sistema completo (IoT + Backend + Frontends) en sus equipos. Respondí dudas técnicas sobre integración y ramas de Git, lo que permitió que nadie quedara excluido del proceso de prueba y que el cumplimiento de los objetivos técnicos fuera accesible para todo el equipo.</p>	
	<p>Oliveira Paucar, Mauricio</p> <p><i>TB1</i></p> <p>Planifiqué y cumplí objetivos específicos en el nivel táctico del DDD, guiando el trabajo colaborativo y sincronizando otras secciones dependientes de los bounded context.</p>	

Criterio específico	Acciones realizadas	Conclusiones
	<p><i>TP</i></p> <p>Durante el desarrollo de los apartados 6.2.1.7 a 6.2.1.9, coordiné la documentación técnica en conjunto con el equipo, promoviendo un ambiente inclusivo y participativo. Establecimos metas claras para la entrega de documentación y pruebas, asegurando la calidad y coherencia del producto final.</p>	
	<p><i>TB2</i></p> <p>Contribuí a la creación de un entorno colaborativo e inclusivo al garantizar la actualización y consistencia de la documentación (TB2) y las conclusiones del equipo. Esto aseguró que todas las contribuciones estuvieran representadas y que el informe final reflejara el trabajo conjunto.</p>	
	<p><i>TF</i></p> <p>En el TF reforzé la colaboración al coordinar la integración de aportes en la documentación final, asegurándome de que cada integrante tuviera visibilidad y reconocimiento de su trabajo en el informe, anexos y presentación. Organicé la revisión conjunta de las secciones clave, promoviendo un ambiente donde todos pudieran proponer cambios y mejoras antes de la entrega final.</p>	
Velarde Luyo, Piero Alberto		
	<p><i>TB1</i></p> <p>Creé un ambiente inclusivo investigando necesidades de usuarios y compartiendo insights que orientan las decisiones del equipo.</p>	
	<p><i>TP</i></p> <p>En la sección 6.2.1.1: Sprint Planning 1, participé activamente en la planificación del sprint, fomentando la inclusión de las ideas de todos los integrantes. Contribuí en la definición de tareas y prioridades, lo que permitió un trabajo equilibrado y enfocado en los objetivos del proyecto.</p>	
	<p><i>TB2</i></p> <p>Participé activamente en la planificación de tareas de testing y la ejecución de Acceptance Tests (6.2.2.5), contribuyendo a un entorno de calidad inclusivo. Establecí las expectativas para la cobertura de pruebas, asegurando que las metas de robustez del código se cumplieran antes de la integración final.</p>	
	<p><i>TF</i></p> <p>Durante el Trabajo Final mantuve un entorno inclusivo coordinando las actividades de validación con usuarios y docentes, invitando a todo el equipo a participar en la definición de preguntas, observaciones y mejoras. Sinteticé las opiniones recogidas y las compartí con el grupo, de forma que las decisiones sobre ajustes finales fueran el resultado de una reflexión colectiva y alineada con los objetivos del TF.</p>	

Capítulo I: Introducción

1.1. Startup Profile

1.1.1. Descripción de la Startup

En el 2020, se reportaron matrículas con aproximadamente 1.3 millones de estudiantes de pregrado y 95 mil de posgrado en universidades peruanas (SUNEDU, 2022). Estos índices, a su vez, provocan una alta demanda de espacios de estudio y problemas con la disponibilidad de las bibliotecas. Un claro ejemplo de ello se puede observar en la Universidad Nacional Mayor de San Marcos. Esta institución universitaria, siendo una de las más grandes del país, cuenta con la infraestructura necesaria para poder atender 2500 estudiantes de manera simultánea (Montoro, 2024). Sin embargo, esto resulta insuficiente debido a la gran cantidad de alumnos. Este problema no está exento de instituciones privadas, ya que universidades como la UPC también han demostrado tener ciertas deficiencias con sus plataformas. Sobre ello, el 31 de marzo de 2025, estudiantes de la UPC denunciaron un colapso total durante el proceso de matrícula en su nueva plataforma conocida como Banner. Esto comenzó el 24 de marzo cuando una gran cantidad de alumnos intentaron acceder al sistema, pero múltiples fallas impidieron completar el proceso, obligando a la universidad a suspender temporalmente la matrícula y forzando a los estudiantes a hacer largas colas presenciales (Diario UNO, 2025). Esta situación logró solventarse gracias a la participación masiva de la manifestación del alumnado y la actualización continua del sistema. No obstante, esta solución solo abarcó las principales funcionalidades de la plataforma y hasta el día de hoy mantiene algunas en espera de mejora, lo cual ocasionó que el personal universitario simplemente tuviera que adaptarse a los cambios.

Una de estas funcionalidades que quedaron deprecadas componen el sistema de reserva de cubículos y espacios de estudio pertenecientes a la biblioteca. Por ello, baje ese contexto, presentamos Bib Flip como una solución tecnológica desarrollada para mejorar la experiencia del uso de cubículos estudiantiles en universidades peruanas. Para ello, la propuesta deberá incluir una aplicación web y móvil con dispositivos IoT integrados. Además, nuestro enfoque es principalmente resolver problemas relacionados con la disponibilidad y la ocupación de cubículos universitarios en tiempo real para reducir tiempos de espera.

Bib Flip brindará a los estudiantes la posibilidad de verificar la disponibilidad de espacios individuales y grupales, hacer reservas y revisar sus detalles. Además, ofrecerá un mapa interactivo para ubicar las sedes disponibles, así como la opción de registrar nuevos usuarios para acceder a los servicios de la aplicación. Además, contará con un panel exclusivo para que administradores seleccionados puedan gestionar los cubículos de estudio y visualizar las reservas activas, permitiendo un control actualizado del uso de cubículos, planificar la distribución de recursos de manera eficiente y optimizar la organización operativa.

Misión: Ofrecer una solución tecnológica eficiente y accesible en la gestión de espacios de estudio para mejorar la experiencia académica y las operaciones de la biblioteca.

Visión: Consolidarse como la principal solución digital en el Perú para la gestión de procesos en bibliotecas universitarias, promoviendo una experiencia de estudio más ordenada, ágil y satisfactoria para los estudiantes, y una gestión más eficiente para las instituciones educativas.

1.1.2. Perfiles de integrantes del equipo

Photo	Description
	<p style="text-align: center;">Aranda Vallejos, Oscar Gabriel</p> <p>Tengo 20 años y soy estudiante de la carrera de Ingeniería de Software, poseo conocimientos básicos en Unity, C++ y diseño web. Me considero una persona preparada y perseverante en cumplir con los objetivos del proyecto; además, siempre estoy dispuesto a aprender nuevos conceptos.</p>
	<p style="text-align: center;">Bernaola Pérez, André Arturo</p> <p>Estoy cursando la carrera de Ingeniería de Software, me gusta jugar videojuegos y aprender cosas nuevas en mis ratos libres. Aspiro a trabajar como desarrollador fullstack y me interesa mantenerme actualizado en nuevas tecnologías, buenas prácticas de programación y metodologías ágiles.</p>
	<p style="text-align: center;">Gutierrez Garcia, José Eduardo</p> <p>Tengo 21 años, actualmente me encuentro cruzando mi 7mo ciclo de la carrera de ingeniería de software en la UPC. Me gusta jugar videojuegos y practicar natación, soy un gran aficionado de la tecnología y del ensamblaje de computadoras. Me considero una persona dispuesta siempre a aprender tecnologías nuevas, creativa y responsable.</p>
	<p style="text-align: center;">Oliveira Paucar, Mauricio</p> <p>Me gusta mucho aprender cosas nuevas sobre mi carrera, trabajar en equipo de manera proactiva y lograr los objetivos junto a mis compañeros. Me considero una persona ambiciosa, ya que mi meta es llegar a obtener un alto cargo en una empresa que me agrade o formar mi propia empresa relacionada al software. Gracias a ello siempre podré trabajar en algo que me guste y llevar una vida cómoda. Cuento con conocimientos de C++ y HTML.</p>
	<p style="text-align: center;">Pedraza Maldonado Joaquín</p> <p>Estudio Ing. Software. Me considero que soy una persona perseverante, entusiasta en aprender cosas nuevas. Me gusta ayudar a los demás y sé trabajar en equipo. Cuento con conocimientos en lenguajes de Programación como C++, Python, CSS, JavaScript.</p>

Photo	Description
	Velarde Luyo, Piero Alberto Soy Piero, estudio la carrera de ingeniería de software en la Universidad Peruana de Ciencias Aplicadas. Escogí esta carrera por mi facilidad en el uso de las computadoras. Asimismo, por mi interés en el funcionamiento de las anteriores mencionadas y todo respecto a la programación y las tecnologías emergentes.
	Soriano Medrano, Diego Estudio la carrera de ingeniería de software. Me considero una persona creativa y paciente. Tengo conocimiento en el uso de diferentes herramientas informáticas y lenguajes de programación. que ayudan a realizar distintos tipos de trabajo y a resolver problemas. Parte de mis habilidades blandas es siempre tomar en cuenta la opinión de mis compañeros, lo cual me facilita el poder trabajar en equipo, para agilizar diferentes actividades.

1.2. Solution Profile

1.2.1. Antecedentes y problemática

What (¿Qué problema abordamos?)

La administración ineficaz de los espacios de estudio en las bibliotecas universitarias peruanas es el problema principal. Los alumnos experimentan incertidumbre con respecto a la disponibilidad de cubículos, se frustran cuando no encuentran espacios libres y deben esperar mucho tiempo. Asimismo, el sistema que las universidades suelen otorgar para la reserva de espacios de estudio conlleva un proceso lento y poco eficiente en la mayoría de situaciones.

Why (¿Por qué existe este problema?)

La raíz del problema surge del crecimiento exponencial de la matrícula universitaria, infraestructura insuficiente para atender la demanda real y procesos deficientes para la administración de reservas.

Where (¿Dónde ocurre el problema?)

El problema surge en las bibliotecas de instituciones universitarias peruanas. Para ello, nos centraremos principalmente en la sede San Miguel de la Universidad Peruana de Ciencias Aplicadas.

When (¿Cuándo se intensifica el problema?)

La problemática se agudiza en diferentes períodos como las semanas de entregas de trabajos parciales y finales, semanas de presentación y envío de avances académicos y en horarios con gran concurrencia de alumnos (mañana y tarde).

Who (¿Quién está afectado?)

Se identifican entre los principales afectados a los estudiantes universitarios que experimentan dificultades para acceder a espacios de estudio cuando los necesitan, especialmente durante períodos altamente concurridos como en semanas de trabajos parciales y finales. En segundo lugar, también tenemos al personal de bibliotecas universitarias que enfrentan desafíos en la gestión de cubículos.

How (¿Cómo?)

La solución se llevará a cabo integrando sensores de medición de peso para cada asiento del cubículo, los cuales estarán vinculados por medio de una red IoT que envíe datos en tiempo real a un sistema centralizado para la gestión bibliotecaria. Además, los estudiantes podrán visualizar la disponibilidad y el personal del establecimiento podrá gestionar la ocupación de manera óptima y recibir alertas en caso de anomalías a través de una aplicación móvil y web.

How much (¿Cuánto?)

¿Cuánto afecta este problema?

La gestión ineficiente de espacios de estudio genera pérdidas académicas significativas para las instituciones universitarias debido a la disminución en la satisfacción estudiantil, lo cual se traduce en una menor productividad académica por parte de los estudiantes.

¿Cuánto costará resolver este problema?

La problemática identificada no requiere una reestructuración completa del sistema bibliotecario universitario, sino una intervención tecnológica específica que resuelva los principales obstáculos: el acceso rápido a información de disponibilidad en tiempo real y una mejor planificación anticipada de espacios de estudio. En este sentido, la solución propuesta no busca reemplazar los procesos académicos tradicionales, sino complementarlos con herramientas digitales que permitan a las bibliotecas universitarias peruanas operar con mayor eficiencia, transparencia y satisfacción estudiantil.

Por ello, el costo de implementación variará según el número de cubículos y la infraestructura tecnológica requerida para la instalación y mantenimiento de los sensores, así como el desarrollo de la plataforma digital integrada al sistema bibliotecario existente.

¿Cuántas personas se beneficiarán?

Esta solución beneficiará directamente a estudiantes universitarios y al personal bibliotecario, con un impacto indirecto en toda la comunidad académica. Se estima que, en bibliotecas universitarias de tamaño mediano, la implementación de esta tecnología podría mejorar la eficiencia en la asignación de espacios en un 35-45%, beneficiando diariamente a cientos de estudiantes y optimizando las labores del personal bibliotecario.

1.2.2. Lean UX Process

1.2.2.1. Lean UX Problem Statements

Problem Statement 1 - Personal de Bibliotecas: Los sistemas utilizados para la administración de cubículos en bibliotecas universitarias no aprovechan las tecnologías actuales con las que contamos y presentan deficiencias en su rendimiento. Por ello, nuestra propuesta abordará esta brecha mediante un panel de administración integral que proporcione una gestión óptima y análisis de datos en tiempo real; además, estará dirigido al personal de la sede San Miguel de la UPC.

Problem Statement 2 - Estudiantes Universitarios: El estado actual para la reserva de espacios de estudio en bibliotecas universitarias se compone principalmente de procesos poco eficientes. Los servicios existentes no permiten que los estudiantes puedan revisar la disponibilidad de los cubículos o realizar reservas de manera rápida y cómoda. Por ello, nuestra propuesta abordará esta brecha mediante una plataforma digital integrada con dispositivos IoT y nuestro enfoque estará dirigido a estudiantes universitarios de la sede San Miguel de la UPC.

1.2.2.2. Lean UX Assumptions

Business Outcomes:

- Aumento del 30% en la eficiencia de utilización de cubículos en las bibliotecas que adopten la solución.
- Reducción del 80% en el tiempo promedio de búsqueda de espacios de estudio disponibles.
- Incremento del 25% en la satisfacción de los estudiantes, medido a través de encuestas de experiencia.
- Reducción del 40% en incidencias por ocupación indebida y problemas de asignación de cubículos.
- Mejora del 35% en la planificación y distribución de recursos bibliotecarios.

Users:

Los usuarios se dividen en estos segmentos:

- Personal de biblioteca: Administradores y operadores de 25 a 55 años responsables de la gestión diaria de cubículos.
- Estudiantes universitarios: Jóvenes de 17 a 35 años que buscan espacios de estudio para optimizar su rendimiento académico y reunirse en equipo.

User Outcomes & Benefits:

Personal de Biblioteca:

- Control óptimo del uso y ocupación de cubículos en tiempo real.
- Acceso a estadísticas e informes detallados que faciliten la toma de decisiones estratégicas.
- Optimización en la distribución de recursos y planificación operativa.
- Escalabilidad para el registro de nuevos cubículos en el sistema.

Estudiantes:

- Información precisa y en tiempo real sobre la disponibilidad de cubículos de estudio.
- Reducción significativa de tiempos de espera y búsqueda de espacios disponibles.
- Mejora en la planificación de sesiones de estudio mediante reservas anticipadas.
- Mayor certeza y tranquilidad al acceder a espacios de estudio cuando los necesitan.

Feature Assumptions:

- Visualización en tiempo real del estado de ocupación de cubículos individuales y grupales.
- Funcionalidad de reservas digitales con confirmación automática y recordatorios.
- Mapas interactivos de ubicación de cubículos disponibles por sede universitaria.
- Panel de gestión integral para administradores con métricas de utilización.
- Sistema de notificaciones push para estudiantes y personal administrativo.
- Integración con dispositivos IoT para detección automática de ocupación.

Business Assumptions:

1. El cliente necesita optimizar la gestión de los cubículos de estudio mediante un sistema automatizado que permita conocer la disponibilidad y administrar reservas en tiempo real, reduciendo la carga operativa del personal y mejorando la eficiencia general del servicio.
2. Las necesidades del cliente se resolverán mediante la implementación de una plataforma digital conectada con sensores IoT que detecten la ocupación de los cubículos, actualicen el estado en tiempo real y generen reportes automáticos para el análisis de uso.
3. Los clientes son (o serán) las instituciones universitarias, en particular el personal administrativo de las bibliotecas encargado de la asignación, control y monitoreo de los espacios de estudio.

4. El cliente quiere una herramienta confiable, rápida y moderna que le permita mejorar la eficiencia en la gestión, reducir errores humanos y ofrecer una mejor experiencia de servicio a los estudiantes.
5. El cliente también puede obtener informes detallados de ocupación, análisis de patrones de uso y datos estratégicos que le permitan planificar mejor la infraestructura y la demanda de espacios.
6. Conseguirán mi base de clientes mediante alianzas con universidades, demostraciones piloto, participación en ferias académicas y estrategias de marketing directo dirigidas a áreas administrativas y tecnológicas de instituciones educativas.
7. La financiación se obtendrá mediante la venta de licencias institucionales para el uso de la plataforma y servicios adicionales de instalación, soporte técnico y mantenimiento de la infraestructura IoT.
8. Mi principal competencia es los sistemas tradicionales de gestión manual de cubículos y las plataformas de reserva básicas que no cuentan con integración IoT ni análisis de datos en tiempo real.
9. Superaremos a la competencia mediante la optimización de procesos con tecnología IoT, una interfaz intuitiva y un sistema de reportes avanzados que permitan una gestión inteligente y una experiencia de usuario superior.
10. Mi mayor riesgo es la resistencia institucional a adoptar nuevas tecnologías y los costos iniciales asociados a la implementación de la solución.
11. Resolveremos este riesgo mediante la realización de pruebas piloto gratuitas, capacitaciones al personal, y la presentación de resultados tangibles que demuestren la eficiencia, el ahorro de tiempo y el retorno de inversión.
12. Los supuestos que, si resultan falsos, harán que el proyecto fracase son que las universidades no estén dispuestas a invertir en la modernización de sus sistemas de gestión, o que el personal administrativo no adopte de forma activa el uso de la plataforma digital.

User Assumptions:

¿Quién es el usuario?

El usuario es tanto el personal administrativo de las bibliotecas universitarias, responsable de gestionar los espacios de estudio, como los estudiantes que utilizan los cubículos para sus actividades académicas diarias.

¿Dónde encaja nuestro producto en la vida (o trabajo) del usuario?

Para el personal administrativo, se integra directamente en sus tareas diarias de control y asignación de espacios; para los estudiantes, se convierte en parte de su rutina académica al facilitar la búsqueda y reserva de cubículos de manera rápida y sencilla.

¿Qué problemas resuelve el producto para el usuario?

Resuelve la falta de información en tiempo real sobre la disponibilidad de cubículos, los tiempos perdidos buscando espacios libres, y la frustración derivada de la desorganización en las reservas, mejorando la eficiencia y la satisfacción general.

¿En qué contexto el usuario utiliza el producto?

Durante los horarios académicos, especialmente en épocas de alta demanda como exámenes o entregas de proyectos, tanto desde sus teléfonos móviles como desde computadoras o terminales ubicadas en la biblioteca.

¿Cómo debería verse y comportarse el producto?

Debe tener una interfaz moderna, limpia y visualmente intuitiva, con navegación simple y tiempos de respuesta menores a dos segundos. El sistema debe ser estable, sincronizar datos entre dispositivos y ofrecer una experiencia fluida y confiable.

1.2.2.3. Lean UX Hypothesis Statements

Hypothesis Statement 1:

Creemos que la eficiencia operativa de las bibliotecas mejorará significativamente si el personal administrativo logra incrementar en un 30% su eficiencia y reducir en un 40% las tareas manuales de gestión mediante una herramienta automatizada que centralice la administración de cubículos y genere reportes dinámicos en tiempo real.

Hypothesis Statement 2:

Creemos que la satisfacción y el uso eficiente de los espacios de estudio aumentarán si los estudiantes universitarios logran reducir en un 30% el tiempo de búsqueda de cubículos y mejorar en un 25% su satisfacción general mediante una aplicación web y móvil que les permita visualizar disponibilidad y realizar reservas fácilmente.

Hypothesis Statement 3:

Creemos que la utilización efectiva de los cubículos y el retorno de inversión de las bibliotecas mejorarán si el personal administrativo logra aumentar en un 30% la ocupación promedio de los espacios mediante la implementación de sensores IoT que transmitan datos en tiempo real a una plataforma centralizada de gestión.

Hypothesis Statement 4:

Creemos que la adopción y satisfacción de los usuarios aumentarán de forma sostenida si el personal administrativo y los estudiantes logran completar tareas básicas en la aplicación sin asistencia en al menos un 85% de los casos y mantener altos niveles de satisfacción mediante una interfaz digital intuitiva y amigable.

Hypothesis Statement 5:

Creemos que la eficiencia en la localización de espacios y la experiencia de los estudiantes mejorará notablemente si los estudiantes universitarios logran reducir en un 60% el tiempo necesario para ubicar cubículos y aumentar en un 70% el uso de la funcionalidad mediante un sistema de mapas interactivos y navegación dentro de las bibliotecas.

1.2.2.4. Lean UX Canvas

Business Problem	Solution Ideas	Business Outcomes
<p>¿Qué problema tiene el negocio que estás tratando de resolver?</p> <p>No poder ser consciente de la disponibilidad verdadera y en tiempo real de cubículos de estudio universitarios</p> <p>Consecuencias</p> <ul style="list-style-type: none"> Ineficiencia en la gestión de espacios Largos tiempos de búsqueda para estudiantes Impacto negativo en la experiencia académica 	<p>¿Qué podemos hacer para resolver nuestro problema de negocio y satisfacer las necesidades de nuestros clientes al mismo tiempo?</p> <ul style="list-style-type: none"> Implementación de sensores IoT en cada cubículo para detectar ocupación en tiempo real Visualización de disponibilidad de cubículos en la aplicación Panel de gestión para administradores y supervisores 	<p>¿Cómo sabrás que resolviste el problema del negocio?</p> <ul style="list-style-type: none"> Reducir el índice de estrés universitario en un 2% Garantizar que el número de cancelaciones de reservas por tardanza sea menor a 10 por mes Lograr un puntaje promedio en encuestas sobre los recursos de la universidad que alcance los 8 de 10 puntos
Users	User Outcomes & Benefits	
<ul style="list-style-type: none"> Personal de bibliotecas universitarias Estudiantes universitarios 	<p>Personal de bibliotecas universitarias</p> <ul style="list-style-type: none"> Control del uso y ocupación de cubículos Gestión de la creación, actualización y eliminación de cubículos Asignación de supervisores para división de responsabilidades <p>Estudiantes</p> <ul style="list-style-type: none"> Información precisa y actualizada sobre disponibilidad de cubículos y horarios Reducción de tiempos de búsqueda 	
Hypothesis		
<p>Creemos que el personal administrativo y los estudiantes universitarios valorarán una herramienta que permita gestionar y conocer la ocupación de cubículos en tiempo real de manera automatizada con sensores IoT. Sabremos que esto es cierto cuando los resultados de las encuestas realizadas por la universidad de manera periódica demuestren una calificación positiva en el 90% de los casos para la sección de recursos y espacios bibliotecarios.</p>		
<p>What's the most important thing we need to learn first?</p> <ul style="list-style-type: none"> Aprender a diagramar la arquitectura IoT del sistema conectado a la aplicación Simular el sistema IoT en Wokwi mediante un ESP32 y sensores de peso Desarrollar un Edge API que se conecte al sistema IoT simulado Integrar el Edge API con el Backend para que todo esté conectado con la aplicación 		<p>What's the least amount of work we need to do?</p> <p>Tener un Producto Mínimo Viable mediante la presentación de un prototipo con sensores y un sistema IoT Edge conectado a nuestra aplicación que demuestre en tiempo real la ocupación de un cubículo piloto.</p> <p>Además, debemos desarrollar la versión inicial de la app web y móvil para la visibilidad de cubículos disponibles</p>

1.3. Segmentos objetivo

Para el desarrollo de nuestra propuesta, se han identificado dos segmentos objetivo principales que representan los stakeholders clave del sistema propuesto.

Segmento 1: Personal de bibliotecas universitarias

Este segmento está conformado por el personal administrativo y operativo de las bibliotecas universitarias, quienes son responsables de la gestión diaria de los espacios de estudio y servicios bibliotecarios. Además, para este segmento se tomará como caso de prueba al personal de la Universidad Peruana de Ciencias Aplicadas en la sede San Miguel. Esto debido a su alta demanda como una de las mejores instituciones privadas en el Perú.

Características Demográficas

- Edad: Personal adulto, entre los 25 y 55 años
- Educación: Técnica o universitaria
- Perfil: Personal con experiencia en gestión de servicios bibliotecarios y atención al usuario

Características Geográficas

- Ubicación: Lima, Perú

Segmento 2: Estudiantes universitarios

Este segmento está compuesto por estudiantes que utilicen regularmente los espacios de estudio de las bibliotecas universitarias. Además, se tomará como caso de prueba a los estudiantes de la UPC. Esto debido a los casos recientes sobre el reporte de fallos con el nuevo sistema.

Características Demográficas

- Edad: Principalmente entre los 17 y 35 años
- Educación: Estudiantes de pregrado y posgrado de diversas carreras profesionales
- Perfil: Estudiantes con familiaridad en el uso de aplicaciones móviles y sistemas digitales

Características Geográficas

- Ubicación: Lima, Perú

Capítulo II: Requirements Elicitation & Analysis

2.1. Competidores

Algunos de los competidores a los que Bib Flip podría enfrentarse son:

- LibCal (Springshare):** Plataforma de gestión de espacios y reservas para bibliotecas que permite a los usuarios reservar salas de estudio, cubículos y equipos. LibCal se destaca por su integración con sistemas de bibliotecas académicas y herramientas de análisis, aunque su sistema depende de reservas manuales y no cuenta con detección automática de ocupación en tiempo real.
- SpaceBooking:** Sistema de reserva de espacios para instituciones educativas que facilita la gestión de aulas, laboratorios y espacios de estudio. Ofrece una interfaz web y móvil para reservas, pero funciona principalmente con confirmación manual de ocupación por parte del personal.
- Midas:** Sistema de reserva de salas y recursos basado en web, diseñado para gestionar la programación y reserva de espacios en diversas organizaciones, incluyendo bibliotecas. MIDAS ofrece gestión de reservas, programación y administración para salas de biblioteca, espacios de estudio, computadoras y otros recursos bibliotecarios, aunque opera principalmente con reservas manuales y no incluye detección automática de ocupación en tiempo real.

2.1.1. Análisis competitivo

Competitive Analysis Landscape					
¿Por qué llevar a cabo este análisis?		Este análisis se lleva a cabo para poder investigar, analizar y comparar el comportamiento de los competidores directos o indirectos en el mercado de gestión de espacios bibliotecarios.			
Nombre	Bib Flip	LibCal	SpaceBooking	Midas	
Logo					
Perfil	Overview	Es una solución tecnológica integral basada en IoT, diseñada específicamente para bibliotecas universitarias. Utiliza sensores de ocupación para detectar automáticamente el uso de cubículos y actualiza esta información en tiempo real en una app web y móvil. El sistema también ofrece una interfaz administrativa completa para el personal de biblioteca, facilitando la gestión del flujo de estudiantes y optimizando la utilización de espacios.	Es una plataforma de gestión de espacios bibliotecarios basada en la nube que permite manejar reservas de salas, cubículos, equipos y recursos. Su enfoque está en ofrecer una experiencia completa para bibliotecas académicas que desean profesionalizar la gestión de sus espacios, utilizando herramientas de análisis, reportes y integración con catálogos bibliotecarios.	Es una solución de reserva de espacios para instituciones educativas que facilita la gestión centralizada de aulas, laboratorios, salas de conferencias y espacios de estudio. Los usuarios pueden buscar espacios disponibles, realizar reservas y gestionar recursos desde una plataforma web, aunque su enfoque principal está en espacios institucionales más que bibliotecarios específicamente.	Es una aplicación que digitaliza el proceso de reserva de espacios académicos, incluyendo cubículos de biblioteca y salas de estudio. Los estudiantes pueden reservar espacios desde sus dispositivos y recibir confirmaciones, aunque el sistema depende de la actualización manual por parte del personal y no ofrece monitoreo automático de ocupación.
Ventaja competitiva ¿Qué valor ofrece a los clientes?	Brinda información precisa y en tiempo real sobre la disponibilidad de cubículos, ayudando a estudiantes a evitar búsquedas infructuosas y optimizando su tiempo de estudio desde antes de llegar a la biblioteca.	Permite gestionar reservas de múltiples recursos bibliotecarios de forma integrada y recibir estadísticas detalladas sobre el uso de espacios, facilitando la toma de decisiones administrativas.	Facilita la reserva centralizada de diversos espacios institucionales de forma rápida y confiable, brindando acceso a una amplia variedad de recursos educativos en una sola plataforma.	Reduce la incertidumbre sobre disponibilidad de espacios de estudio, permitiendo a los estudiantes planificar mejor sus sesiones académicas mediante reservas anticipadas.	
Perfil de Marketing	Mercado objetivo	Bibliotecas universitarias que buscan innovar en la gestión de espacios de estudio y mejorar la experiencia académica de sus estudiantes mediante tecnología avanzada e integración IoT.	Bibliotecas académicas y públicas que buscan mejorar su gestión de recursos y espacios. Su mercado abarca desde pequeñas bibliotecas universitarias hasta grandes sistemas bibliotecarios institucionales.	Instituciones educativas (universidades, colegios, institutos) que buscan gestionar eficientemente sus espacios académicos y optimizar la utilización de recursos físicos.	Universidades y bibliotecas académicas que buscan digitalizar sus procesos de reserva de espacios y mejorar la experiencia estudiantil en el acceso a recursos de estudio.
Estrategias de Marketing	Demostraciones en vivo para	Marketing directo a	Marketing directo a	Pruebas gratuitas y	

	personal bibliotecario.	bibliotecarios.	instituciones educativas.	pilotos.	
	Alianzas estratégicas con universidades.	Participación en eventos del sector.	Demostraciones en ferias educativas.	Testimonios de universidades.	
	Marketing de contenidos educativos.	Programa de referencias.	Alianzas con proveedores de tecnología educativa.	Marketing digital dirigido a estudiantes.	
	Participación en conferencias bibliotecarias.				
Perfil de producto	Productos y Servicios	Solución basada en IoT que utiliza sensores de ocupación para detectar el uso de cubículos en tiempo real. Esta información se actualiza automáticamente en una app web y móvil, permitiendo a estudiantes ver disponibilidad antes de llegar. Incluye panel administrativo para gestión de espacios, reportes de utilización y análisis de patrones de uso estudiantil.	Proporciona un sistema de reservas bibliotecarias, gestión de equipos, herramientas de análisis de uso, integración con sistemas de bibliotecas (ILS), y funciones de calendario para eventos. También incluye módulos para gestión de salas de estudio grupales y recursos especializados.	Ofrece gestión centralizada de espacios institucionales, sistema de reservas multi-recurso, reportes de utilización, integración con sistemas académicos, y herramientas de administración para diferentes tipos de usuarios (estudiantes, profesores, personal).	Proporciona una aplicación móvil y web para reservas de espacios de estudio, sistema de notificaciones, gestión básica de disponibilidad, y herramientas simples de reporte para el personal bibliotecario.
	Precios y Costos	Ofrece un modelo de suscripción anual por biblioteca, con precios que varían según el número de cubículos y funcionalidades requeridas. Incluye costo inicial por instalación de sensores IoT. No cobra comisiones por reserva, siendo más rentable a largo plazo para instituciones con alto volumen de uso.	Sigue un modelo de suscripción anual, con precios desde \$3,000 anuales dependiendo del tamaño de la biblioteca y módulos contratados. Los precios pueden incluir costos de implementación y capacitación del personal.	Opera con modelo de suscripción institucional, comenzando desde \$2,500 anuales para instituciones pequeñas. Precios varían según número de usuarios y espacios gestionados. Incluye soporte e implementación básica.	Modelo de suscripción mensual desde \$150 para bibliotecas pequeñas. Precios escalables según número de usuarios activos. No incluye hardware adicional, siendo la opción más económica pero con funcionalidades limitadas.
	Canales de distribución	- Página web institucional - Aplicaciones móviles iOS y Android - Panel web para administradores	- Página web - Integración con sistemas bibliotecarios existentes - Aplicaciones móviles	- Plataforma web institucional - APIs para integración - Aplicaciones móviles	- Página web - Aplicaciones móviles iOS y Android
Análisis FODA	Fortalezas	- Detección automática de ocupación en tiempo real. - Especialización en bibliotecas universitarias. - App ligera con actualizaciones rápidas. - Instalación simple de sensores IoT.	- Plataforma madura y establecida. - Amplia integración con sistemas bibliotecarios. - Funcionalidades completas de gestión. - Base sólida de clientes académicos.	- Versatilidad para diferentes tipos de espacios. - Interfaz robusta y escalable. - Buena integración institucional. - Soporte técnico establecido.	- Fácil de implementar y usar. - Costos accesibles. - Enfoque simple y directo. - Sin requerimientos de hardware.
	Debilidades	- Requiere inversión en hardware IoT. - Producto nuevo en el mercado. - Necesidad de validación en más instituciones.	- Sin detección automática de ocupación. - Costos elevados para bibliotecas pequeñas. - Interfaz compleja para usuarios básicos.	- No especializado en bibliotecas. - Falta de integración IoT. - Menos enfocado en experiencia estudiantil.	- Dependiente de actualización manual. - Funcionalidades limitadas. - Sin monitoreo automático. - Reportes básicos.
	Oportunidades	- Expansión a bibliotecas públicas. - Integración con sistemas académicos.	- Adopción de tecnologías IoT. - Expansión en mercados emergentes.	- Especialización en sector bibliotecario. - Integración de tecnologías emergentes.	- Mejora con detección automática. - Expansión de funcionalidades.

	- Desarrollo de analíticas predictivas.	- Nuevos módulos especializados.	- Alianzas con proveedores IoT.	- Integración con sistemas bibliotecarios.
Amenazas	- Competencia de plataformas establecidas.	- Nuevas tecnologías disruptivas.	- Competencia especializada en bibliotecas.	- Soluciones con detección automática.
	- Resistencia al cambio en instituciones conservadoras.	- Soluciones IoT más avanzadas.	- Tecnologías de automatización.	- Plataformas más completas.
	- Posible copia del modelo por competidores.	- Reducción de presupuestos bibliotecarios.	- Preferencias por soluciones integradas.	- Cambios en preferencias institucionales.

2.1.2. Estrategias y tácticas frente a competidores

Hemos empleado un análisis FODA para identificar las posibilidades y desafíos en el mercado de gestión de espacios bibliotecarios, así como para evaluar nuestras fortalezas y debilidades internas. Esta metodología nos ha permitido concebir estrategias y tácticas que se ajusten de manera coherente a nuestro entorno y a los recursos disponibles.

Estrategia de diferenciación tecnológica:

La principal estrategia de Bib Flip es posicionarse como la única solución en el mercado bibliotecario que incorpora tecnología IoT para detectar en tiempo real la ocupación de cubículos, eliminando completamente la dependencia de actualizaciones manuales o estimaciones. Esto permite ofrecer a estudiantes y personal bibliotecario información confiable y constante sobre disponibilidad, algo que nuestros competidores no logran con la misma precisión ni automatización.

Estrategia de enfoque especializado en bibliotecas universitarias:

Una estrategia clave es enfocarse específicamente en el área de bibliotecas universitarias, a diferencia de competidores que abordan múltiples tipos de espacios institucionales. Bib Flip se especializa en entender las necesidades únicas del entorno bibliotecario universitario, donde la alta rotación estudiantil, los períodos de exámenes y la necesidad de espacios silenciosos son fundamentales. De esta manera, podemos enfocarnos en priorizar la experiencia y otorgarle muchos más beneficios a nuestro público objetivo original.

Estrategia de experiencia estudiantil superior:

Bib Flip adopta una estrategia centrada en la experiencia del estudiante, ofreciendo una aplicación móvil intuitiva que no solo muestra disponibilidad sino que predice patrones de uso y sugiere mejores horarios. Además, se implementan notificaciones inteligentes y reservas con recordatorios automáticos.

2.2. Entrevistas

2.2.1. Diseño de entrevistas

Personal de bibliotecas universitarias

Buenas tardes, gracias por asistir a esta entrevista

Preguntas de presentación:

1. ¿Podrías indicarnos tu nombre y edad?
2. ¿Qué puesto ocupas dentro del personal de la biblioteca?
3. ¿Desde cuándo trabajas en esta biblioteca y cuál es su capacidad aproximada?

Preguntas Específicas:

1. ¿Cómo se gestionan actualmente las reservas y la disponibilidad de los cubículos?
2. ¿Cuáles son los problemas más comunes que enfrenta la biblioteca con la ocupación de espacios durante épocas de alta demanda (exámenes, entregas)?
3. ¿Cuáles consideras que son las características más relevantes que debería incluir una plataforma de gestión de aforo para bibliotecas?
4. ¿Le parece interesante una solución tecnológica con dispositivos IOT para gestionar los espacios de estudio en tiempo real?
5. ¿Qué nivel de importancia le darías a la facilidad de instalación y mantenimiento del sistema?
6. ¿Cómo crees que la tecnología puede mejorar la experiencia de los estudiantes y optimizar la gestión de espacios de la biblioteca?
7. ¿Qué datos o métricas les gustaría obtener sobre el uso de los cubículos estudiantiles?

Estudiantes universitarios

Buenas tardes, gracias por asistir a esta entrevista

Preguntas de presentación:

1. ¿Podrías indicarnos tu nombre y carrera?
2. ¿Cuál es tu edad?
3. ¿Con qué frecuencia sueles usar la biblioteca para estudiar?

Preguntas Específicas:

1. ¿Cómo sueles verificar la disponibilidad de espacios de estudio antes de ir a la biblioteca?
2. ¿Para ti qué características debería tener una app de biblioteca para que la uses con frecuencia?
3. ¿Qué importancia le darías a la rapidez y facilidad de uso en una aplicación diseñada para gestionar tu experiencia de estudio en la reserva de cubículos estudiantiles?
4. ¿Te interesaría recibir notificaciones sobre la disponibilidad de tu cubículo o computadora preferida?
5. ¿Has tenido experiencias frustrantes buscando lugar para estudiar? ¿Podrías contarnos sobre ellas?
6. ¿Algo más que consideras importante para mejorar tu experiencia como usuario de la biblioteca?

2.2.2. Registro de entrevistas

Se realizaron tres entrevistas para cada segmento objetivo y estas se encuentran registradas en el siguiente enlace: https://upcedupe-my.sharepoint.com/:v/g/personal/u202218167_upc_edu_pe/EbipDl3neqpBkcsL2Ewh0tIBwZL3w3QbSfnV5e7A5g833w?#nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAIoijPbmVEcm12ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6Ild1YiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOijNeUZpbGVzTGlua0NvcHkifX0&e=h1Nh0

Entrevistas para el Segmento Objetivo 1 - Personal Bibliotecario

Entrevista N°1:

Entrevistado: Juan Carlos Villanueva

Sexo: Masculino

Edad: 39 años

Domicilio: Lima

Inicio de la Entrevista: 0:12

Fin de la Entrevista: 4:44



Resumen de la Entrevista:

Durante la entrevista con Juan Carlos Villanueva, miembro del personal administrativo de la biblioteca, se nos comenta que para la reserva de cubículos en la biblioteca de la UPC se viene utilizando la misma plataforma web desde hace unos años. Por ello, él considera interesante la adición de tecnologías IoT para mejorar la gestión y reserva de cubículos. Sin embargo, menciona que debería hacerse un estudio previo para analizar si la relación entre el beneficio y el costo es realmente positiva.

Entrevista N°2:

Entrevistado: Alonso Rafael

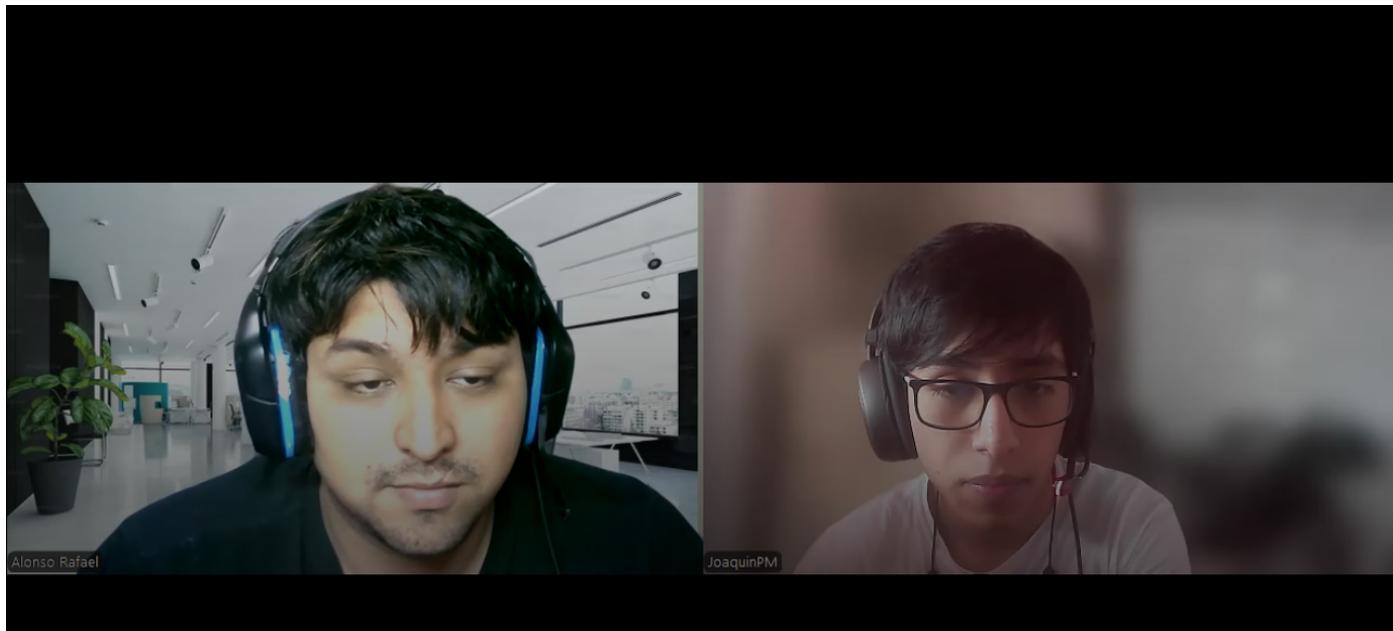
Sexo: Masculino

Edad: 26 años

Domicilio: Lima

Inicio de la Entrevista: 4:45

Fin de la Entrevista: 7:50



Resumen de la Entrevista:

Durante la entrevista con Alonso Rafael, de 26 años y miembro del personal administrativo de la biblioteca, se nos indicó que uno de los aspectos que más le interesaría mejorar es la visualización del aforo en cada cubículo en tiempo real. Alonso considera que una solución tecnológica que permita conocer la disponibilidad inmediata de los espacios sería muy útil, especialmente en épocas de alta demanda. Según comenta, esto ayudaría a reducir significativamente los tiempos de espera para los estudiantes y permitiría una mejor organización del uso de los espacios dentro de la biblioteca.

Entrevista N°3:

Entrevistado: Jorge Luis Céspedes

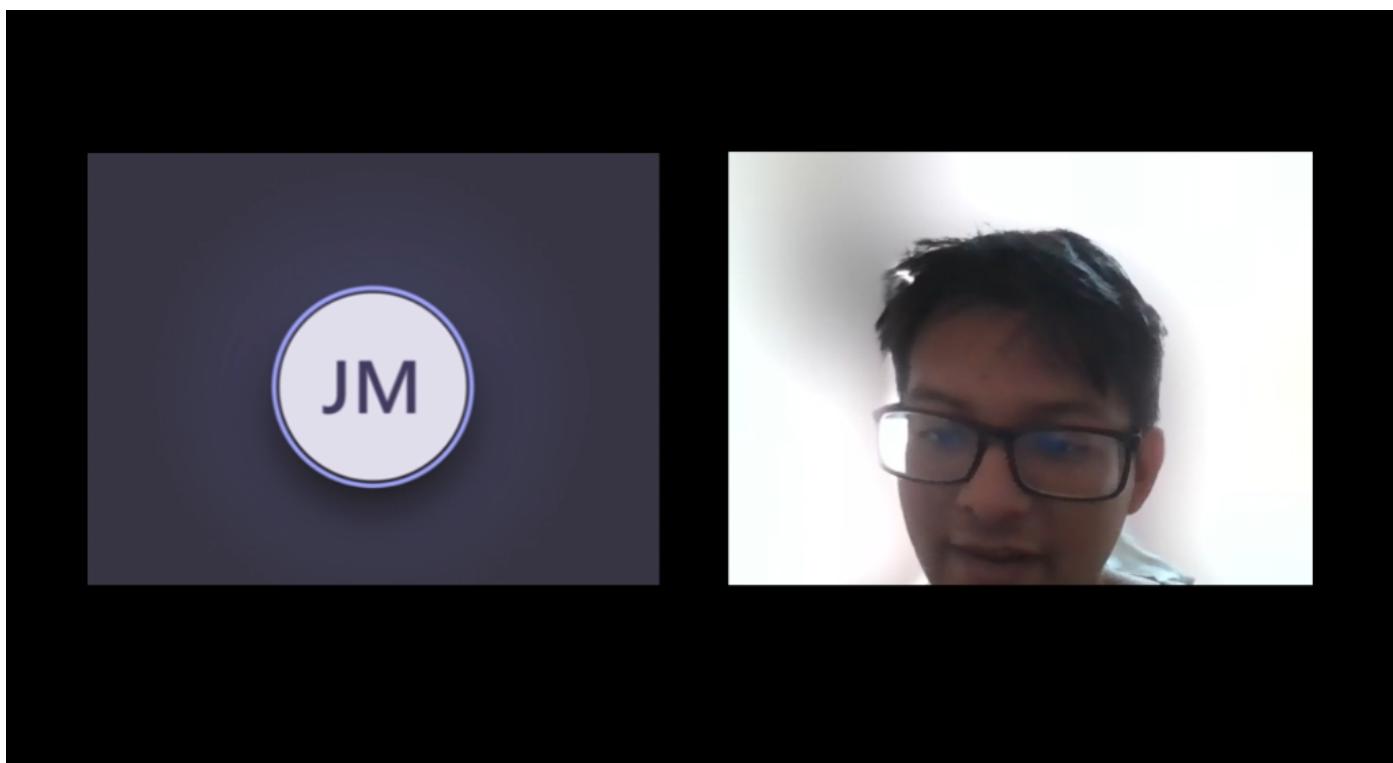
Sexo: Masculino

Edad: 46 años

Domicilio: Lima

Inicio de la Entrevista: 7:51

Fin de la Entrevista: 10:57



Resumen de la Entrevista:

Durante la entrevista con Jorge Luis Céspedes, de 46 años y parte del equipo de atención a estudiantes de la biblioteca, se destacó la necesidad de contar con un sistema más ágil y confiable para la gestión de espacios. Jorge, quien trabaja en la biblioteca desde hace 3 años, señaló que los principales problemas durante épocas de alta demanda incluyen la lentitud del sistema actual, reservaciones no utilizadas y reclamos de estudiantes que no logran acceder a espacios reservados. Considera muy importante que una solución IoT sea fácil de instalar y mantener, y destaca la necesidad de obtener métricas detalladas como horarios de mayor demanda, porcentaje de ocupación real y tiempo promedio de permanencia para optimizar la gestión de los espacios de estudio.

Entrevista para el Segmento Objetivo 2 - Estudiantes Universitarios

Entrevista N°1:

Entrevistado: Nasthya del Carpio

Sexo: Femenino

Edad: 21 años

Domicilio: La Perla

Inicio de la Entrevista: 10:58

Fin de la Entrevista: 15:32



Resumen de la Entrevista:

La entrevista con Nasthya del Carpio, estudiante de 21 años de Diseño de Interiores en la UPC, mostró que suele usar la biblioteca entre 2 y 4 veces por semana, pero enfrenta dificultades para verificar la disponibilidad de cubículos debido a la falta de información confiable. Destacó que una app debería ser rápida, sencilla y mostrar la ocupación en tiempo real, con funciones de reserva, cancelación y notificaciones personalizables. Compartió experiencias frustrantes al no encontrar espacios libres o al hallar ocupado un cubículo reservado, y resaltó la importancia de contar con un sistema de feedback y de garantizar la privacidad de los usuarios.

Entrevista N°2:

Entrevistado: David Rivas

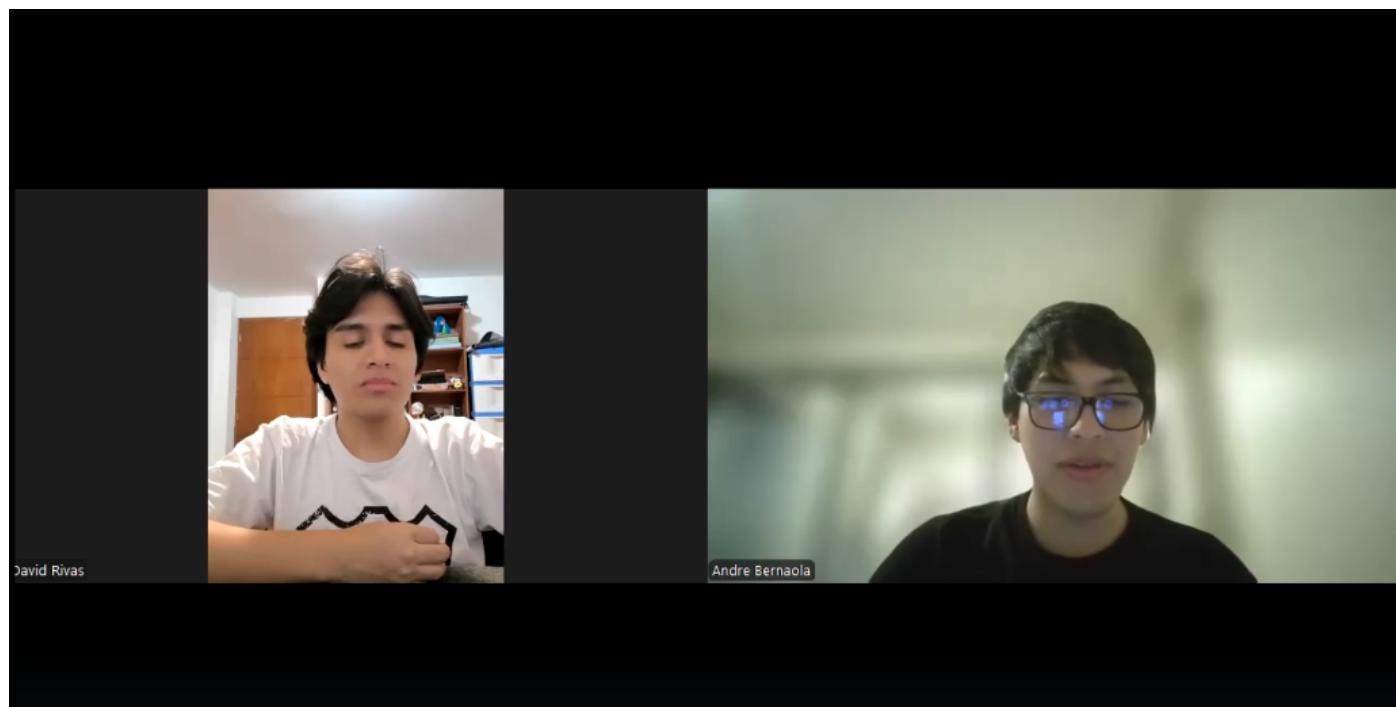
Sexo: Masculino

Edad: 22 años

Domicilio: San Miguel

Inicio de la Entrevista: 15:37

Fin de la Entrevista: 19:28



Resumen de la Entrevista:

David Rivas, estudiante de 22 años de Ingeniería de Sistemas, utiliza la biblioteca principalmente durante épocas de exámenes y proyectos finales, visitándola aproximadamente 3 veces por semana. Su principal problema es la pérdida de tiempo al buscar cubículos disponibles, especialmente en horas pico. Considera esencial que una aplicación permita visualizar mapas de la biblioteca con disponibilidad en tiempo real y la posibilidad de reservar espacios por períodos específicos. Mencionó la importancia de tener filtros por tipo de espacio (silencioso, colaborativo, individual) y sugirió implementar un sistema de penalización para usuarios que no cancelen sus reservas. También valoró la integración con el calendario académico para anticipar períodos de alta demanda.

Entrevista N°3:

Entrevistado: Renzo Silva

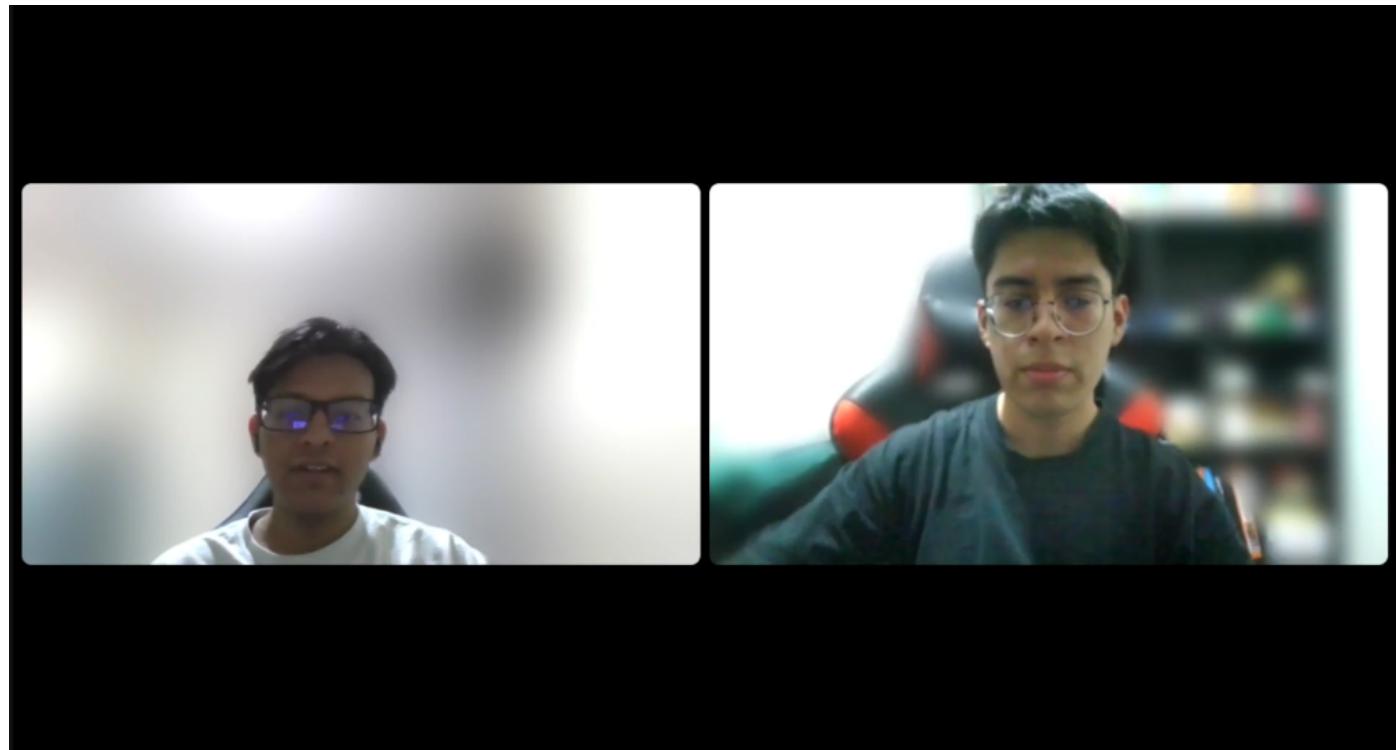
Sexo: Masculino

Edad: 20 años

Domicilio: Pueblo Libre

Inicio de la Entrevista: 19:30

Fin de la Entrevista: 22:27



Resumen de la Entrevista:

Renzo Silva, estudiante de 20 años de Ing. de Software, frecuenta la biblioteca diariamente para estudios grupales e individuales. Su mayor frustración es encontrar cubículos reservados pero vacíos, lo que considera una pérdida de recursos. Propuso que la aplicación incluya verificación de ocupación mediante sensores o check-in manual, y un sistema de liberación automática si el usuario no confirma su llegada en un tiempo determinado. Destacó la necesidad de poder extender reservas desde la app y recibir recordatorios antes del vencimiento. También sugirió funciones sociales como compartir mesa con compañeros conocidos y un sistema de reputación para usuarios responsables. Consideró importante que la app funcione offline para mostrar reservas activas.

2.2.3. Análisis de entrevistas

Segmento Objetivo 1: Personal Bibliotecario

Insights principales:

- Tecnología actual limitada: El personal bibliotecario utiliza la misma plataforma web desde hace años, lo que indica una oportunidad de modernización.
- Apertura a la innovación: Juan Carlos muestra interés en la implementación de tecnologías IoT para mejorar la gestión de cubículos.
- Enfoque en cost-benefit: El personal administrativo prioriza un análisis previo que demuestre que la relación beneficio-costo es positiva antes de implementar nuevas soluciones.
- Experiencia operativa: Su conocimiento del sistema actual les permite identificar las limitaciones y oportunidades de mejora.

Necesidades identificadas:

- Sistema más eficiente para gestionar reservas
- Herramientas de monitoreo en tiempo real
- Justificación clara del retorno de inversión
- Integración con procesos administrativos existentes

Segmento Objetivo 2: Estudiantes Universitarios

Insights principales:

- Uso frecuente: Nasthya utiliza la biblioteca 2-4 veces por semana, indicando una alta demanda de estos espacios.
- Frustración con la disponibilidad: La falta de información confiable sobre ocupación genera experiencias negativas recurrentes.
- Expectativas de inmediatez: Los estudiantes esperan información en tiempo real y procesos rápidos y sencillos.
- Experiencias problemáticas: Situaciones donde encuentran cubículos ocupados a pesar de aparecer disponibles, o viceversa.

Necesidades identificadas:

Las necesidades fundamentales que se han detectado se enfocan en disponer de información real y actualizada sobre la disponibilidad de un cubículo, junto con un procedimiento de reserva eficaz y sencillo que permita la actualización del estado del cubículo. Además, es necesario tener la opción de personalizar las notificaciones, contar con un sistema de retroalimentación que facilite el reporte de problemas e incidencias y disponer de garantías robustas en cuanto a la privacidad y seguridad del manejo de los datos del usuario.

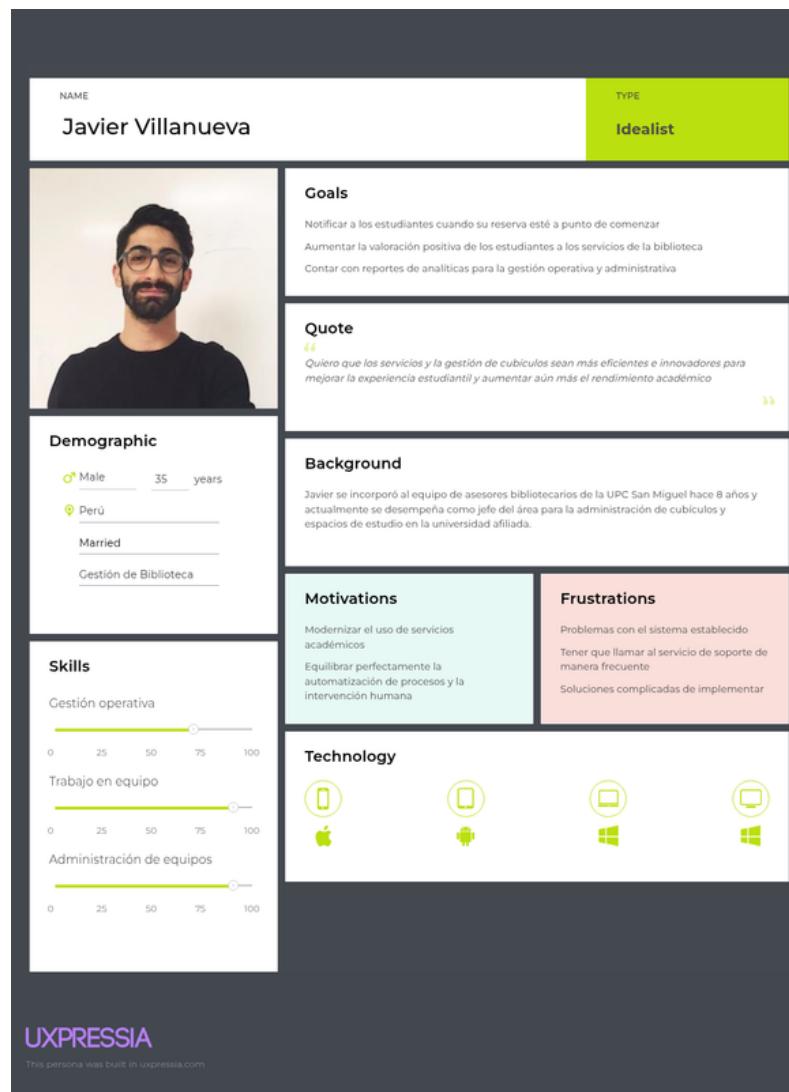
Patrones comunes entre segmentos:

Ambos reconocen las limitaciones del sistema actual. Existe disposición para adoptar nuevas tecnologías. Se valora la eficiencia y la confiabilidad del sistema. Hay una necesidad compartida de información en tiempo real.

2.3. Needfinding

2.3.1. User Personas

Personal Bibliotecario



Estudiantes Universitarios

Miriam Ramírez

NAME: Miriam Ramírez

TYPE: Guardian

Demographic	Goals
Female 22 years Perú Single Estudiante de Pregado	Poder liberar el espacio de un cubículo con facilidad si ya no lo voy a usar Recibir alarmas o recordatorios del cubículo reservado y del tiempo restante Optimizar el proceso de reserva para contar con pocos pasos y con una interfaz más directa
Skills	Background
Plataformas digitales: 85 Creatividad: 80 Gestión del tiempo: 80	Miriam se integró a la comunidad upecina en el año 2022, es decir, hace 3 años. Durante sus primeros ciclos, solía utilizar los cubículos y los espacios de la biblioteca para diferentes actividades. Actualmente se encuentra cursando su 8vo ciclo y tiene que improvisar diferentes opciones porque los cubículos raramente están desocupados.
Motivations	Frustrations
Aprovechar bien su tiempo libre para descansar de las actividades académicas y laborales Ir a lugares donde se sienta cómoda y pueda estudiar tranquila Compartir sus experiencias con sus amigos en redes sociales	Interfaz de la universidad con demasiados procesos Pérdida de tiempo en la búsqueda de cubículos Cancelaciones arbitrarias de cubículos reservados por no haber escrito manualmente un código de confirmación
Technology	
Iconos de Apple, Windows, escritorio y portátil.	

UXPRESSIA
This persona was built in uxpressia.com

2.3.2. User Task Matrix

Personal Bibliotecario (Javier)

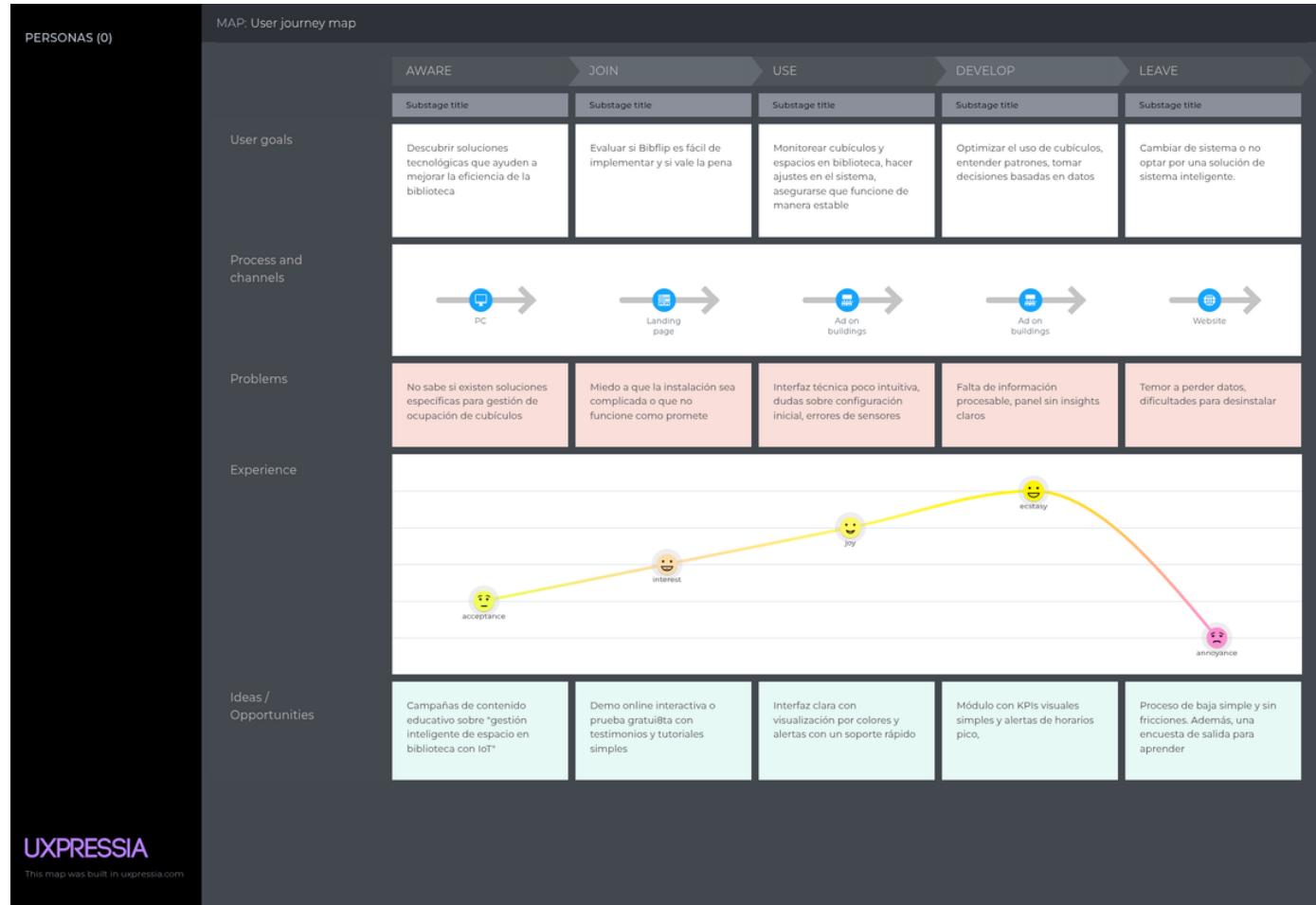
Tarea	Importancia	Frecuencia
Consultar el estado de ocupación de cubículos en tiempo real	Alta	Alta
Agregar/modificar cubículos desde el panel administrativo	Media	Media
Gestionar reservas y cancelaciones desde el panel	Alta	Alta
Configurar parámetros del sistema (horarios, límites de tiempo, etc.)	Media	Baja
Supervisar que el sistema de reservas funcione correctamente	Alta	Alta
Generar reportes de uso y estadísticas para gestión operativa	Alta	Media
Enviar notificaciones automáticas a estudiantes sobre sus reservas	Media	Media

Estudiantes Universitarios (Miriam)

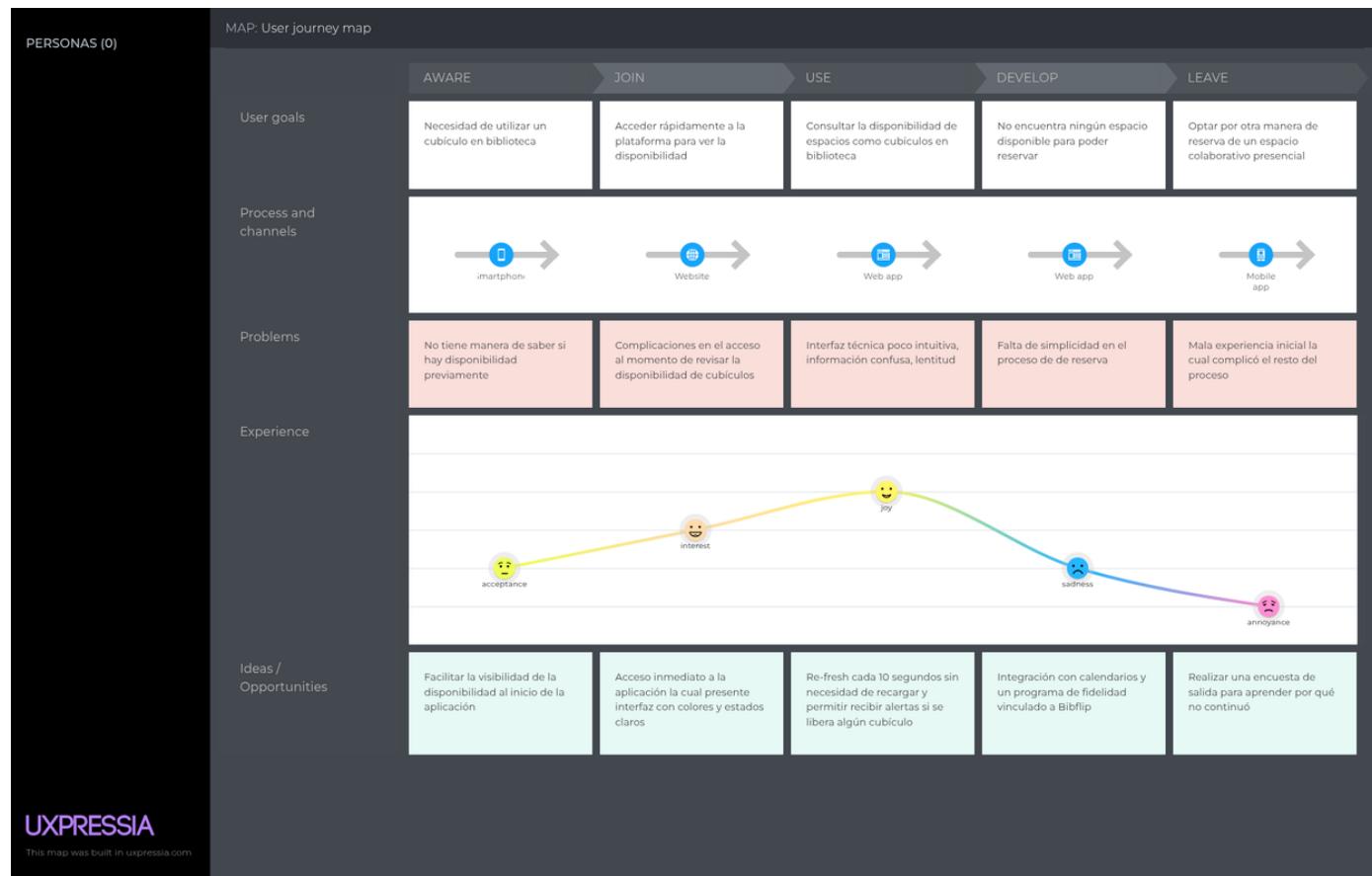
Tarea	Importancia	Frecuencia
Consultar disponibilidad de cubículos en tiempo real	Alta	Alta
Reservar cubículos de estudio individual rápidamente	Alta	Alta
Acceder al sistema desde el celular de manera fácil	Alta	Alta
Recibir recordatorios de su reserva y tiempo restante	Alta	Media
Cancelar o modificar reservas existentes	Media	Media
Liberar cubículo antes del tiempo límite si terminan antes	Media	Media
Buscar cubículos disponibles por ubicación en la biblioteca	Media	Media
Disminuir la cantidad de pasos requeridos para efectuar la reserva	Alta	Alta

2.3.3. User Journey Mapping

Personal Bibliotecario



Estudiantes Universitarios



2.3.4. Empathy Mapping

PERSONA: Empathy map - Javier

1.WHO are we empathizing with?

Jefe de gestión de cubículos en la biblioteca. Administra reservas con un sistema obsoleto que falla seguido. Su rol es garantizar un servicio eficiente para los estudiantes.

7.What do they THINK and FEEL?

Cree que el sistema está desactualizado y siente cansancio por los problemas, pero también motivación para innovar y mejorar la experiencia de los usuarios.

2.What do they need to DO?

Necesita modernizar la gestión con herramientas simples y automáticas. Quiere reducir los problemas técnicos y generar reportes útiles. Será exitoso cuando los estudiantes estén satisfechos y requiera menos soporte.

3.What do they SEE?

Ve estudiantes insatisfechos, un sistema anticuado y presión de la universidad para mejorar. También observa que en otros lugares existen soluciones más modernas.

6.What do they HEAR?

Escucha reclamos constantes de estudiantes, exigencias de la universidad y comentarios de colegas sobre lo complicado que es el sistema. También oye sobre soluciones más avanzadas en otras instituciones.

5.What do they DO?

- Atiende quejas de estudiantes.
- Gestiona manualmente cancelaciones y reservas.
- Pide soporte técnico con frecuencia.

PAINS

- Problemas técnicos frecuentes.
- Dependencia del soporte técnico.
- Complejidad para implementar nuevas soluciones.

GAINS

- Panel integral con analíticas en tiempo real.
- Automatización equilibrada con intervención humana.
- Mayor eficiencia y satisfacción estudiantil.

4.What do they SAY?

- "Necesitamos un sistema más eficiente e innovador."
- "El sistema actual falla demasiado."
- "Con reportes podríamos mejorar la gestión."

UXPRESSIA
This persona was built in upressoia.com

Estudiantes Universitarios

PERSONA: Empathy map - Miriam

1.WHO are we empathizing with? Estudiante universitaria que necesita reservar cubículos para estudiar, pero se enfrenta a un sistema lento e inefficiente. Su rol es el de usuaria final que busca rapidez y confiabilidad.	7.What do they THINK and FEEL? <i>Cree que el sistema es inefficiente y siente frustración al perder tiempo. Lo que más desea es un proceso rápido y confiable que le dé tranquilidad.</i>	2.What do they need to DO? Necesita reservar y liberar cubículos de manera sencilla, evitando procesos largos. Quiere asegurar un espacio disponible y recibir notificaciones. Será exitosa cuando pueda hacerlo en minutos sin perder tiempo.
6.What do they HEAR? <ul style="list-style-type: none"> Quejas de amigos sobre la disponibilidad. Recomendaciones de otros estudiantes para estudiar fuera de la biblioteca. Opiniones de que la universidad debería modernizar sus procesos. 		3.What do they SEE? Ve procesos engorrosos en la universidad y compañeros frustrados. Observa que muchos prefieren no usar los cubículos y nota que en otros lugares existen apps modernas que facilitan estas tareas.
5.What do they DO? <ul style="list-style-type: none"> Improvisa en otros lugares de estudio cuando no consigue cubículo. Dedica tiempo extra a buscar disponibilidad. Comparte experiencias en redes sociales. 		
PAINS <ul style="list-style-type: none"> Procesos complicados de la universidad. Pérdida de tiempo buscando cubículos. Cancelaciones arbitrarias de reservas. 	GAINS <ul style="list-style-type: none"> Proceso simple y rápido de reserva. Notificaciones claras y confiables. Sentirse tranquila y segura de que el cubículo estará disponible. 	4.What do they SAY? <ul style="list-style-type: none"> "Siempre están ocupados los cubículos." "Es mejor no usarlos que perder tiempo buscándolos."

UXPRESSIA
This persona was built in uspressia.com

2.3.5. As-is Scenario Mapping

Este mapeo "As-Is" presenta el escenario actual antes de la implementación de Bibflip, describiendo las experiencias problemáticas para nuestros dos segmentos. Para ello, se detallan acciones, pensamientos y emociones que reflejan las frustraciones y limitaciones del proceso actual.

¿Qué hace el Personal de la Biblioteca?

Phases	Gestión matutina	Supervisión de reservas	Atención al estudiante	Generación de reportes
Doing	<ul style="list-style-type: none"> Llega y revisa el estado de cubículos. Accede al sistema web convencional. Verifica reservas pendientes del día. 	<ul style="list-style-type: none"> Realiza rondas físicas por la biblioteca. Supervisa el uso adecuado de los cubículos. Contrasta información del sistema con la realidad. 	<ul style="list-style-type: none"> Recibe consultas sobre disponibilidad de cubículos. Media conflictos por dobles reservas. Proporciona información desactualizada. 	<ul style="list-style-type: none"> Compila estadísticas basándose en datos incompletos. Intenta generar reportes con información manual. Prepara informes para administración.
Thinking	<p>"Necesito verificar manualmente todo porque el sistema no es confiable."</p> <p>"Este proceso consume mucho tiempo cada mañana."</p>	<p>"El sistema no refleja la ocupación real de los cubículos."</p> <p>"Tengo que caminar constantemente para verificar el uso apropiado."</p>	<p>"No puedo dar información precisa a los estudiantes."</p> <p>"Estos conflictos se repiten por fallas del sistema."</p>	<p>"Los datos que manejo no reflejan la realidad del uso."</p> <p>"¿Cómo justificar mejoras sin datos precisos?"</p>
Feeling	<ul style="list-style-type: none"> Resignación 	<ul style="list-style-type: none"> Cansancio 	<ul style="list-style-type: none"> Estrés 	<ul style="list-style-type: none"> Insatisfacción Preocupación

¿Qué hace el Estudiante?

Phases	Descubrimiento del sistema	Proceso de reserva	Búsqueda física de cubículos	Uso del cubículo
--------	----------------------------	--------------------	------------------------------	------------------

Phases	Descubrimiento del sistema	Proceso de reserva	Búsqueda física de cubículos	Uso del cubículo
Doing	<ul style="list-style-type: none"> Accede al sistema universitario complejo. Navega por múltiples interfaces. 	<ul style="list-style-type: none"> Completa formularios con muchos pasos. Escribe manualmente códigos de confirmación. Espera confirmaciones del sistema lento. 	<ul style="list-style-type: none"> Camina por toda la biblioteca buscando cubículos libres. Pregunta al personal sobre disponibilidad. Pierde tiempo verificando espacios ocupados. 	<ul style="list-style-type: none"> Llega y encuentra su cubículo ocupado por otros. Espera a que desocupen o busca alternativas. Negocia con otros estudiantes el uso del espacio.
Thinking	<p>"Este sistema es muy complicado para algo tan simple."</p> <p>"¿Por qué tantos pasos para reservar un cubículo?"</p>	<p>"Espero no olvidarme de escribir el código de confirmación."</p> <p>"El proceso es demasiado largo."</p>	<p>"Ojalá hubiera una forma de saber qué está disponible sin caminar tanto."</p> <p>"Estoy perdiendo tiempo valioso buscando."</p>	<p>"¿Cómo es posible que mi cubículo reservado esté ocupado?"</p> <p>"En época de exámenes esto puede afectarme mucho."</p>
Feeling	<ul style="list-style-type: none"> Confusión Frustración 	<ul style="list-style-type: none"> Ansiedad Impaciencia 	<ul style="list-style-type: none"> Cansancio Desesperación 	<ul style="list-style-type: none"> Enojo Estrés

2.4. Ubiquitous Language

Término	Definición
Student	Alumno de pregrado o posgrado que utiliza los servicios de la biblioteca universitaria para realizar actividades académicas y de estudio.
Booking	Reservación de un cubículo realizada por un estudiante para garantizar el uso exclusivo del espacio durante un período de tiempo determinado.
Cubicle	Cubículo Universitario (espacio académico compuesto por una mesa de trabajo, una computadora y 6 sillas como máximo) destinado al estudio individual o grupal dentro de las instalaciones de la biblioteca.
Administrator	Personal de la Biblioteca encargado de la gestión operativa de los cubículos, incluyendo la supervisión de reservas, mantenimiento de espacios y atención a usuarios.
SuperAdmin	Equipo de desarrollo encargado de pruebas y mantenimiento
Branch	Sede o campus universitario que cuenta con su propia biblioteca y sistema de cubículos independiente, permitiendo la gestión descentralizada de espacios de estudio.
Library	Biblioteca Universitaria que alberga los cubículos y proporciona el entorno académico para el estudio.
Time Slot	Franja horaria específica durante la cual un cubículo puede ser reservado y utilizado por un estudiante.
Availability	Disponibilidad en tiempo real de los cubículos, indicando cuáles están libres, ocupados o reservados.
Session	Sesión de uso efectivo de un cubículo por parte de un estudiante durante el tiempo reservado.
Notification	Notificación automática enviada a los estudiantes sobre el estado de su reserva, recordatorios o cambios importantes.
Occupancy Status	Estado de ocupación que indica si un cubículo está disponible, reservado, ocupado o fuera de servicio.
Reservation Period	Período de tiempo durante el cual una reserva es válida y el cubículo permanece asignado al estudiante.

Capítulo III: Requirements Specification

3.1. To-Be Scenario Mapping

Este mapeo "To-Be" presenta el escenario ideal posterior a la implementación de Bibflip, describiendo cómo cambiarán las experiencias del propietario del centro de estudio y del usuario gracias a la solución IoT propuesta. Se detallan acciones, pensamientos y emociones alineadas a los beneficios que aporta el sistema.

¿Qué hace el Administrador biblioteca universitarios?

Phases	Doing	Thinking	Feeling
Registro del local	<ul style="list-style-type: none"> Instala la app Bibflip. Crea su cuenta como dueño del centro. Registra su centro de estudio y sedes. 	<ul style="list-style-type: none"> "Qué bueno que puedo registrar todas mis sedes fácil." "Ahora tengo visibilidad online." 	<ul style="list-style-type: none"> Aliviado Organizado
Gestión del aforo	<ul style="list-style-type: none"> Consulta la ocupación en tiempo real de su sede. Revisa cada sede desde su celular o computadora. 	<ul style="list-style-type: none"> "Qué fácil es ver cuántos cubículos están ocupados." "Ya no dependo del personal para esto." "Qué accesible que pueda ver desde cualquier dispositivo la app de Bibflip." 	<ul style="list-style-type: none"> En control Efficiente
Gestión del tiempo de espera	<ul style="list-style-type: none"> Ve cómo los estudiantes reservan online. Evita filas innecesarias en la entrada. 	<ul style="list-style-type: none"> "Qué bueno que ya no hay colas innecesarias." "Mis usuarios llegan más contentos." 	<ul style="list-style-type: none"> Relajado Optimista

Phases	Doing	Thinking	Feeling
Monitoreo de cubículos	<ul style="list-style-type: none"> Agrega o elimina cubículos. Coordina instalación de sensores IoT. 	<ul style="list-style-type: none"> "Wow, es sencillo gestionar los cubículos con ayuda." "No imaginé tanta flexibilidad." 	<ul style="list-style-type: none"> • Empoderado • Innovador

¿Qué hace el Usuario?

Phases	Descubrimiento de la aplicación	Registro de datos personales	Búsqueda de cubículos	Reserva de cubículo
Doing	<ul style="list-style-type: none"> Descarga Bibflip desde la tienda de aplicaciones. Explora las funcionalidades principales. 	<ul style="list-style-type: none"> Registra sus datos en Bibflip. Configura sus centros favoritos. 	<ul style="list-style-type: none"> Utiliza la función de búsqueda por ubicación. Revisa la disponibilidad en tiempo real. 	<ul style="list-style-type: none"> Selecciona un cubículo disponible. Confirma la reserva para una hora específica.
Thinking	"Esta app me ayudará a no perder tiempo buscando cubículos disponibles."	"El proceso de registro es rápido."	"Me encanta poder ver exactamente dónde hay espacio antes de ir."	"La reserva fue sencilla y ahora tengo mi lugar asegurado."
Feeling	Curiosidad	Comodidad	Alivio	Satisfacción

3.2. User Stories

Epics

Epic ID	Name
EPIC-LANDING-001	Funcionalidades landing page
EPIC-CLIENT-002	Funcionalidades user estudiante
EPIC-ADMIN-003	Funcionalidades user administrador
EPIC-SUPERADMIN-004	Funcionalidades user superadmin
EPIC-BOOKING-TECH	Funcionalidades de desarrolladores

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
US001	Ver sección "¿Qué es Bibflip?"	Como visitante de la página, quiero entender qué es Bibflip desde la landing page para conocer su propósito y cómo puede ayudarme como estudiante o administrador de un centro de estudio.	EPIC-LANDING-001	Visitante landing page	<p>Escenario 1:</p> <p>Dado que el visitante accede a la landing page,</p> <p>Cuando hace scroll en "¿Qué es Bibflip?",</p> <p>Entonces el sistema le muestra la sección donde se describe claramente el objetivo de la solución.</p> <p>Escenario 2:</p> <p>Dado que el visitante accede a la landing page,</p> <p>Cuando hace click en "¿Qué es Bibflip?",</p> <p>Entonces el sistema le muestra la sección donde se describe claramente el objetivo de la solución.</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
US002	Ver sección "Características clave"	Como visitante, quiero revisar una lista de características principales de Bibflip para evaluar si cubre mis necesidades como usuario final o administrador.	EPIC-LANDING-001	Visitante landing page	<p>Escenario 1:</p> <p>Dado que el usuario navega la landing, Cuando cuando hace scroll a la sección "Características clave", Entonces el sistema le muestra un resumen de las funcionalidades más importantes del sistema.</p> <p>Escenario 2:</p> <p>Dado que el visitante está en la landing, Cuando da click sobre el botón "Características clave", Entonces el sistema le muestra un resumen de las funcionalidades más importantes del sistema.</p>
US003	Ver la galería virtual	Como visitante, quiero ver una galería virtual en la landing para visualizar cómo luce Bibflip, su interfaz y los sensores en acción.	EPIC-LANDING-001	Visitante landing page	<p>Escenario 1:</p> <p>Dado que el usuario navega la landing, Cuando cuando hace scroll a la sección "Galería Virtual" Entonces visualiza el contenido visual ilustrativo del sistema en uso</p> <p>Escenario 2:</p> <p>Dado que el usuario navega la landing, Cuando cuando hace click a la sección "Galería Virtual" Entonces visualiza el contenido visual ilustrativo del sistema en uso</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
US004	Usar el formulario de contacto	Como visitante interesado, quiero tener acceso a un formulario o medio de contacto en la landing para comunicarme con el equipo de Bibflip.	EPIC-LANDING-001	Visitante landing page	<p>Escenario 1: Dado que el visitante accede a "Contáctanos" Cuando completa el formulario con sus datos Entonces el sistema debe notificar al equipo de Bibflip</p> <p>Escenario 2: Dado que el visitante intenta enviar el formulario, Cuando deja campos obligatorios vacíos, Entonces el sistema debe mostrar mensajes de validación para completar la información.</p>
US005	Ver menú digital de una sede	Como estudiante, quiero visualizar el reglamento o servicios del centro de estudio para conocer la oferta disponible antes de asistir.	EPIC-CLIENT-002	Estudiante	<p>Escenario 1: Dado que el estudiante accede a la plataforma, Cuando le da click a la sección "Servicios", Entonces el sistema debe mostrar la información completa de la sede.</p> <p>Escenario 2: Dado que el estudiante visualiza la lista, Cuando selecciona una categoría como "Cubículos" o "Salas grupales", Entonces el sistema filtra los servicios mostrados según esa categoría.</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
US006	Visualizar reserva activa	Como estudiante, quiero poder visualizar mi reserva activa desde la aplicación, para confirmar los detalles de mi cubículo y el horario reservado antes de llegar al centro de estudio.	EPIC-CLIENT-002	Estudiante	<p>Escenario 1:</p> <p>Dado que el estudiante ha iniciado sesión en la plataforma</p> <p>Y tiene al menos una reserva activa,</p> <p>Cuando accede a la sección principal,</p> <p>Entonces el sistema le muestra la información de la reserva actual incluyendo horario, sede y número de cubículo.</p> <p>Escenario 2:</p> <p>Dado que el estudiante visualiza su reserva,</p> <p>Cuando esta ya ha expirado,</p> <p>Entonces el sistema la remueve de la sección activa.</p>
US007	Ver disponibilidad de cubículos en tiempo real	Como estudiante, quiero conocer la disponibilidad de cubículos para decidir a qué sede acudir.	EPIC-CLIENT-002	Estudiante	<p>Escenario 1:</p> <p>Dado que el estudiante accede a la plataforma</p> <p> Cuando le da click a la sección "Explorar Sedes"</p> <p>Entonces el sistema debe mostrarle la información de las sedes con sus respectivos aforos disponibles</p> <p>Escenario 2:</p> <p>Dado que el estudiante visualiza el estado de aforo de las sedes</p> <p>Cuando hay una actualización desde los sensores</p> <p>Entonces el sistema debe reflejar el nuevo estado en menos de 5 segundos</p> <p>Escenario 3:</p> <p>Dado que todos los cubículos están ocupados en una sede</p> <p>Cuando el estudiante visualiza la disponibilidad</p> <p>Entonces el sistema debe informar que no hay cubículos disponibles actualmente</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
US008	Reservar un cubículo	Como estudiante, quiero reservar un cubículo para asegurarme un lugar al llegar al centro de estudio.	EPIC-CLIENT-002	Estudiante	<p>Escenario 1:</p> <p>Dado que el estudiante selecciona una sede en la sección "Explorar Sedes"</p> <p>Cuando el estudiante selecciona la hora aproximada de estadía</p> <p>Y selecciona un cubículo disponible</p> <p>Entonces el sistema debe permitir realizar la reserva</p>
US009	Visualizar más sedes en un mapa	Como estudiante, quiero poder ver más sedes en el mapa para poder identificar la más cercana.	EPIC-CLIENT-002	Estudiante	<p>Escenario 1:</p> <p>Dado que el estudiante quiere ver más sedes</p> <p>Cuando le da click a la sección "Sedes"</p> <p>Entonces el sistema debe mostrar todas las sedes del centro de estudio</p> <p>Escenario 2:</p> <p>Dado que el estudiante se encuentra en la sección "Sedes"</p> <p>Y puede ver todas las sedes</p> <p>Cuando le da click al botón de "Ver en mapa"</p> <p>Entonces el sistema muestra un pop up con la dirección en un mapa interactivo</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
US010	Registro de cuenta de estudiante	Como visitante, quiero poder registrarme como estudiante en el sistema Bibflip para poder acceder a las funcionalidades de reserva y visualización de cubículos.	EPIC-CLIENT-002	Visitante	<p>Escenario 1:</p> <p>Dado que un visitante accede a la plataforma</p> <p>Cuando completa el formulario de registro con nombre y contraseña</p> <p>Y acepta los términos y condiciones</p> <p>Entonces el sistema debe crear una cuenta nueva</p> <p>Y permitir el acceso a las funcionalidades de estudiante</p>
US011	Inicio de sesión de cuenta de estudiante	Como estudiante registrado, quiero poder iniciar sesión en el sistema para acceder a mis reservas y funcionalidades personalizadas.	EPIC-CLIENT-002	Estudiante	<p>Escenario 2:</p> <p>Dado que un visitante intenta registrarse con un nombre de usuario ya usado,</p> <p>Cuando envía el formulario,</p> <p>Entonces el sistema debe impedir el registro y mostrar un mensaje de error indicando que el correo ya está registrado.</p> <p>Escenario 1:</p> <p>Dado que un estudiante registrado accede a la pantalla de login</p> <p>Cuando ingresa sus credenciales correctas (email y contraseña)</p> <p>Entonces el sistema debe autenticarlo</p> <p>Y redirigirlo al dashboard principal</p> <p>Escenario 2:</p> <p>Dado que un usuario ingresa credenciales incorrectas</p> <p>Cuando intenta iniciar sesión</p> <p>Entonces el sistema debe mostrar un mensaje de error</p> <p>Y permitir reintentar</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
US012	Agregar nuevos cubículos	Como administrador de sede, quiero agregar cubículos al sistema para gestionar mejor la ocupación.	EPIC-ADMIN-003	Administrador	<p>Escenario 1: Dado que el admin accede al panel de su sede Cuando le da click a la sección "Gestión de Cubículos" Y posteriormente a la opción "Agregar un nuevo cubículo" Entonces el sistema le brinda un formulario simple de registro de cubículo.</p> <p>Escenario 2: Dado que el admin ingresa los datos correctos del cubículo como su identificador y capacidad Y le da click al botón "Guardar" Entonces el sistema guarda con su identificador y capacidad el cubículo registrado</p>
US013	Visualizar reservas realizadas	Como administrador, quiero ver las reservas registradas en mi sede para planificar el servicio.	EPIC-ADMIN-003	Administrador	<p>Escenario 1: Dado que el administrador accede a su sede Cuando le da clic a la sección "Visualizar Reservas" Entonces el sistema debe mostrar las reservas con usuario, hora y cubículo</p> <p>Escenario 2: Dado que el administrador ha filtrado por persona o cubículo Cuando aplica el filtro Entonces el sistema debe mostrar solo las reservas que coinciden con los criterios seleccionados</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
US014	Ver la sede asignada	Como administrador, quiero confirmar qué sede tengo asignada para asegurarme de gestionar la correcta.	EPIC-ADMIN-003	Administrador	<p>Escenario 1:</p> <p>Dado que el administrador inicia sesión</p> <p>Cuando accede a su información</p> <p>Entonces el sistema debe mostrar la sede asociada a su cuenta</p>
US015	Eliminar cubículos	Como administrador, quiero poder eliminar cubículos que ya no están en uso en mi sede, para mantener el sistema actualizado y evitar asignaciones incorrectas.	EPIC-ADMIN-003	Administrador	<p>Escenario 1:</p> <p>Dado que el administrador tiene asignada una sede</p> <p>Cuando intenta acceder a otra sede que no le corresponde</p> <p>Entonces el sistema le impide el acceso</p> <p>Escenario 2:</p> <p>Dado que el administrador ha iniciado sesión</p> <p>Y se encuentra en la sección "Ver Cubículos"</p> <p>Cuando selecciona un cubículo existente</p> <p>Y confirma su eliminación</p> <p>Entonces el cubículo es eliminado del sistema</p> <p>Y deja de aparecer en la lista</p> <p>Escenario 2:</p> <p>Dado que el cubículo tiene una reserva activa</p> <p>Cuando el administrador intenta eliminarlo</p> <p>Entonces el sistema debe impedir la eliminación</p> <p>Y mostrar un mensaje indicando que no puede eliminar cubículos con reservas activas</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
US016	Inicio de sesión de administrador	Como administrador, quiero poder iniciar sesión en el sistema con mis credenciales específicas para acceder al panel de administración de mi sede.	EPIC-ADMIN-003	Administrador	<p>Escenario 1:</p> <p>Dado que un administrador accede a la pantalla de login Cuando ingresa sus credenciales válidas Entonces el sistema debe autenticarlo Y mostrar el panel de administración con las funcionalidades asociadas a su sede</p> <p>Escenario 2:</p> <p>Dado que un administrador ingresa credenciales incorrectas Cuando intenta acceder Entonces el sistema debe mostrar un mensaje de error Y registrar el intento fallido por seguridad</p>
US017	Registro de cuenta de administrador	Como superadministrador, quiero poder registrar administradores como administradores de una sede en el sistema Bibflip para que puedan acceder a las funcionalidades de ver reservas y la gestión de cubículos.	EPIC-SUPERADMIN-004	Superadmin	<p>Escenario 1:</p> <p>Dado que un superadmin se encuentra en su cuenta Cuando se dirija a la sección de "Administradores" Y da click a agregar administrador Entonces puede registrar un administrador y asignarle una sede</p> <p>Escenario 2:</p> <p>Dado que un superadmin ha registrado a un administrador Cuando se dirige a la sección de "Administradores" Entonces puede visualizar los administradores que ha registrado</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
US018	Acceder a todos los módulos del sistema	Como superadmin, quiero tener acceso a todas las funcionalidades del sistema para realizar pruebas, crear sedes y admins.	EPIC-SUPERADMIN-004	Superadmin	<p>Escenario 1:</p> <p>Dado que el superadmin accede al sistema</p> <p>Cuando desea visualizar sedes o usuarios</p> <p>Entonces el sistema debe permitirle ver toda la información</p> <p>Escenario 2:</p> <p>Dado que el superadmin crea un nuevo admin o sede</p> <p>Cuando registra los datos</p> <p>Entonces el sistema debe almacenarlos y mostrar confirmación</p>

Technical Stories

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
TS001	Implementar Endpoint para Obtener Cubículos	Como Developer, quiero implementar un endpoint de API que permita a las aplicaciones frontend obtener la lista de cubículos disponibles en una biblioteca específica, para que los usuarios puedan ver el estado de ocupación.	EPIC-BOOKING-TECH	Developer	<p>Escenario 1:</p> <p>Obtener cubículos disponibles exitosamente</p> <p>Dado que existe una biblioteca con cubículos registrados</p> <p>Cuando un cliente solicita la lista de cubículos para esa biblioteca</p> <p>Entonces el sistema responde con un código 200 OK y una lista de cubículos con su estado de ocupación.</p> <p>Escenario 2:</p> <p>Biblioteca no encontrada</p> <p>Dado que se solicita una biblioteca que no existe</p> <p>Cuando el sistema intenta obtener los cubículos</p> <p>Entonces el sistema responde con un código 404 Not Found.</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
TS002	Procesar Datos de Sensor de Ocupación	Como Developer, quiero implementar la lógica de backend para recibir y procesar los datos de ocupación enviados por los sensores IoT, para que el sistema actualice el estado de los cubículos en tiempo real.	EPIC-IOT-PROCESSING	Developer	<p>Escenario 1: Datos de ocupación recibidos y procesados</p> <p>Dado que el Edge API recibe datos de ocupación de un sensor</p> <p>Cuando el backend procesa estos datos</p> <p>Entonces el estado del cubículo correspondiente se actualiza en la base de datos.</p>
TS003	Implementar Lógica de Reserva de Cubículo API	Como Developer, quiero implementar la lógica de backend para gestionar el proceso de reserva de cubículos a través de la API, asegurando la disponibilidad y registrando la reserva correctamente.	EPIC-BOOKING-TECH	Developer	<p>Escenario 1: Reserva de cubículo exitosa</p> <p>Dado que un usuario solicita reservar un cubículo disponible</p> <p>Cuando la lógica de reserva valida la disponibilidad y registra la reserva</p> <p>Entonces la reserva se crea en la base de datos y el estado del cubículo se actualiza a "Reservado".</p> <p>Escenario 2: Cubículo no disponible</p> <p>Dado que un usuario intenta reservar un cubículo ya ocupado o reservado</p> <p>Cuando la lógica de reserva valida la disponibilidad</p> <p>Entonces el sistema rechaza la reserva y responde con un mensaje de error de disponibilidad.</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
TS004	Implementar Firmware para Dispositivo IoT	Como Developer, quiero desarrollar el firmware para el dispositivo IoT que detecta la ocupación de cubículos, para transmitir datos al Edge API.	EPIC-IOT-DEVICE	Developer	<p>Escenario 1: Lectura de sensor correcta</p> <p>Dado que el dispositivo está conectado correctamente</p> <p>Cuando el sensor detecta presencia</p> <p>Entonces el firmware genera un paquete de datos con el estado de ocupación y lo transmite al Edge API.</p> <p>Escenario 2: Fallo de lectura del sensor</p> <p>Dado un fallo de hardware</p> <p>Cuando se intenta leer el sensor</p> <p>Entonces el firmware registra el error sin transmitir datos.</p>
TS005	Integrar Firmware con Plataforma Embebida	Como Developer, quiero integrar el firmware en la plataforma embebida para que pueda iniciar, procesar datos y comunicarse vía red con el Edge.	EPIC-IOT-DEVICE	Developer	<p>Escenario 1: Inicio correcto del sistema</p> <p>Dado que el sistema embebido arranca con configuración válida</p> <p>Cuando se enciende el dispositivo</p> <p>Entonces el sistema inicializa sensores y red sin errores.</p> <p>Escenario 2: Error de red</p> <p>Dado que la conexión de red falla</p> <p>Cuando el sistema intenta establecer conexión</p> <p>Entonces se lanza un mensaje de error y se reinicia el intento.</p>

Story ID	Título	Descripción	Epic ID	Usuario	Criterios de aceptación
TS006	Implementar Recepción de Datos en Edge API	Como Developer, quiero implementar la lógica en el Edge API que reciba los datos enviados por los sensores IoT, para enviarlos al backend.	EPIC-IOT-PROCESSING	Developer	<p>Escenario 1: Datos recibidos correctamente</p> <p>Dado que un dispositivo IoT transmite datos válidos</p> <p>Cuando el Edge API recibe el paquete</p> <p>Entonces lo reenvía al backend y responde con un 200 OK.</p> <p>Escenario 2: Datos con formato incorrecto</p> <p>Dado que se recibe un paquete con formato inválido</p> <p>Cuando se intenta procesarlo</p> <p>Entonces el sistema registra el error y descarta el paquete.</p>
TS007	Simular Sensores en Wokwi	Como Developer, quiero simular el comportamiento del dispositivo IoT en Wokwi, para validar la lectura del sensor y la comunicación con el Edge API.	EPIC-IOT-DEVICE	Developer	<p>Escenario 1: Simulación de detección de ocupación</p> <p>Dado que se ejecuta el circuito simulado en Wokwi</p> <p>Cuando un peso es detectado por el sensor virtual</p> <p>Entonces el sistema envía correctamente el estado al Edge API simulado.</p> <p>Escenario 2: Sensor inactivo</p> <p>Dado que el sensor no detecta peso</p> <p>Cuando se revisa el estado</p> <p>Entonces no se envía ninguna actualización.</p>

3.3. Impact Mapping

Segmento 1: Personal de biblioteca universitaria

Segmento Objetivo: Personal Bibliotecario/Administradores

Optimizar la gestión operativa y administrativa de espacios de estudio en un 50%



Administrador de Sede

Mejorar planificación de servicios
Como administrador, quiero visualizar todas las reservas para planificar mejor los servicios y recursos.

US013

Optimizar gestión de inventario de cubículos
Como administrador, quiero agregar y eliminar cubículos según las necesidades operativas.

US012 **US015**

Mantener control sobre sede asignada
Como administrador, quiero confirmar mi sede asignada para gestionar correctamente mi área de responsabilidad.

US014

Facilitar acceso administrativo seguro
Como administrador, quiero un sistema de login específico para acceder a funcionalidades administrativas.

US016

Panel de gestión de reservas
Vista completa de reservas con filtros por usuario, fecha y cubículo

Módulo de gestión de cubículos
CRUD completo para agregar, editar y eliminar cubículos de la sede

Dashboard administrativo
Panel principal con información de sede asignada y métricas clave

Sistema de autenticación administrativa
Login específico para administradores con permisos diferenciados

Módulo de registro de administradores
Funcionalidad para que superadmin registre nuevos administradores por sede

Activar Windows
[Ver Configuración para activar Windows](#)

Segmento 2: Estudiantes universitarios

Impact Mappings - Bibflip

Incrementar la satisfacción y eficiencia de los estudiantes al gestionar espacios de estudio en un 40%



Estudiante Universitario

Reducir tiempo perdido buscando espacios disponibles
Como estudiante, quiero conocer la disponibilidad de cubículos en tiempo real para no perder tiempo visitando sedes llenas.

US007

Garantizar disponibilidad de espacio al llegar
Como estudiante, quiero reservar un cubículo con anticipación para asegurar mi espacio de estudio.

US008

Facilitar la selección de sede más conveniente
Como estudiante, quiero ver todas las sedes en un mapa para elegir la más cercana o conveniente.

US009

Mejorar control y seguimiento de reservas
Como estudiante, quiero visualizar mi reserva activa para confirmar detalles antes de llegar.

US006

Dashboard en tiempo real
Visualización de disponibilidad de cubículos con actualización cada 5 segundos

Sistema de reservas
Funcionalidad para reservar cubículos con selección de horario y duración

Mapa interactivo de sedes
Visualización geográfica de todas las sedes con información de disponibilidad

Panel de reservas activas
Sección para ver y gestionar reservas actuales del estudiante

Sistema de autenticación
Registro e inicio de sesión para estudiantes con validación de credenciales

Activar Windows

3.4. Product Backlog

#	Story ID	Título	Descripción	Story Points
1	US008	Reservar un cubículo	Como usuario, quiero reservar un cubículo para asegurarme un espacio al llegar a la biblioteca.	13
2	US018	Acceder a todos los módulos del sistema	Como superadmin, quiero tener acceso a todas las funcionalidades del sistema para realizar pruebas, crear bibliotecas y admins.	13
3	US007	Ver disponibilidad de cubículos en tiempo real	Como usuario, quiero conocer la disponibilidad de cubículos para decidir qué biblioteca visitar.	8

#	Story ID	Título	Descripción	Story Points
4	US012	Agregar nuevos cubículos	Como administrador de biblioteca, quiero agregar cubículos al sistema para gestionar mejor la ocupación.	5
5	US009	Visualizar más bibliotecas en un mapa	Como usuario, quiero poder ver más bibliotecas en el mapa para poder identificar la más cercana.	5
6	US006	Visualizar reserva activa	Como usuario, quiero poder visualizar mi reserva activa desde la aplicación, para confirmar los detalles de mi cubículo y el horario reservado.	5
7	US013	Visualizar reservas realizadas	Como administrador, quiero ver las reservas registradas en mi biblioteca para planificar el servicio.	5
8	US017	Registro de cuenta de administrador	Como superadmin, quiero poder registrar administradores como responsables de una biblioteca en el sistema Bib Flip para gestionar reservas y cubículos.	5
9	US005	Ver catálogo digital de la biblioteca	Como usuario, quiero visualizar el catálogo de la biblioteca para conocer los recursos disponibles antes de asistir.	3
10	US004	Usar el formulario de contacto	Como visitante interesado, quiero tener acceso a un formulario o medio de contacto en la landing para comunicarme con el equipo de Bib Flip.	3
11	US014	Ver la biblioteca asignada	Como administrador, quiero confirmar qué biblioteca tengo asignada para asegurarme de gestionarla correctamente.	3
12	US015	Eliminar cubículos	Como administrador, quiero poder eliminar cubículos que ya no están en uso en mi biblioteca, para mantener el sistema actualizado y evitar asignaciones incorrectas.	3
13	US010	Registro de cuenta de usuario	Como visitante, quiero poder registrarme como usuario en el sistema Bib Flip para poder acceder a las funcionalidades de reserva y visualización de cubículos.	3
14	US011	Inicio de sesión de usuario	Como usuario registrado, quiero poder iniciar sesión en el sistema para acceder a mis reservas y funcionalidades personalizadas.	3
15	US016	Inicio de sesión de administrador	Como administrador, quiero poder iniciar sesión en el sistema con mis credenciales específicas para acceder al panel de administración de mi biblioteca.	3
16	US001	Ver sección "¿Qué es Bib Flip?"	Como visitante de la página, quiero entender qué es Bib Flip desde la landing page para conocer su propósito y cómo puede ayudarme como usuario o administrador de biblioteca.	2
17	US002	Ver sección "Características clave"	Como visitante, quiero revisar una lista de características principales de Bib Flip para evaluar si cubre mis necesidades como usuario o administrador.	2
18	US003	Ver la galería virtual	Como visitante, quiero ver una galería virtual en la landing para visualizar cómo luce Bib Flip, su interfaz y los sensores en acción.	2

Capítulo IV: Solution Software Design

4.1. Strategic-Level Domain-Driven Design

4.1.1. Event Storming

Con el objetivo de comprender en profundidad el dominio del sistema, se llevó a cabo una sesión de Event Storming de aproximadamente 1 hora. Esto permitió que nosotros como equipo organizáramos nuestras ideas y pensamientos sobre sistema desde múltiples perspectivas: negocio, usuario final, administración y experiencia. A través de esta dinámica, identificamos eventos clave, comandos, usuarios y agregados que nos ayudaron a esbozar una primera visión integral del sistema.

Pasos del Event Storming

Paso 1: Collect domain events

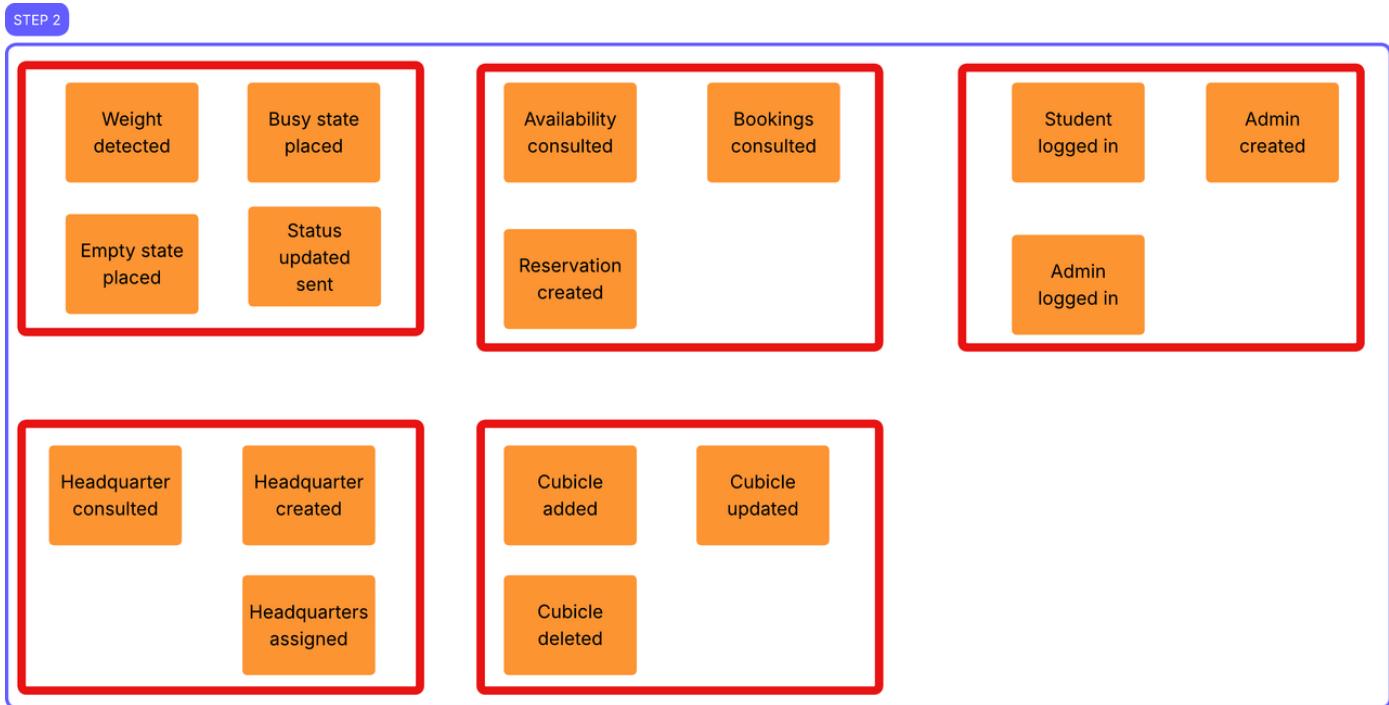
En el primer paso se identifican los eventos principales que ocurren en el sistema, plasmándolos de manera aislada y ordenada.

Esta fase ayuda a comprender la diversidad de acciones e hitos, sin preocuparse aún por su secuencia o agrupación, lo que permite descubrir todo el vocabulario de dominio relevante.



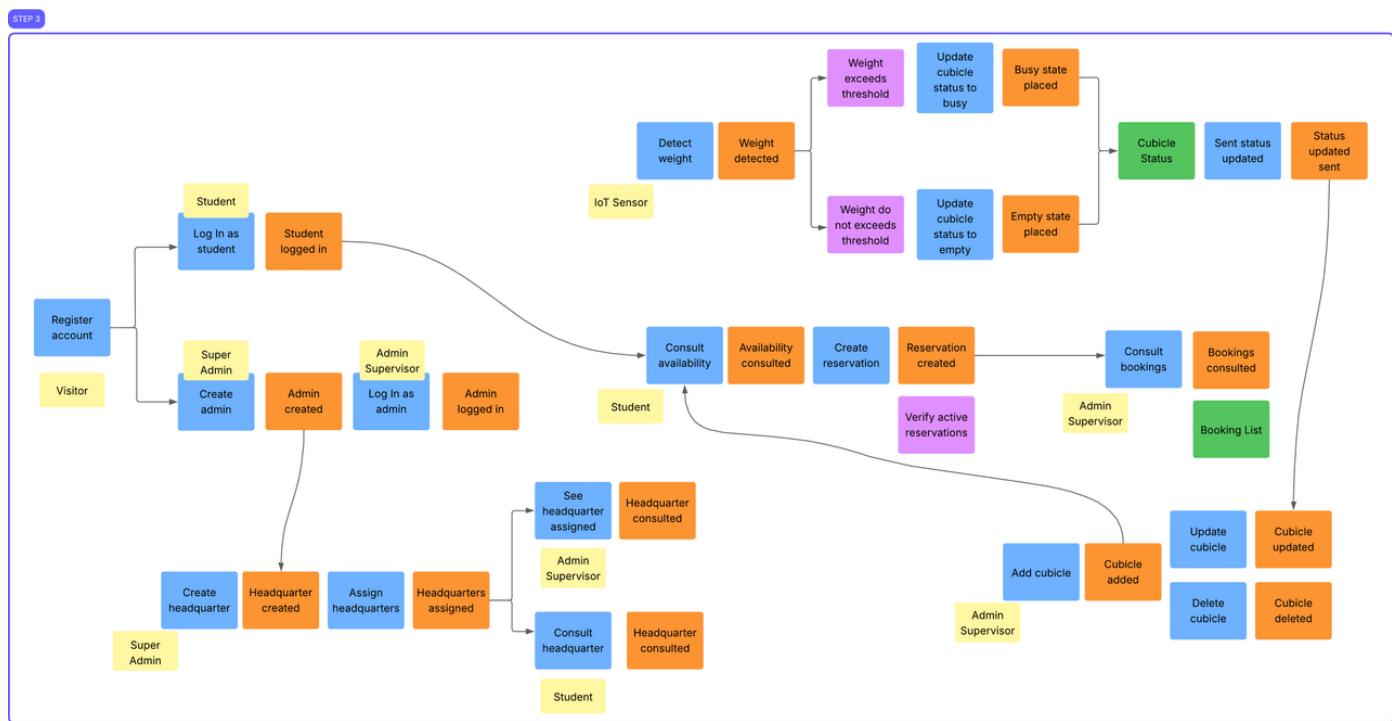
Paso 2: Refine domain events

El segundo paso consiste en clasificar los eventos detectados y organizarlos en bloques según el tipo de actividad o área funcional. Este agrupamiento facilita visualizar patrones, dependencias y la relación entre procesos, preparando el terreno para explorar el flujo narrativo y el impacto en el negocio.



Paso 3: Track causes

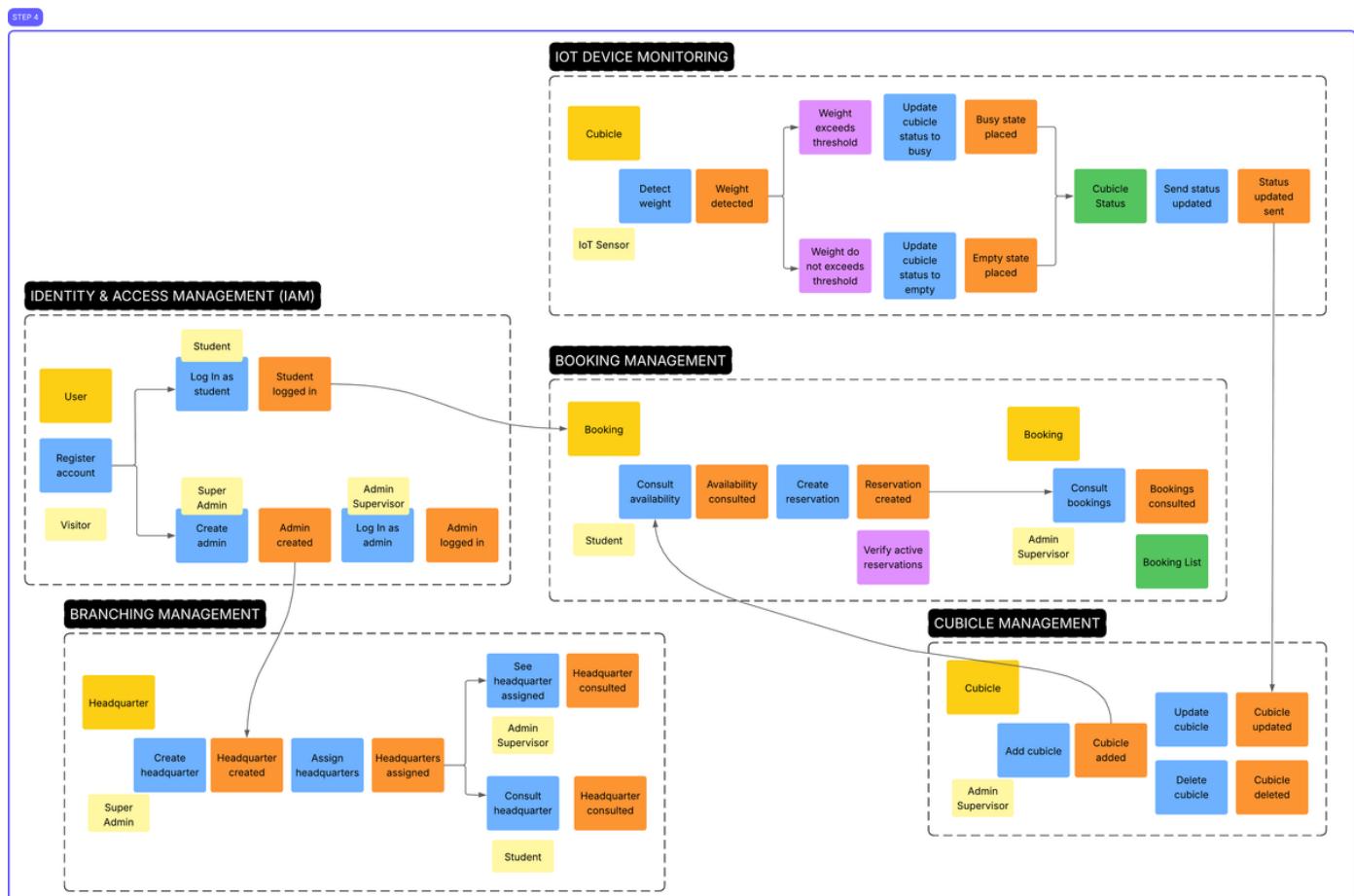
Durante el tercer paso se introducen los actores, comandos y decisiones que intervienen en los flujos narrados. Aquí se conectan los eventos a través de relaciones y se representan secuencias lógicas, dando claridad a los caminos que recorren los datos y las acciones desde el inicio hasta su resultado final.



Paso 4: Find aggregates & re-sort them

Finalmente, en el cuarto paso se delimitan los subdominios y contextos dentro del sistema, agrupando elementos relacionados bajo áreas temáticas o módulos funcionales.

Este mapeo ayuda a definir límites claros, responsabilidades y dependencias estructurales para guiar el diseño estratégico y la futura arquitectura del sistema.



Se vieron los siguientes puntos en la reunión:

- Exploración del dominio general

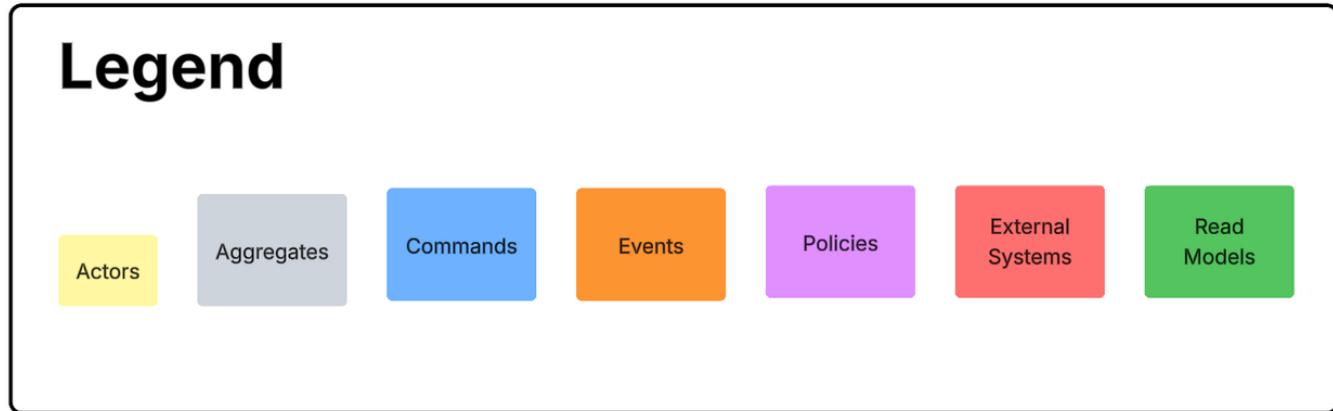
Se partió desde la experiencia del visitante en la landing page, avanzando por el flujo de registro e inicio de sesión como estudiante o administrador, hasta la reserva de cubículos, visualización en tiempo real y gestión desde el panel administrativo.

- Identificación de eventos y comandos clave

Se colocaron notas naranjas para eventos, y se complementaron con comandos en azul. El equipo seguía de las User Stories previamente realizadas, lo que aseguró la coherencia y el flujo de la solución.

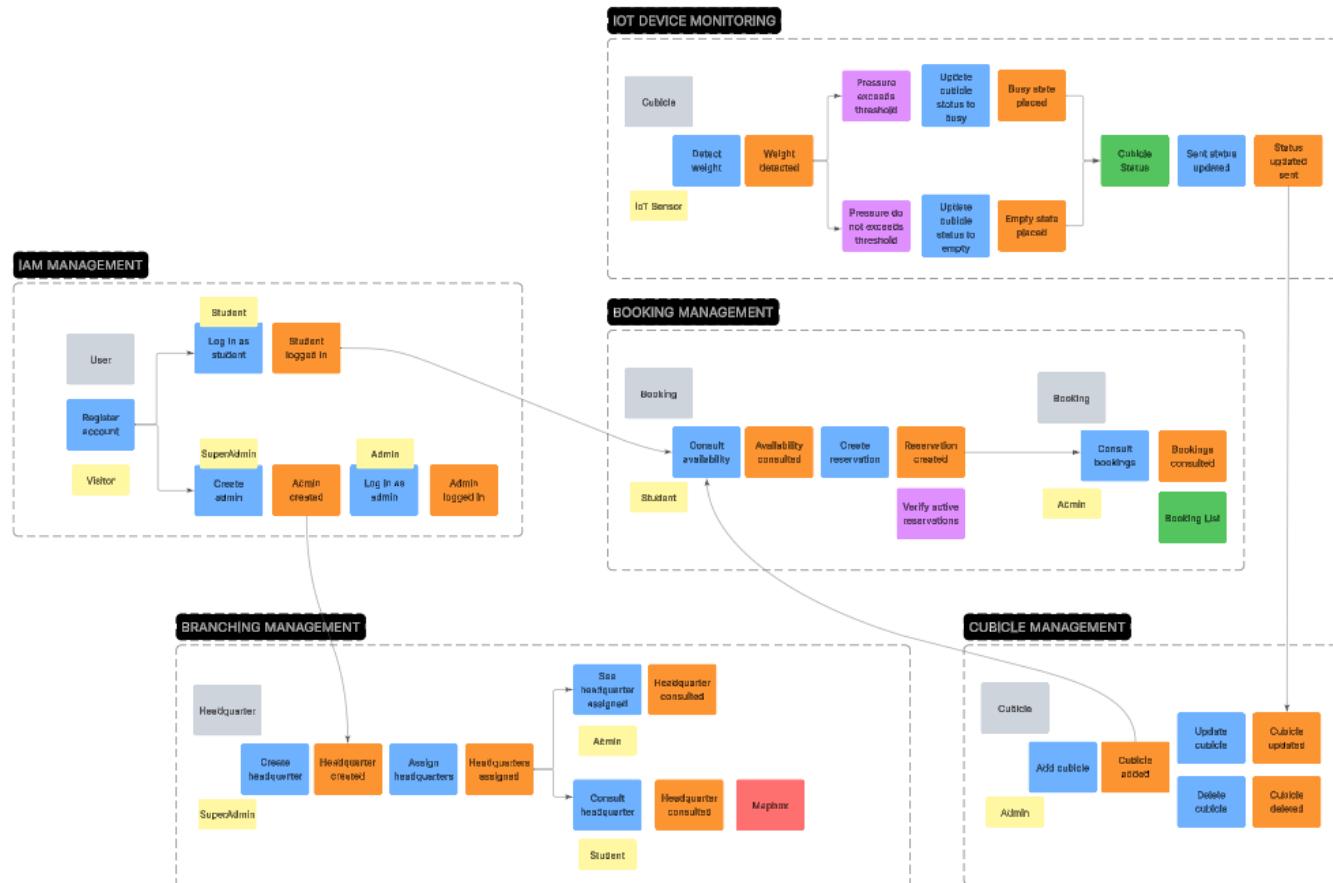
- Asignación de roles y responsables

Se diferenciaron los actores como: estudiante, administrador, superadmin, visitante, para asociar claramente qué parte del sistema controlan o en qué puntos interactúan. Al dividirlo de esta manera es más fácil detectar posibles conflictos o áreas de mejora en la experiencia.



4.1.1.1 Candidate Context Discovery

Identificar contextos candidatos es un paso clave para gestionar la complejidad en el desarrollo de sistemas. Se trata de un análisis minucioso que busca entender los elementos centrales del sistema y sus interconexiones. A partir de ahí, se procede a agrupar estos elementos en 'contextos delimitados' lógicos y coherentes. Esta separación no solo facilita el diseño y la implementación, sino que también tiene como meta principal potenciar la escalabilidad, el desempeño y la mantenibilidad del sistema resultante.



Para una mejor visualización del conjunto de diagramas, se proporciona el siguiente enlace: https://lucid.app/lucidchart/e553dbed-f6de-4913-994b-ed85dcd83d41/edit?viewport_loc=-3331%2C-1097%2C8294%2C3700%2CVlUR_3p1AnhQ&invitationId=inv_a52a3e8b-f40f-4bae-ac7e-ddae54aa97c9

4.1.1.2 Domain Message Flows Modeling

Para dar inicio con el diseño de nuestro software, es imperativo el uso del Modelado de Flujos de Mensajes de Dominio, un método que ilustra la transferencia de información entre componentes mediante mensajes. Este proceso se centra en especificar los mensajes enviados y recibidos por los diferentes actores del sistema y en descifrar sus relaciones. El uso de esta metodología aporta claridad para entender y representar las vías de información del sistema, permitiendo detectar problemas potenciales más fácilmente y optimizar la estructura del diseño. Por ello, a continuación se presenta el siguiente modelado:

Scenario: Registro de usuarios

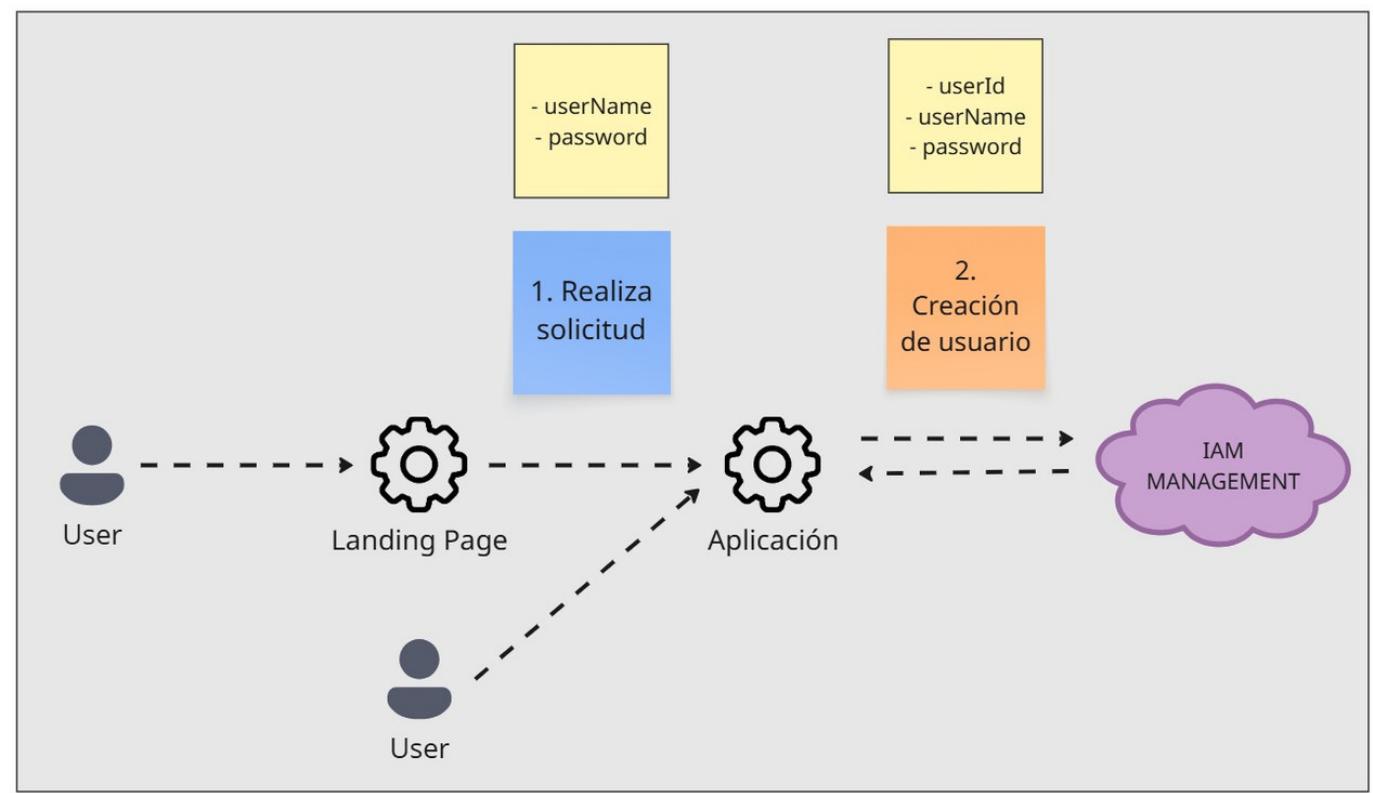
Actores:

- Identificación de los usuarios dentro de nuestra solución teniendo dos actores que se pueden identificar como **Personal de bibliotecas universitarias** o **Estudiantes universitarios**.

Explicación del proceso y definición de los eventos:

1. El usuario llega a la página de inicio de sesión de la aplicación y a través de las opciones se realiza la gestión de roles para cada tipo de usuario.
2. El usuario realiza el registro en el sistema como identificándose como uno de los segmentos.
3. Registra sus datos (username, password) en los campos correspondientes para la creación de su cuenta.
4. Se realizan las validaciones correspondientes al registro de su contraseña.
5. Se obtiene el registro del usuario en la aplicación.
6. El usuario debe realizar la autenticación de sus datos la primera vez que ingresa a la aplicación.

Visualización del flujo:



Escenario: Autenticación y acceso a módulos

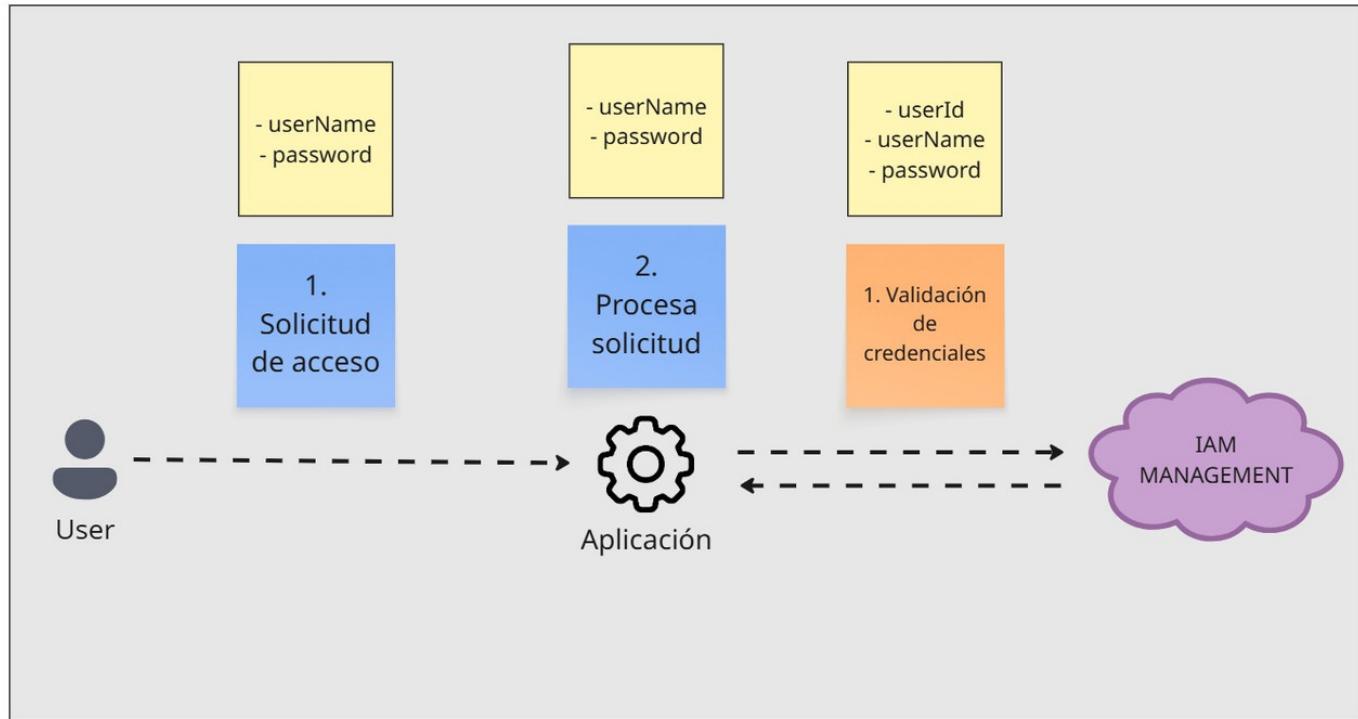
Actores:

- Principal: Usuario (Estudiante o Personal Administrativo)
- Principal: Personal de bibliotecas con permisos superiores

Explicación del proceso y definición de los eventos:

1. El usuario o administrador solicita acceso en la plataforma Bib Flip.
2. El sistema IAM valida credenciales.
3. Se determina el rol y se conceden permisos según perfil (usuario, administrador, superadmin).
4. Se notifica al usuario sobre el resultado (acceso concedido o denegado).
5. El usuario accede a las funcionalidades asignadas.

Visualización del flujo:



Escenario: Reserva de cubículos

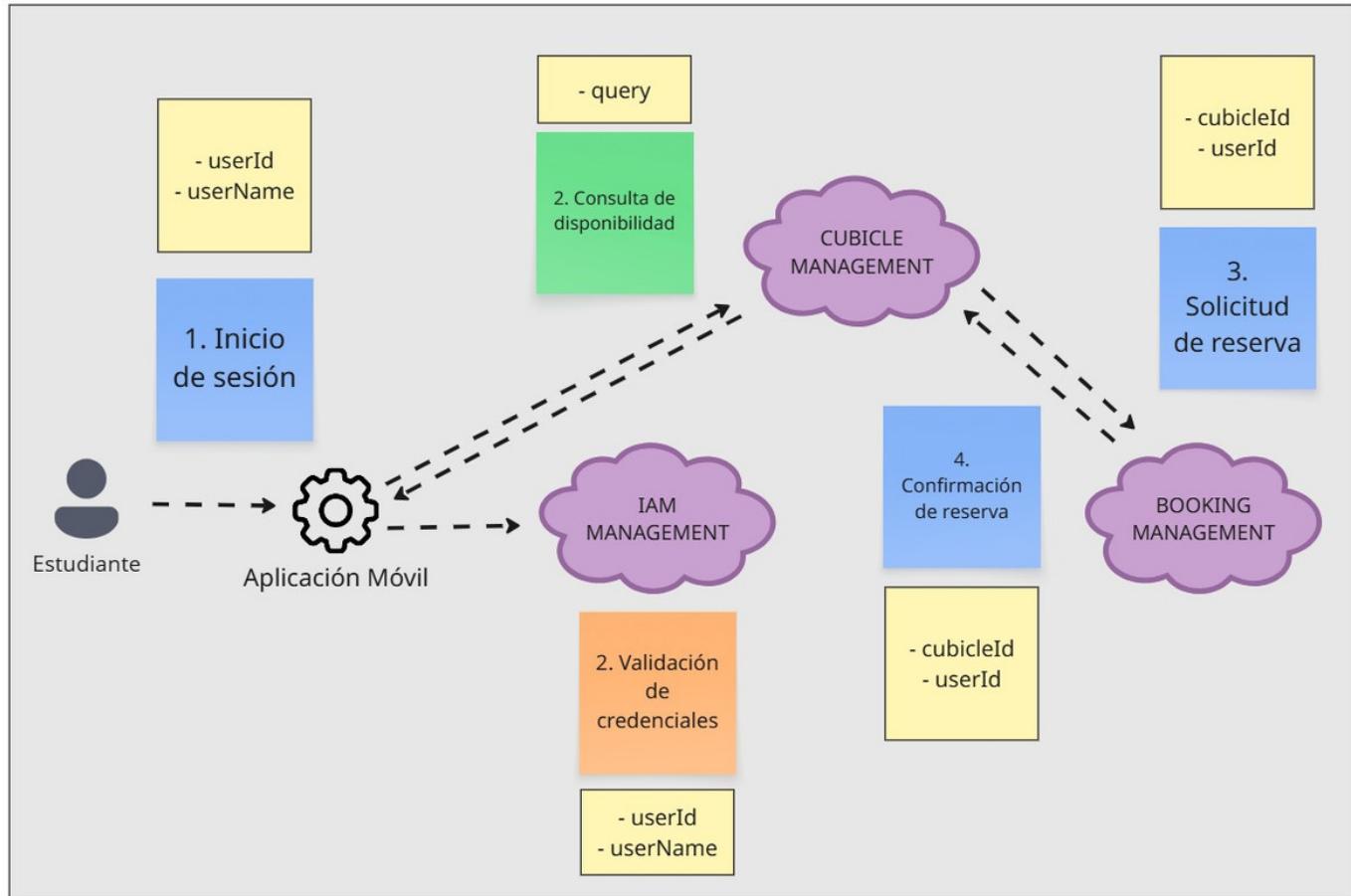
Actores:

- Principal: Usuario Estudiante universitario

Explicación del proceso y definición de los eventos:

1. Usuario inicia sesión y solicita ver disponibilidad en tiempo real.
2. CUBICLE MANAGEMENT consulta y muestra disponibilidad.
3. Usuario selecciona cubículo y crea solicitud de reserva.
4. BOOKING MANAGEMENT valida y registra la reserva.
5. CUBICLE MANAGEMENT actualiza el estado del cubículo.
6. Usuario visualiza reserva activa confirmada.
7. En caso de cancelación o finalización, la reserva se libera.

Visualización del flujo:



Escenario: Gestión administrativa de bibliotecas y cubículos

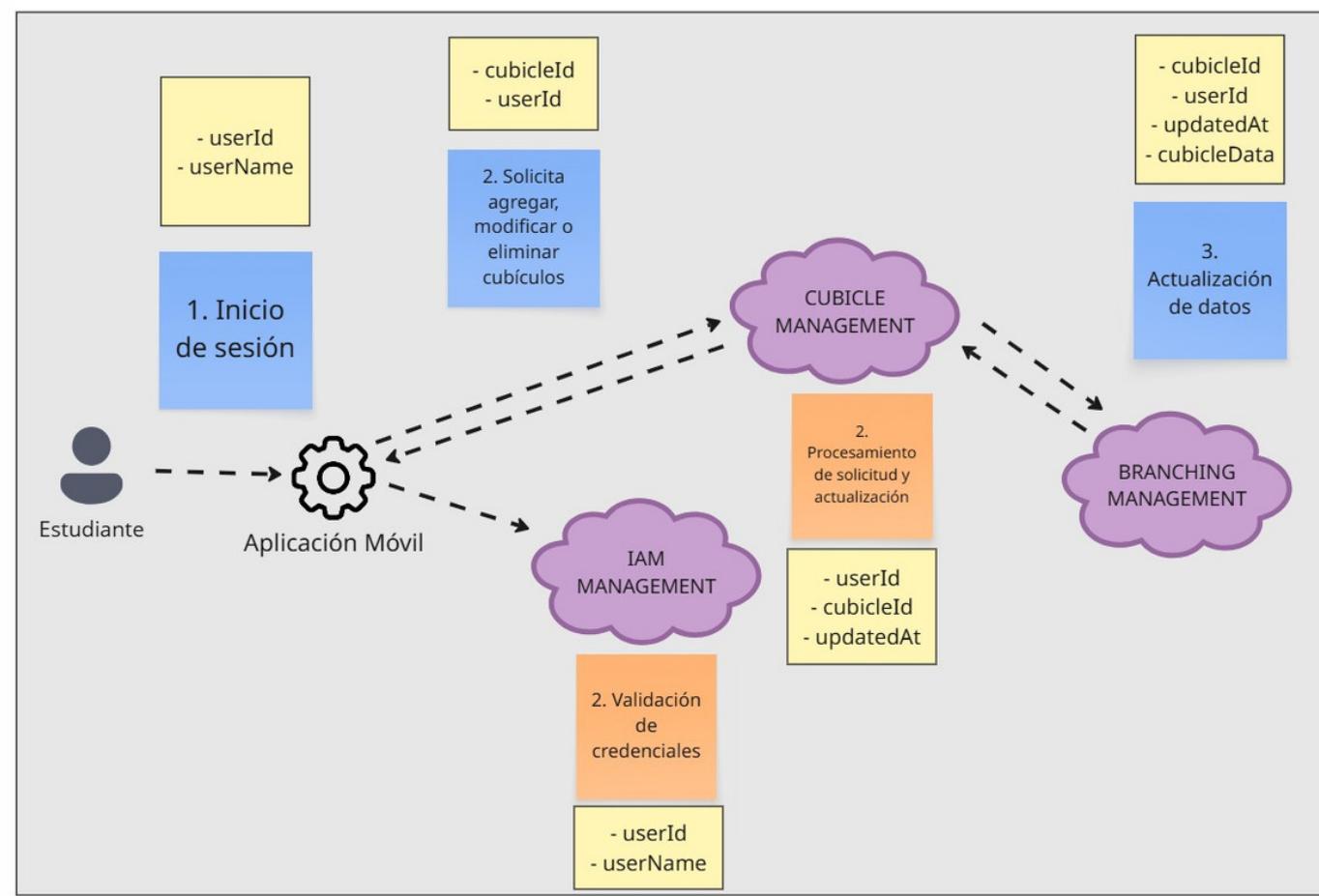
Actores:

- Principal: Administrador Personal de bibliotecas

Explicación del proceso y definición de los eventos:

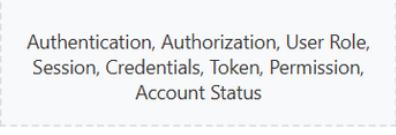
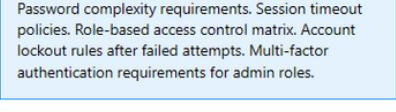
1. Administrador inicia sesión y accede al panel administrativo.
2. Sigue el flujo de datos descrito en el diagrama.
3. CUBICLE MANAGEMENT procesa cambios y actualiza estado.
4. ADMINISTRADOR visualiza biblioteca asignada.
5. BRANCHING MANAGEMENT actualiza la gestión de sucursales o bibliotecas.
6. Cambios se reflejan en la disponibilidad para usuarios.

Visualización del flujo:

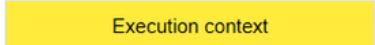
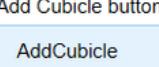
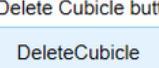
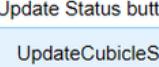
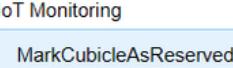
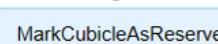
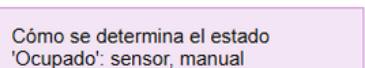
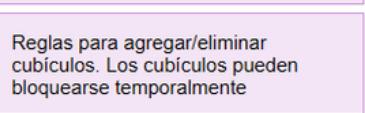
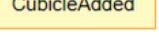


4.1.1.3 Bounded Context Canvases

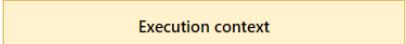
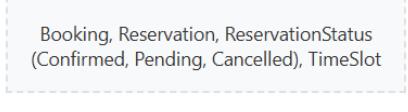
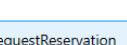
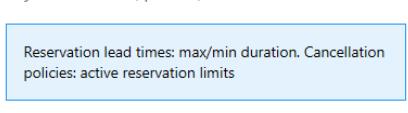
IAM

Name: IAM Management			V4.0 github.com/ddd-crew/bounded-context-canvas
Purpose	Strategic Classification	Domain Roles	
Manages user authentication, authorization, and account lifecycle. Handles user registration, login processes for different user types (Student, Admin, SuperAdmin), and maintains user roles and permissions across the library system.	Domain Generic Business Model Compliance Evolution Product	Role types Gateway context	
Inbound Communication	Ubiquitous Language	Outbound Communication	
Collaborator  User  Visitor  SuperAdmin  Admin	Messages Log in as student Register account Create admin Log in as admin	Collaborator  User  User  User  Branching Management	Messages Student logged in Admin created Admin logged in User authenticated
Ubiquitous Language Context-specific domain terminology		Business Decisions Key business rules, policies, and decisions	
			

Cubicle Management

Name: Cubicle Management			V4.0 github.com/ddd-crew/bounded-context-canvas					
Purpose Gestiona el inventario de cubículos físicos por ubicación y proporciona información sobre su estado y disponibilidad en tiempo real	Strategic Classification <table border="1"> <thead> <tr> <th>Domain</th><th>Business Model</th><th>Evolution</th></tr> </thead> <tbody> <tr> <td>Core</td><td>Compliance</td><td>Commodity</td></tr> </tbody> </table>	Domain	Business Model	Evolution	Core	Compliance	Commodity	Domain Roles Role types 
Domain	Business Model	Evolution						
Core	Compliance	Commodity						
Inbound Communication Collaborator  Add Cubicle button  Delete Cubicle button  Update Status button  IoT Monitoring	Ubiquitous Language Context-specific domain terminology 	Outbound Communication Collaborator  Booking Management	Messages  AddCubicle  DeleteCubicle  UpdateCubicleStatus  MarkCubicleAsReserved  MarkCubicleAsFree					
Business Decisions Key business rules, policies, and decisions  	Collaborator  Booking Management	Messages  CubicleStatusChanged  CubicleAdded  CubicleDeleted						

Booking Management

Name: Booking Management			V4.0 github.com/ddd-crew/bounded-context-canvas					
Purpose Gestionar el ciclo de vida completo de las reservas de cubículos: creación, visualización y cancelación	Strategic Classification <table border="1"> <thead> <tr> <th>Domain</th><th>Business Model</th><th>Evolution</th></tr> </thead> <tbody> <tr> <td>Core</td><td>Compliance</td><td>Product</td></tr> </tbody> </table>	Domain	Business Model	Evolution	Core	Compliance	Product	Domain Roles Role types 
Domain	Business Model	Evolution						
Core	Compliance	Product						
Inbound Communication Collaborator  Confirm Reservation button  Cancel Reservation button  Reservations in Admin Panel	Ubiquitous Language Context-specific domain terminology 	Outbound Communication Collaborator  Cubicle Management	Messages  RequestReservation  CancelReservation  GetAllReservations					
Business Decisions Key business rules, policies, and decisions 	Collaborator  Cubicle Management	Messages  ReservationCreated  ReservationFailed  SuccessfulReservation						

Branching Management

Name: Branching Management			V4.0 github.com/ddd-crew/bounded-context-canvas								
Purpose	Strategic Classification	Domain Roles									
Manages information about library branches, librarian accounts, and possibly global configurations or cross-context reporting.	Strategic Classification <table> <tr> <th>Domain</th> <th>Business Model</th> <th>Evolution</th> </tr> <tr> <td>Core</td> <td>Compliance</td> <td>Commodity</td> </tr> </table>	Domain	Business Model	Evolution	Core	Compliance	Commodity	Domain Roles <table> <tr> <td>Role types</td> </tr> <tr> <td>Execution context</td> </tr> </table>	Role types	Execution context	
Domain	Business Model	Evolution									
Core	Compliance	Commodity									
Role types											
Execution context											
Inbound Communication	Ubiquitous Language	Outbound Communication									
Collaborator  User Messages Consult headquarter  Admin See headquarter assigned  Superadmin Create headquarter  Superadmin Assign headquarter	Ubiquitous Language Context-specific domain terminology <div style="border: 1px dashed gray; padding: 5px; margin-top: 10px;"> Location(Sede), Admin, Superadmin, UserRole, Credentials </div>	Outbound Communication Collaborator  User Admin Messages GetAssignedLocation  GetAllocations  User Admin LocationCreated									
Inbound Communication	Business Decisions										
	Key business rules, policies, and decisions <div style="border: 1px solid blue; padding: 5px; margin-top: 10px;"> Criteria for adding new library branches. Password policies for Librarians. What data is aggregated for reporting? Access control rules for Superadmins </div>										

IoT Device Monitoring

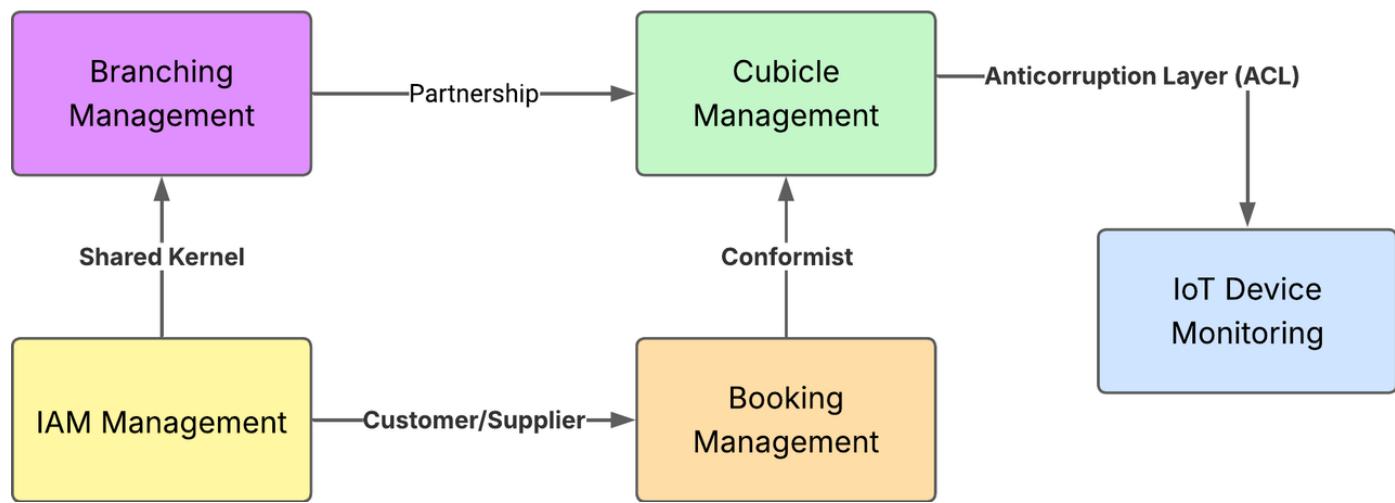
Name: IoT Device Monitoring			V4.0 github.com/ddd-crew/bounded-context-canvas								
Purpose	Strategic Classification	Domain Roles									
To receive data from physical sensors associated with cubicles, interpret this data to determine cubicle occupancy status, and communicate status changes. Primarily supports the real-time availability feature	Strategic Classification <table> <tr> <th>Domain</th> <th>Business Model</th> <th>Evolution</th> </tr> <tr> <td>Core</td> <td>Compliance</td> <td>Commodity</td> </tr> </table>	Domain	Business Model	Evolution	Core	Compliance	Commodity	Domain Roles <table> <tr> <td>Role types</td> </tr> <tr> <td>Execution context</td> </tr> </table>	Role types	Execution context	
Domain	Business Model	Evolution									
Core	Compliance	Commodity									
Role types											
Execution context											
Inbound Communication	Ubiquitous Language	Outbound Communication									
Collaborator  IoT Gateway Messages SensorDataReceived  Admin / Superadmin ConfigureSensor	Ubiquitous Language Context-specific domain terminology <div style="border: 1px dashed gray; padding: 5px; margin-top: 10px;"> Sensor, OccupancyStatus, SensorReading, Heartbeat </div>	Outbound Communication Collaborator  Cubicle Management Messages CubicleOccupancyChanged  SensorHeartbeat  Cubicle Management SensorStatusUpdated									
Inbound Communication	Business Decisions										
	Key business rules, policies, and decisions <div style="border: 1px solid blue; padding: 5px; margin-top: 10px;"> Rules for interpreting sensor data (thresholds, algorithms). How to handle conflicting sensor readings. Frequency of status updates. Fallback mechanisms if sensors fail </div>										

4.1.2. Context Mapping

En esta sección desarrollamos un conjunto de context maps para visualizar las relaciones entre los bounded contexts del sistema. A partir de la información recolectada, exploramos distintas alternativas de diseño, cuestionando cómo cambiaría la estructura si reubicamos, dividimos o agrupamos capabilities. Finalmente, evaluamos cada propuesta considerando patrones como Anti-corruption Layer, Conformist, Customer/Supplier y Shared Kernel, con el fin de definir la mejor aproximación para la arquitectura del dominio. A continuación presentaremos las opciones que contemplamos para el sistema y la estructura final.

Opción 1

Esta alternativa propone una arquitectura compuesta por cinco bounded contexts bien definidos, con relaciones claras entre ellos. La estructura busca equilibrar la separación de responsabilidades, para permitir que el sistema escale y se mantenga con facilidad. Además, facilita la instalación sin asistencia técnica y asegura tiempos de respuesta adecuados, lo que contribuye directamente a mejorar la experiencia de los estudiantes y administradores en el uso del sistema.

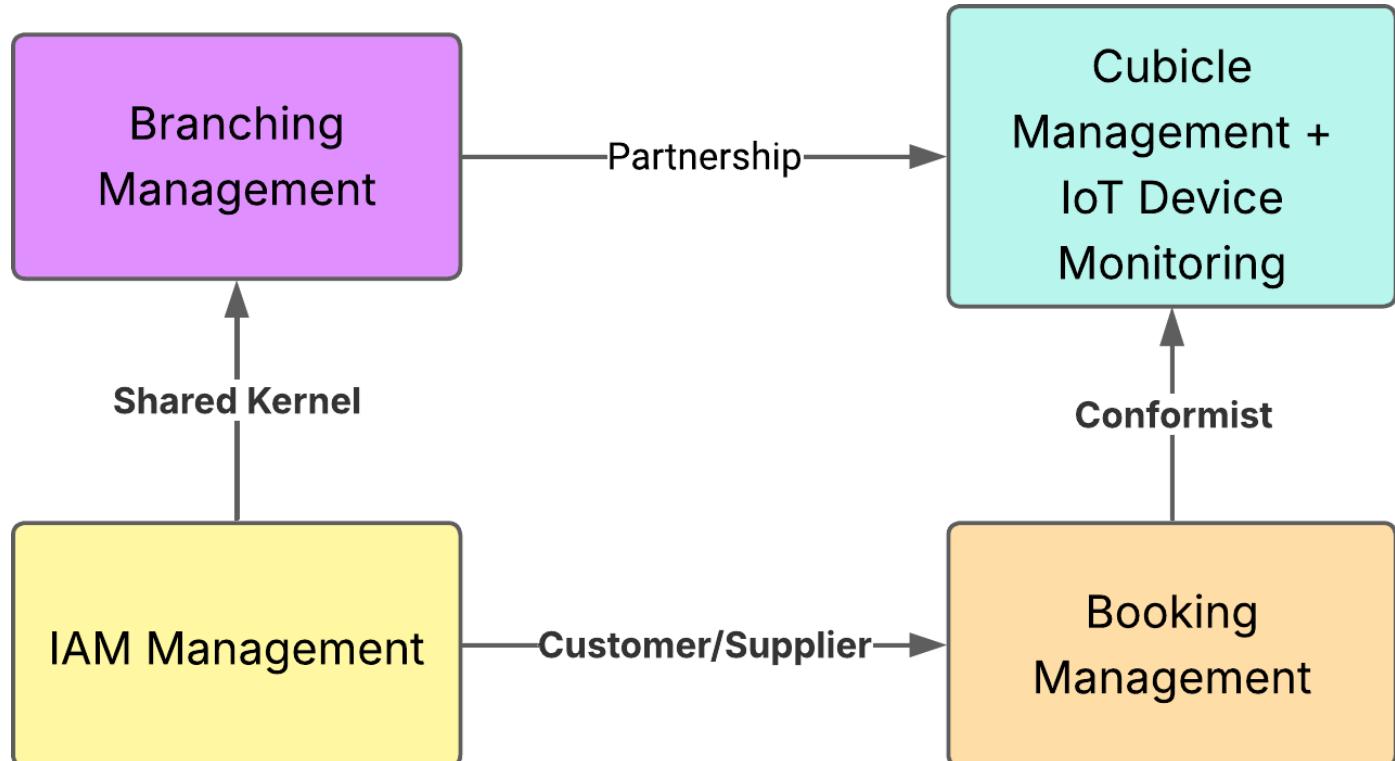


Opción 2

Esta alternativa propone unir los contextos de IoT Monitoring y Cubicle Management en un solo bounded context. Al hacerlo, se elimina la necesidad de sincronización externa entre ambos, manteniendo relaciones similares con los demás contextos del sistema.

Esta combinación presenta ventajas como la simplificación de la arquitectura al disminuir la cantidad de bounded contexts, una comunicación más directa entre la detección de ocupación de los cubículos y la gestión de reservas.

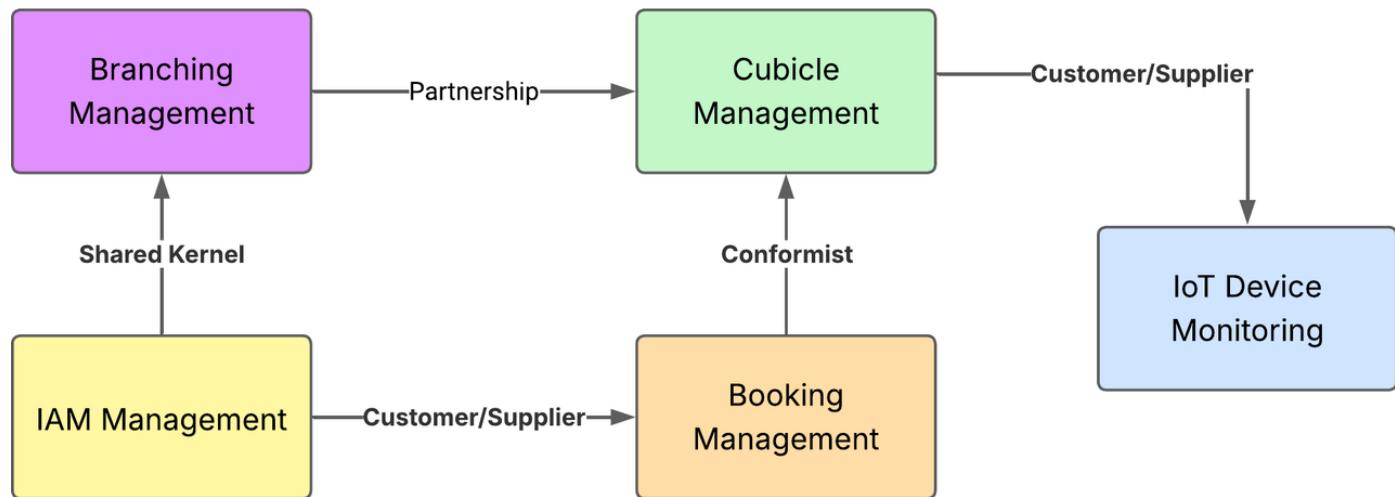
No obstante, una desventaja es la combinación de responsabilidades distintas, ya que una parte se enfoca en la infraestructura de sensores y la otra en procesos administrativos. Esto podría dificultar que el personal de la universidad realice la instalación del sistema en menos de una hora sin ayuda técnica, y además genera el riesgo de que un solo contexto asuma demasiadas funciones.



Opcion 3

En esta estructura mantenemos los cinco bounded contexts separados con relaciones claramente definidas. Las ventajas de este tipo de contexto son por un lado la clara separación de responsabilidades y por otro lado, se especifica que cada contexto se enfoca en una funcionalidad específica. Una de las principales desventajas

es que hay una mayor complejidad en la sincronización entre contextos.



Elección

Elegimos la opción 1, ya que proporciona el mejor equilibrio entre la separación de responsabilidades, la fácil de implementación y el cumplimiento de los requisitos del sistema, tomando en cuenta lo puntos detallados a continuación:

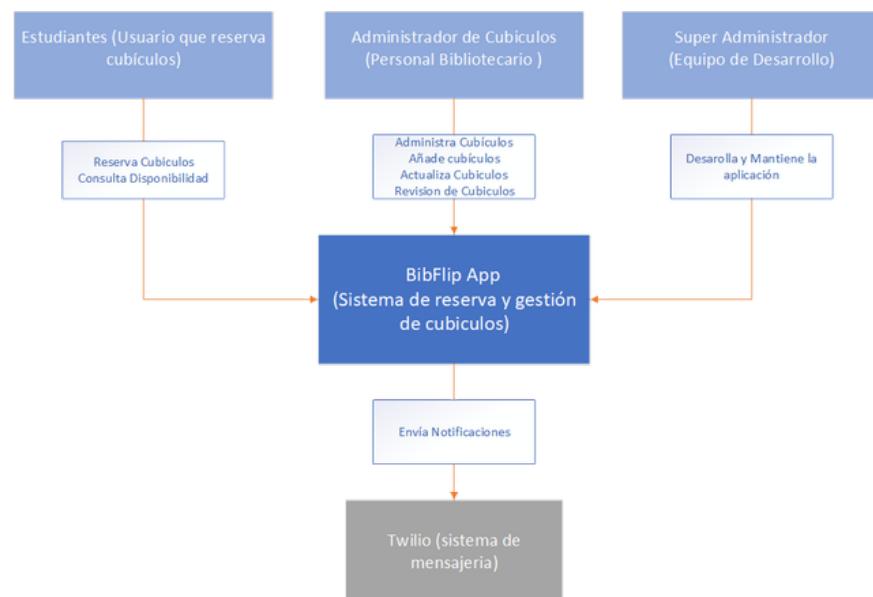
- Booking Management depende de IAM para autenticar y autorizar usuarios.
 - IAM Management y Branching Management comparten un núcleo común para datos de usuario y permisos administrativos.
 - Booking Management sigue las reglas y estado definidos por Cubicle Management sobre espacios.
 - IOT provee datos de sensores que Cubicle Management adapta para no contaminar su modelo.
 - Branching Management y Cubicle Management colaboran para gestionar cubículos y asignaciones.
 - IAM Management y IOT Device Monitoring operan independientes sin dependencia directa.

4.1.3. Software Architecture

En esta sección, se presenta y explica la representación de la arquitectura de software para el sistema utilizando el C4 Model. A través de estos diagramas, se busca proporcionar una comprensión clara y accesible de la arquitectura, permitiendo a los miembros del equipo, stakeholders y futuros desarrolladores entender cómo se organiza y comunica el sistema a diferentes niveles.

4.1.3.1. Software Architecture System Landscape Diagram

Este diagrama muestra que el sistema opera en un ecosistema compuesto por tres tipos de usuarios: Estudiantes, Administradores de Cubículos y Super Administradores. Los estudiantes utilizan el sistema para reservar cubículos y consultar su disponibilidad en tiempo real. Los Administradores de Cubículos gestionan la asignación de espacios y verifican que se cumpla la regla de ocupación mínima (no se pueden reservar cubículos con menos de 4 estudiantes). Por su parte, el Super Administrador se encargan de mantener y actualizar el sistema. El sistema también se conecta con un servicio externo, como Twilio, para enviar notificaciones a los usuarios.



4.1.3.2. Software Architecture Context Level Diagrams

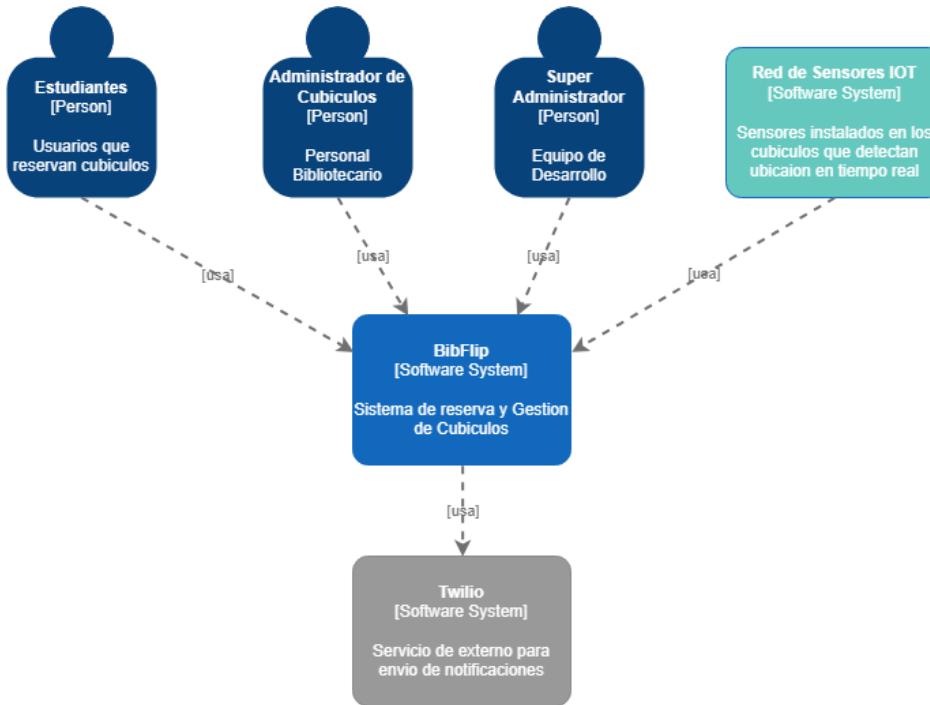
Este diagrama muestra que el sistema (representado como una única entidad) interactúa con los tres tipos de usuarios:

- Estudiantes, que buscan consultar cubículos disponibles y realizar reservas.

- Administradores de Cubículos, que gestionan los espacios disponibles y validan las reservas.
- Super Administradores, que mantienen el sistema en funcionamiento y realizan configuraciones generales.

El sistema también se comunica con dos servicios externos:

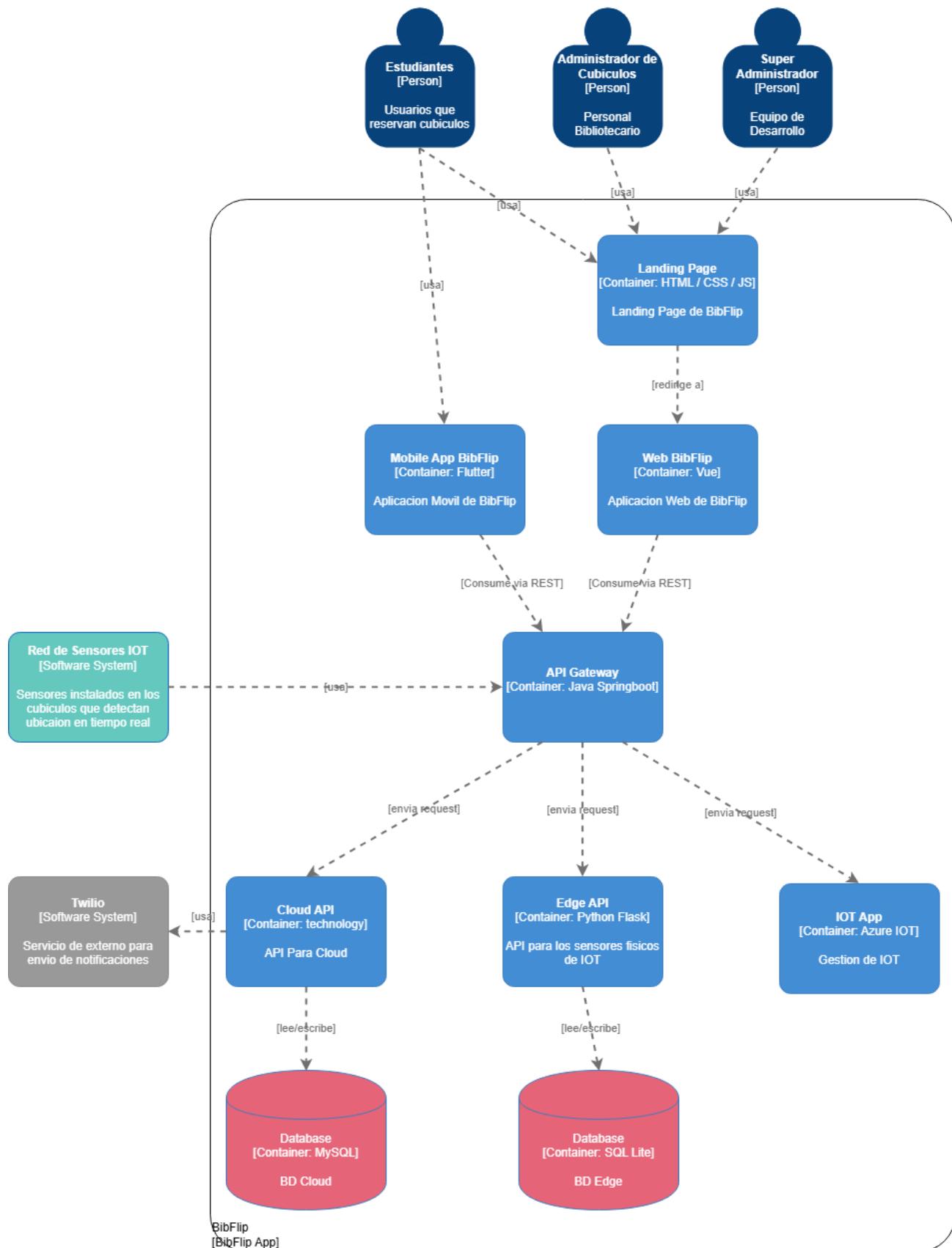
- Twilio, que permite enviar notificaciones a los usuarios.
- Sensores IoT, que proveen datos en tiempo real sobre la ocupación de cubículos.



4.1.3.3. Software Architecture Container Level Diagrams

Este diagrama muestra que el sistema está compuesto por x contenedores principales:

- Una Landing Page para presentar nuestro producto a cualquier usuario.
- Una aplicación web accesible desde navegadores.
- Una aplicación móvil para estudiantes en dispositivos portátiles.
- Una API Gateway para repartir los request hacia el servicio correspondiente.
- Una Cloud API que concentra toda la lógica de negocio, incluyendo la validación de reglas de ocupación mínima y reservas.
- Una Edge API para hacer los calculos y análisis necesario antes de mandarlo a cloud
- IOT App para gestionar todo los sistemas físicos.
- Dos base de datos que almacena información de estudiantes, cubículos y reservas.

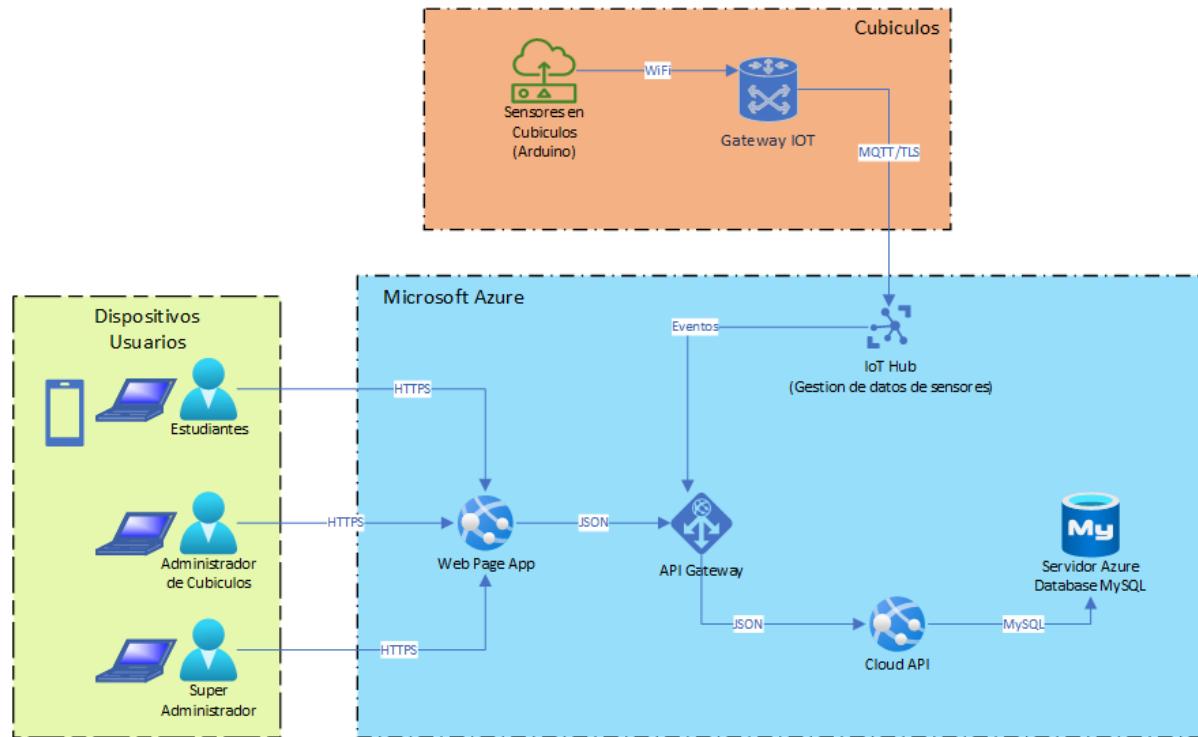


4.1.3.4. Software Architecture Deployment Diagrams

Este diagrama muestra que el sistema se despliega en tres entornos principales: Microsoft Azure Cloud, Dispositivos Cliente y Sedes físicas de la universidad.

En Azure, el sistema utiliza App Service para alojar la aplicación web frontend, otro App Service para la API (con documentación Swagger), Azure Database for MySQL para los datos, y Azure IoT Hub para gestionar las comunicaciones con los sensores de cubículos.

Los estudiantes acceden al sistema desde navegadores web o dispositivos Android. En cada sede, los cubículos cuentan con sensores IoT conectados a un Gateway local, que transmite la información al IoT Hub en Azure. Todas las comunicaciones se realizan mediante protocolos seguros como HTTPS, MQTT/TLS o conexiones MySQL.



4.2. Tactical-Level Domain-Driven Design

4.2.1. Bounded Context: IAM

El **IAM (Identity and Access Management) Bounded Context** es responsable de gestionar la autenticación y la creación de usuarios en el sistema. Este contexto asegura que los usuarios (principalmente Estudiantes, Administradores de Cubículos y Super Administradores) puedan registrarse, iniciar sesión y gestionar sus credenciales de manera segura, cumpliendo con los estándares de seguridad y privacidad.

4.2.1.1. Domain Layer

La Domain Layer del IAM Bounded Context encapsula la lógica de negocio relacionada con la autenticación y la gestión de usuarios. En esta capa se definen los elementos principales del dominio: agregados, entidades y objetos de valor que representan los conceptos clave del sistema.

Aggregates

1. User

- **Propósito:** El agregado principal es el usuario (**User**), que encapsula la lógica de negocio relacionada con la autenticación y la gestión de roles.
- **Atributos:**
 - **username:** Identificador único del usuario, validado para cumplir con restricciones de longitud y unicidad.
 - **password:** Contraseña del usuario, almacenada de forma segura.
 - **roles:** Conjunto de roles asociados al usuario, representados como una relación **ManyToMany** con la entidad **Role**.
- **Métodos:**
 - **addRole(Role role):** Agrega un rol al usuario.
 - **addRoles(List<Role> roles):** Agrega múltiples roles al usuario, validando que sean roles válidos.
- **Características:**
 - Extiende **AuditableAbstractAggregateRoot**, lo que permite auditar cambios en los usuarios.
 - Gestiona la relación entre usuarios y roles, asegurando consistencia y validación.

Entities

1. Role

- **Propósito:** La entidad **Role** representa los roles que pueden ser asignados a los usuarios.
- **Atributos:**
 - **id:** Identificador único del rol.
 - **name:** Nombre del rol, representado como un valor enumerado (**Roles**).
- **Métodos:**
 - **getDefaultRole():** Devuelve el rol predeterminado (**ROLE_USER**).
 - **toRoleFromName(String name):** Convierte un nombre de rol en una instancia de **Role**.
 - **validateRoleSet(List<Role> roles):** Valida un conjunto de roles, asignando un rol predeterminado si el conjunto está vacío o es nulo.
- **Características:**
 - Define una relación única entre los nombres de roles y sus representaciones en la base de datos.
 - Facilita la validación y asignación de roles a los usuarios.

Value Objects:

1. Roles

- **Propósito:** El objeto de valor `Roles` es una enumeración que define los roles disponibles en el sistema.
- **Valores:**
 - `ROLE_ADMIN`: Rol de administrador.
 - `ROLE_SUPERVISOR`: Rol de supervisor.
 - `ROLE_USER`: Rol de usuario estándar.
- **Características:**
 - Representa roles como valores inmutables, asegurando consistencia en su uso dentro del dominio.

En conjunto, estos elementos permiten modelar de manera robusta y segura la lógica de negocio relacionada con la autenticación y la gestión de usuarios, asegurando que las reglas del dominio se cumplan de manera consistente.

4.2.1.2. Interface Layer

La **Interface Layer** del IAM Bounded Context expone los puntos de entrada al sistema a través de controladores REST. Estos controladores permiten la interacción con las entidades del dominio mediante solicitudes HTTP, facilitando la comunicación entre los clientes y el sistema. Además, esta capa incluye recursos y transformadores que aseguran una representación adecuada de los datos y su conversión entre las capas de la aplicación.

Controllers

Los controladores son responsables de manejar las solicitudes HTTP y delegar la lógica de negocio a los servicios correspondientes. A continuación, se describen los principales controladores:

1. UsersController

- **Propósito:** Gestiona las operaciones relacionadas con los usuarios.
- **Endpoints:**
 - `GET /api/v1/users`: Obtiene la lista de todos los usuarios.
 - `GET /api/v1/users/{userId}`: Obtiene los detalles de un usuario específico por su ID.
- **Dependencias:**
 - `UserQueryService`: Servicio encargado de manejar las consultas relacionadas con los usuarios.

2. RolesController

- **Propósito:** Gestiona las operaciones relacionadas con los roles.
- **Endpoints:**
 - `GET /api/v1/roles`: Obtiene la lista de todos los roles disponibles.
- **Dependencias:**
 - `RoleQueryService`: Servicio encargado de manejar las consultas relacionadas con los roles.

3. AuthenticationController

- **Propósito:** Gestiona las operaciones de autenticación y registro de usuarios.
- **Endpoints:**
 - `POST /api/v1/authentication/sign-up`: Registra un nuevo usuario en el sistema.
 - `POST /api/v1/authentication/sign-in`: Autentica a un usuario y genera un token de acceso.
- **Dependencias:**
 - `UserCommandService`: Servicio encargado de manejar los comandos relacionados con los usuarios.

Resources

Los recursos representan los datos que se exponen a través de la API REST. Estos recursos son utilizados para estructurar las respuestas de los controladores y asegurar una representación clara y consistente de los datos. A continuación, se describen los principales recursos:

1. UserResource

- Representa un usuario en el sistema.
- **Atributos:**
 - `id`: Identificador único del usuario.
 - `username`: Nombre de usuario.
 - `roles`: Lista de roles asociados al usuario.

2. RoleResource

- Representa un rol en el sistema.
- **Atributos:**
 - `id`: Identificador único del rol.
 - `name`: Nombre del rol.

3. AuthenticatedUserResource

- Representa un usuario autenticado junto con su token de acceso.
- **Atributos:**
 - `id`: Identificador único del usuario.
 - `username`: Nombre de usuario.
 - `token`: Token de acceso generado.

4. **SignUpResource** Representa los datos necesarios para registrar un nuevo usuario.

- **Atributos:**

- `username`: Nombre de usuario.
- `password`: Contraseña.
- `roles`: Lista de roles asignados al usuario.

5. **SignInResource** Representa los datos necesarios para autenticar a un usuario.

- **Atributos:**

- `username`: Nombre de usuario.
- `password`: Contraseña.

Transformers

Los transformadores son responsables de convertir las entidades del dominio en recursos y viceversa. Esto asegura que los datos expuestos a través de la API REST sean consistentes y estén en el formato esperado. A continuación, se describen los principales transformadores:

1. **UserResourceFromEntityAssembler** Convierte una entidad `User` en un recurso `UserResource`.

- **Método principal:**

- `toResourceFromEntity(User entity)`: Transforma un usuario del dominio en un recurso.

2. **RoleResourceFromEntityAssembler** Convierte una entidad `Role` en un recurso `RoleResource`.

- **Método principal:**

- `toResourceFromEntity(Role entity)`: Transforma un rol del dominio en un recurso.

3. **AuthenticatedUserResourceFromEntityAssembler** Convierte una entidad `User` y un token en un recurso `AuthenticatedUserResource`.

- **Método principal:**

- `toResourceFromEntity(User entity, String token)`: Transforma un usuario autenticado en un recurso.

4. **SignUpCommandFromResourceAssembler** Convierte un recurso `SignUpResource` en un comando `SignUpCommand`.

- **Método principal:**

- `toCommandFromResource(SignUpResource resource)`: Transforma los datos de registro en un comando.

5. **SignInCommandFromResourceAssembler** Convierte un recurso `SignInResource` en un comando `SignInCommand`.

- **Método principal:**

- `toCommandFromResource(SignInResource resource)`: Transforma los datos de inicio de sesión en un comando.

Relaciones entre componentes

- Los controladores utilizan los servicios de consulta (`UserQueryService`, `RoleQueryService`) y de comandos (`UserCommandService`) para delegar la lógica de negocio.
- Los transformadores convierten las entidades del dominio en recursos para las respuestas HTTP y viceversa para las solicitudes entrantes.
- Los recursos estructuran los datos expuestos a los clientes, asegurando una representación clara y consistente.

Esta estructura asegura que la **Interface Layer** sea modular, reutilizable y fácil de mantener, facilitando la interacción entre los clientes y el sistema.

4.2.1.3 Application Layer

La **Application Layer** del IAM Bounded Context actúa como un intermediario entre la **Domain Layer** y las capas externas, como la **Interface Layer** y la **Infrastructure Layer**. Su propósito principal es coordinar las operaciones de negocio, manejar comandos y consultas, y orquestar la lógica de aplicación sin exponer directamente los detalles del dominio.

Command Services

Los servicios de comandos son responsables de ejecutar operaciones que modifican el estado del sistema. A continuación, se describen los principales servicios de comandos:

1. **RoleCommandServiceImpl**

- **Propósito:** Gestiona las operaciones relacionadas con los roles, como la creación inicial de roles en el sistema.

- **Métodos principales:**

- `handle(SeedRolesCommand command)`: Verifica si los roles predefinidos existen en el sistema y los crea si no están presentes.

- **Dependencias:**

- `RoleRepository`: Interactúa con la base de datos para verificar y guardar roles.

2. **UserCommandServiceImpl**

- **Propósito:** Gestiona las operaciones relacionadas con los usuarios, como el registro y la autenticación.

- **Métodos principales:**

- `handle(SignUpCommand command)`: Registra un nuevo usuario en el sistema, asignándole roles y almacenando su contraseña de forma segura.

- `handle(SignInCommand command)`: Autentica a un usuario verificando sus credenciales y generando un token de acceso.
- **Dependencias:**
 - `UserRepository`: Interactúa con la base de datos para guardar y recuperar usuarios.
 - `HashingService`: Codifica y verifica contraseñas.
 - `TokenService`: Genera y valida tokens de acceso.
 - `RoleRepository`: Recupera roles asignados a los usuarios.

Query Services

Los servicios de consultas son responsables de recuperar información del sistema sin modificar su estado. A continuación, se describen los principales servicios de consultas:

1. RoleQueryServiceImpl

- **Propósito:** Gestiona las consultas relacionadas con los roles.
- **Métodos principales:**
 - `handle(GetAllRolesQuery query)`: Recupera todos los roles disponibles en el sistema.
 - `handle(GetRoleByNameQuery query)`: Recupera un rol específico por su nombre.
- **Dependencias:**
 - `RoleRepository`: Interactúa con la base de datos para recuperar roles.

2. UserQueryServiceImpl

- **Propósito:** Gestiona las consultas relacionadas con los usuarios.
- **Métodos principales:**
 - `handle(GetAllUsersQuery query)`: Recupera todos los usuarios registrados en el sistema.
 - `handle(GetUserByIdQuery query)`: Recupera un usuario específico por su ID.
 - `handle(GetUserByUsernameQuery query)`: Recupera un usuario específico por su nombre de usuario.
- **Dependencias:**
 - `UserRepository`: Interactúa con la base de datos para recuperar usuarios.

Relaciones entre componentes

- Los **Command Services** interactúan con los repositorios para modificar el estado del sistema y con los servicios externos para operaciones auxiliares, como el hashing de contraseñas y la generación de tokens.
- Los **Query Services** interactúan únicamente con los repositorios para recuperar información del sistema.

Esta estructura asegura que la **Application Layer** sea modular, reutilizable y fácil de mantener, permitiendo una separación clara de responsabilidades y facilitando la evolución del sistema.

4.2.1.4. Infrastructure Layer

La **Infrastructure Layer** del IAM Bounded Context proporciona las implementaciones técnicas necesarias para soportar las operaciones del sistema. Esta capa incluye configuraciones de seguridad, servicios de hashing, manejo de tokens, repositorios para la persistencia de datos y componentes relacionados con la autorización y autenticación. Su objetivo principal es conectar la lógica de negocio con los recursos externos, como bases de datos, servicios de seguridad y APIs externas.

Persistencia (JPA Repositories)

1. RoleRepository

- **Propósito:** Proporciona métodos para interactuar con la base de datos de roles.
- **Métodos principales:**
 - `findByName`: Busca un rol por su nombre.
 - `existsByName`: Verifica si un rol existe en la base de datos.

2. UserRepository

- **Propósito:** Proporciona métodos para interactuar con la base de datos de usuarios.
- **Métodos principales:**
 - `findByUsername`: Busca un usuario por su nombre de usuario.
 - `existsByUsername`: Verifica si un usuario existe en la base de datos.

Relaciones entre componentes

- **Persistencia:** Los repositorios (`RoleRepository`, `UserRepository`) proporcionan acceso a los datos almacenados en la base de datos, permitiendo a las capas superiores interactuar con las entidades del dominio.

Esta estructura asegura que la **Infrastructure Layer** sea modular, reutilizable y fácil de mantener, facilitando la integración con otros sistemas y servicios externos.

4.2.1.5. Bounded Context Software Architecture Component Level Diagrams

En esta sección se presenta el diagrama de componentes del **IAM Bounded Context**, el cual detalla los principales módulos y sus interacciones dentro del contexto delimitado. Este diagrama sigue el enfoque del C4 Model para representar los componentes clave, como servicios de aplicación, controladores, repositorios y

servicios externos, junto con sus relaciones.

El propósito de este diagrama es proporcionar una visión clara y estructurada de cómo se organizan los componentes dentro del contexto, facilitando la comprensión de su arquitectura y permitiendo identificar puntos de integración y responsabilidades.

El **IAM Bounded Context** está compuesto por los siguientes módulos principales:

1. Application Layer:

- Coordina las operaciones de negocio relacionadas con la autenticación y la gestión de usuarios.
- Incluye servicios de comandos y consultas que interactúan con la **Domain Layer** y la **Infrastructure Layer**.
- Maneja eventos relacionados con el registro de usuarios, la asignación de roles y la autenticación.

2. Interface Layer:

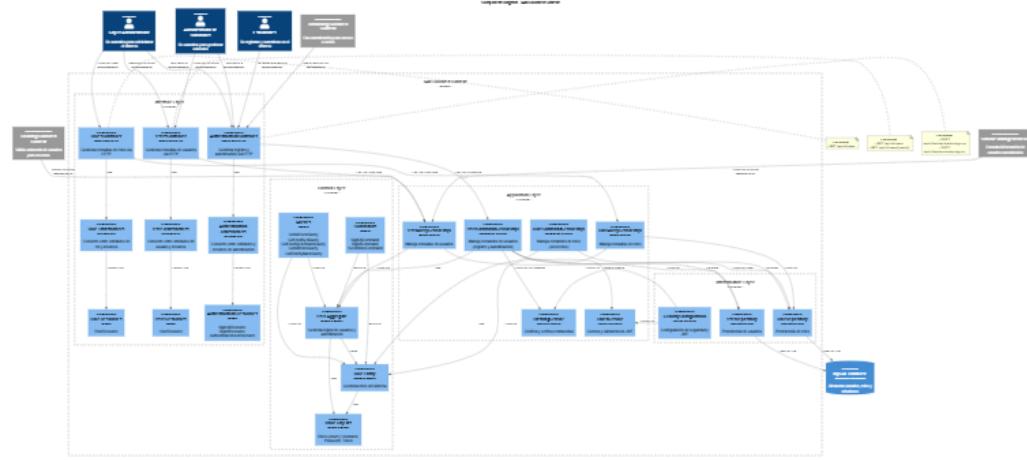
- Expone los puntos de entrada al sistema a través de controladores REST.
- Incluye recursos y transformadores que aseguran una representación adecuada de los datos y su conversión entre las capas de la aplicación.
- Proporciona endpoints para operaciones como el registro de usuarios, la autenticación y la consulta de roles.

3. Domain Layer:

- Encapsula la lógica de negocio relacionada con la autenticación y la gestión de usuarios.
- Define los agregados, entidades y objetos de valor que representan los conceptos clave del dominio, como usuarios, roles y permisos.
- Asegura que las reglas de negocio, como la validación de contraseñas y la asignación de roles, se cumplan de manera consistente.

4. Infrastructure Layer:

- Proporciona las implementaciones técnicas necesarias para soportar las operaciones del sistema.
- Incluye repositorios para la persistencia de datos y componentes que conectan la lógica de negocio con los recursos externos, como bases de datos y servicios de seguridad.
- Implementa servicios auxiliares como el hashing de contraseñas y la generación de tokens de autenticación.

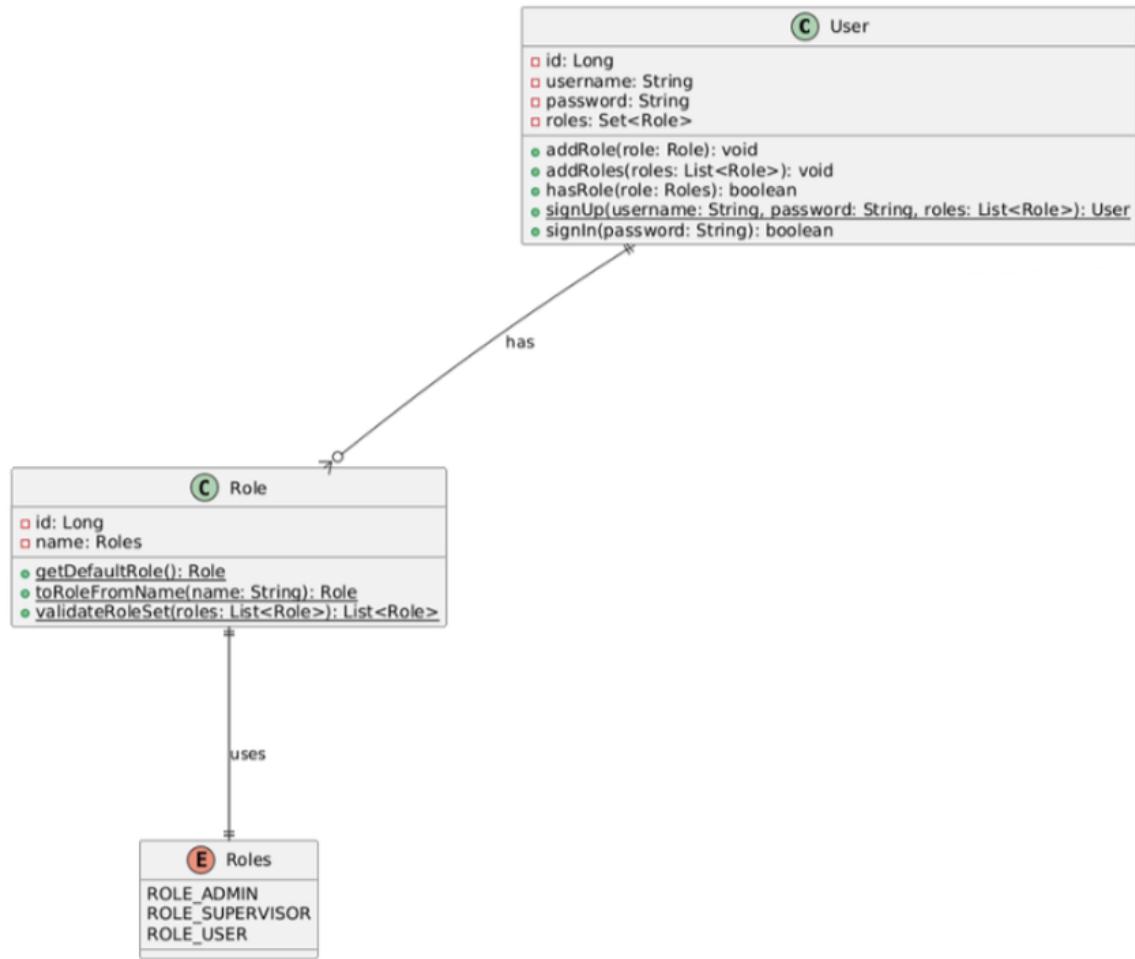


4.2.1.6. Bounded Context Software Architecture Code Level Diagrams

En este apartado se presentan los diagramas que ofrecen un mayor nivel de detalle sobre la implementación de los componentes del **IAM Bounded Context**. Estos diagramas están diseñados para ilustrar cómo se estructuran las clases, interfaces y relaciones dentro de las capas del contexto, proporcionando una visión técnica que facilita el desarrollo, mantenimiento y evolución del sistema.

4.2.1.6.1. Bounded Context Domain Layer Class Diagrams

El diagrama de clases correspondiente a la **Domain Layer** del **IAM Bounded Context** incluye las clases principales, como agregados, entidades y objetos de valor, así como las interfaces y enumeraciones que definen el comportamiento del dominio. También se destacan las relaciones entre estos elementos, como asociaciones, composiciones y dependencias.



Elementos principales del diagrama:

1. Aggregates:

- **User**: Agregado principal que encapsula la lógica de negocio relacionada con la autenticación y la gestión de usuarios. Incluye atributos como `username`, `password` y `roles`.
 - **Atributos**:
 - `username`: Identificador único del usuario.
 - `password`: Contraseña almacenada de forma segura.
 - `roles`: Lista de roles asignados al usuario.
 - **Métodos**:
 - `addRole(Role role)`: Agrega un rol al usuario.
 - `addRoles(List<Role> roles)`: Agrega múltiples roles al usuario, validando que sean válidos.

2. Entities:

- **Role**: Representa los roles que pueden ser asignados a los usuarios.
 - **Atributos**:
 - `id`: Identificador único del rol.
 - `name`: Nombre del rol.
 - **Métodos**:
 - `getDefaultRole()`: Devuelve el rol predeterminado (`ROLE_USER`).
 - `validateRoleSet(List<Role> roles)`: Valida un conjunto de roles asignados.

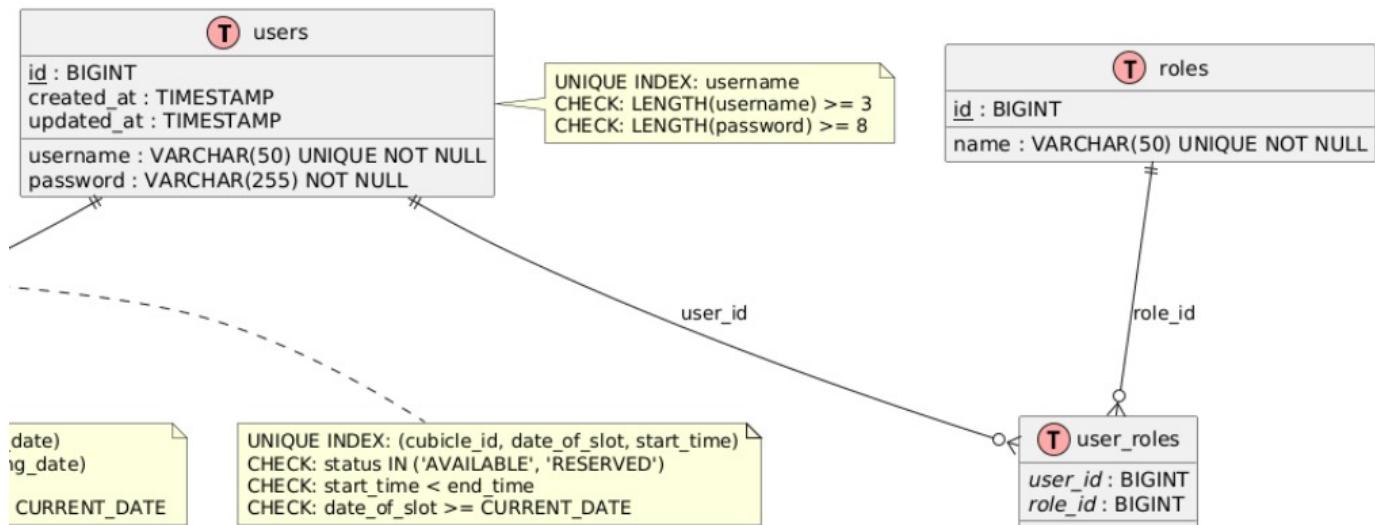
3. Value Objects:

- **Roles**: Enumeración que define los roles disponibles en el sistema (`ROLE_ADMIN`, `ROLE_SUPERVISOR`, `ROLE_USER`).
- **Password**: Representa la contraseña del usuario, asegurando que cumpla con las políticas de seguridad.
- **Token**: Representa un token de autenticación generado para un usuario.

Relaciones destacadas:

- El agregado **User** gestiona las relaciones con la entidad **Role**, asegurando que los usuarios tengan roles válidos asignados.
- Los objetos de valor encapsulan datos inmutables y validaciones específicas, como la seguridad de contraseñas y la unicidad de roles.
- La entidad **Role** define los permisos y responsabilidades que pueden ser asignados a los usuarios.

El diseño de la base de datos para el **IAM Bounded Context** refleja la estructura del dominio, asegurando que las entidades y relaciones definidas en la **Domain Layer** se representen de manera eficiente en el modelo relacional.



Este diseño incluye las siguientes tablas principales:

1. Users:

- Representa los usuarios registrados en el sistema.
- **Atributos principales:**
 - **id:** Identificador único del usuario.
 - **username:** Nombre de usuario único.
 - **password:** Contraseña almacenada de forma segura.
 - **created_at:** Fecha de creación del usuario.

2. Roles:

- Representa los roles disponibles en el sistema.
- **Atributos principales:**
 - **id:** Identificador único del rol.
 - **name:** Nombre del rol.

3. UserRoles:

- Representa la relación entre usuarios y roles.
- **Atributos principales:**
 - **user_id:** Identificador del usuario.
 - **role_id:** Identificador del rol.

Este diseño asegura que las operaciones relacionadas con la autenticación y la gestión de usuarios sean eficientes y consistentes, facilitando la integración con la lógica de negocio definida en la **Domain Layer**. Voy a completar el análisis táctico de Domain-Driven Design para los bounded contexts faltantes: Cubicle Management y IoT Monitoring. Desarrollaré cada sección siguiendo la estructura que ya tiene tu informe.

4.2.2. Bounded Context: Cubicle Management

El **Cubicle Management Bounded Context** es responsable de gestionar la información y el estado de los cubículos en las sedes de la universidad. Este contexto se encarga de registrar, actualizar y consultar el estado de los cubículos, así como su capacidad, ubicación dentro de la sede y otros atributos relevantes. Es el núcleo central que permite la visualización en tiempo real del estado de ocupación de los cubículos para los usuarios.

4.2.2.1. Domain Layer

La **Domain Layer** del Cubicle Management Bounded Context encapsula la lógica de negocio relacionada con la gestión de cubículos. En esta capa, se definen los elementos principales del dominio, como agregados, entidades y objetos de valor, que representan los conceptos clave del sistema.

Aggregates

1. Cubicle

- **Propósito:** El agregado principal es el cubículo (**Cubicle**), que encapsula la lógica de negocio relacionada con la gestión del cubículo y su estado de ocupación.
- **Atributos:**
 - **cubicleNumber:** Número identificativo del cubículo, representado como un objeto de valor **CubicleNumber**.
 - **headquarterId:** Identificador de la sede a la que pertenece el cubículo, representado como un objeto de valor **HeadquarterId**.
 - **capacity:** Capacidad del cubículo (máx. 4 o 5 estudiantes), representada como un objeto de valor **Capacity**.

- **status:** Estado actual del cubículo, representado como un objeto de valor `CubicleStatus`.
- **location:** Ubicación física del cubículo dentro de la sede, representada como un objeto de valor `Location`.
- **Métodos:**
 - `updateStatus(CubicleStatus status)`: Actualiza el estado del cubículo.
 - `assignToHeadquarter(HeadquarterId headquarterId)`: Asigna el cubículo a una sede específica.
 - `changeCapacity(Capacity capacity)`: Modifica la capacidad del cubículo.
 - `relocate(Location location)`: Actualiza la ubicación del cubículo dentro de la sede.
- **Características:**
 - Extiende `AuditableAbstractAggregateRoot`, lo que permite auditar cambios en los cubículos.
 - Gestiona la validación de estados y cambios según reglas de negocio.

Entities

1. Seat

- **Propósito:** La entidad `Seat` representa cada asiento asociado a un cubículo, con su estado de ocupación.
- **Atributos:**
 - `id`: Identificador único del asiento.
 - `seatNumber`: Número identificativo del asiento dentro del cubículo.
 - `status`: Estado actual del asiento (ocupado o libre), representado como un objeto de valor `SeatStatus`.
- **Métodos:**
 - `updateStatus(SeatStatus status)`: Actualiza el estado del asiento.
 - `isOccupied()`: Verifica si el asiento está ocupado.

Value Objects

1. CubicleNumber

- Representa el número identificativo de un cubículo.
- Validaciones: debe ser positivo y único dentro de una sede.

2. HeadquarterId

- Representa el identificador único de una sede.
- Validaciones: no puede ser nulo ni negativo.

3. Capacity

- Representa la capacidad de un cubículo en términos de estudiantes.
- Validaciones:
 - La capacidad debe ser un número positivo.
 - El rango permitido es entre 4 y 5 estudiantes.

4. CubicleStatus

- Enumera los estados posibles de un cubículo.
- Valores:
 - `AVAILABLE`: El cubículo está disponible para reservar.
 - `OCCUPIED`: El cubículo está actualmente ocupado.
 - `RESERVED`: El cubículo está reservado.
 - `MAINTENANCE`: El cubículo está fuera de servicio temporalmente.

5. SeatStatus

- Enumera los estados posibles de un asiento.
- Valores:
 - `OCCUPIED`: El asiento está ocupado por un estudiante.
 - `FREE`: El asiento está libre.

6. Location

- Representa la ubicación física de un cubículo dentro de la sede.
- Atributos:
 - `zone`: Zona de la universidad (piso, bloque, biblioteca, etc.).
 - `coordinates`: Coordenadas relativas dentro del plano de la sede.

Commands

1. CreateCubicleCommand

- Crea un nuevo cubículo en una sede específica.
- Atributos: `cubicleNumber`, `headquarterId`, `capacity`, `location`.

2. UpdateCubicleStatusCommand

- Actualiza el estado de un cubículo existente.
- Atributos: `cubicleId`, `status`.

3. DeleteCubicleCommand

- Elimina un cubículo existente.
- Atributos: `cubicleId`, `headquarterId`.

Queries

1. GetCubicleByIdQuery

- Recupera un cubículo específico por su ID.
- Atributos: `cubicleId`.

2. GetCubiclesByHeadquarterQuery

- Recupera todos los cubículos asociados a una sede.
- Atributos: `headquarterId`.

3. GetCubiclesByStatusQuery

- Recupera todos los cubículos con un estado específico.
- Atributos: `status`, `headquarterId`.

Events

1. CubicleCreatedEvent

- Evento que se dispara cuando se crea un nuevo cubículo.
- Atributos: `cubicleId`, `headquarterId`.

2. CubicleStatusChangedEvent

- Evento que se dispara cuando cambia el estado de un cubículo.
- Atributos: `cubicleId`, `previousStatus`, `newStatus`, `timestamp`.

4.2.2.2. Interface Layer

La **Interface Layer** del Cubicle Management Bounded Context expone los puntos de entrada al sistema a través de controladores REST. Esta capa permite la interacción con las entidades del dominio mediante solicitudes HTTP.

Access Control Layer (ACL)

1. CubicleManagementContextFacade

- **Propósito:** Proporciona una interfaz simplificada para interactuar con el dominio desde otros contextos.
- **Métodos principales:**
 - `getCubicleStatus(Long cubicleId)`
 - `getCubiclesByStatus(Long headquarterId, String status)`
 - `existsCubicle(Long cubicleId)`
- **Dependencias:** `CubicleQueryService`

Controllers

1. CubicleController

- **Endpoints:**
 - `POST /api/v1/cubicles`
 - `GET /api/v1/cubicles/{cubicleId}`
 - `GET /api/v1/cubicles`
 - `GET /api/v1/headquarters/{headquarterId}/cubicles`
 - `PUT /api/v1/cubicles/{cubicleId}/status`
 - `DELETE /api/v1/cubicles/{cubicleId}`

Resources

1. CreateCubicleResource

- Atributos: `cubicleNumber`, `headquarterId`, `capacity`, `zone`, `xPosition`, `yPosition`

2. CubicleResource

- Atributos: `id`, `cubicleNumber`, `headquarterId`, `capacity`, `status`, `zone`, `position`

3. UpdateCubicleStatusResource

- Atributos: `status`

Transformers

1. [CreateCubicleCommandFromResourceAssembler](#)
 2. [CubicleResourceFromEntityAssembler](#)
 3. [UpdateCubicleStatusCommandFromResourceAssembler](#)

4.2.2.3. Application Layer

La **Application Layer** actúa como intermediaria entre la **Domain Layer** y capas externas

Command Services

- **CubicleCommandServiceImpl**
 - Métodos: `handle(CreateCubicleCommand)`, `handle(UpdateCubicleStatusCommand)`, `handle(DeleteCubicleCommand)`
 - Validaciones:
 - Capacidad entre 4 y 5 estudiantes.
 - Verificación de sede con `ExternalHeadquarterService`.
 - Evitar duplicados en la misma sede.
 - Dependencias: `CubicleRepository`, `ExternalHeadquarterService`

Query Services

- **CubicleQueryServiceImpl**
 - Métodos: `handle(GetCubicleByIdQuery)`, `handle(GetCubiclesByHeadquarterQuery)`, `handle(GetCubiclesByStatusQuery)`
 - Dependencias: `CubicleRepository`

Event Handlers

- **CubicleStatusChangedEventHandler**
 - Maneja `CubicleStatusChangedEventArgs`
 - Dependencias: `NotificationService`, `StatisticsService`

Outbound Services (ACL)

- **ExternalHeadquarterService**
 - existsHeadquarter(Long headquarterId)

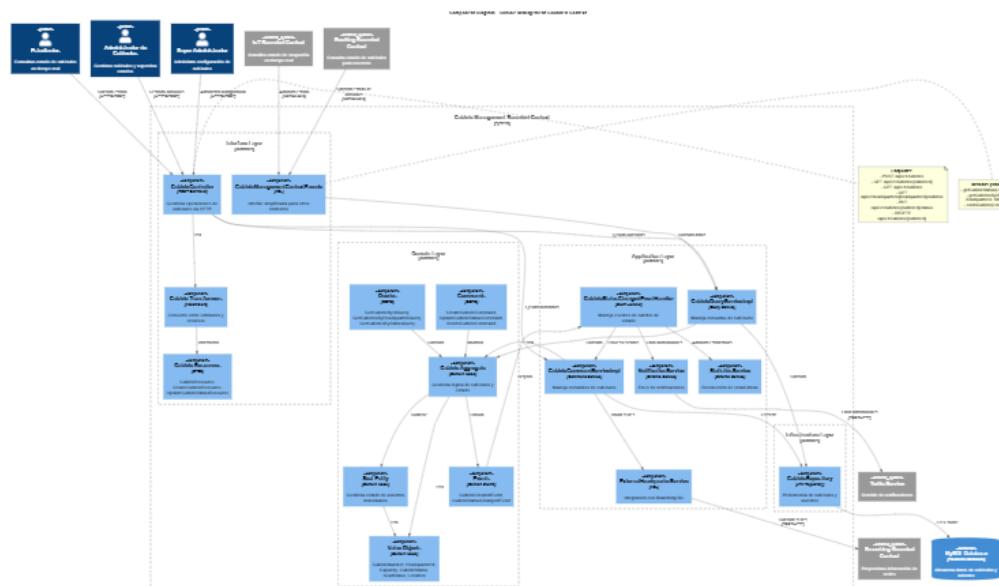
4.2.2.4. Infrastructure Layer

La **Infrastructure Layer** del Cubicle Management Bounded Context proporciona implementaciones técnicas y repositorios.

Persistencia (JPA Repositories)

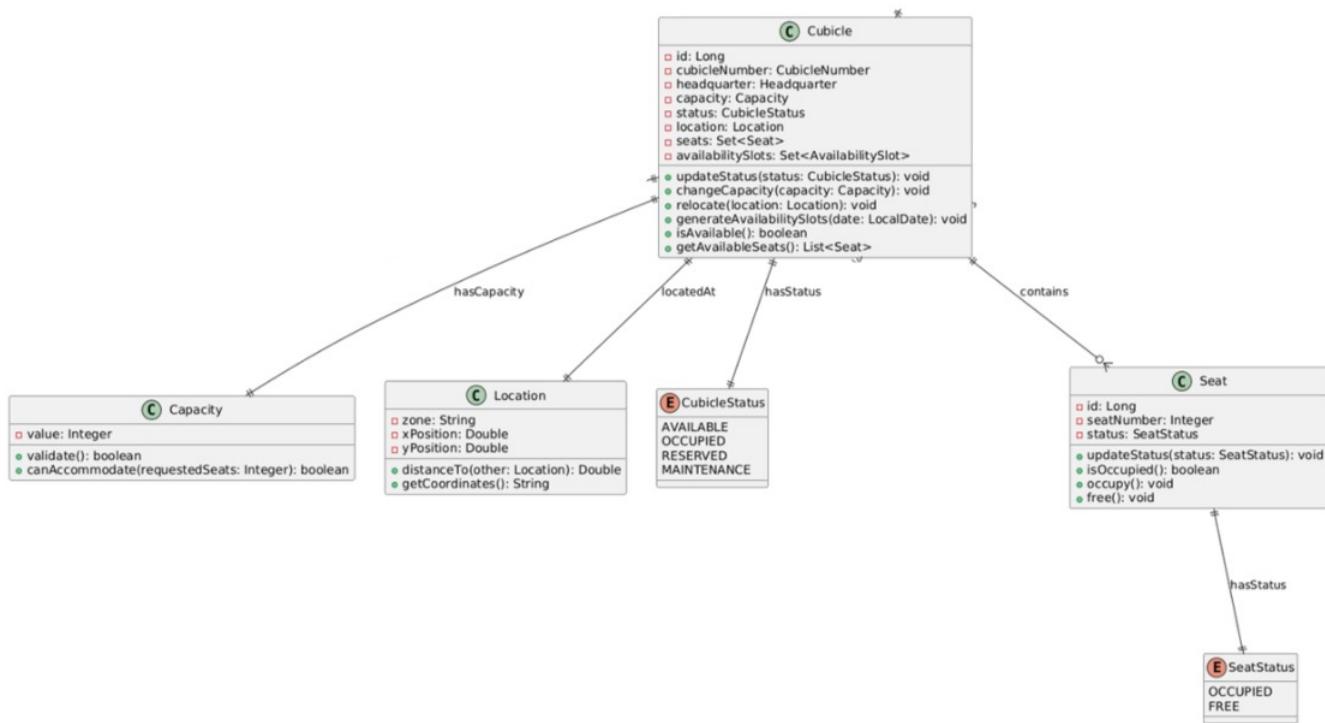
- 1. **CubicleRepository**
 - o `findByHeadquarterId(Long headquarterId)`
 - o `findByHeadquarterIdAndStatus(Long headquarterId, CubicleStatus status)`
 - o `existsByHeadquarterIdAndCubicleNumber(Long headquarterId, Integer cubicleNumber)`

4.2.2.5. Bounded Context Software Architecture Component Level Diagrams



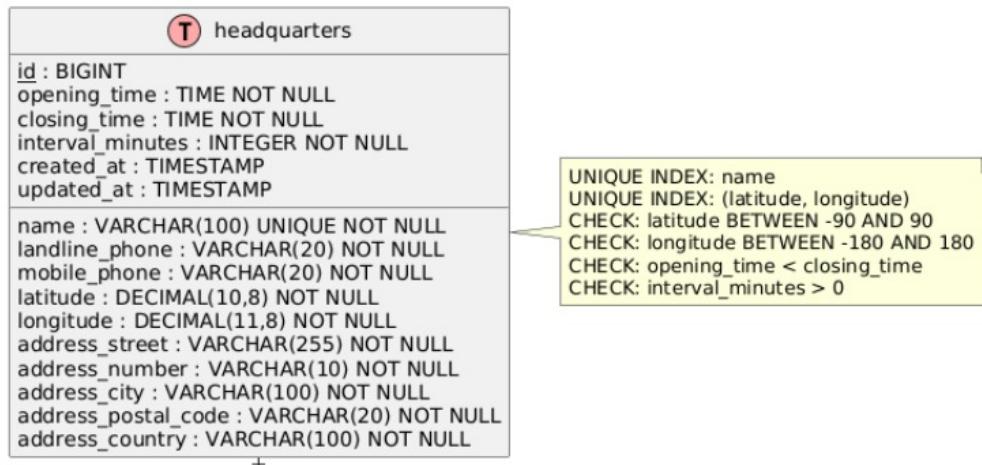
4.2.2.6. Bounded Context Software Architecture Code Level Diagrams

4.2.2.6.1. Bounded Context Domain Layer Class Diagrams



4.2.2.6.2. Bounded Context Database Design Diagram

El diseño de la base de datos para el **Cubicle Management Bounded Context** refleja la estructura del dominio, asegurando que las entidades y relaciones definidas en la **Domain Layer** se representen de manera eficiente en el modelo relacional.



Este diseño incluye las siguientes tablas principales:

1. Cubicles:

- Representa los cubículos en el sistema.
- **Atributos principales:**
 - **id**: Identificador único del cubículo.
 - **cubicle_number**: Número del cubículo dentro de la sede.
 - **headquarter_id**: Identificador de la sede a la que pertenece el cubículo.
 - **capacity**: Capacidad del cubículo en términos de estudiantes.
 - **status**: Estado actual del cubículo (AVAILABLE, OCCUPIED, RESERVED, MAINTENANCE).
 - **zone**: Zona de la sede donde está ubicado el cubículo.
 - **x_position**: Coordenada X del cubículo en el plano.
 - **y_position**: Coordenada Y del cubículo en el plano.

2. Seats:

- Representa los asientos asociados a los cubículos.
- **Atributos principales:**
 - `id`: Identificador único del asiento.
 - `cubicle_id`: Identificador del cubículo al que pertenece el asiento.
 - `seat_number`: Número del asiento dentro del cubículo.
 - `status`: Estado actual del asiento (OCCUPIED, FREE).

4.2.3. Bounded Context: Branching

El **Branching Bounded Context** es responsable de gestionar la información de las sedes (headquarters) del sistema. Este contexto asegura que las sedes puedan ser registradas, actualizadas y gestionadas de manera eficiente, incluyendo detalles como horarios de atención, ubicación geográfica, información de contacto y dirección. Este contexto es clave para garantizar que las operaciones relacionadas con las sedes sean consistentes y cumplan con los estándares de calidad y precisión requeridos.

4.2.3.1. Domain Layer

La **Domain Layer** del Branching Bounded Context encapsula la lógica de negocio relacionada con la gestión de sedes. En esta capa, se definen los elementos principales del dominio, como agregados, entidades y objetos de valor, que representan los conceptos clave del sistema.

Aggregates

1. Headquarter

- **Propósito:** El agregado principal es la sede (`Headquarter`), que encapsula la lógica de negocio relacionada con la gestión de información de las sedes.
- **Atributos:**
 - `name`: Nombre de la sede, representado como un objeto de valor `NameHeadquarter`.
 - `contactNumbers`: Números de contacto de la sede, representados como un objeto de valor `ContactNumbers`.
 - `coordinates`: Coordenadas geográficas de la sede, representadas como un objeto de valor `Coordinates`.
 - `schedule`: Horarios de atención de la sede, representados como una entidad `Schedule`.
 - `address`: Dirección de la sede, representada como un objeto de valor `StreetAddress`.
- **Métodos:**
 - `getOpeningTime()`: Devuelve la hora de apertura de la sede.
 - `getClosingTime()`: Devuelve la hora de cierre de la sede.
 - `getIntervalMinutes()`: Devuelve el intervalo de tiempo en minutos entre reservas.
- **Características:**
 - Extiende `AuditableAbstractAggregateRoot`, lo que permite auditar cambios en las sedes.
 - Gestiona la relación entre los horarios, la dirección y los datos de contacto de la sede, asegurando consistencia y validación.

Entities

1. Schedule

- **Propósito:** Representa los horarios de atención de una sede.
- **Atributos:**
 - `businessHours`: Horarios de apertura y cierre, representados como un objeto de valor `BusinessHours`.
 - `intervalMinutes`: Intervalo de tiempo en minutos entre reservas.
- **Características:**
 - Permite definir y validar los horarios de atención de la sede.
 - Incluye validaciones para garantizar que los horarios sean consistentes (por ejemplo, la hora de apertura no puede ser posterior a la hora de cierre).

Value Objects

1. BusinessHours

- **Propósito:** Representa los horarios de apertura y cierre de una sede.
- **Atributos:**
 - `openingTime`: Hora de apertura.
 - `closingTime`: Hora de cierre.
- **Validaciones:**
 - La hora de apertura no puede ser nula.
 - La hora de cierre no puede ser nula.
 - La hora de apertura no puede ser posterior a la hora de cierre.

2. ContactNumbers

- **Propósito:** Representa los números de contacto de una sede.
- **Atributos:**
 - `landlinePhone`: Número de teléfono fijo.
 - `mobilePhone`: Número de teléfono móvil.
- **Validaciones:**
 - Ambos números deben ser válidos y no nulos.

3. Coordinates

- **Propósito:** Representa las coordenadas geográficas de una sede.
- **Atributos:**
 - `latitude`: Latitud de la sede.
 - `longitude`: Longitud de la sede.
- **Validaciones:**
 - La latitud y la longitud no pueden ser nulas.

4. NameHeadquarter

- **Propósito:** Representa el nombre de una sede.
- **Atributos:**
 - `name`: Nombre de la sede.
- **Validaciones:**
 - El nombre no puede ser nulo ni vacío.
 - El nombre debe cumplir con un formato válido (solo letras y espacios).

5. StreetAddress

- **Propósito:** Representa la dirección de una sede.
- **Atributos:**
 - `street`: Calle.
 - `number`: Número de la dirección.
 - `city`: Ciudad.
 - `postalCode`: Código postal.
 - `country`: País.
- **Validaciones:**
 - Los campos `street`, `city`, `postalCode` y `country` no pueden ser nulos ni vacíos.
 - La calle debe cumplir con un formato válido (solo letras, espacios y puntos).

Relaciones entre componentes

- El agregado `Headquarter` actúa como el núcleo del dominio, gestionando las relaciones con los objetos de valor (`NameHeadquarter`, `ContactNumbers`, `Coordinates`, `StreetAddress`) y la entidad `Schedule`.
- Los objetos de valor encapsulan datos inmutables y validaciones específicas, asegurando consistencia en el dominio.
- La entidad `Schedule` permite modelar horarios complejos, incluyendo intervalos de tiempo entre reservas.

4.2.3.2. Interface Layer

La **Interface Layer** del Branching Bounded Context expone los puntos de entrada al sistema a través de controladores REST y una ACL (Access Control Layer). Esta capa permite la interacción con las entidades del dominio mediante solicitudes HTTP, facilitando la comunicación entre los clientes y el sistema. Además, incluye recursos y transformadores que aseguran una representación adecuada de los datos y su conversión entre las capas de la aplicación.

Access Control Layer (ACL)

1. HeadquarterContextFacade

- **Propósito:** Proporciona una interfaz simplificada para interactuar con el dominio del Branching Bounded Context desde otros contextos. Permite consultar información clave de las sedes, como horarios de apertura y cierre, intervalos de servicio y existencia de una sede.
- **Métodos principales:**
 - `getOpeningTime(Long headquarterId)`: Devuelve la hora de apertura de una sede.
 - `getClosingTime(Long headquarterId)`: Devuelve la hora de cierre de una sede.
 - `getIntervalMinutes(Long headquarterId)`: Devuelve el intervalo de servicio en minutos de una sede.
 - `existsHeadquarter(Long headquarterId)`: Verifica si una sede existe en el sistema.
- **Dependencias:**
 - `HeadquarterQueryService`: Servicio encargado de manejar las consultas relacionadas con las sedes.

Controllers

1. HeadquarterController

- **Propósito:** Gestiona las operaciones relacionadas con las sedes.
- **Endpoints:**
 - `POST /api/v1/headquarters`: Crea una nueva sede.
 - `GET /api/v1/headquarters/{headquarterId}`: Obtiene los detalles de una sede específica por su ID.
 - `GET /api/v1/headquarters`: Obtiene la lista de todas las sedes.
- **Dependencias:**
 - `HeadquarterCommandService`: Servicio encargado de manejar los comandos relacionados con las sedes.
 - `HeadquarterQueryService`: Servicio encargado de manejar las consultas relacionadas con las sedes.

Resources

1. CreateHeadquarterResource

- Atributos:
 - name, landlinePhone, mobilePhone, latitude, longitude, street, number, city, postalCode, country, openingTime, closingTime, intervalMinutes.

2. HeadquarterResource

- Atributos:
 - id, name, landlinePhone, mobilePhone, latitude, longitude, streetAddress, openingTime, closingTime, intervalMinutes.

Transformers

1. CreateHeadquarterCommandFromResourceAssembler

- Convierte `CreateHeadquarterResource` en `CreateHeadquarterCommand`.

2. HeadquarterResourceFromEntityAssembler

- Convierte `Headquarter` en `HeadquarterResource`.

4.2.3.3. Application Layer

4.2.3.4. Infrastructure Layer

La **Infrastructure Layer** del Branching Bounded Context proporciona las implementaciones técnicas necesarias para soportar las operaciones del sistema relacionadas con la gestión de sedes. Esta capa incluye repositorios para la persistencia de datos y componentes que conectan la lógica de negocio con los recursos externos, como bases de datos. Su objetivo principal es garantizar que las operaciones de almacenamiento y recuperación de información sean eficientes, consistentes y seguras.

Persistencia (JPA Repositories)

1. HeadquarterRepository

- Propósito: Proporciona métodos para interactuar con la base de datos de sedes.
- Métodos principales:
 - `existsByName(NameHeadquarter name)`: Verifica si existe una sede con un nombre específico.
 - `existsByCoordinates(Coordinates coordinates)`: Verifica si existe una sede en una ubicación geográfica específica.
- Características:
 - Extiende `JpaRepository`, lo que permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre las entidades `Headquarter`.
 - Facilita la validación de unicidad para nombres y coordenadas de sedes, asegurando que no haya duplicados en el sistema.

Relaciones entre componentes

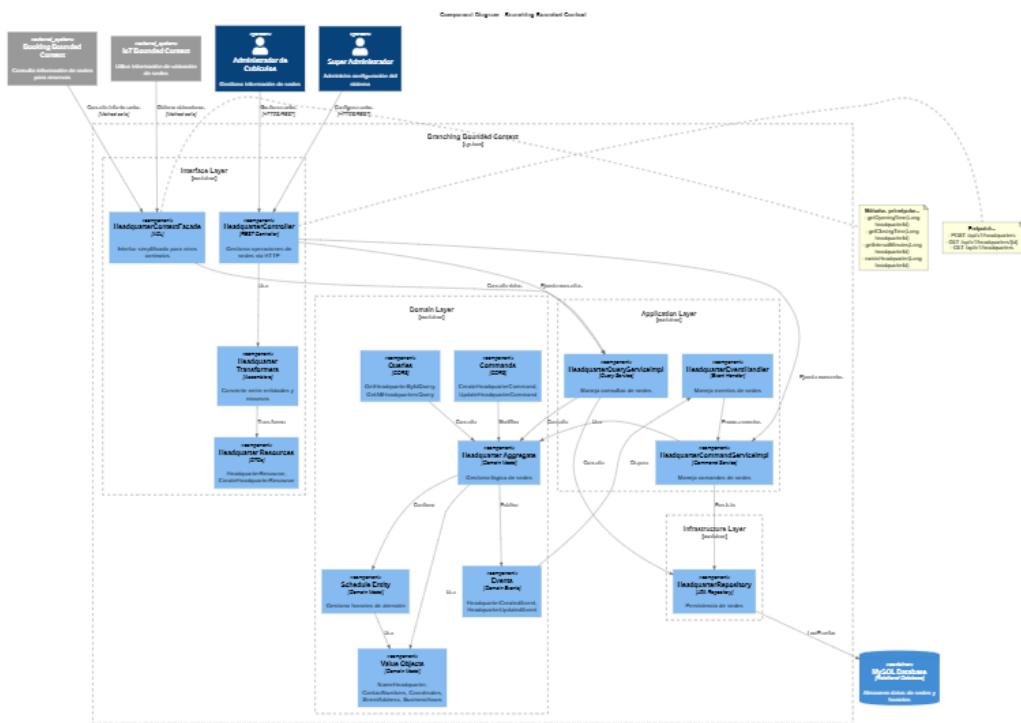
- Persistencia: El repositorio `HeadquarterRepository` proporciona acceso a los datos almacenados en la base de datos, permitiendo a las capas superiores (como la **Application Layer**) interactuar con las entidades del dominio.
- Validación: Los métodos `existsByName` y `existsByCoordinates` son utilizados para validar la unicidad de las sedes durante las operaciones de creación o actualización, asegurando la consistencia de los datos.

La **Infrastructure Layer** del Branching Bounded Context asegura que las operaciones relacionadas con la persistencia de datos sean robustas y confiables. Al proporcionar un repositorio especializado para las sedes, esta capa facilita la integración con la base de datos y garantiza que las reglas de negocio, como la unicidad de nombres y ubicaciones, se cumplan de manera eficiente. Esta estructura modular y reutilizable permite que el sistema sea escalable y fácil de mantener.

4.2.3.5. Bounded Context Software Architecture Component Level Diagrams

En esta sección se presenta el diagrama de componentes del **Branching Bounded Context**, el cual detalla los principales módulos y sus interacciones dentro del contexto delimitado. Este diagrama sigue el enfoque del C4 Model para representar los componentes clave, como servicios de aplicación, controladores, repositorios y servicios externos, junto con sus relaciones.

El propósito de este diagrama es proporcionar una visión clara y estructurada de cómo se organizan los componentes dentro del contexto, facilitando la comprensión de su arquitectura y permitiendo identificar puntos de integración y responsabilidades.



El **Branching Bounded Context** está compuesto por los siguientes módulos principales:

1. Application Layer:

- Coordina las operaciones de negocio relacionadas con la gestión de sedes.
- Incluye servicios de comandos y consultas que interactúan con la **Domain Layer** y la **Infrastructure Layer**.
- Maneja eventos relacionados con la creación y actualización de sedes.

2. Interface Layer:

- Expone los puntos de entrada al sistema a través de controladores REST.
- Incluye recursos y transformadores que aseguran una representación adecuada de los datos y su conversión entre las capas de la aplicación.

3. Domain Layer:

- Encapsula la lógica de negocio relacionada con la gestión de sedes.
- Define los agregados, entidades y objetos de valor que representan los conceptos clave del dominio.

4. Infrastructure Layer:

- Proporciona las implementaciones técnicas necesarias para soportar las operaciones del sistema.
- Incluye repositorios para la persistencia de datos y componentes que conectan la lógica de negocio con los recursos externos, como bases de datos.

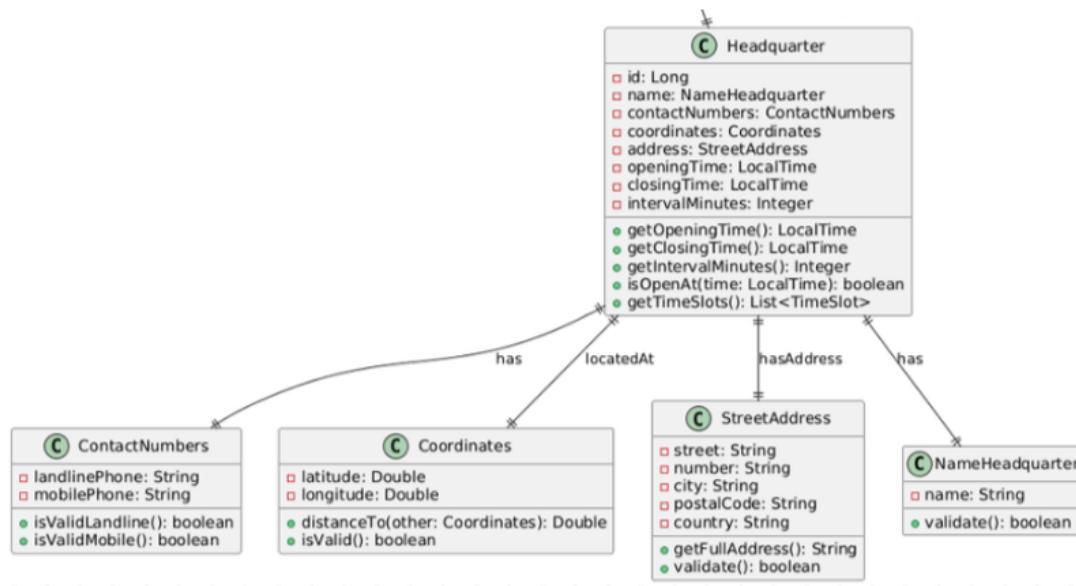
4.2.3.6. Bounded Context Software Architecture Code Level Diagrams

En este apartado se presentan los diagramas que ofrecen un mayor nivel de detalle sobre la implementación de los componentes del **Branching Bounded Context**. Estos diagramas están diseñados para ilustrar cómo se estructuran las clases, interfaces y relaciones dentro de las capas del contexto, proporcionando una visión técnica que facilita el desarrollo, mantenimiento y evolución del sistema.

4.2.3.6.1. Bounded Context Domain Layer Class Diagrams

El diagrama de clases correspondiente a la **Domain Layer** del **Branching Bounded Context** incluye las clases principales, como agregados, entidades y objetos de valor, así como las interfaces y enumeraciones que definen el comportamiento del dominio. También se destacan las relaciones entre estos elementos, como asociaciones, composiciones y dependencias.

El objetivo de este diagrama es proporcionar una representación detallada de la lógica de negocio encapsulada en la capa del dominio, asegurando que las reglas del negocio estén claramente definidas y alineadas con los requisitos del sistema.



Elementos principales del diagrama:

1. Aggregates:

- Headquarter**: Agregado principal que encapsula la lógica de negocio relacionada con la gestión de sedes. Incluye atributos como **name**, **contactNumbers**, **coordinates**, **schedule** y **address**.

2. Entities:

- Schedule**: Representa los horarios de atención de una sede, incluyendo atributos como **businessHours** e **intervalMinutes**.

3. Value Objects:

- NameHeadquarter**: Representa el nombre de una sede.
- ContactNumbers**: Representa los números de contacto de una sede.
- Coordinates**: Representa las coordenadas geográficas de una sede.
- StreetAddress**: Representa la dirección de una sede.
- BusinessHours**: Representa los horarios de apertura y cierre de una sede.

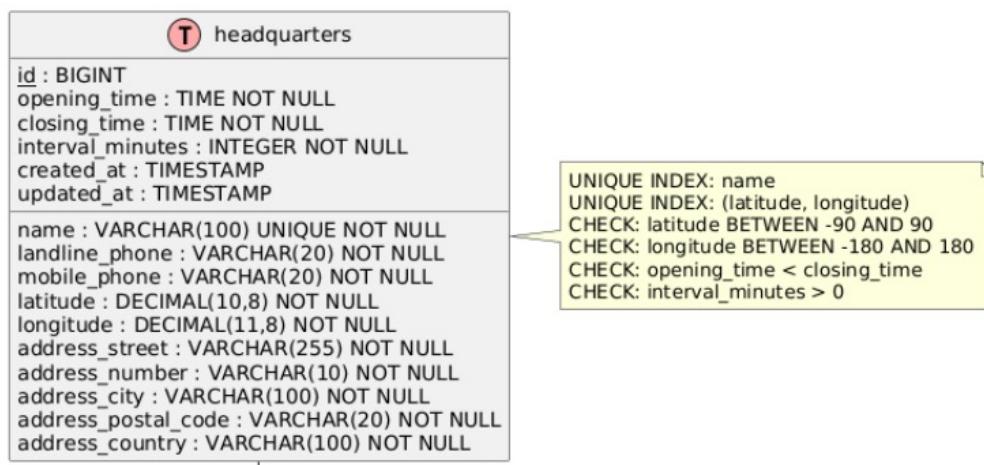
Relaciones destacadas:

- El agregado **Headquarter** actúa como el núcleo del dominio, gestionando las relaciones con los objetos de valor (**NameHeadquarter**, **ContactNumbers**, **Coordinates**, **StreetAddress**) y la entidad **Schedule**.
- Los objetos de valor encapsulan datos inmutables y validaciones específicas, asegurando consistencia en el dominio.
- La entidad **Schedule** permite modelar horarios complejos, incluyendo intervalos de tiempo entre reservas.

4.2.3.6.2. Bounded Context Database Design Diagram.

4.2.3.6.2. Bounded Context Database Design Diagram

El diseño de la base de datos para el **Branching Bounded Context** refleja la estructura del dominio, asegurando que las entidades y relaciones definidas en la **Domain Layer** se representen de manera eficiente en el modelo relacional.



Este diseño incluye las siguientes tablas principales:

1. Headquarters

- **Propósito:** Representa las sedes registradas en el sistema.
- **Atributos principales:**
 - `id`: Identificador único de la sede.
 - `name`: Nombre de la sede.
 - `landline_phone`: Número de teléfono fijo de la sede.
 - `mobile_phone`: Número de teléfono móvil de la sede.
 - `latitude`: Latitud de la ubicación de la sede.
 - `longitude`: Longitud de la ubicación de la sede.
 - `address_street`: Calle de la dirección de la sede.
 - `address_number`: Número de la dirección.
 - `address_city`: Ciudad donde se encuentra la sede.
 - `address_postal_code`: Código postal de la sede.
 - `address_country`: País donde se encuentra la sede.
 - `schedule_id`: Relación con la tabla `Schedules`.

2. Schedules

- **Propósito:** Representa los horarios de atención de las sedes.
- **Atributos principales:**
 - `id`: Identificador único del horario.
 - `opening_time`: Hora de apertura de la sede.
 - `closing_time`: Hora de cierre de la sede.
 - `interval_minutes`: Intervalo de tiempo en minutos entre reservas.

Este diseño asegura que las operaciones relacionadas con la gestión de sedes sean eficientes y consistentes, facilitando la integración con la lógica de negocio definida en la **Domain Layer**.

4.2.4 Bounded Context: Booking Bounded Context

El Booking Bounded Context es responsable de gestionar las reservas de cubículos en el sistema. Este contexto asegura que los estudiantes puedan realizar reservas de manera eficiente, gestionando la disponibilidad de cubículos, horarios y slots de tiempo. Además, permite a los administradores supervisar y gestionar las reservas activas, asegurando que las operaciones relacionadas con las reservas sean consistentes y cumplan con los estándares de calidad y precisión requeridos.

4.2.4.1. Domain Layer

La Domain Layer del Booking Bounded Context encapsula la lógica de negocio relacionada con la gestión de reservas de cubículos. En esta capa, se definen los elementos principales del dominio, como agregados, entidades, objetos de valor, comandos, consultas y eventos, que representan los conceptos clave del sistema.

Aggregates

1. Booking

- **Propósito:** Representa una reserva realizada por un grupo de estudiantes para un cubículo específico en una fecha y horario determinado.
- **Atributos:**
 - `userId`: Identificador del estudiante que genera la reserva, representado como un objeto de valor `UserId`.
 - `cubicleId`: Cubículo reservado, representado como una referencia a la entidad `Cubicle`.
 - `bookingDate`: Fecha de la reserva.
 - `bookingSlots`: Conjunto de intervalos de tiempo reservados, representados como una colección de entidades `BookingSlot`.
- **Características:**
 - Extiende `AuditableAbstractAggregateRoot`, lo que permite auditar cambios en las reservas.
 - Gestiona la relación entre el estudiante que genera la reserva, el cubículo y los slots reservados, asegurando consistencia y validación.

2. Cubicle

- **Propósito:** Representa un cubículo en una sede específica, incluyendo su disponibilidad y detalles.
- **Atributos:**
 - `cubicleDetails`: Detalles del cubículo, como número de cubículo y cantidad de asientos, representados como un objeto de valor `CubicleDetails`.
 - `headquarterId`: Identificador de la sede a la que pertenece el cubículo, representado como un objeto de valor `HeadquarterId`.
 - `status`: Estado actual del cubículo (`AVAILABLE`, `RESERVED`, `OCCUPIED`), representado como un objeto de valor `TableStatus`.
 - `availabilitySlots`: Conjunto de slots de disponibilidad generados para el cubículo, representados como una colección de entidades `AvailabilitySlot`.
- **Características:**
 - Extiende `AuditableAbstractAggregateRoot`, lo que permite auditar cambios en los cubículos.
 - Gestiona la generación y actualización de slots de disponibilidad.

Entities

1. AvailabilitySlot

- **Propósito:** Representa un intervalo de tiempo disponible para un cubículo en una fecha específica.

- **Atributos:**
 - `dateOfSlot`: Fecha del slot.
 - `timeInterval`: Intervalo de tiempo del slot, representado como un objeto de valor `TimeSlot`.
 - `status`: Estado del slot (`AVAILABLE`, `RESERVED`), representado como un objeto de valor `ScheduleSlotStatus`.
- **Métodos:**
 - `updateStatus(ScheduleSlotStatus status)`: Actualiza el estado del slot.

2. BookingSlot

- **Propósito:** Representa un intervalo de tiempo reservado por un estudiante o grupo de estudiantes.
- **Atributos:**
 - `timeInterval`: Intervalo de tiempo reservado, representado como un objeto de valor `TimeSlot`.

Value Objects

1. UserId

- **Propósito:** Representa el identificador único de un estudiante.
- **Validaciones:**
 - El identificador no puede ser negativo.

2. HeadquarterId

- **Propósito:** Representa el identificador único de una sede.
- **Validaciones:**
 - El identificador no puede ser nulo ni negativo.

3. CubicleDetails

- **Propósito:** Representa los detalles de un cubículo, como su número y cantidad de asientos.
- **Validaciones:**
 - El número de cubículo no puede ser negativo.
 - El número no puede ser negativo y la capacidad debe ser mayor a cero.

4. CubicleStatus

- **Propósito:** Enumera los estados posibles de un cubículo (`AVAILABLE`, `RESERVED`, `OCCUPIED`).

5. TimeSlot

- **Propósito:** Representa un intervalo de tiempo con una hora de inicio y una hora de fin.
- **Validaciones:**
 - La hora de inicio y la hora de fin no pueden ser nulas.
 - La hora de inicio debe ser anterior a la hora de fin.

6. ScheduleSlotStatus

- **Propósito:** Enumera los estados posibles de un slot de disponibilidad (`AVAILABLE`, `RESERVED`).

7. MaximumDuration

- **Propósito:** Representa la duración máxima permitida para una reserva.
- **Validaciones:**
 - La duración debe ser un número positivo.
 - La duración no puede exceder las 2 horas.

Commands

1. CreateBookingCommand

- **Propósito:** Representa la solicitud para crear una nueva reserva.
- **Atributos:**
 - `clientId`: Identificador del estudiante.
 - `cubicleId`: Identificador del cubículo.
 - `bookingDate`: Fecha de la reserva.
 - `slotIds`: Lista de identificadores de slots reservados.

2. CreateCubicleCommand

- **Propósito:** Representa la solicitud para crear un nuevo cubículo.
- **Atributos:**
 - `cubicleNumber`: Número del cubículo.
 - `seats`: Cantidad de asientos.
 - `headquartersId`: Identificador de la sede.

3. CreateTableScheduleCommand

- **Propósito:** Representa la solicitud para generar los slots de disponibilidad de un cubículo.
- **Atributos:**
 - `cubicleId`: Identificador del cubículo.

Queries

1. GetAllBookingsQuery

- **Propósito:** Recupera todas las reservas registradas en el sistema.

2. GetAllCubiclesQuery

- **Propósito:** Recupera todos los cubículos registradas en el sistema.

3. GetBookingByIdQuery

- **Propósito:** Recupera una reserva específica por su identificador.

4. GetCubicleByIdQuery

- **Propósito:** Recupera un cubículo específica por su identificador.

5. GetCubicleScheduleByIdAndDateQuery

- **Propósito:** Recupera los slots de disponibilidad de un cubículo para una fecha específica.

Events

1. SingleCubicleAvailabilitySlotsGeneratedEvent

- **Propósito:** Evento que se dispara cuando se generan los slots de disponibilidad para un cubículo.
- **Atributos:**
 - `tableId`: Identificador del cubículo.

Relaciones entre componentes

- El agregado `Booking` actúa como el núcleo del dominio, gestionando las relaciones con las entidades `BookingSlot` y `Cubicle`.
- El agregado `Cubicle` gestiona la generación y actualización de los slots de disponibilidad (`AvailabilitySlot`).
- Los objetos de valor encapsulan datos inmutables y validaciones específicas, asegurando consistencia en el dominio.
- Los comandos y consultas permiten interactuar con el sistema de manera estructurada, facilitando la creación de reservas y la recuperación de información.
- Los eventos aseguran que las operaciones críticas, como la generación de slots de disponibilidad, sean comunicadas de manera eficiente a otros componentes del sistema.

Esta estructura asegura que la lógica de negocio relacionada con la gestión de reservas sea robusta, consistente y fácil de mantener, cumpliendo con los requisitos del sistema.

4.2.4.2. Interface Layer

La **Interface Layer** del Booking Bounded Context expone los puntos de entrada al sistema a través de controladores REST. Esta capa permite la interacción con las entidades del dominio mediante solicitudes HTTP, facilitando la comunicación entre los clientes y el sistema. Además, incluye recursos y transformadores que aseguran una representación adecuada de los datos y su conversión entre las capas de la aplicación.

Controllers

Los controladores son responsables de manejar las solicitudes HTTP y delegar la lógica de negocio a los servicios correspondientes. A continuación, se describen los principales controladores:

1. BookingController

- **Propósito:** Gestiona las operaciones relacionadas con las reservas de cubículos.
- **Endpoints:**
 - `POST /api/v1/bookings`: Crea una nueva reserva.
 - `GET /api/v1/bookings/{id}`: Obtiene los detalles de una reserva específica por su ID.
 - `GET /api/v1/bookings`: Obtiene la lista de todas las reservas.
- **Dependencias:**
 - `BookingCommandService`: Servicio encargado de manejar los comandos relacionados con las reservas.
 - `BookingQueryService`: Servicio encargado de manejar las consultas relacionadas con las reservas.

2. CubicleController

- **Propósito:** Gestiona las operaciones relacionadas con los cubículos.
- **Endpoints:**
 - `POST /api/v1/cubicles`: Crea un nuevo cubículo.
 - `GET /api/v1/cubicles/{cubicleId}`: Obtiene los detalles de un cubículo específico por su ID.
 - `GET /api/v1/cubicles`: Obtiene la lista de todos los cubículos.
 - `GET /api/v1/cubicles/{cubicleId}/schedule`: Obtiene los slots de disponibilidad de un cubículo para una fecha específica.

- **Dependencias:**

- **CubicleCommandService:** Servicio encargado de manejar los comandos relacionados con los cubículos.
- **CubicleQueryService:** Servicio encargado de manejar las consultas relacionadas con los cubículos.

Resources

Los recursos representan los datos que se exponen a través de la API REST. Estos recursos son utilizados para estructurar las respuestas de los controladores y asegurar una representación clara y consistente de los datos. A continuación, se describen los principales recursos:

1. AvailabilitySlotResource

- **Propósito:** Representa un slot de disponibilidad de un cubículo.
- **Atributos:**
 - **id:** Identificador único del slot.
 - **date:** Fecha del slot.
 - **startTime:** Hora de inicio del slot.
 - **endTime:** Hora de fin del slot.
 - **status:** Estado del slot (**AVAILABLE**, **RESERVED**).

2. BookingResource

- **Propósito:** Representa una reserva en el sistema.
- **Atributos:**
 - **id:** Identificador único de la reserva.
 - **clientId:** Identificador del estudiante que realizó la reserva.
 - **cubicleId:** Identificador del cubículo reservado.
 - **bookingDate:** Fecha de la reserva.
 - **bookingSlots:** Lista de slots reservados.

3. BookingSlotResource

- **Propósito:** Representa un slot reservado por un estudiante.
- **Atributos:**
 - **startTime:** Hora de inicio del slot reservado.
 - **endTime:** Hora de fin del slot reservado.

4. CreateBookingResource

- **Propósito:** Representa los datos necesarios para crear una nueva reserva.
- **Atributos:**
 - **clientId:** Identificador del estudiante.
 - **cubicleId:** Identificador del cubículo.
 - **bookingDate:** Fecha de la reserva.
 - **slotIds:** Lista de identificadores de slots reservados.

5. CreateCubicleResource

- **Propósito:** Representa los datos necesarios para crear un nuevo cubículo.
- **Atributos:**
 - **headquarterId:** Identificador de la sede a la que pertenece el cubículo.
 - **cubicleNumber:** Número del cubículo.
 - **capacity:** Capacidad máxima del cubículo.

6. CubicleResource

- **Propósito:** Representa un cubículo en el sistema.
- **Atributos:**
 - **id:** Identificador único del cubículo.
 - **headquarterId:** Identificador de la sede a la que pertenece el cubículo.
 - **cubicleNumber:** Número del cubículo.
 - **capacity:** Capacidad máxima del cubículo.
 - **status:** Estado actual del cubículo (**AVAILABLE**, **RESERVED**, **OCCUPIED**, **MAINTENANCE**).

Transformers

Los transformadores son responsables de convertir las entidades del dominio en recursos y viceversa. Esto asegura que los datos expuestos a través de la API REST sean consistentes y estén en el formato esperado. A continuación, se describen los principales transformadores:

1. AvailabilitySlotResourceFromEntityAssembler

- **Propósito:** Convierte una entidad **AvailabilitySlot** en un recurso **AvailabilitySlotResource**.
- **Métodos principales:**
 - **toResourceFromEntity(AvailabilitySlot entity):** Transforma un slot de disponibilidad en un recurso.
 - **toResourceListFromEntities(List<AvailabilitySlot> entities):** Transforma una lista de slots de disponibilidad en una lista de recursos.

2. BookingResourceFromEntityAssembler

- **Propósito:** Convierte una entidad `Booking` en un recurso `BookingResource`.
- **Método principal:**
 - `toResourceFromEntity(Booking booking)`: Transforma una reserva en un recurso.

3. CreateBookingCommandFromResourceAssembler

- **Propósito:** Convierte un recurso `CreateBookingResource` en un comando `CreateBookingCommand`.
- **Método principal:**
 - `toCommandFromResource(CreateBookingResource resource)`: Transforma los datos de creación de una reserva en un comando.

4. CreateCubicleCommandFromResourceAssembler

- **Propósito:** Convierte un recurso `CreateCubicleResource` en un comando `CreateCubicleCommand`.
- **Método principal:**
 - `toCommandFromResource(CreateCubicleResource resource)`: Transforma los datos de creación de un cubículo en un comando.

5. CubicleResourceFromEntityAssembler

- **Propósito:** Convierte una entidad `Cubicle` en un recurso `CubicleResource`.
- **Método principal:**
 - `toResourceFromEntity(Cubicle entity)`: Transforma un cubículo en un recurso.

Relaciones entre componentes

- Los controladores (`BookingController`, `CubicleController`) utilizan los servicios de comandos y consultas para delegar la lógica de negocio.
- Los transformadores convierten las entidades del dominio en recursos para las respuestas HTTP y viceversa para las solicitudes entrantes.
- Los recursos estructuran los datos expuestos a los clientes, asegurando una representación clara y consistente.

Esta estructura asegura que la **Interface Layer** sea modular, reutilizable y fácil de mantener, facilitando la interacción entre los clientes y el sistema.

4.2.4.3. Application Layer

La **Application Layer** del Booking Bounded Context actúa como un intermediario entre la **Domain Layer** y las capas externas, como la **Interface Layer** y la **Infrastructure Layer**. Su propósito principal es coordinar las operaciones de negocio, manejar comandos y consultas, orquestar la lógica de aplicación y garantizar que las reglas del dominio se cumplan de manera consistente. Además, esta capa incluye manejadores de eventos y servicios externos (ACL) para interactuar con otros contextos delimitados.

Command Services

Los servicios de comandos son responsables de ejecutar operaciones que modifican el estado del sistema. A continuación, se describen los principales servicios de comandos:

1. BookingCommandServiceImpl

- **Propósito:** Gestiona las operaciones relacionadas con la creación de reservas de cubículos.
- **Métodos principales:**
 - `handle(CreateBookingCommand command)`: Crea una nueva reserva, validando la disponibilidad de los slots y asegurando que las reglas de negocio, como la duración máxima de la reserva y la capacidad mínima del cubículo, se cumplan.
- **Validaciones:**
 - Verifica que el estudiante exista utilizando el servicio externo `ExternalUserService`.
 - Valida que los slots solicitados estén disponibles y sean consecutivos.
 - Asegura que la duración total de la reserva no exceda las 2 horas.
 - Valida que la reserva incluya al menos 3 estudiantes.
- **Dependencias:**
 - `BookingRepository`: Persistencia de reservas.
 - `CubicleRepository`: Gestión de cubículos y sus slots de disponibilidad.
 - `ExternalUserService`: Verifica la existencia de los estudiantes.

2. CubicleCommandServiceImpl

- **Propósito:** Gestiona las operaciones relacionadas con la creación de cubículos y la generación de slots de disponibilidad.
- **Métodos principales:**
 - `handle(CreateCubicleCommand command)`: Crea un nuevo cubículo en una sede específica, validando que la sede exista y que no haya duplicados.
 - `handle(CreateCubicleScheduleCommand command)`: Genera los slots de disponibilidad para un cubículo, basándose en los horarios y el intervalo de la sede.
- **Validaciones:**
 - Verifica que la sede exista utilizando el servicio externo `ExternalHeadquarterService`.
 - Asegura que no existan cubículos duplicados en la misma sede.
- **Dependencias:**
 - `CubicleRepository`: Persistencia de cubículos y sus slots de disponibilidad.
 - `ExternalHeadquarterService`: Obtiene información de la sede, como horarios y intervalos.

Query Services

Los servicios de consultas son responsables de recuperar información del sistema sin modificar su estado. A continuación, se describen los principales servicios de consultas:

1. BookingQueryServiceImpl

- **Propósito:** Gestiona las consultas relacionadas con las reservas.
- **Métodos principales:**
 - `handle(GetBookingByIdQuery query)`: Recupera una reserva específica por su ID.
 - `handle(GetAllBookingsQuery query)`: Recupera todas las reservas registradas en el sistema.
- **Dependencias:**
 - `BookingRepository`: Persistencia de reservas.

2. CubicleQueryServiceImpl

- **Propósito:** Gestiona las consultas relacionadas con los cubículos y sus slots de disponibilidad.
- **Métodos principales:**
 - `handle(GetCubicleByIdQuery query)`: Recupera un cubículo específico por su ID.
 - `handle(GetAllCubiclesQuery query)`: Recupera todos los cubículos registrados en el sistema.
 - `handle(GetCubicleScheduleByIdAndDateQuery query)`: Recupera los slots de disponibilidad de un cubículo para una fecha específica.
- **Dependencias:**
 - `CubicleRepository`: Persistencia de cubículos y sus slots de disponibilidad.

Event Handlers

Los manejadores de eventos son responsables de reaccionar a eventos específicos del sistema. A continuación, se describe el principal manejador de eventos:

1. SingleCubicleAvailabilitySlotsGeneratedEventHandler

- **Propósito:** Maneja el evento `SingleCubicleAvailabilitySlotsGeneratedEvent`, que se dispara cuando se generan los slots de disponibilidad para un cubículo.
- **Método principal:**
 - `on(SingleCubicleAvailabilitySlotsGeneratedEvent event)`: Genera los slots de disponibilidad para un cubículo específico.
- **Dependencias:**
 - `CubicleCommandService`: Servicio encargado de manejar los comandos relacionados con los cubículos.

Outbound Services (ACL)

Los servicios externos proporcionan funcionalidades auxiliares que no forman parte del dominio principal. A continuación, se describen los principales servicios externos:

1. ExternalHeadquarterService

- **Propósito:** Interactúa con el Branching Bounded Context para obtener información de las sedes.
- **Métodos principales:**
 - `getHeadquarterOpeningTime(Long headquarterId)`: Obtiene la hora de apertura de una sede.
 - `getHeadquarterClosingTime(Long headquarterId)`: Obtiene la hora de cierre de una sede.
 - `getHeadquarterIntervalMinutes(Long headquarterId)`: Obtiene el intervalo de servicio en minutos de una sede.
 - `existsHeadquarter(Long headquarterId)`: Verifica si una sede existe.

2. ExternalUserService

- **Propósito:** Interactúa con el IAM Bounded Context para verificar la existencia de estudiantes.
- **Método principal:**
 - `existUserById(Long userId)`: Verifica si un estudiante existe en el sistema.

Relaciones entre componentes

- Los **Command Services** interactúan con los repositorios para modificar el estado del sistema y con los servicios externos (ACL) para validar información de otros contextos.
- Los **Query Services** interactúan únicamente con los repositorios para recuperar información del sistema.
- Los **Event Handlers** reaccionan a eventos del dominio para ejecutar lógica adicional, como la generación de slots de disponibilidad.
- Los **Outbound Services** (ACL) permiten la integración con otros contextos delimitados, como el Branching Bounded Context y el IAM Bounded Context.

La **Application Layer** del Booking Bounded Context asegura que las operaciones relacionadas con la gestión de reservas y cubículos sean robustas, consistentes y fáciles de mantener. Al coordinar la lógica de negocio, manejar eventos y facilitar la integración con otros contextos, esta capa garantiza que las reglas del dominio se cumplan de manera eficiente y que el sistema sea escalable y extensible.

4.2.4.4. Infrastructure Layer

La **Infrastructure Layer** del Booking Bounded Context proporciona las implementaciones técnicas necesarias para soportar las operaciones del sistema relacionadas con la gestión de reservas y cubículos. Esta capa incluye repositorios para la persistencia de datos y componentes que conectan la lógica de negocio con los recursos

externos, como bases de datos. Su objetivo principal es garantizar que las operaciones de almacenamiento y recuperación de información sean eficientes, consistentes y seguras.

Persistencia (JPA Repositories)

1. BookingRepository

- **Propósito:** Proporciona métodos para interactuar con la base de datos de reservas.
- **Características:**
 - Extiende `JpaRepository`, lo que permite realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre las entidades `Booking`.
 - Facilita la persistencia y recuperación de reservas, asegurando que las operaciones relacionadas con las reservas sean eficientes y confiables.

2. CubicleRepository

- **Propósito:** Proporciona métodos para interactuar con la base de datos de cubículos y sus slots de disponibilidad.
- **Métodos principales:**
 - `existsByHeadquarterIdAndCubicleDetails_CubicleNumber(HeadquarterId headquarterId, Integer cubicleDetails_cubicleNumber)`: Verifica si existe un cubículo con un número específico en una sede.
 - `findAvailabilitySlotsByCubicleIdAndDate(Long cubicleId, LocalDate date)`: Recupera los slots de disponibilidad de un cubículo para una fecha específica.
 - `findByIdWithSlotsForUpdate(Long id)`: Recupera un cubículo junto con sus slots de disponibilidad utilizando un bloqueo pesimista para evitar modificaciones concurrentes.
- **Características:**
 - Extiende `JpaRepository`, lo que permite realizar operaciones CRUD sobre las entidades `Cubicle`.
 - Incluye consultas personalizadas para manejar la relación entre cubículos y sus slots de disponibilidad.
 - Utiliza un bloqueo pesimista (`PESSIMISTIC_WRITE`) para garantizar la consistencia de los datos durante las operaciones críticas, como la creación de reservas.

Relaciones entre componentes

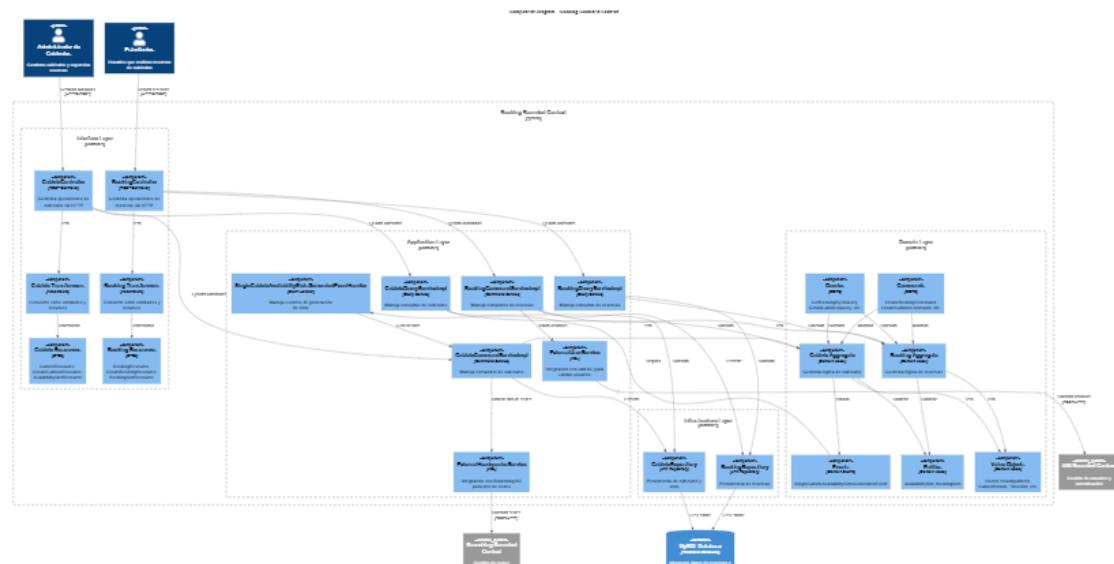
- **Persistencia:** Los repositorios `BookingRepository` y `CubicleRepository` proporcionan acceso a los datos almacenados en la base de datos, permitiendo a las capas superiores (como la **Application Layer**) interactuar con las entidades del dominio.
- **Validación:** Los métodos personalizados en `CubicleRepository` son utilizados para validar la existencia de cubículos y recuperar información específica, como los slots de disponibilidad, asegurando la consistencia de los datos durante las operaciones de negocio.

La **Infrastructure Layer** del Booking Bounded Context asegura que las operaciones relacionadas con la persistencia de datos sean robustas y confiables. Al proporcionar repositorios especializados para las reservas y cubículos, esta capa facilita la integración con la base de datos y garantiza que las reglas de negocio, como la validación de slots de disponibilidad y la unicidad de cubículos, se cumplan de manera eficiente. Esta estructura modular y reutilizable permite que el sistema sea escalable y fácil de mantener.

4.2.4.5. Bounded Context Software Architecture Component Level Diagrams

En esta sección se presenta el diagrama de componentes del **Booking Bounded Context**, el cual detalla los principales módulos y sus interacciones dentro del contexto delimitado. Este diagrama sigue el enfoque del C4 Model para representar los componentes clave, como servicios de aplicación, controladores, repositorios y servicios externos, junto con sus relaciones.

El propósito de este diagrama es proporcionar una visión clara y estructurada de cómo se organizan los componentes dentro del contexto, facilitando la comprensión de su arquitectura y permitiendo identificar puntos de integración y responsabilidades.



El **Booking Bounded Context** está compuesto por los siguientes módulos principales:

1. Application Layer:

- Coordina las operaciones de negocio relacionadas con la gestión de reservas y cubículos.
- Incluye servicios de comandos y consultas que interactúan con la **Domain Layer** y la **Infrastructure Layer**.
- Maneja eventos relacionados con la creación de reservas y la generación de slots de disponibilidad.

2. Interface Layer:

- Expone los puntos de entrada al sistema a través de controladores REST.
- Incluye recursos y transformadores que aseguran una representación adecuada de los datos y su conversión entre las capas de la aplicación.

3. Domain Layer:

- Encapsula la lógica de negocio relacionada con la gestión de reservas y cubículos.
- Define los agregados, entidades y objetos de valor que representan los conceptos clave del dominio.

4. Infrastructure Layer:

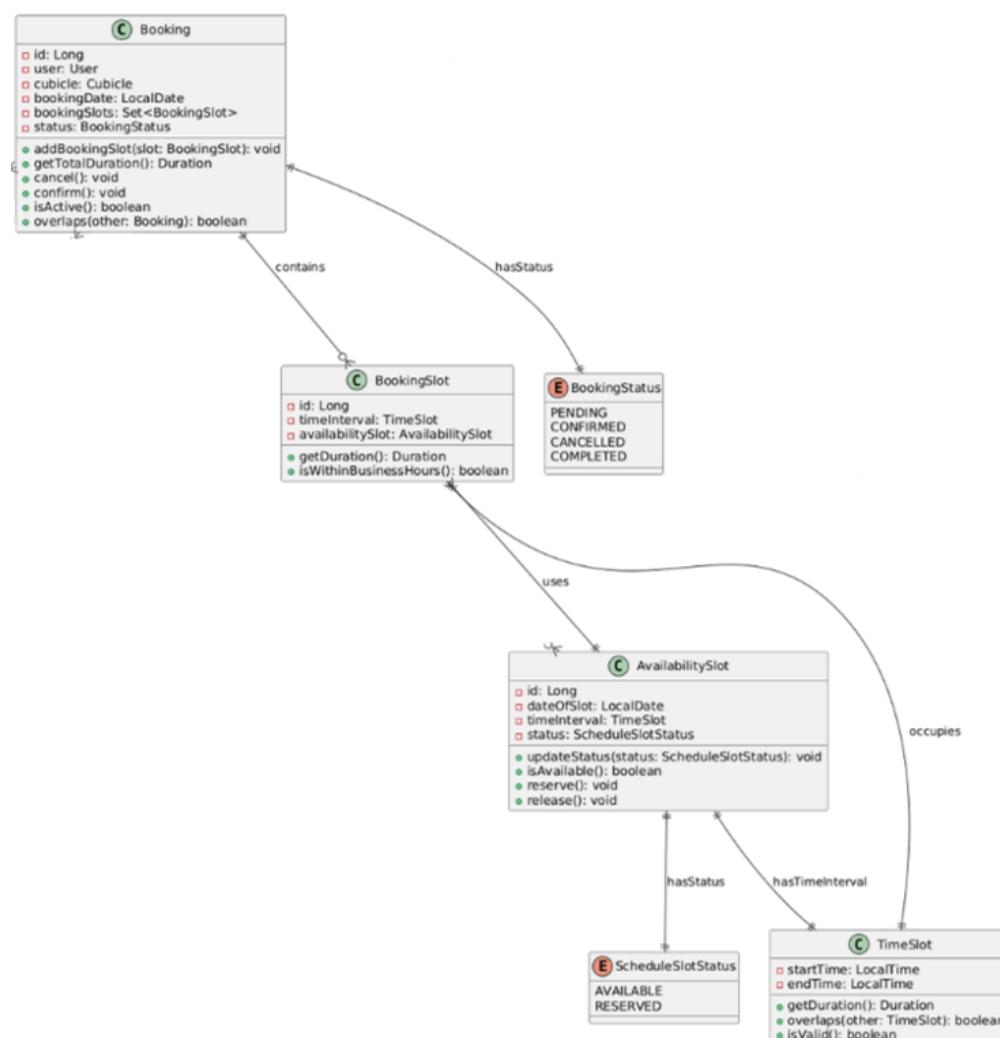
- Proporciona las implementaciones técnicas necesarias para soportar las operaciones del sistema.
- Incluye repositorios para la persistencia de datos y componentes que conectan la lógica de negocio con los recursos externos, como bases de datos.

4.2.4.6. Bounded Context Software Architecture Code Level Diagrams

En este apartado se presentan los diagramas que ofrecen un mayor nivel de detalle sobre la implementación de los componentes del **Booking Bounded Context**. Estos diagramas están diseñados para ilustrar cómo se estructuran las clases, interfaces y relaciones dentro de las capas del contexto, proporcionando una visión técnica que facilita el desarrollo, mantenimiento y evolución del sistema.

4.2.4.6.1. Bounded Context Domain Layer Class Diagrams

El diagrama de clases correspondiente a la **Domain Layer** del **Booking Bounded Context** incluye las clases principales, como agregados, entidades y objetos de valor, así como las interfaces y enumeraciones que definen el comportamiento del dominio. También se destacan las relaciones entre estos elementos, como asociaciones, composiciones y dependencias.



Elementos principales del diagrama:

1. Aggregates:

- **Booking:** Agregado principal que encapsula la lógica de negocio relacionada con las reservas. Incluye atributos como `userId`, `cubicleId`, `bookingDate` y `bookingSlots`.
- **Cubicle:** Agregado que representa un cubículo en una sede específica, incluyendo su disponibilidad y detalles.

2. Entities:

- **AvailabilitySlot:** Representa un intervalo de tiempo disponible para un cubículo en una fecha específica.
- **BookingSlot:** Representa un intervalo de tiempo reservado por un estudiante.

3. Value Objects:

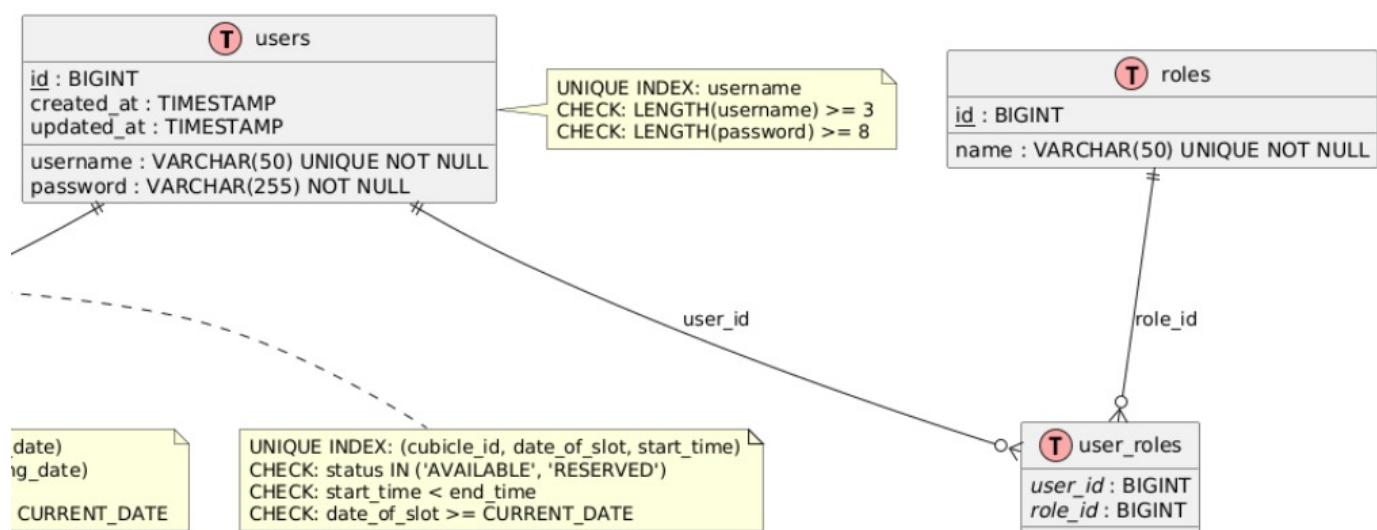
- **UserId:** Representa el identificador único de un estudiante.
- **HeadquarterId:** Representa el identificador único de una sede.
- **CubicleDetails:** Representa los detalles de un cubículo, como su número y cantidad de asientos.
- **TimeSlot:** Representa un intervalo de tiempo con una hora de inicio y una hora de fin.
- **CubicleStatus:** Enumera los estados posibles de un cubículo (`AVAILABLE`, `RESERVED`, `OCCUPIED`, `MAINTENANCE`).
- **ScheduleSlotStatus:** Enumera los estados posibles de un slot de disponibilidad (`AVAILABLE`, `RESERVED`).

Relaciones destacadas:

- El agregado `Booking` gestiona las relaciones con las entidades `BookingSlot` y `Cubicle`.
- El agregado `Cubicle` gestiona la generación y actualización de los slots de disponibilidad (`AvailabilitySlot`).
- Los objetos de valor encapsulan datos inmutables y validaciones específicas, asegurando consistencia en el dominio.

4.2.4.6.2. Bounded Context Database Design Diagram.

El diseño de la base de datos para el **Booking Bounded Context** refleja la estructura del dominio, asegurando que las entidades y relaciones definidas en la **Domain Layer** se representen de manera eficiente en el modelo relacional.



Este diseño incluye las siguientes tablas principales:

1. Bookings:

- Representa las reservas realizadas por los estudiantes.
- Atributos principales:
 - `id`: Identificador único de la reserva.
 - `user_id`: Identificador del estudiante que realizó la reserva.
 - `cubicle_id`: Identificador del cubículo reservado.
 - `booking_date`: Fecha de la reserva.

2. Cubicles:

- Representa los cubículos disponibles en las sedes.
- Atributos principales:
 - `id`: Identificador único del cubículo.
 - `headquarter_id`: Identificador de la sede a la que pertenece el cubículo.
 - `cubicle_number`: Número del cubículo.
 - `seats`: Cantidad de asientos.
 - `status`: Estado actual del cubículo.

3. AvailabilitySlots:

- Representa los slots de disponibilidad de los cubículos.

- Atributos principales:
 - **`id`**: Identificador único del slot.
 - **`cubicle_id`**: Identificador del cubículo asociado.
 - **`date_of_slot`**: Fecha del slot.
 - **`start_time`**: Hora de inicio del slot.
 - **`end_time`**: Hora de fin del slot.
 - **`status`**: Estado del slot (**AVAILABLE**, **RESERVED**).

4. BookingSlots:

- Representa los slots reservados por los estudiantes.
- Atributos principales:
 - **`id`**: Identificador único del slot reservado.
 - **`booking_id`**: Identificador de la reserva asociada.
 - **`start_time`**: Hora de inicio del slot reservado.
 - **`end_time`**: Hora de fin del slot reservado.

4.2.5. Bounded Context: IoT Device Monitoring

El **IoT Monitoring Bounded Context** es responsable de gestionar la interacción con los dispositivos IoT instalados en los cubículos de estudio, como los sensores de peso en los asientos. Este contexto se encarga de recopilar, procesar e interpretar los datos enviados por estos dispositivos para determinar el estado de disponibilidad de los cubículos en tiempo real. Además, proporciona mecanismos para la configuración y monitoreo de los dispositivos, asegurando su correcto funcionamiento y la precisión de los datos recopilados.

Capítulo V: Solution UI/UX Design

5.1. Style Guidelines

En este apartado se establecen los principios de interacción y visuales que regulan la experiencia completa en la plataforma Bib Flip, garantizando así una coherencia entre las distintas plataformas. Utilizando la tipografía, los colores, el espaciado, la iconografía y un tono de comunicación unificado, establecemos una identidad visual definida.

5.1.1. General Style Guidelines

Branding

Para representar nuestro producto Bibflip, se desarrolló un logo que fusiona elementos clave de nuestra propuesta de valor: un libro abierto que simboliza el conocimiento y los espacios académicos, junto con una estrella de cuatro puntas en su interior. Esta combinación visual transmite de forma inmediata la naturaleza del producto como una plataforma de gestión de reservas de cubículos estudiantiles con un enfoque innovador y accesible.

El libro, elemento universal asociado al estudio, la biblioteca y el aprendizaje, se convierte en un símbolo dinámico gracias a la estrella que lo corona. Esta estrella comunica la idea de excelencia, destacar y facilitar la experiencia del usuario. Juntos, estos elementos representan a Bibflip como la solución que ilumina el camino hacia espacios de estudio organizados y fácilmente reservables.



Variantes del Logo

- **Versión Light**: Fondo gris claro (#D3D3D3) con icono en negro, transmitiendo claridad y accesibilidad
- **Versión Primary**: Fondo en tonalidad naranja/marrón cálido (#D4844A), reflejando calidez académica y profesionalismo
- **Versión Dark**: Fondo oscuro (#2A2A2A) con icono en blanco cremoso (#F5F5DC), para contextos digitales premium

Aplicación

El logo de Bibflip representa la modernidad aplicada a la gestión educativa: simplificando la reserva de espacios de estudio y permitiendo que los estudiantes se enfoquen en lo que realmente importa: su aprendizaje.

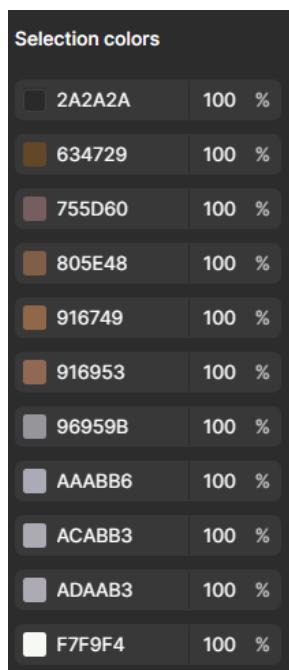
Tipografía

- Fuente: Inter

- Tamaños: Seleccionar entre los tamaños disponibles para asegurar una legibilidad adecuada en diferentes dispositivos y tamaños de pantalla.
- Espaciado entre letras y líneas: Ajustar según sea necesario para mejorar la legibilidad, especialmente en textos largos.



Paleta de colores



Colores Principales

Nombre	Código Hex	Uso	Opacidad
Dark Charcoal	#2A2A2A	Textos principales, encabezados	100%
Brown Primary	#634729	Botones principales, elementos interactivos	100%
Mauve	#755D60	Acentos secundarios, bordes	100%

Nombre	Código Hex	Uso	Opacidad
Brown Secondary	#805E48	Hover states, variaciones de botones	100%
Warm Tan	#916749	Estados activos, elementos destacados	100%
Light Tan	#916953	Fondos claros, backgrounds secundarios	100%

Colores Neutros

Nombre	Código Hex	Uso	Opacidad
Dark Gray	#96959B	Textos secundarios, placeholders	100%
Blanco	#AABB66	Fondos alternativos, spaces	100%
Light Gray	#ACABBB3	Bordes, divisores	100%
Very Light Gray	#ADAAB3	Fondos muy claros	100%
Off-White	#F7F9F4	Fondos principales, espacios limpios	100%

5.1.2. Web, Mobile and IoT Style Guidelines

Estados y Componentes

Botones y Acciones:

- Color base: #634729 (Brown Primary)
- Hover: #805E48 (Brown Secondary)
- Activo: #916749 (Warm Tan)

Tags/Badges (Disponible):

- Background: #916953 (Light Tan)
- Texto: Blanco (#FFFFFF)

Fondos y Containers:

- Fondo principal: #F7F9F4 (Off-White)
- Cards/Containers: Blanco (#FFFFFF)
- Bordes: #96959B (Dark Gray)

Significado y Uso

La paleta de **tonos tierra y marrón** transmite calidez, confianza y profesionalismo, perfecta para la temática de biblioteca y espacios compartidos. Los tonos neutros proporcionan equilibrio visual y facilitan la legibilidad. Los acentos en colores tierra crean jerarquía visual sin ser intrusivos.

Ejemplo de Interfaz

Bibflip - San Miguel Cubículos:

- Navbarra: Blanco (#FFFFFF)
- Headers: Dark Charcoal (#2A2A2A)
- Botones "Reservar": #634729 (Brown Primary)
- Tags "Disponible": #916953 (Light Tan) con texto blanco
- Fondos de página: #F7F9F4 (Off-White)
- Cards de cubículos: Blanco (#FFFFFF) con bordes sutiles

[← Volver a sedes](#)

San Miguel - Cubiculos

⌚ La Marina 2810, Lima, 1508, Perú

Filtros

Capacidad:

Todas las capacidades

Limpiar filtros

Cubiculo 1

Disponible

Capacidad: 5 personas

Reservar

Cubiculo 2

Disponible

Capacidad: 4 personas

Reservar

Cubiculo 3

Disponible

Capacidad: 5 personas

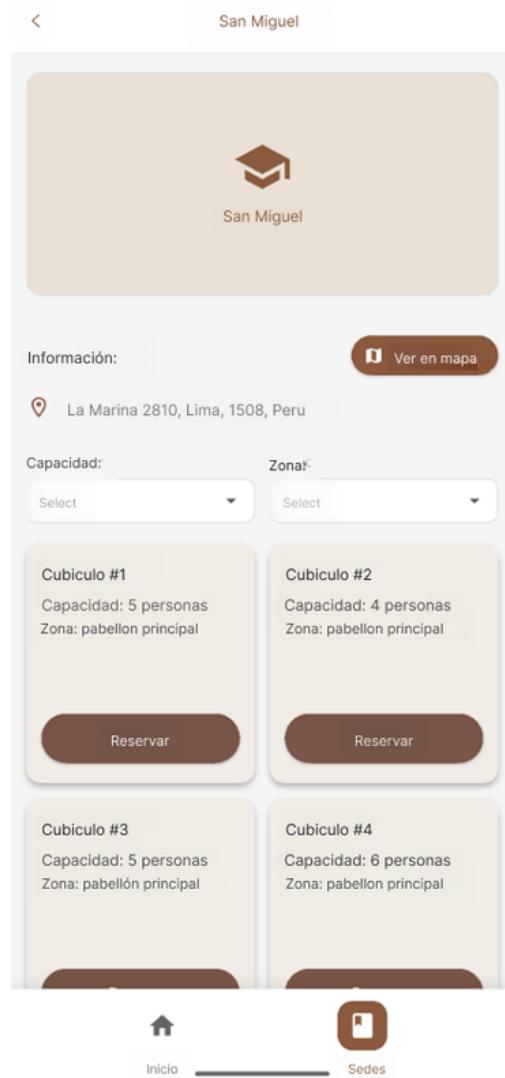
Reservar

Cubiculo 4

Disponible

Capacidad: 6 personas

Reservar



5.2. Information Architecture

5.2.1. Organization Systems

Para Bibflip, hemos implementado sistemas de organización que facilitan la navegación y acceso a la información tanto para estudiantes como para administradores. Estos sistemas están diseñados para proporcionar una experiencia de usuario intuitiva y eficiente.

Sistema jerárquico

Organizamos la información en Bibflip siguiendo una estructura jerárquica clara:

Para estudiantes:

El proceso comienza en la Landing Page, donde los usuarios son redirigidos a la aplicación web o móvil según su preferencia. Una vez allí, los estudiantes pueden registrarse o iniciar sesión con sus credenciales, accediendo a un panel principal que les muestra información relevante como reservas activas, sedes disponibles y servicios de la biblioteca, incluyendo información sobre cubículos y salas grupales. Para realizar una reserva, los estudiantes seleccionan una sede, visualizan los cubículos disponibles en tiempo real y completan un formulario con los detalles de su reserva, como fecha, hora aproximada de estadía y cubículo específico. Además, pueden ubicar la sede en un mapa integrado para mayor comodidad.

Para administradores:

Por otro lado, los administradores acceden a un panel de gestión después de iniciar sesión. Desde allí, pueden administrar los cubículos de las sedes, agregando nuevos o eliminando los existentes mediante un formulario sencillo. También tienen acceso a información detallada de las sedes y pueden visualizar las reservas de diferentes maneras: por cubículo, por estudiante o en una lista completa. Esto les permite mantener un control eficiente sobre las operaciones del centro de estudio.

Para superadministradores:

Los superadministradores tienen acceso completo al sistema. Pueden registrar nuevos administradores asignándoles sedes específicas, crear nuevas sedes en el sistema, y visualizar toda la información consolidada de usuarios, sedes y reservas para realizar pruebas y monitoreo general del sistema.

Sistema secuencial

También implementamos una organización secuencial para procesos específicos:

Proceso de reserva:

1. Selección de sede
2. Visualización de cubículos disponibles en tiempo real
3. Selección de cubículo
4. Selección de hora aproximada de estadía
5. Confirmación de reserva

Proceso de gestión de cubículos:

1. Visualización del inventario actual de cubículos
2. Selección de acción (agregar/eliminar)
3. Introducción de datos del cubículo (identificador, capacidad)
4. Confirmación de la operación

Proceso de registro de administrador:

1. Acceso del superadmin al módulo de administradores
2. Selección de "Aregar administrador"
3. Ingreso de credenciales del nuevo administrador
4. Asignación de sede específica
5. Confirmación del registro

5.2.2. Labeling Systems

Antes de implementar las etiquetas en nuestra plataforma, determinamos qué requisitos necesitamos cumplir. Las etiquetas nos permiten agregar información contextual a diferentes elementos de la plataforma Bibflip. A continuación, se detalla el sistema de etiquetado implementado:

Etiquetas para cubículos

Etiqueta	Descripción
[DISPONIBLE]	Cubículo que puede ser reservado en este momento
[OCUPADO]	Cubículo actualmente en uso por estudiantes
[RESERVADO]	Cubículo con reserva activa pendiente de ocupación

Etiquetas para reservas

Etiqueta	Descripción
[ACTIVA]	Reserva en curso (dentro del rango horario seleccionado)
[EXPIRADA]	Reserva que ya ha finalizado su período de vigencia
[PENDIENTE]	Reserva programada para uso futuro

Etiquetas para servicios de biblioteca

Etiqueta	Descripción
[CUBÍCULOS]	Espacios individuales de estudio
[SALAS GRUPALES]	Espacios para trabajo colaborativo
[COMPUTADORAS]	Equipos de cómputo disponibles

Etiquetas para nivel de ocupación

Etiqueta	Descripción
[ALTA DISPONIBILIDAD]	Más del 50% de cubículos disponibles
[DISPONIBILIDAD MEDIA]	Entre 20% y 50% de cubículos disponibles
[BAJA DISPONIBILIDAD]	Menos del 20% de cubículos disponibles
[COMPLETO]	Sin cubículos disponibles

Etiquetas para notificaciones de sistema

Etiqueta	Descripción
[ÉXITO]	Confirmación de operación completada exitosamente
[ERROR]	Problema durante una operación
[ADVERTENCIA]	Mensaje de precaución o limitación del sistema

Etiqueta	Descripción
[ACTUALIZACIÓN]	Cambio en disponibilidad detectado por sensores

Este sistema de etiquetas proporciona contexto visual inmediato y mejora la comprensión de la información presentada tanto para estudiantes como para administradores.

5.2.3. SEO Tags and Meta Tags

La implementación adecuada de etiquetas SEO es crucial para mejorar la visibilidad de Bibflip en los motores de búsqueda.

Título

Indica el tema de la página, debe ser corto y descriptivo, manteniéndose entre 55-60 caracteres.

```
<title>Bibflip - Sistema de Gestión de Cubículos para Bibliotecas Universitarias</title>
```

Descripción

Breve descripción del propósito de la plataforma.

```
<meta name="description" content="Bibflip optimiza la gestión de espacios en bibliotecas universitarias con tecnología IoT, visualización en tiempo real de cubículos disponibles y reservas digitales. ¡Mejora la experiencia de estudio!" />
```

Robots

Indican a los motores de búsqueda qué hacer con la página.

```
<meta name="robots" content="index, follow">
```

Tipo de contenido

Útil para que los motores de búsqueda identifiquen el idioma de la página.

```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<meta http-equiv="Content-Language" content="es">
```

Viewport Meta Tag

Crucial para asegurar que el contenido se vea bien en dispositivos móviles.

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Open Graph Tags

Para mejorar la apariencia cuando se comparte en redes sociales.

```
<meta property="og:title" content="Bibflip - Gestión Inteligente para Bibliotecas Universitarias">
<meta property="og:description" content="Sistema IoT para optimizar la ocupación de cubículos en centros de estudio">
<meta property="og:image" content="https://bibflip.com/images/bibflip-preview.jpg">
<meta property="og:url" content="https://bibflip.com">
<meta property="og:type" content="website">
```

Keywords

Aunque tienen menor importancia actualmente, aún pueden ser útiles.

```
<meta name="keywords" content="reservas bibliotecas, cubículos disponibles, IoT bibliotecas, gestión centros estudio, sistema reservas tiempo real, ocupación cubículos, bibliotecas universitarias, espacios estudio">
```

5.2.4. Searching Systems

El sistema de búsqueda propuesto para Bibflip permitirá a los usuarios encontrar fácilmente información relevante según sus necesidades específicas.

Sistema de búsqueda para estudiantes

Nombre del filtro	Descripción
Ubicación	El usuario podrá filtrar bibliotecas por distrito o zona de la ciudad
Disponibilidad inmediata	Muestra sólo las bibliotecas con cubículos disponibles en el momento actual
Capacidad	Permite filtrar por tipo de espacio (individual, grupal, con computadora)
Horario de atención	Filtrar por bibliotecas abiertas en un rango horario específico
Distancia	Ordena los resultados desde la ubicación actual del usuario
Nivel de ocupación	Filtra sedes por porcentaje de disponibilidad (alta, media, baja)

Sistema de búsqueda para administradores

Nombre del filtro	Descripción
Búsqueda por número de cubículo	Permite encontrar rápidamente información de un cubículo específico
Búsqueda por nombre de estudiante	Localiza reservas realizadas por un estudiante específico
Filtro por fecha	Busca reservas en un rango de fechas específico
Estado de cubículo	Filtrar por estado: disponible, ocupado, reservado
Historial de reservas	Busca en el histórico de reservas completadas

Sistema de búsqueda para superadministradores

Nombre del filtro	Descripción
Búsqueda de administradores	Localiza administradores por nombre o sede asignada
Búsqueda de sedes	Encuentra sedes específicas en el sistema
Visualización global	Accede a estadísticas consolidadas de todas las sedes

Características adicionales del sistema de búsqueda

- Búsqueda en tiempo real:** Los resultados se actualizan automáticamente cada 5 segundos mediante datos de sensores IoT.
- Historial de búsquedas:** Mantiene un registro de búsquedas recientes del usuario.
- Sugerencias automáticas:** Ofrece resultados mientras el usuario escribe.
- Filtros combinados:** Permite aplicar múltiples filtros simultáneamente.

Este sistema de búsqueda está diseñado para ser intuitivo y proporcionar resultados precisos, mejorando así la experiencia del usuario y la eficiencia operativa de la plataforma Bibflip.

5.2.5. Navigation Systems

El sistema de navegación de la plataforma Bibflip guiará a los usuarios a través de las distintas secciones y pantallas, proporcionándoles acceso intuitivo a todas las funcionalidades. A continuación, se detallan los elementos que facilitan la navegación de los usuarios:

Navegación global

Nombre	Descripción
Inicio	Página principal que muestra un resumen de la información relevante según el tipo de usuario
Perfil	Permite al usuario gestionar su información personal y preferencias
Notificaciones	Alertas sobre reservas, actualizaciones de disponibilidad o mensajes del sistema

Navegación para estudiantes

Nombre	Descripción
Explorar sedes	Muestra todas las bibliotecas disponibles con información sobre ubicación y disponibilidad en tiempo real
Mi reserva	Permite al usuario ver su reserva activa con detalles de horario, sede y número de cubículo
Sedes	Visualización de todas las sedes del centro de estudio con opción de ver en mapa
Ver servicios	Acceso al reglamento y servicios del centro de estudio (cubículos, salas grupales, etc.)
Mapa	Visualización geográfica de las sedes en un mapa interactivo

Navegación para administradores

Nombre	Descripción
Dashboard	Panel principal con indicadores clave sobre la operación de la sede asignada
Gestión de cubículos	Interfaz para agregar, eliminar o modificar cubículos en la sede asignada
Visualizar reservas	Visualización y gestión de las reservas actuales y futuras con filtros por estudiante o cubículo
Mi sede	Información detallada de la sede asignada al administrador

Navegación para superadministradores

Nombre	Descripción
Panel completo	Acceso a todas las funcionalidades del sistema
Administradores	Módulo para registrar y gestionar administradores con asignación de sedes
Gestión de sedes	Crear, editar y visualizar todas las sedes del sistema
Usuarios	Visualización completa de estudiantes, administradores y sus actividades

Elementos de navegación

- **Menú principal:** Barra de navegación fija en la parte superior que contiene los enlaces principales según el rol del usuario.
- **Menú lateral desplegable:** Para acceso a funciones secundarias, configuraciones y cierre de sesión.
- **Botones de acción flotantes:** Para acciones principales como "Realizar reserva" o "Aregar cubículo".
- **Breadcrumbs:** Indicadores de ubicación dentro de la jerarquía de navegación.
- **Gestos táctiles:** En la versión móvil, se implementan gestos como deslizar para ver más opciones o pellizcar para ampliar el mapa.

Patrones de navegación

- **Hub and spoke:** La pantalla principal actúa como centro desde donde se accede a las diferentes funcionalidades específicas del rol.
- **Navegación por pestanas:** Para alternar entre diferentes vistas de una misma categoría (ej: cubículos disponibles vs. ocupados).
- **Navegación jerárquica:** Para profundizar en detalles (sede → cubículos → reservas → detalles de estudiante).
- **Desplazamiento infinito:** En listados de sedes o cubículos para cargar más resultados al desplazarse.
- **Navegación contextual:** Botones y acciones que aparecen según el estado del sistema (ej: "Reservar" solo si hay disponibilidad).

Características de accesibilidad

- **Indicadores visuales claros:** Uso de colores y etiquetas para mostrar estados de disponibilidad.
- **Actualización automática:** Los indicadores de disponibilidad se actualizan en menos de 5 segundos según datos de sensores.
- **Prevención de errores:** El sistema impide reservar si ya existe una reserva activa del estudiante.
- **Confirmaciones:** Mensajes de confirmación para acciones críticas como eliminar cubículos o cancelar reservas.

Este sistema de navegación está diseñado para ser consistente, intuitivo y adaptable a diferentes dispositivos, asegurando que los usuarios puedan encontrar rápidamente la información o funcionalidad que necesitan sin esfuerzo excesivo. La estructura se adapta dinámicamente según el rol del usuario (estudiante, administrador o superadministrador), mostrando únicamente las opciones relevantes para cada perfil.

5.3. Landing Page UI Design

5.3.1. Landing Page Wireframe

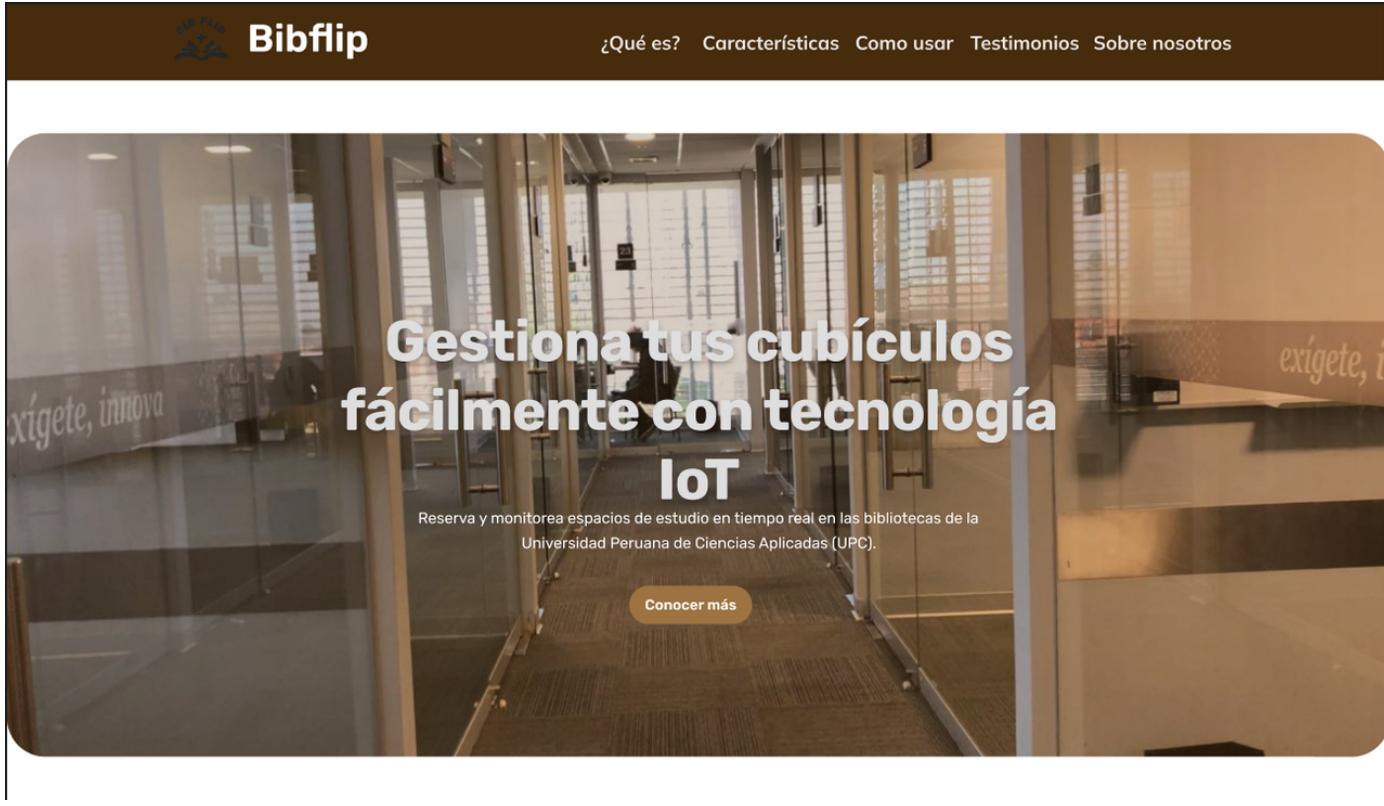
En esta sección se presenta el wireframe de la landing page de Bibflip, una representación estructural inicial que organiza los principales bloques de contenido del sitio sin incluir aún detalles gráficos. Su propósito es definir la jerarquía informativa y el flujo de navegación que guiará al usuario durante su experiencia en la página. Este esquema permite visualizar la distribución de secciones como el encabezado, la introducción al proyecto, las características principales, los beneficios y la sección de contacto. El diseño se centra en la claridad y coherencia de la información, garantizando que el visitante pueda comprender rápidamente el objetivo de Bibflip: ofrecer una plataforma que optimiza la reserva de cubículos mediante tecnología IoT dentro de la Universidad Peruana de Ciencias Aplicadas (UPC).

El wireframe fue elaborado en Figma y define la estructura inicial de la landing page.

[Ver Wireframe en Figma](#)

5.3.2. Landing Page Mock-up

El mockup de la landing page de Bibflip muestra una versión visual refinada y cercana al diseño final. En esta etapa se aplicó una tipografía legible y una composición equilibrada. Este diseño busca comunicar el propósito central del proyecto: mejorar la experiencia de los estudiantes en la gestión de espacios de estudio mediante una interfaz moderna, intuitiva y atractiva. Las secciones del mockup incluyen una presentación clara de la solución propuesta, los beneficios de Bibflip y un llamado a la acción enfocado en conocer más sobre la aplicación. Con ello se refleja una propuesta visual profesional y funcional, alineada con los valores de innovación y comunidad universitaria.



The screenshot shows the Bibflip website homepage. At the top, there's a dark header bar with the Bibflip logo (a blue stylized book icon) and the word "Bibflip" in white. To the right of the logo are navigation links: "¿Qué es?", "Características", "Como usar", "Testimonios", and "Sobre nosotros". Below the header is a large banner image of a modern library interior with glass partitions and study carrels. Overlaid on the banner is the main headline: "Gestiona tus cubículos fácilmente con tecnología IoT". Below this, a sub-headline reads: "Reserva y monitorea espacios de estudio en tiempo real en las bibliotecas de la Universidad Peruana de Ciencias Aplicadas (UPC)". A "Conocer más" button is visible at the bottom of the banner.

¿Qué es Bibflip?

Bibflip es una solución tecnológica desarrollada para optimizar la experiencia de los estudiantes de la UPC al momento de usar cubículos de estudio. Integra sensores IoT y una plataforma digital que permite visualizar la disponibilidad de espacios en tiempo real y realizar reservas de manera sencilla y rápida.



Encuentra rápido

Reduce el tiempo que pasas buscando estacionamientos.



Gestión eficiente

Facilita la gestión y el monitoreo para el personal de biblioteca.



Mejor experiencia

Optimiza la experiencia de estudio de los alumnos de la UPC.

Características clave



Detección automática de ocupación mediante sensores IoT.



Reservas digitales desde la web o aplicación móvil.



Mapas interactivos por sede UPC.



Panel administrativo para personal de biblioteca.



Notificaciones automáticas y recordatorios para los estudiantes.

Fácil y simple de usar

Bibflip fue diseñado pensando en la comodidad del estudiante UPC. Con solo unos clics podrás reservar tu espacio ideal.



Revisa la disponibilidad

Consulta qué cubículos están libres.



Reserva tu espacio

Selecciona la hora y sede que prefieras.



Estudia sin preocupaciones

Disfruta de un ambiente adecuado y gestionado en tiempo real.

Con la confianza de estudiantes y bibliotecarios



"Recomiendo Bibflip a todos mis compañeros, es muy fácil de usar y realmente útil."

— Diego Ramos, Estudiante de Arquitectura, UPC Villa



"Como bibliotecario, puedo gestionar los cubículos de forma más eficiente y rápida."

— Oscar Caceres, Estudiante de Ingeniería, UPC San Miguel



"La reserva online me permite organizar mi grupo de estudio sin complicaciones."

— Piero Landa, Estudiante de Ingeniería, UPC Monterrico



Conoce al equipo de Bibflip

Somos estudiantes de Ingeniería de Software de la Universidad Peruana de Ciencias Aplicadas (UPC) comprometidos con mejorar la experiencia académica mediante soluciones innovadoras.



Mauricio Oliveira
Ingeniero de Software



Oscar Aranda Vallejos
Ingeniero de Software



André Bernaola Pérez
Ingeniero de Software



José Eduardo Gutiérrez García
Ingeniero de Software



Joaquín Pedraza Maldonado
Ingeniero de Software



Diego Soriano Medrano
Ingeniero de Software



Piero Velarde Luyo
Ingeniero de Software



Bibflip

©2025 Bibflip. Todos los derechos reservados

El mockup fue elaborado en Figma y presenta la propuesta visual final antes de la implementación.

[Ver Mock-up en Figma](#)

5.4. Applications UX/UI Design

En esta sección se documenta el diseño de la aplicación de Bib Flip, la cual cuenta con su versión web y móvil. La versión web contará con interfaces y páginas personalizadas para cada tipo de usuario (Administrador, Supervisor y Estudiante); sin embargo, la aplicación móvil solo contará con pantallas para los estudiantes. Además, la adición de esta versión de la aplicación para dispositivos móviles es necesaria porque permite cumplir con los objetivos del proyecto y con las historias de usuario de los sprints correspondientes.

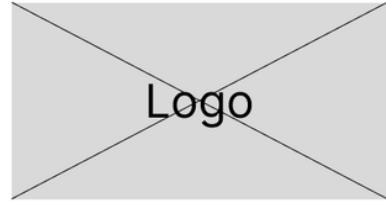
5.4.1. Applications Wireframes

En esta sección se presentan los wireframes de la aplicación Bibflip, tanto para la versión web como para la versión móvil. Los wireframes son esquemas visuales de baja fidelidad que muestran la estructura básica de las pantallas, la disposición de los elementos y la jerarquía de la información, sin incluir detalles gráficos finales. Su objetivo es definir la organización de los contenidos y la navegación entre las diferentes secciones, permitiendo validar la lógica de interacción antes de avanzar a etapas de diseño visual más detalladas.

Web Application Wireframes

Los wireframes de la aplicación web contemplan las vistas principales para cada tipo de usuario: Administrador, Supervisor y Estudiante. Incluyen pantallas como el panel principal (dashboard), gestión de cubículos, visualización de reservas, perfil de usuario y navegación entre sedes. Cada wireframe muestra la ubicación de menús, botones de acción, tablas de información y formularios, asegurando una experiencia de usuario clara y eficiente para cada rol sin entrar demasiado en detalle. Para poder visualizar el conjunto de wireframes se proporciona el siguiente enlace: [Enlace-Web-Application-Wireframes-Figma](#)

Pantallas Generales



Iniciar Sesión

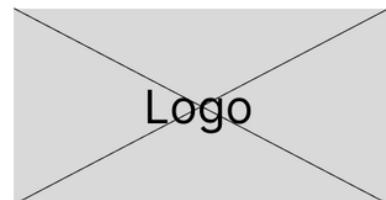
Usuario

Contraseña

Iniciar sesión

¿No tienes una cuenta todavía? [Crear cuenta](#)

Los 3 tipos de usuario podrán iniciar sesión ingresando su usuario y contraseña previamente registrados.



Registrarse

Usuario

Contraseña

Confirmar Contraseña

Registrarse

¿Ya tienes una cuenta? [Iniciar sesión](#)

Al ser esta una pantalla totalmente pública y accedida por cualquier usuario de internet, los administradores y supervisores no podrán registrarse de esta manera por razones de seguridad. De esta manera se evita que cualquier agente externo pueda crearse una cuenta de administrador y sabotear el sistema. Por lo tanto, esta pantalla de registro tendrá la funcionalidad de registrar nuevos estudiantes.

Pantallas para Administrador

The screenshot shows the Bibflip Admin Dashboard. At the top, there is a header bar with the Bibflip Logo, a back arrow, and navigation links for Inicio, Dashboard, Usuario, and another back arrow. Below the header is a large central panel titled "Panel Administrativo" with the subtitle "Tu centro de gestión de creación de sedes y supervisores". This panel contains two sections: "Estadísticas" and "Acciones Rápidas". The "Estadísticas" section displays two items: "Sedes registradas" (with an icon of a building) and "Supervisores activos" (with an icon of a person). The "Acciones Rápidas" section contains two buttons: "Gestionar Sedes" and "Gestionar Supervisores".

Este es el menú inicial del administrador que observa tras iniciar sesión.

[Inicio](#)[Reservar](#)[Usuario](#)[Sedes](#)[Supervisores](#)

Sedes Disponibles

[+ Nueva Sede](#)[Buscar Sedes](#)

San Miguel

Activa

This large rectangular area represents the main content of the page, showing a placeholder for the list of available sites.

Al presionar el botón "Gestionar Sedes", el administrador es dirigido a esta pantalla donde podrá agregar una sede nueva.

[Inicio](#)[Reservar](#)[Usuario](#)

Bibflip

Gestión de Sedes

[Sedes](#)[Supervisores](#)

Sedes Disponibles

[+ Nueva Sede](#)[Buscar Sedes](#)

San Miguel

Activa

Registro de Supervisores

Crear una nueva cuenta de supervisor

[Registrar Supervisor](#)

SUPERVISOR

Al presionar el botón de supervisores, el administrador podrá crearle una cuenta al supervisor y asignarle una sede.

The screenshot shows the Bibflip software interface. At the top, there is a navigation bar with four buttons: 'Inicio' (Home), 'Reservar' (Reserve), 'Usuario' (User), and a back arrow. Below the navigation bar, the main title 'Bibflip' and subtitle 'Gestión de Sedes' (Site Management) are displayed. To the right of the title are two buttons: 'Sedes' (Sites) and 'Supervisores' (Supervisors). The 'Supervisores' button is highlighted with a dark background. The main content area is divided into two sections. On the left, under 'Sedes Disponibles' (Available Sites), there is a '+ Nueva Sede' (New Site) button and a 'Buscar Sedes' (Search Site) input field. Below these are the site names 'San Miguel' and 'Activa'. On the right, under 'Supervisores', it shows 'San Miguel' assigned as a 'SUPERVISOR Supervisor Activo' (Active Supervisor). A note 'Máximo 1 supervisor por sede' (Maximum 1 supervisor per site) is displayed above the supervisor entry. There is also a trash can icon next to the supervisor entry.

En esta pantalla el administrador podrá ver al supervisor por sede asignado.

Pantallas para Supervisor

The screenshot shows a top navigation bar with five items: 'Inicio' (dark grey), 'Gestión de Cubículos' (light grey), 'Ver Reservas' (dark grey), 'Usuario' (light grey), and a back arrow icon. Below this is a large grey rounded rectangle containing the title 'Panel de Supervisor' and the subtitle 'Tu centro de gestión de reservas y cubículos'. To the left of this is a sidebar with the heading 'Sede Asignada' and the location 'San Miguel'. Under 'San Miguel', there are four radio buttons with the following options: 'La Marina 2810, Lima, 1508, Perú', '6303333', '962222408', and '06:00 - 23:00'. To the right of the sidebar is a section titled 'Acciones Rápidas' with two dark grey buttons labeled 'Gestionar Reservas' and 'Gestionar Cubículos'.

Este es el menú inicial que verá el supervisor tras iniciar sesión en la aplicación web.

The screenshot shows a top navigation bar identical to the previous one. Below it is a section titled 'Reservas de la Sede' with the subtitle 'Administra y visualiza las reservas en tu sede asignada'. Underneath this are two buttons: 'Por Estudiante' (dark grey) and 'Por Cubículo' (light grey). A large rounded rectangle below contains a header 'Estefano' and a list of reservation details: 'Cubículo: 1', 'Capacidad: 5 personas', 'Hora: 08:00 - 08:30', and 'Fecha: 2025-10-06'.

En este apartado el supervisor podrá visualizar las reservas por estudiante.



Reservas de la Sede

Administra y visualiza las reservas en tu sede asignada

Por Estudiante

Por Cubículo

Cubículo 1

Estudiante: Estefano
Hora: 08:00 - 08:30
Fecha: 2025-10-06

En este apartado el supervisor podrá visualizar las reservas por cubículo.

[Inicio](#)[Gestión de Cubículos](#)[Ver Reservas](#)[Usuario](#)

Gestión de Cubículos

Administra los cubículos disponibles en tu sede asignada

[Agregar Cubículo](#)

Cubiculo 1 Capacidad: 5 personas Reservar	Cubiculo 2 Capacidad: 5 personas Eliminar	Cubiculo 3 Capacidad: 6 personas Reservar
Cubiculo 4 Capacidad: 4 personas Eliminar	Cubiculo 5 Capacidad: 4 personas Eliminar	Cubiculo 6 Capacidad: 6 personas Eliminar

En esta pantalla podrá gestionar los cubículos, es decir agregarlos, editarlos o eliminarlos.

Pantallas para estudiante

[Inicio](#)[Reservar](#)[Usuario](#)

Mis Reservas

BJ Lunes, 6 de octubre de 2025

EB Cubículo 1

Sede: San Miguel

08:00 - 08:30

[Ver](#)[Eliminar](#)

Este será el menú inicial que verá el estudiante tras iniciar sesión. En este apartado podrá ver sus reservas más recientes y cancelarlas.

Detalles de la Reserva ×

Fecha y Hora

Fecha:

Hora:

Ubicación

Sede:

Cubículo:

Información Adicional

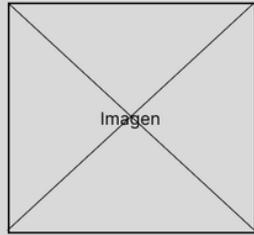
ID de Reserva:

Cerrar Cancelar Reserva

Al seleccionar la opción de ver una reserva, el estudiante podrá ver estos detalles.

[Inicio](#)[Reservar](#)[Usuario](#)

Conoce nuestras sedes y reserva!



San Miguel

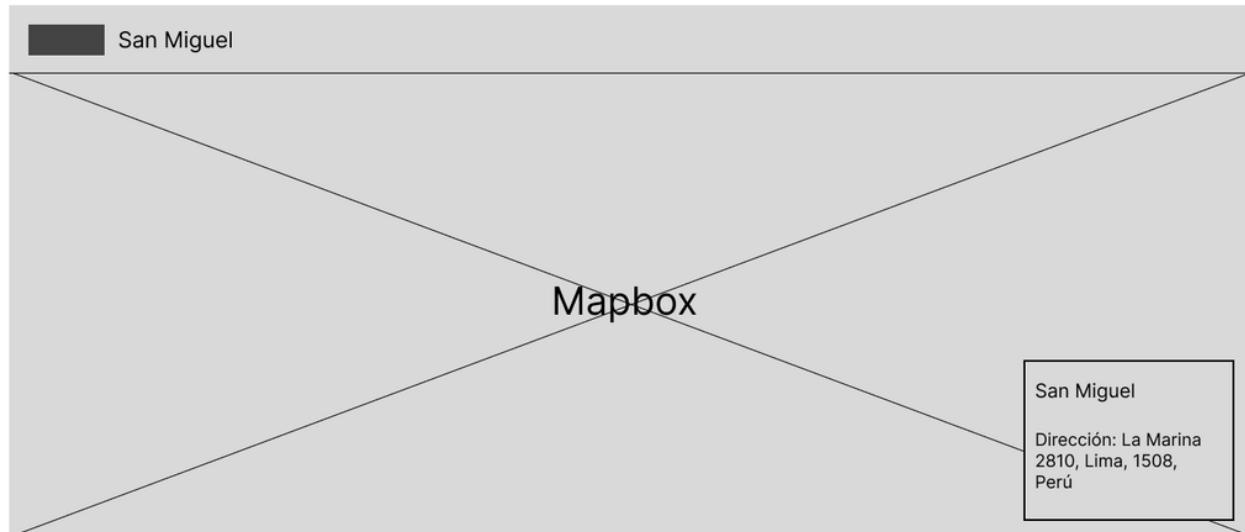
Dirección: La Marina 2810, Lima, 1508, Perú

Horario:

- Lunes - Viernes: 9:00 am - 8 pm
- Sábado: 10:00 am - 6 pm
- Domingo: 10:00 am - 5 pm

[Ver cubículos](#)[Ver en mapa](#)

Al seleccionar el botón del navbar que dice "Reservar", el estudiante verá este apartado. En esta página tendrá la opción de dirigirse a ver el mapa de la sede o continuar a la reserva de cubículos.

[Inicio](#)[Reservar](#)[Usuario](#)

Este apartado muestra el mapa de la sede para que nuevos estudiantes puedan ubicarse mejor.

[Inicio](#)[Reservar](#)[Usuario](#)[Volver a sedes](#)

San Miguel - Cubiculos

Filtros

Capacidad:

Todas las capacidades ~~

[Limpiar filtros](#)

Cubiculo 1

Capacidad: 5 personas

[Reservar](#)

Cubiculo 2

Capacidad: 5 personas

[Reservar](#)

Cubiculo 3

Capacidad: 6 personas

[Reservar](#)

Cubiculo 4

Capacidad: 4 personas

[Reservar](#)

Cubiculo 5

Capacidad: 4 personas

[Reservar](#)

Cubiculo 6

Capacidad: 6 personas

[Reservar](#)

Este apartado muestra los cubículos disponibles para reservar.

Reservar Cubiculo

Cubiculo: Cubiculo 1

Capacidad: 5 personas

Fecha:

06/10/2025

Horarios disponibles

Cancelar **Confirmar**

Al querer reservar un cubículo, el estudiante podrá ver los horarios disponibles y efectuar la reserva.

Mobile Application Wireframes

Los wireframes de la aplicación móvil están enfocados en la experiencia del estudiante. Incluyen pantallas como inicio de sesión, panel principal, exploración de sedes, visualización de cubículos disponibles, detalles de reserva, historial de reservas y perfil. El diseño prioriza la simplicidad y la facilidad de uso en dispositivos móviles, con menús accesibles, botones grandes y navegación intuitiva adaptada a pantallas táctiles. Para poder visualizar el conjunto de wireframes se proporciona el siguiente enlace: [Enlace-Mobile-Application-Wireframes-Figma](#)



¡Bienvenido!

Iniciar Sesión

Usuario

Contraseña

Iniciar Sesión

¿No tienes cuenta? [Regístrate](#)

En este apartado el inicio de sesión es exclusivo para estudiantes.



¡Bienvenido!

Regístrate

Usuario

Contraseña

Confirmar contraseña

Registrarse

¿Ya tienes cuenta? [Inicia sesión](#)

Esta imagen expresa la pantalla de registro para la aplicación móvil con los mismos parámetros que la aplicación web. El registro por este medio seguirá siendo igualmente solo para estudiantes.

Bib-Flip

Inicio

¡Bienvenido!



Nuestras sedes

Ver todas



Inicio

Sedes

Esta será la pantalla inicial que verá el estudiante tras iniciar sesión. En ella podrá darle click a la sede San Miguel para acceder a los cubículos. Además, podrá seleccionar la tarjeta de reservas activas para ver todas sus reservas.

Sedes

San Miguel

Horario: 06:00 - 23:00

Dirección: La Marina 2810, Lima, 1508, Perú

Mesas disponibles: 6

[Inicio](#)

[Sedes](#)

Este será un apartado secundario para acceder a los cubículos de la sede San Miguel.

<

San Miguel

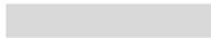
San Miguel

Información:

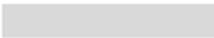
[Ver en mapa](#)

© La Marina 2810, Lima, 1508, Peru

Capacidad



Zona:



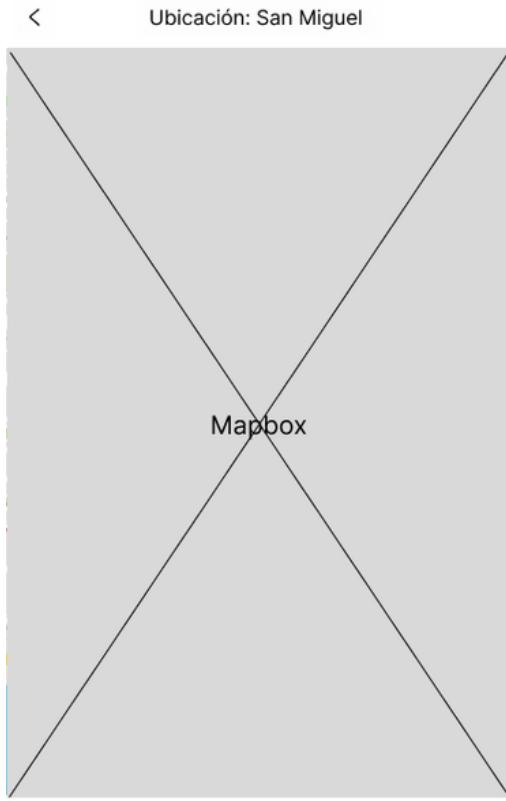
Cubiculo #1

Capacidad: 5 personas
Zona: pabellon principal[Reservar](#)

Cubiculo #2

Capacidad: 4 personas
Zona: pabellon principal[Reservar](#)[Inicio](#)[Sedes](#)

Este apartado se utilizará para todos los cubículos disponibles en la sede San Miguel.



San Miguel

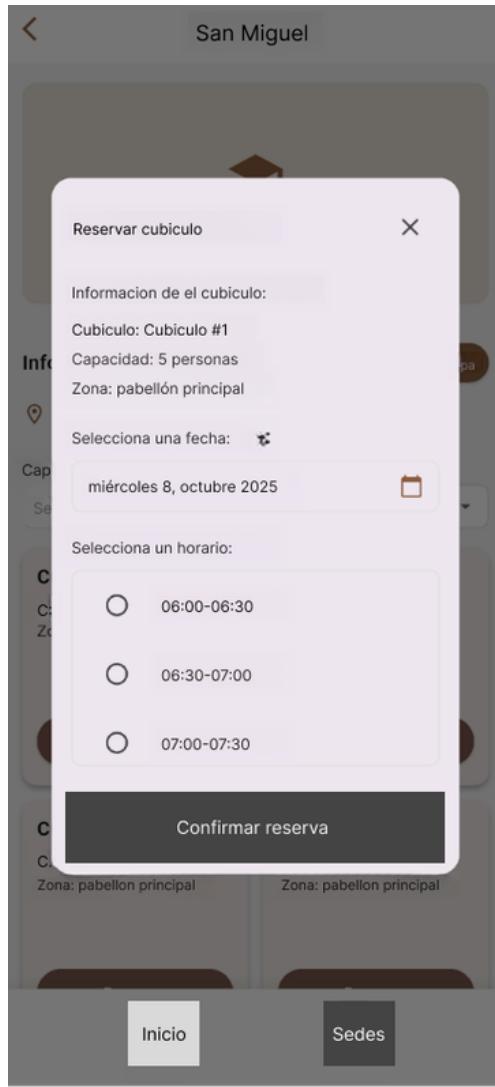
La Marina 2810, Lima, 1508, Perú

Cómo llegar

Inicio

Sedes

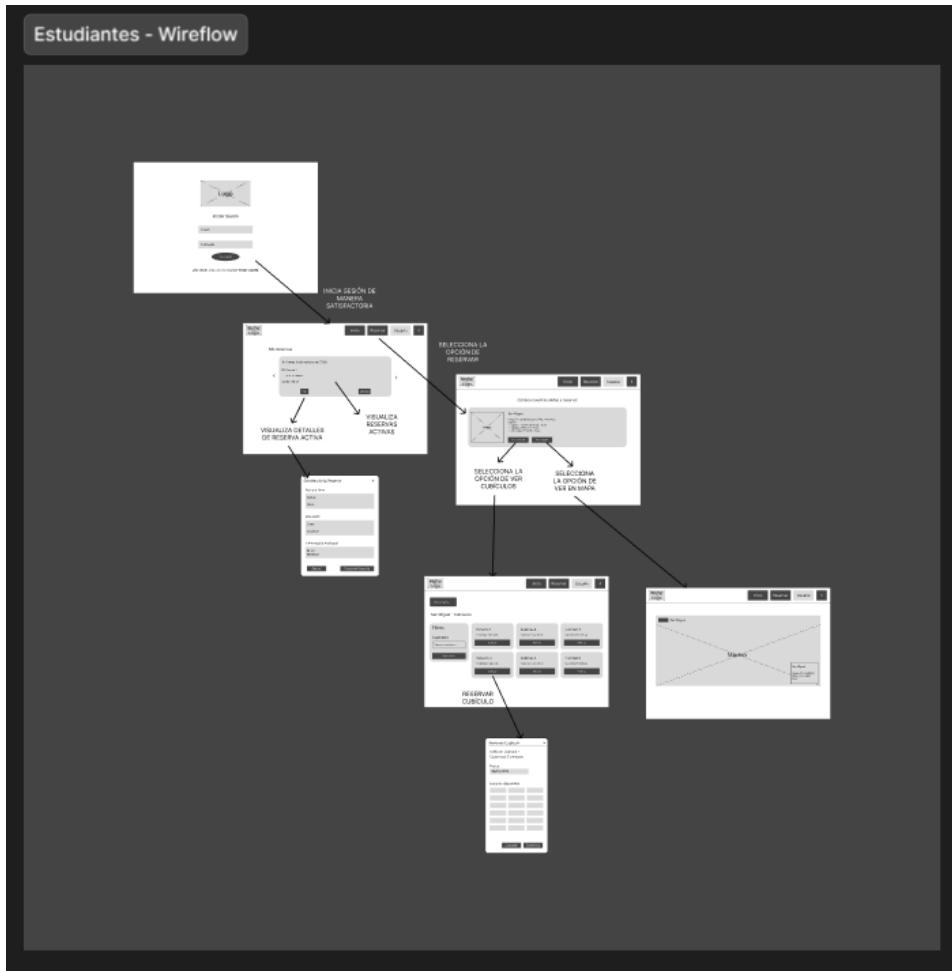
Esta pantalla se utilizará para visualizar el mapa de la sede San Miguel.



Esta pantalla se utilizará para reservar cubículos en un horario disponible.

5.4.2. Applications Wireflow Diagrams

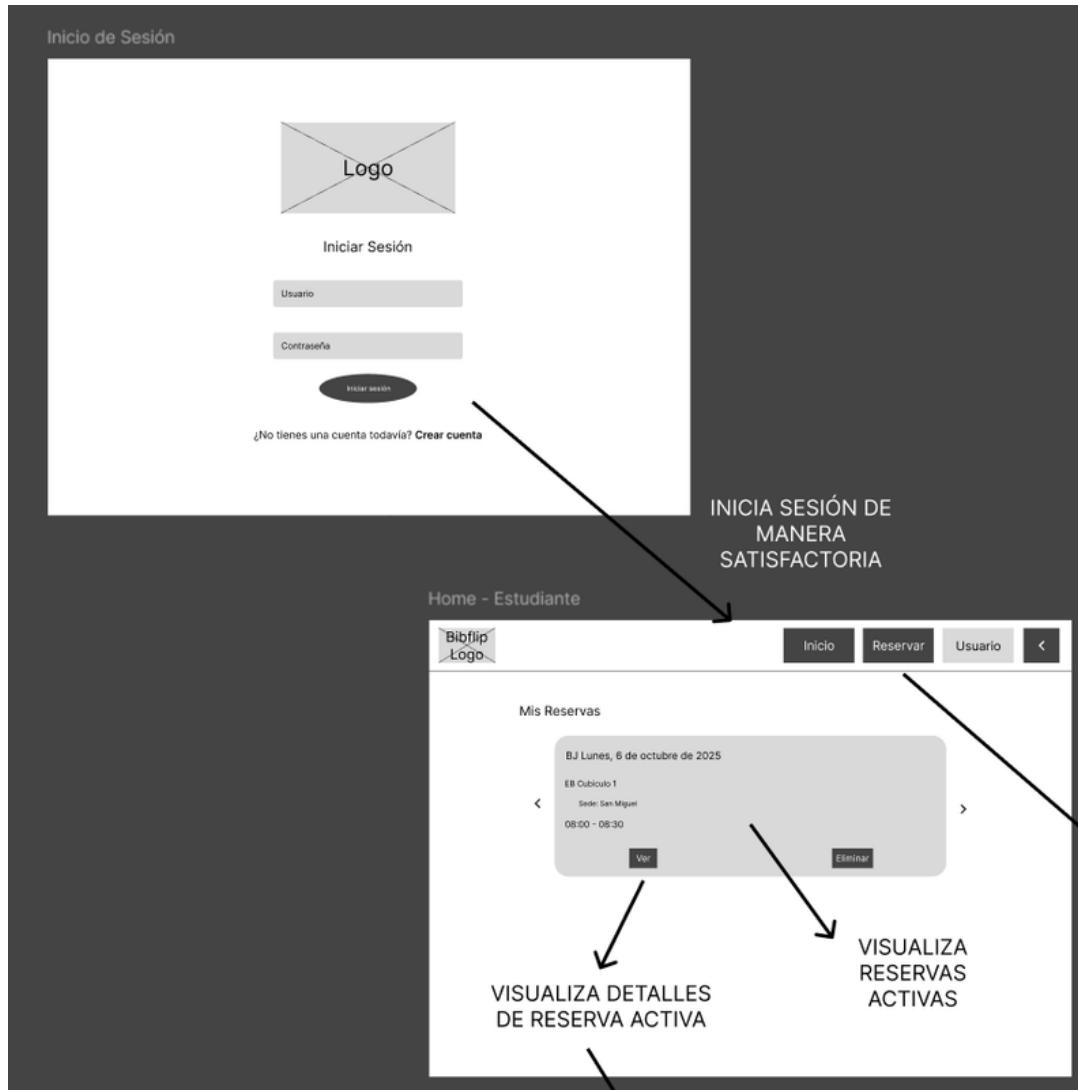
En esta sección se presentan los wireflows de la aplicación, que combinan wireframes con diagramas de flujo para ilustrar cómo los usuarios navegan entre las diferentes pantallas y funcionalidades. Los wireflows permiten visualizar los posibles caminos de interacción, las decisiones del usuario y las respuestas del sistema, facilitando la identificación de mejoras en la experiencia de usuario y la detección de posibles bloqueos o redundancias en la navegación.



El enlace de los web application wireflow diagrams es el siguiente: <https://www.figma.com/design/36XnbJB9ljpI3viqruRiML/Bib-Flip?node-id=38-393>

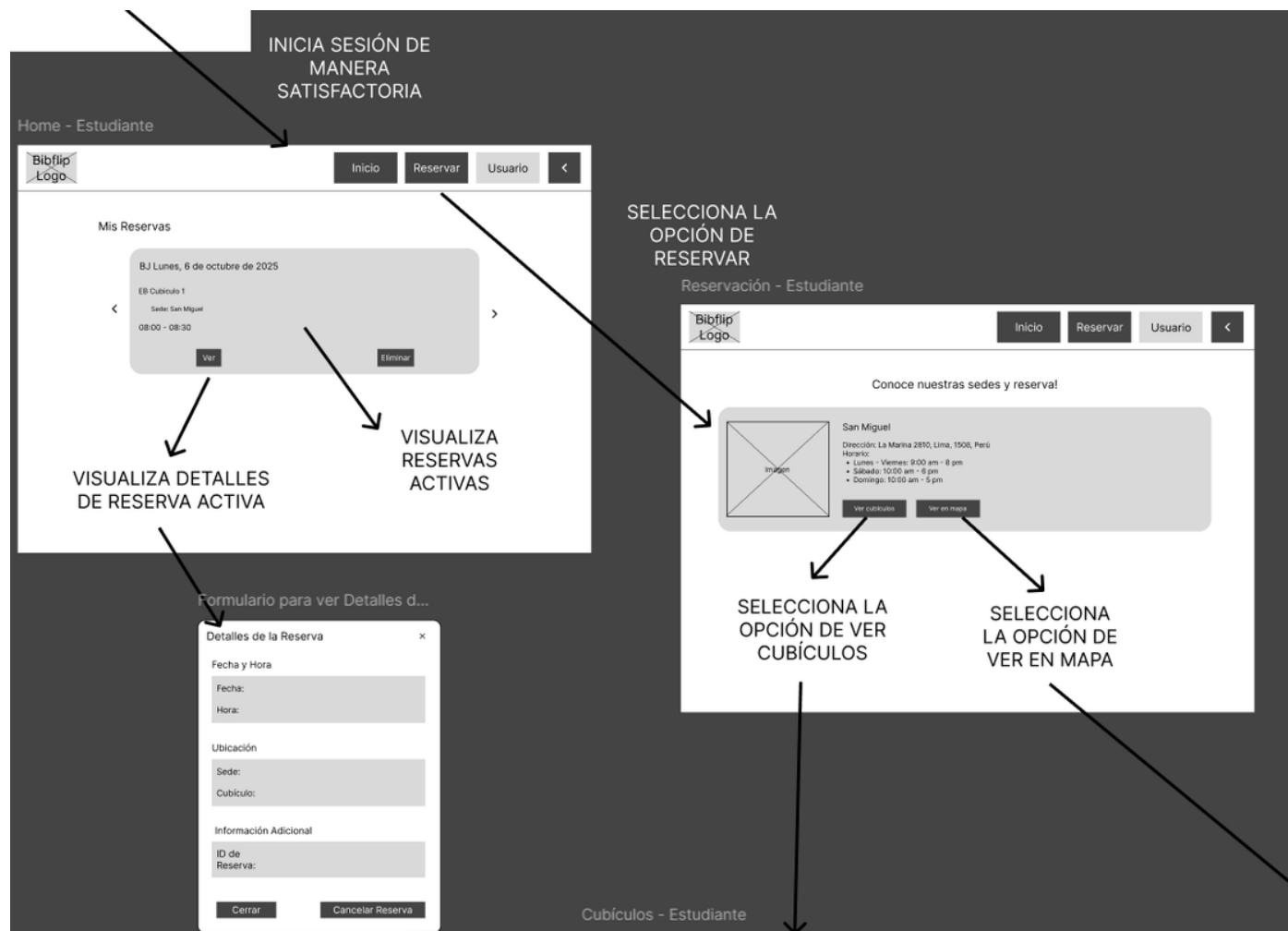
User Persona: Estudiante

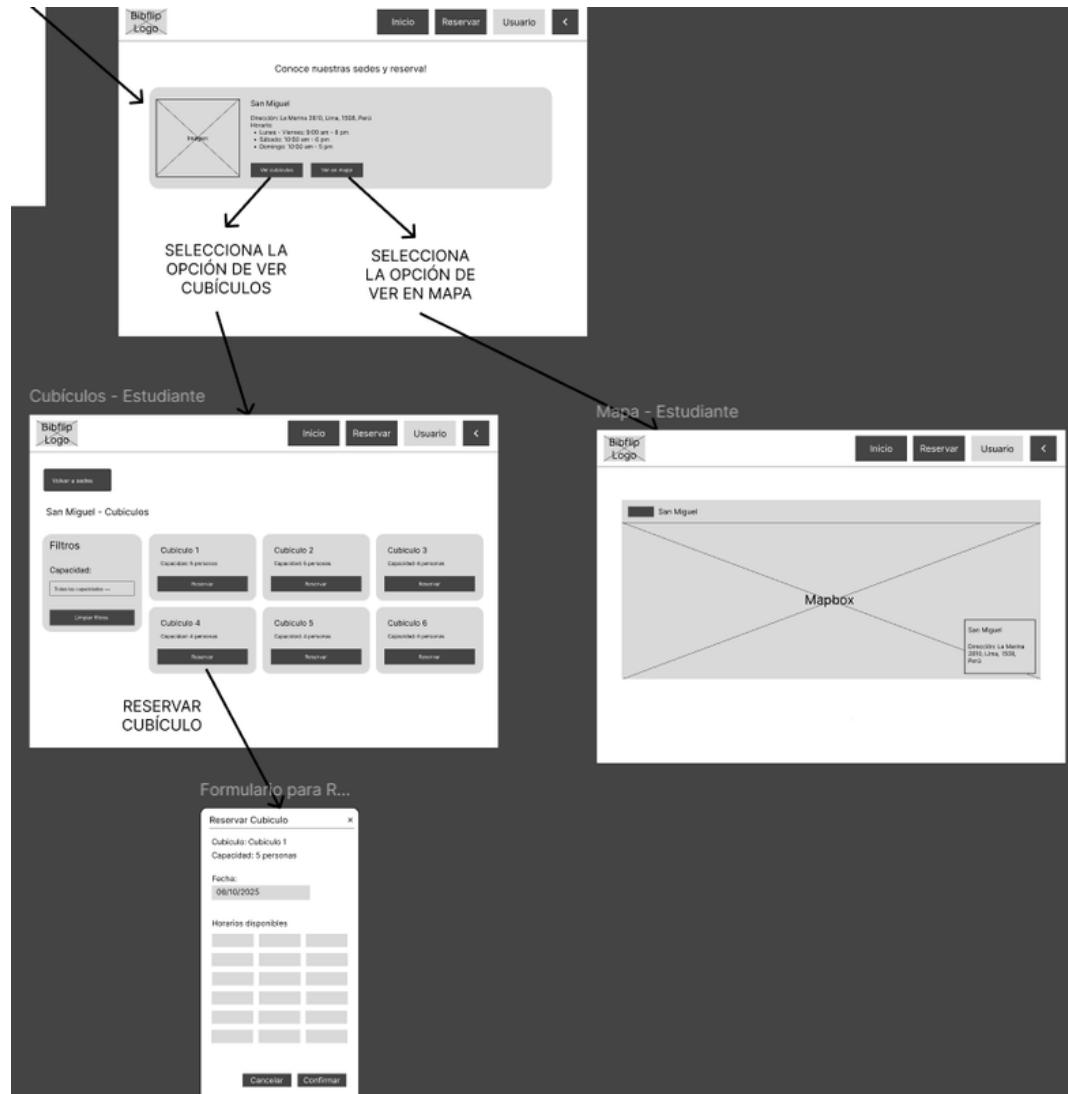
User Goal 1: Acceder rápidamente para ver la disponibilidad



Este flujo representa el objetivo principal del estudiante: consultar de manera inmediata la disponibilidad de cubículos en la sede de su preferencia. El usuario inicia sesión y, tras la autenticación, accede directamente al panel principal donde se muestran todas las sedes disponibles. Al seleccionar una sede, puede visualizar en tiempo real el estado de los cubículos (disponibles, ocupados o reservados), junto con detalles como capacidad y horarios. El diseño prioriza la rapidez de acceso, la actualización automática de la información y la claridad visual, permitiendo que el estudiante tome decisiones informadas en menos de 30 segundos desde el ingreso a la plataforma. Además, se contemplan mensajes claros ante errores de autenticación o problemas de conexión, asegurando una experiencia fluida y sin frustraciones.

User Goal 2: Realizar una reserva de cubículo

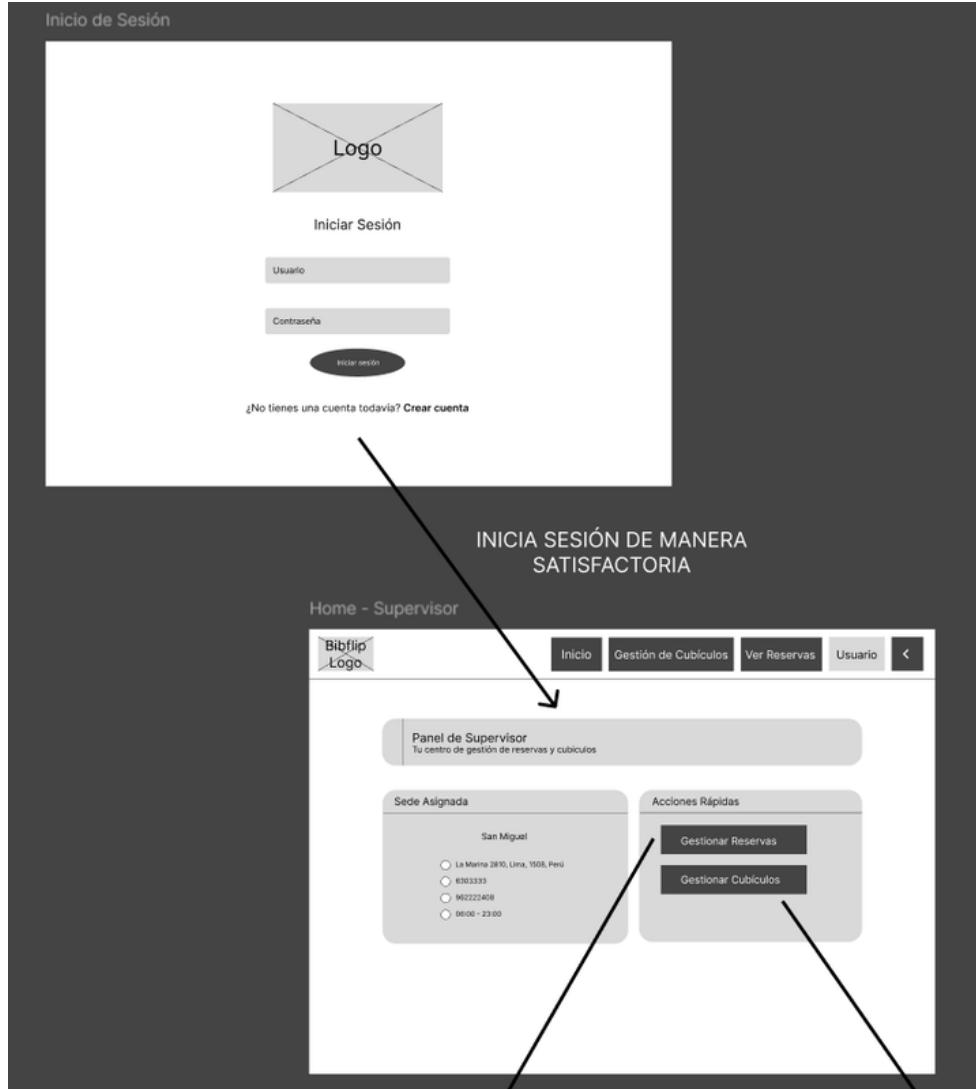


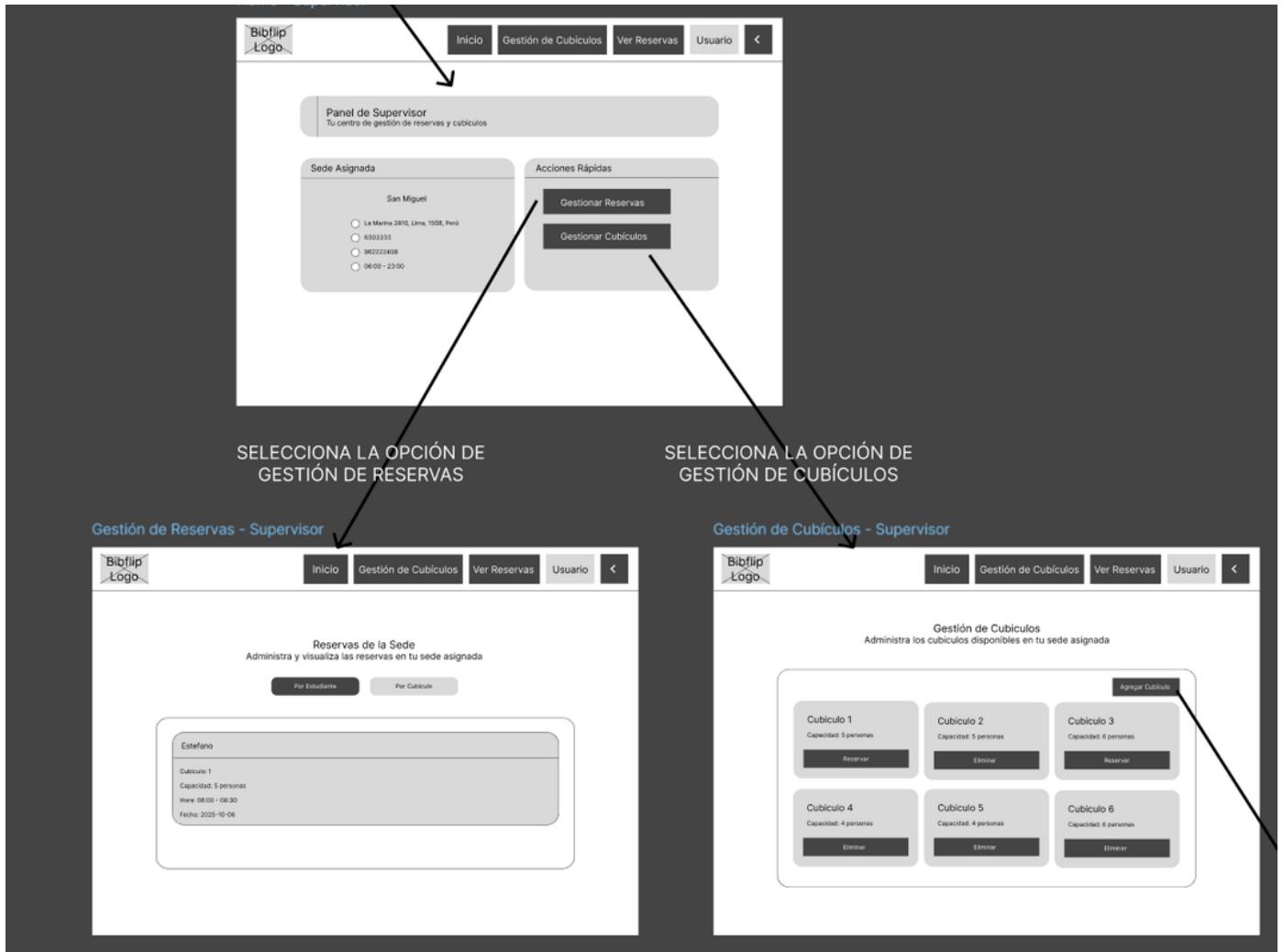


Este flujo ilustra el proceso que sigue un estudiante para reservar un cubículo de estudio en la plataforma. Tras iniciar sesión y visualizar la disponibilidad, el usuario selecciona la sede y el cubículo que mejor se adapta a sus necesidades. A continuación, el sistema muestra los horarios disponibles para ese cubículo en tiempo real. El estudiante elige el horario deseado y confirma la reserva. El sistema valida que no existan conflictos de horario ni reservas activas previas, y muestra una confirmación inmediata de la reserva exitosa. Si ocurre algún inconveniente, como la ocupación repentina del horario seleccionado o el límite de reservas alcanzado, el sistema notifica al usuario y le permite seleccionar otra opción. El diseño prioriza la simplicidad, la retroalimentación clara y la actualización instantánea del estado de los cubículos, asegurando que el proceso de reserva sea rápido, intuitivo y sin pasos innecesarios.

User Persona: Personal Bibliotecario

User Goal 3: Gestionar cubículos con facilidad







Este flujo está orientado a la experiencia del supervisor o administrador encargado de la gestión de cubículos en una sede. El proceso inicia con el acceso al panel de gestión, donde se visualiza el listado completo de cubículos existentes, junto con información relevante como número, capacidad y ubicación. El supervisor puede agregar nuevos cubículos mediante un formulario sencillo, editar los datos de los existentes o eliminarlos si es necesario. El sistema valida los datos ingresados y previene acciones que puedan afectar reservas activas, mostrando advertencias y confirmaciones cuando corresponde. Todas las operaciones se reflejan en tiempo real, permitiendo una administración eficiente y transparente. El diseño prioriza la facilidad de uso, la retroalimentación inmediata y la reducción de errores, asegurando que la gestión de cubículos sea ágil, segura y centralizada en una sola interfaz.

5.4.3. Applications Mock-ups

Aquí se muestran los mockups de alta fidelidad de la aplicación Bibflip, tanto para la versión web como móvil. Los mockups representan el diseño visual final, incluyendo tipografía, colores, iconografía y disposición precisa de los elementos.

Web App Mock-ups

En el siguiente enlace se desarrollaron los mock-ups: [Enlace-Vista-Web-App-Mockups](#)

Pantallas Generales



iBienvenido de vuelta!

Iniciar Sesión

Usuario

TESTUser

Contraseña

.....

Iniciar sesión

¿No tienes una cuenta todavía? [Crear cuenta](#)

Los 3 tipos de usuario podrán iniciar sesión ingresando su usuario y contraseña previamente registrados.



iBienvenido!

Registrarse

Usuario

Crea tu nombre de usuario

Contraseña

Crea tu contraseña

Confirmar Contraseña

Confirma tu contraseña

Crear cuenta

¿Ya tienes una cuenta? [Iniciar sesión](#)

Al ser esta una pantalla totalmente pública y accedida por cualquier usuario de internet, los administradores y supervisores no podrán registrarse de esta manera por razones de seguridad. De esta manera se evita que cualquier agente externo pueda crearse una cuenta de administrador y sabotear el sistema. Por lo tanto, esta pantalla de registro tendrá la funcionalidad de registrar nuevos estudiantes.

Pantallas para Administrador

The screenshot shows the Bibflip Admin Dashboard. At the top, there is a navigation bar with the Bibflip logo, the word "Bibflip", and links for "Inicio", "Dashboard", and "ADMIN". The main area is titled "Panel Administrativo" with the subtitle "Tu centro de gestión de creación de sedes y supervisores". Below this, there are two sections: "Estadísticas del Sistema" and "Acciones Rapidas". The "Estadísticas del Sistema" section displays two metrics: "Sedes registradas" (1) and "Supervisores activos" (1). The "Acciones Rapidas" section contains two buttons: "Gestionar Sedes" and "Gestionar Supervisores".

Estadísticas del Sistema	
	1 Sedes registradas
	1 Supervisores activos

Acciones Rapidas	
Gestionar Sedes	
Gestionar Supervisores	

Este es el menú inicial del administrador que observa tras iniciar sesión.

Sedes Disponibles

- + Nueva Sede
- Buscar sedes...
- San Miguel (Activa)

Registro de Supervisores

Crear una nueva cuenta de supervisor

Nombre de Usuario
Ingrese nombre de usuario

Contraseña

Confirmar Contraseña

Registrar Supervisor

Supervisores Registrados

SUPERVISOR

Al presionar el botón de supervisores, el administrador podrá crearle una cuenta al supervisor y asignarle una sede.

Bibflip Gestión de Sedes

Sedes Disponibles

- + Nueva Sede
- Buscar sedes...
- San Miguel (Activa)

Supervisores

+ Máximo 1 supervisor por sede

San Miguel

SUPERVISOR
Supervisor Activo

En esta pantalla el administrador puede gestionar y visualizar al supervisor que fue asignado a una sede.

Pantallas para Supervisor

Bibflip

Inicio

Gestion de Cubiculos

Ver Reservas

SUPERVISOR

The screenshot shows the Bibflip Supervisor Dashboard. At the top, there's a header bar with the Bibflip logo and navigation links: Inicio, Gestion de Cubiculos, Ver Reservas, and SUPERVISOR. Below the header is a section titled "Panel de Supervisor" with the sub-instruction "Tu centro de gestión de reservas y cubículos". The main content area is divided into two columns. The left column, titled "Mi Sede Asignada", displays information for "San Miguel": address (La Marina 2810, Lima, 1508, Peru), phone number (6303333), fax number (962222408), and operating hours (06:00-23:00). The right column, titled "Acciones Rapidas", contains two buttons: "Gestionar Reservas" and "Gestionar Cubiculos".

Este es el menú inicial que verá el supervisor tras iniciar sesión en la aplicación web.



Reservas de la Sede

Administra y visualiza las reservas en tu sede asignada

[Por Cliente](#)
[Por Cubículo](#)

TESTuser

[Reserva](#)

Cubículo: 1
Capacidad: 5 personas
Hora: 08:00 - 08:30
Fecha: 2025-10-06

TESTuser

[Reserva](#)

Cubículo: 2
Capacidad: 4 personas
Hora: 07:30 - 08:00
Fecha: 2025-10-08

En este apartado el supervisor podrá visualizar las reservas por estudiante.



Reservas de la Sede

Administra y visualiza las reservas en tu sede asignada

[Por Cliente](#)
[Por Cubículo](#)

Cubículo 1

5 personas

Cliente: TESTuser
Hora: 08:00 - 08:30
Fecha: 2025-10-06

Cubículo 2

4 personas

Cliente: TESTuser
Hora: 07:30 - 08:00
Fecha: 2025-10-08

Cubículo 3

5 personas

Cliente: TESTuser
Hora: 08:00 - 08:30
Fecha: 2025-10-08

En este apartado el supervisor podrá visualizar las reservas por cubículo.

Bibflip

Inicio Gestión de Cubiculos Ver Reservas SUPERVISOR [→]

Gestión de Cubiculos

Administra los cubiculos disponibles en tu sede asignada

Cubiculo #1	Cubiculo #2	Cubiculo #3
5 personas pabellón principal Eliminar	4 personas pabellón principal Eliminar	5 personas pabellón principal Eliminar
Cubiculo #4	Cubiculo #5	Cubiculo #6
6 personas pabellón principal Eliminar	6 personas pabellón principal Eliminar	6 personas pabellón principal Eliminar

+ Agregar cubiculo

Bibflip

Inicio Gestión de Cubiculos Ver Reservas SUPERVISOR [→]

Gestión de Cubiculos

Administra los cubiculos disponibles en tu sede asignada

Cubiculo #1	Cubiculo #2	Cubiculo #3
5 personas pabellón principal Eliminar	4 personas pabellón principal Eliminar	5 personas pabellón principal Eliminar
Cubiculo #4	Cubiculo #5	Cubiculo #6
6 personas pabellón principal Eliminar	6 personas pabellón principal Eliminar	6 personas pabellón principal Eliminar

+ Agregar cubiculo

Agregar cubiculo

Número de cubiculo

Capacidad (personas)

Zona

[Cancelar](#) [Agregar](#)

En esta pantalla el supervisor podrá gestionar los cubiculos, es decir agregarlos, editarlos o eliminarlos.



Mis Reservas

Lunes, 6 de octubre
de 2025

Cubiculo 1

Sede: San Miguel

08:00-08:30

Miércoles, 8 de
octubre de 2025

Cubiculo 2

Sede: San Miguel

07:30-08:00

Miércoles, 8 de
octubre de 2025

Cubiculo 2

Sede: San Miguel

22:30-23:00

Este será el menú inicial que verá el estudiante tras iniciar sesión. En este apartado podrá ver sus reservas más recientes y cancelarlas.

Bibflip

Inicio Reservar TESTuser

Detalles de la Reserva

Fecha y Hora

Fecha:	Hora:
Lunes, 6 de octubre de 2025	08:00 - 08:30

Ubicación

Sede:	Cubículo:
San Miguel	1

Información Adicional

ID de Reserva:	1
----------------	---

Cerrar Cancelar Reserva

Al seleccionar la opción de ver una reserva, el estudiante podrá ver estos detalles.



Conoce nuestras sedes y reserva!

San Miguel

Dirección: La Marina 2810, Lima, 1508, Perú

Horario:

Lunes - Viernes: 9 : 00 am - 8 pm

Sábado: 10 : 00 am - 6 pm

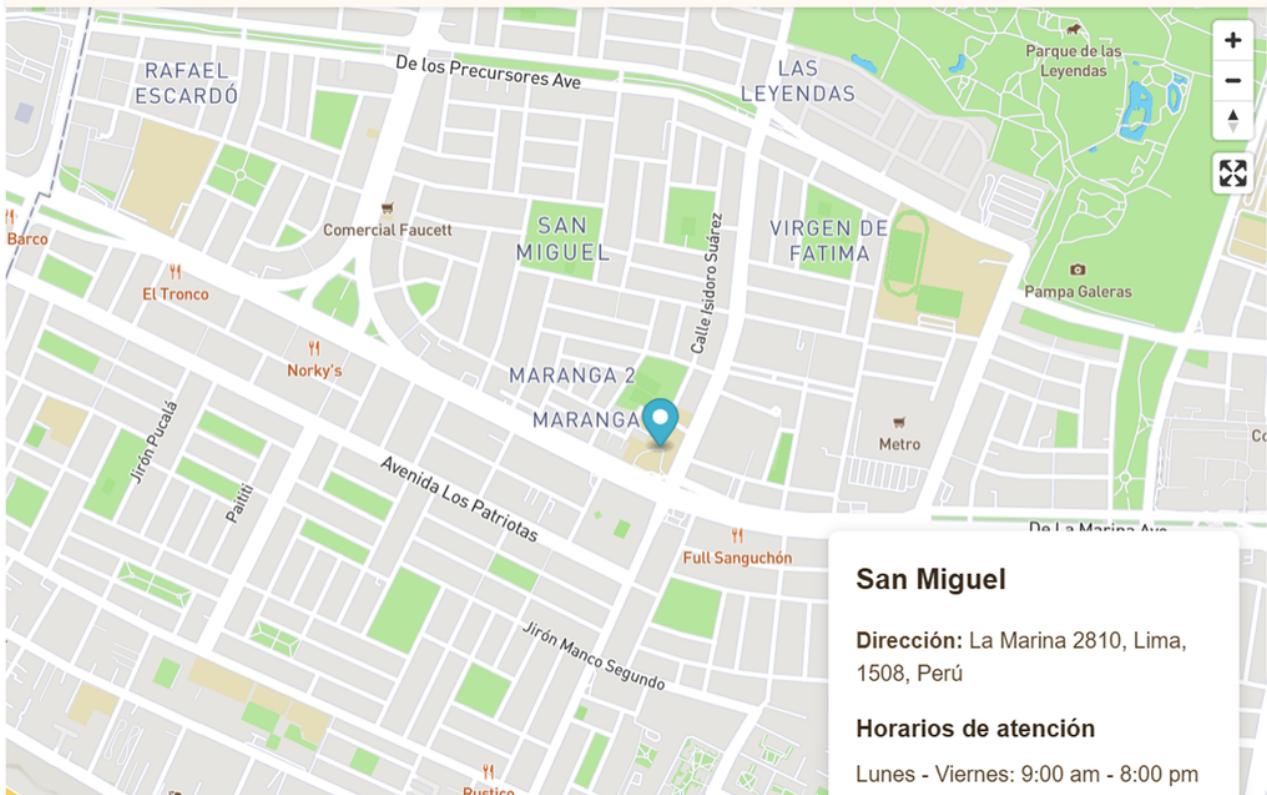
Domingo: 10 : 00 am - 5 pm

[≡ Ver cubículos](#) [Ver en Mapa](#)

Al seleccionar en la opción del navbar que dice "Reservar", el estudiante verá este apartado. En esta página podrá seleccionar la opción de ver el mapa de la sede o continuar a la reserva de cubículos.

[← Volver](#)

San Miguel



San Miguel

Dirección: La Marina 2810, Lima, 1508, Perú

Horarios de atención

Lunes - Viernes: 9:00 am - 8:00 pm

Este apartado muestra el mapa de la sede para que nuevos estudiantes puedan ubicarse mejor.

[← Volver a sedes](#)

San Miguel - Cubiculos

La Marina 2810, Lima, 1508, Perú

Filtros

Capacidad:

Todas las capacidades

Limpiar filtros

Cubiculo 1

Disponible

Capacidad: 5 personas

Reservar

Cubiculo 2

Disponible

Capacidad: 4 personas

Reservar

Cubiculo 3

Disponible

Capacidad: 5 personas

Reservar

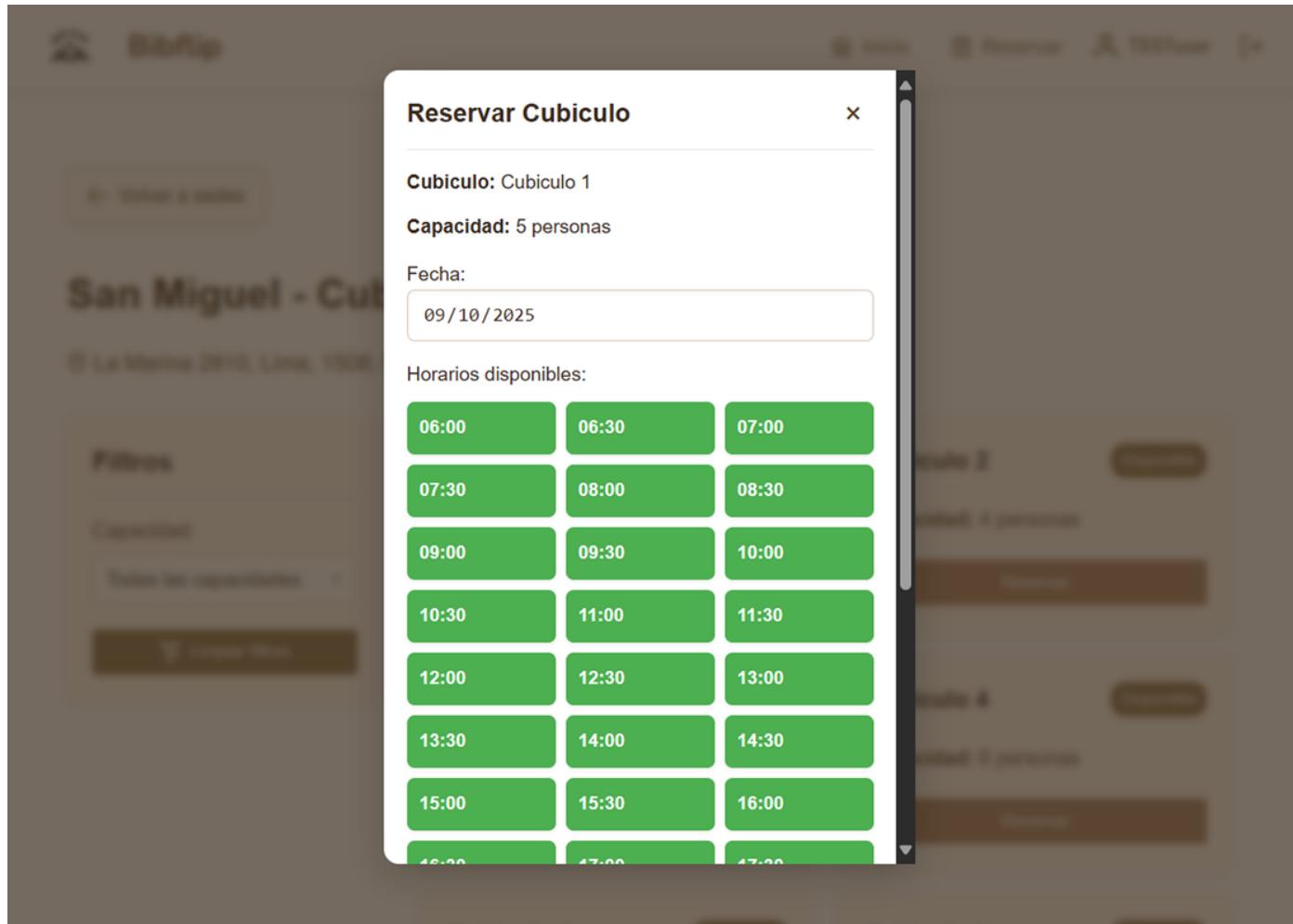
Cubiculo 4

Disponible

Capacidad: 6 personas

Reservar

Este apartado muestra los cubículos disponibles para reservar.



Al querer reservar un cubículo, el estudiante podrá ver los horarios disponibles y efectuar la reserva.

Mobile Application Mock-ups

En el siguiente enlace se desarrollaron los mock-ups: [Enlace-Vista-Mobile-App-Mockups](#)



¡Bienvenido!

Iniciar Sesión

Usuario

Contraseña

Iniciar Sesión

¿No tienes cuenta? [Regístrate](#)

En este apartado el inicio de sesión es exclusivo para estudiantes.



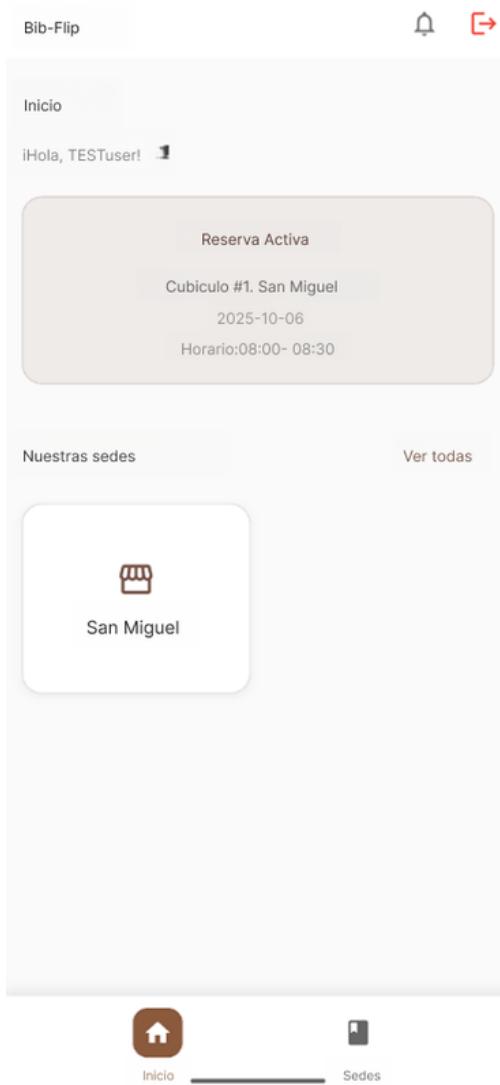
¡Bienvenido!
Regístrate

Usuario
Contraseña
Confirmar contraseña

Registrarse

¿Ya tienes cuenta? [Inicia sesión](#)

Esta imagen expresa la pantalla de registro para la aplicación móvil con los mismos parámetros que la aplicación web. El registro por este medio seguirá siendo igualmente solo para estudiantes.



Esta será la pantalla inicial que verá el estudiante tras iniciar sesión. En ella podrá darle click a la sede San Miguel para acceder a los cubículos. Además, podrá seleccionar la tarjeta de reservas activas para ver todas sus reservas.

Sedes**San Miguel**

Horario:06:00 - 23:00

Direccion: La Marina 2810, Lima, 1508, Peru

Mesas disponibles: 6



Inicio



Sedes

Este será un apartado secundario para acceder a los cubículos de la sede San Miguel.

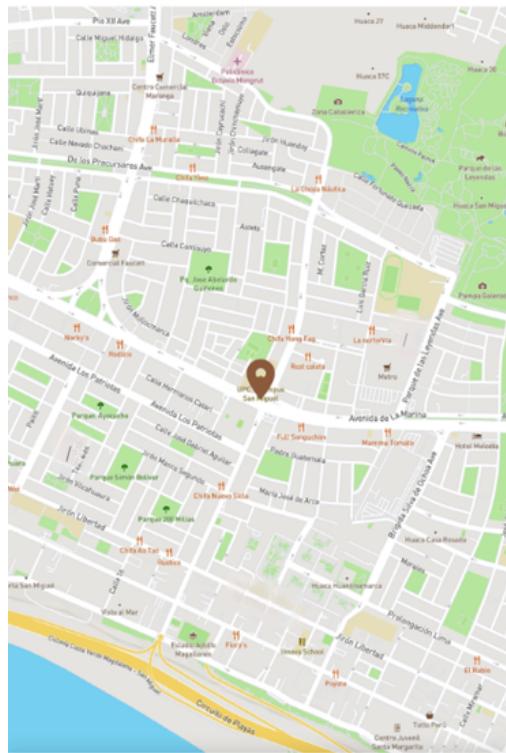
The screenshot shows a mobile application interface for 'San Miguel'. At the top, there is a back arrow icon and the text 'San Miguel'. Below this is a large orange header section containing a building icon and the text 'San Miguel'. The main content area is titled 'Información:' and includes a 'Ver en mapa' button. A location pin icon indicates the address: 'La Marina 2810, Lima, 1508, Peru'. There are two dropdown menus for 'Capacidad:' and 'Zona:'. Below these are four booking cards:

- Cubiculo #1**
Capacidad: 5 personas
Zona: pabellón principal
- Cubiculo #2**
Capacidad: 4 personas
Zona: pabellón principal
- Cubiculo #3**
Capacidad: 5 personas
Zona: pabellón principal
- Cubiculo #4**
Capacidad: 6 personas
Zona: pabellón principal

Each card has a 'Reservar' button at the bottom. At the bottom of the screen, there is a navigation bar with icons for 'Inicio' (home) and 'Sedes' (locations), with 'Sedes' being the active tab.

Este apartado se utilizará para todos los cubículos disponibles en la sede San Miguel.

Ubicación: San Miguel



San Miguel

La Marina 2810, Lima, 1508, Perú

Como llegar

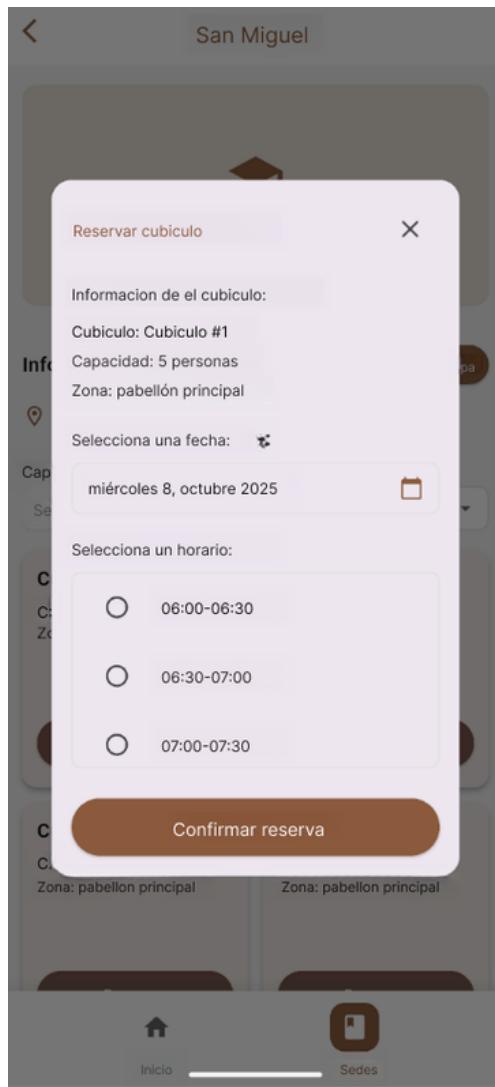


Inicio



Sedes

Esta pantalla se utilizará para visualizar el mapa de la sede San Miguel.



Esta pantalla se utilizará para reservar cubículos en un horario disponible.

5.4.4. Applications User Flow Diagrams

En esta sección se documentan los diagramas de flujo de usuario (user flows) para los principales procesos dentro de la aplicación, como la reserva de cubículos, la gestión de sedes y la administración de usuarios. Los user flows muestran los pasos que sigue un usuario desde el inicio hasta la finalización de una tarea específica, identificando puntos clave de decisión, validaciones y retroalimentación del sistema. Estos diagramas ayudan a optimizar la experiencia y asegurar que los procesos sean intuitivos y eficientes. Estos flows se encuentran en el siguiente enlace: [Enlace-User-Flows](#)

Estudiantes

User Goal 1: Acceder rápidamente para ver la disponibilidad

Descripción: El usuario necesita ingresar a la plataforma de forma ágil para consultar inmediatamente la disponibilidad de cubículos de estudio en la sede elegida.

Happy Path

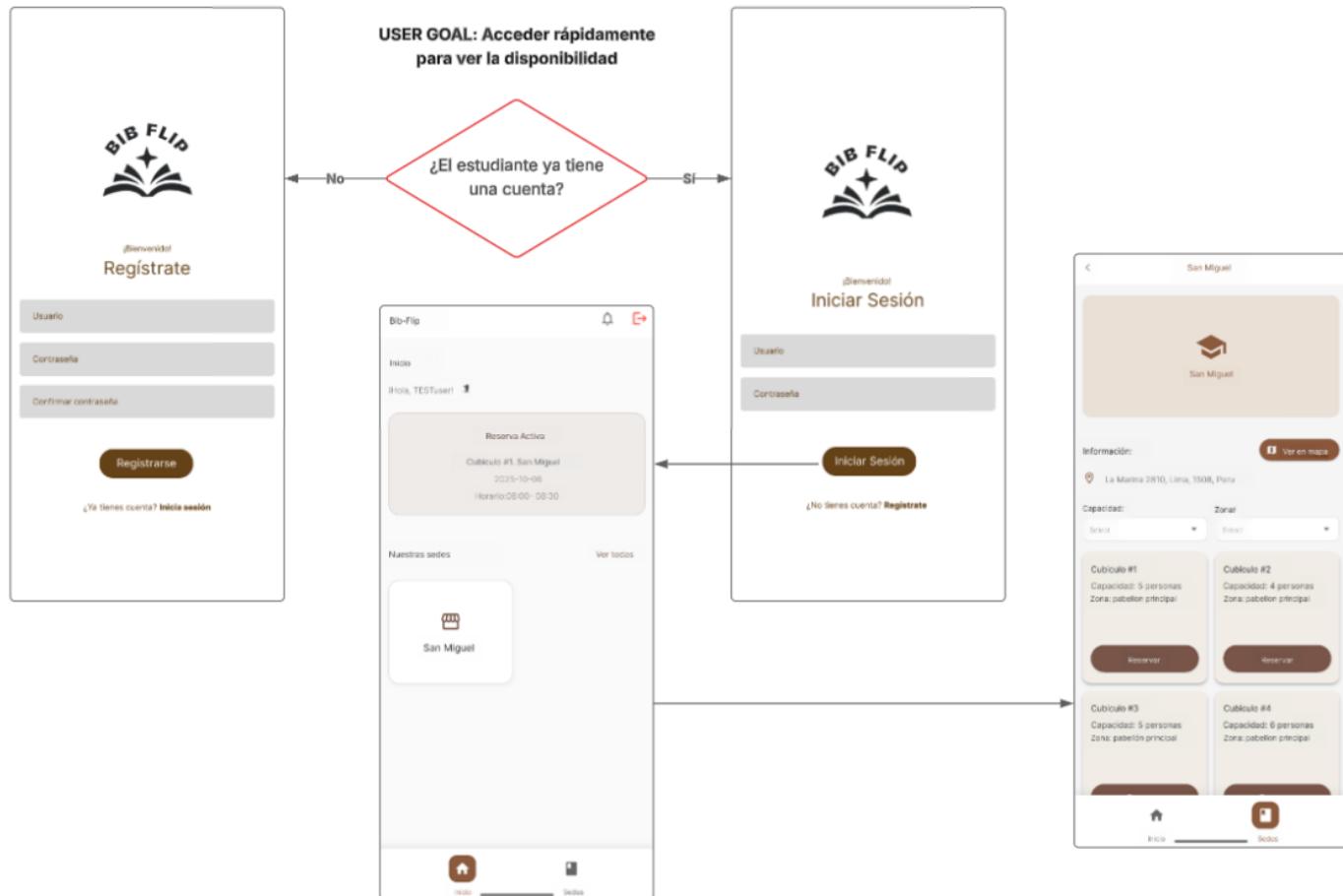
- El usuario llega a la pantalla de bienvenida y visualiza el formulario de "Iniciar Sesión"
- Ingresa sus credenciales (usuario y contraseña) correctamente
- El sistema valida la información y lo redirige al dashboard principal
- Se muestra la lista de sedes disponibles (ej: San Miguel)
- El usuario selecciona una sede y accede inmediatamente al listado de cubículos
- Puede visualizar detalles: capacidad, horario de funcionamiento y realizar una reserva en menos de 30 segundos
- La experiencia es fluida, sin interrupciones ni errores de validación

Unhappy Path

- El usuario ingresa credenciales incorrectas y el sistema muestra un mensaje de error
- Puede intentar nuevamente o usar la opción "¿No tienes cuenta? Regístrate"
- Si selecciona un cubículo sin horarios disponibles, puede visualizarlo claramente
- Si la conexión falla durante la consulta de disponibilidad, se muestra un estado de carga o error amigable
- El usuario puede reintentar sin perder el progreso

Condiciones Especificadas

- El tiempo de carga debe ser menor a 5 segundos desde la autenticación
- La disponibilidad debe actualizarse en tiempo real
- El diseño debe permitir acceso rápido sin pasos innecesarios
- Debe ser intuitivo para usuarios nuevos y recurrentes



User Goal 2: Realizar una reserva de cubículo

Descripción: El usuario desea reservar un cubículo específico para estudiar en un horario determinado, completando el proceso de forma sencilla y recibiendo confirmación inmediata.

Happy Path

- El usuario visualiza los cubículos y horarios disponibles con sus características (capacidad, ubicación)
- Selecciona un cubículo que se adapta a sus necesidades
- El sistema le muestra opciones de horario disponibles para ese día
- Confirma la reserva y puede verla en la sección "Reservas Activas" del perfil

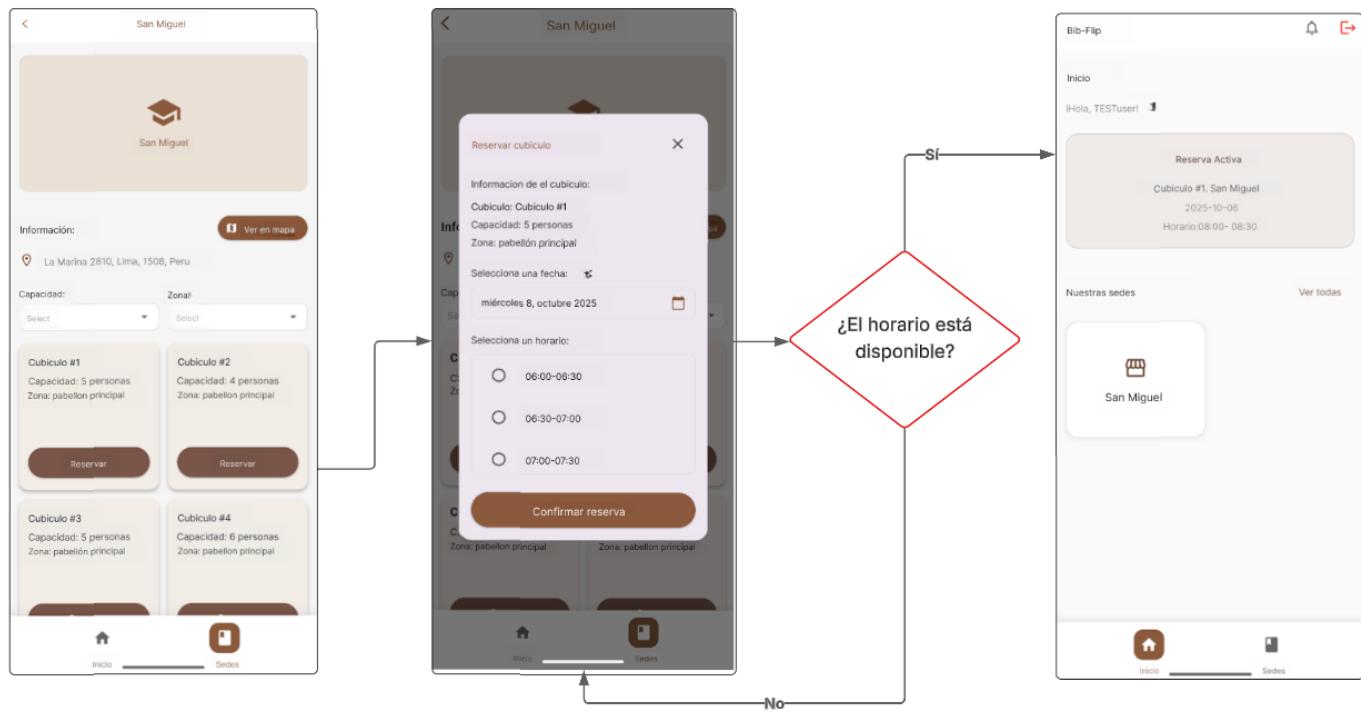
Unhappy Path

- El horario seleccionado fue reservado justo antes de completar el proceso
- El estudiante tiene que reservar en otro horario
- Si tiene reservas conflictivas o límite de reservas activas, el sistema lo notifica
- El usuario puede cancelar la reserva actual o ajustar horarios

Condiciones Especificadas

- El proceso de reserva no debe exceder 5 pasos
- Confirmación instantánea por completar la acción
- Opción de cancelación sin penalización con antelación de hasta 5 minutos máximo.
- Sincronización en tiempo real del inventario de cubículos

USER GOAL: Realizar una reserva de cubículo



Persona Bibliotecario de la Universidad

User Goal 3: Gestionar cubículos con facilidad

Descripción: El supervisor/administrador necesita gestionar de forma rápida y eficiente todos los cubículos disponibles en una sede, incluyendo agregar nuevos cubículos, eliminar existentes y actualizar información de ocupación y disponibilidad.

Happy Path

- El supervisor accede al Panel de Supervisor desde su cuenta autenticada
- Visualiza "Mi Sede Asignada" (ej: San Miguel) con toda la información: ubicación, teléfono, horario
- Navega a "Acciones Rápidas" y selecciona "Gestionar Cubículos"
- El sistema carga la vista de gestión mostrando todos los cubículos existentes en la sede con sus detalles:
 - Número de cubículo
 - Capacidad (cantidad de personas)
 - Pabellón/Ubicación principal
 - Botón de acción "Eliminar"
- El supervisor puede agregar un nuevo cubículo haciendo clic en "+ Agregar cubículo"
- Se abre un modal con campos para:
 - Nombre del cubículo
 - Capacidad (número de personas)
 - Zona/Pabellón de ubicación
- Completa los datos y confirma la acción
- El nuevo cubículo se agrega inmediatamente a la lista y el sistema actualiza la disponibilidad
- El proceso es rápido, intuitivo y sin pasos innecesarios

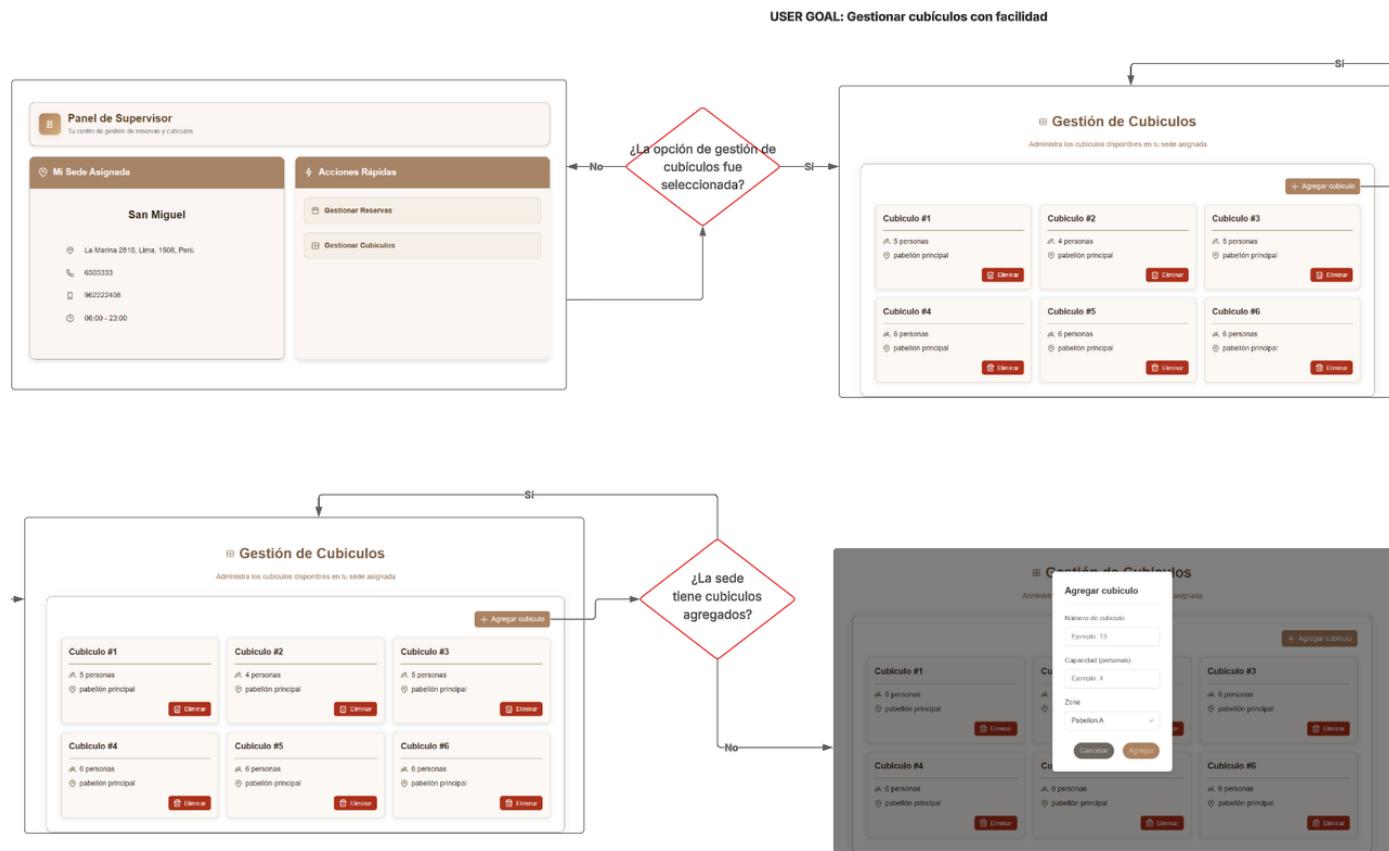
Unhappy Path

- El supervisor intenta agregar un cubículo con capacidad inválida o campos vacíos
- El sistema valida los datos y muestra un mensaje de error indicando los campos requeridos
- El supervisor corrige los datos y reintenta
- Si intenta eliminar un cubículo con reservas activas, el sistema lo advierte y sugiere esperar a que se libere
- Si la conexión falla durante la operación, se muestra un mensaje de error y la acción se guarda en cola para reintentarse
- El supervisor puede reintentar sin perder los datos ingresados
- Si selecciona accidentalmente eliminar, aparece un diálogo de confirmación antes de ejecutar la acción

Condiciones Especificadas

- El panel debe cargar en menos de 2 segundos
- Agregar cubículo no debe tomar más de 5 campos de entrada
- Confirmación visual inmediata al agregar o eliminar cubículos
- Los cambios se reflejan en tiempo real para todos los usuarios

- No se permite eliminar cubículos con reservas activas sin confirmación
 - Validación de capacidad (mínimo 1 persona, máximo 10 personas)
 - El supervisor debe pertenecer a la sede para gestionar sus cubículos
 - Historial de cambios (auditoría) opcional para usuarios administrativos



5.5. Applications Prototyping

En esta sección se describen los prototipos desarrollados para Bibflip. Estos permiten simular la navegación y las principales funcionalidades tanto en la versión web como móvil.

https://upcedu-my.sharepoint.com/:g/personal/u202218167_upc_edu_pe/EY73O0cRc6dLsHxQofhu7e8BDWhNknpIkjdJtQr4BNVXQfA?
nav=eyJzZWlcnJhbEluZm8iOnsicVmZXJyYWxhC1AiOiPbmVeclm2ZUZvckJ1c2luZXNzliwicVmZXJyYWxhC1HBQbGF0Zm9ybSI6IlldIYlsInJlZmVycmFsTW9kZSI6lnZpZ
XciLCjyZWlcnJhbFZpZXciOiJNeUzbpGVzTGlua0NvcKhifX0&e=OfAHMM

Capítulo VI: Product Implementation, Validation & Deployment

6.1. Software Configuration Management

En esta sección se establece el proceso de implementación, comprobación, despliegue y validación de la solución compuesta por los productos digitales de nuestro alcance. Este capítulo, alineado al enunciado y rúbrica del curso, cubre: configuración del entorno de desarrollo, control de código fuente, guías de estilo y la configuración de despliegue de **Landing Page**, **Web Applications**, **Web Services** y **Mobile Applications**.

6.1.1. Software Development Environment Configuration

A continuación se especifica cada uno de los productos de software, que utilizamos para elaborar en el ciclo de vida y desarrollo de los productos para nuestra solución, considerando todas las herramientas utilizadas en la documentación, especificación de nuestra solución y desarrollo de software, respetando las restricciones indicadas sobre los productos de software y herramientas que debemos utilizar.

Project Management

Producto de Software	Descripción	Ruta de referencia o de descarga
----------------------	-------------	----------------------------------

Producto de Software	Descripción	Ruta de referencia o de descarga
Trello	Para el control del proyecto, asignación de tareas y actividades de cada integrante del equipo en base a nuestras User Stories.	https://www.trello.com/
Google Meet	Plataforma virtual para realizar reuniones del grupo para coordinar y asignar actividades.	https://meet.google.com/
Discord	Plataforma virtual para realizar reuniones del grupo para coordinar y asignar actividades.	https://discord.com/

Requirements Management

Producto de Software	Descripción	Ruta de referencia o de descarga
UXPressia	Elaboración de User Personas, Empathy Maps, Journey Maps e Impact Maps.	https://uxpressia.com/
Miro	Elaboración de As-Is y To-Be Scenario Maps	https://miro.com/
Microsoft Stream	Publicación del video que contiene todas las entrevistas realizadas	https://www.microsoft.com/es-es/microsoft-365/microsoft-stream

Product UX/UI Design

Producto de Software	Descripción	Ruta de referencia o de descarga
Figma	Elaboración de Wireframes, Mock-ups y Prototypes.	https://www.figma.com

Software Development

Producto de Software	Descripción	Ruta de referencia o de descarga
IntelliJ	Idea Entorno de Desarrollo Integrado (IDE) para codificación y desarrollo del lado Backend de nuestra solución.	https://www.jetbrains.com/idea/
Visual Studio Code	Editor de código fuente para codificación y desarrollo de nuestra aplicación móvil.	https://code.visualstudio.com/
GIT	Almacenamiento y control de versiones de código.	https://git-scm.com
GitHub	Gestionar GIT en un repositorio común.	https://github.com

Software Testing

Producto de Software	Descripción	Ruta de referencia o de descarga
Gherkin	Lenguaje de especificación de pruebas utilizado para escribir nuestros Test en formato más legible.	https://cucumber.io/docs/gherkin/

Software Deployment

Producto de Software	Descripción	Ruta de referencia o de descarga
GitHub Pages	Despliegue de nuestra Landing Page.	https://pages.github.com/
Firebase	Despliegue de nuestra aplicación web.	https://firebase.google.com/
Microsoft Azure	Despliegue de servicios web.	https://azure.microsoft.com/

Software Documentation

Producto de Software	Descripción	Ruta de referencia o de descarga
Swagger	Documentación de Web Services usando OpenAPI Specification.	https://swagger.io

6.1.2. Source Code Management

Para el seguimiento de modificaciones en el desarrollo de nuestra solución, utilizamos la plataforma GitHub para alojar tanto la documentación como el código de nuestros productos de software. Esta plataforma nos permite realizar seguimiento de las modificaciones en cada parte de los productos desarrollados y también la

utilizaremos como sistema de control de versiones. Para asegurar que todo el equipo pueda acceder a la plataforma y que los repositorios que alojarán diferentes partes de nuestra solución sean accesibles por una misma ruta, por lo cual hemos creado una organización en GitHub con el nombre de nuestro producto.

Organización	URL
upc-pre-202520-1asi0572-3355-BibFlip	https://github.com/upc-pre-202520-1asi0572-3355-BibFlip

Dentro de nuestra organización en GitHub, se encuentran los repositorios correspondientes a cada uno de los productos a desarrollar.

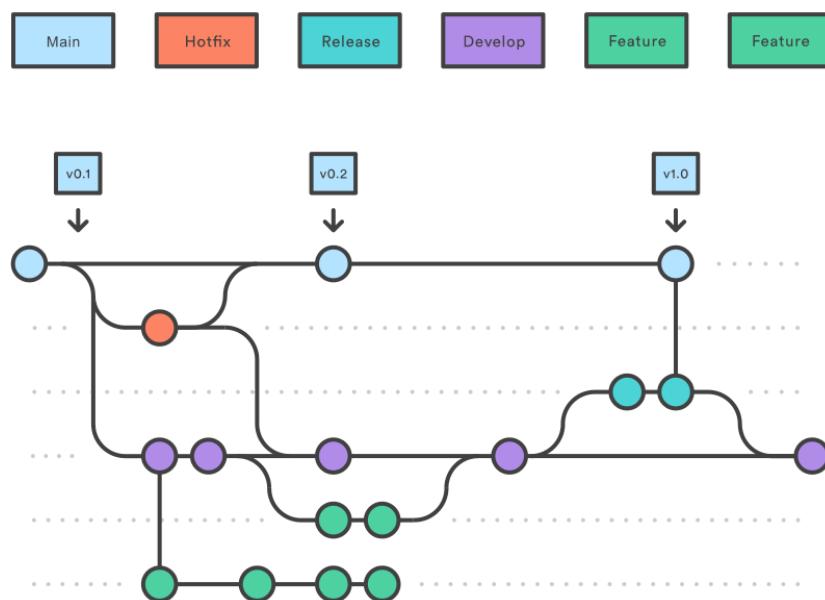
Producto	URL del Repositorio
Landing Page	https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/LandingPage
Web Service	https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Backend
Web Application	https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Web-Frontend
Mobile Application	https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Mobile-Frontend

Implementación de GitFlow

Para el desarrollo de nuestro proyecto, implementaremos GitFlow propuesto por Vincent Driessen en "A successful Git branching model". Utilizaremos este flujo de trabajo en todos los repositorios que integran nuestra solución.

Para implementar GitFlow crearemos las siguientes ramas de trabajo en cada repositorio:

Rama	Descripción
main	La rama principal del repositorio contiene el código inicial y estable, y solo se actualizará con versiones estables del software. Cada cambio en esta rama deberá ser revisado por todos los integrantes del equipo.
develop	Esta rama contendrá el código más reciente desarrollado y servirá como base para futuras versiones. Se fusionará con ramas de características "feature" y principal "main" cuando sea necesario. Los miembros del equipo podrán realizar fusiones a esta rama si lo consideran conveniente.
feature	Ramas que pueden crear los integrantes del equipo para el desarrollo de nuevas funciones o características. Estas ramas existen mientras la funcionalidad esté en desarrollo, pero eventualmente se deben fusionar con la rama "develop" para añadir definitivamente la nueva funcionalidad a próximas versiones. La convención de nomenclatura para esta rama puede ser cualquier nombre excepto master, develop, release o hotfix.
release	Esta rama se utilizará para preparar el código que incluye varias características nuevas para una nueva versión. Podrá fusionarse con las ramas "develop" y "main" después de una revisión parcial por parte del equipo. La convención de nomenclatura para esta rama debe ser release-<número>, donde <número> debe ser una secuencia de dígitos y puntos.
hotfixes	Esta rama nos permitirá corregir errores en el código de la rama "main". Una vez realizadas las correcciones, éstas se fusionarán tanto en la rama "main" como en la rama "develop". La convención de nomenclatura para esta rama debe ser hotfix-<número>, donde <número> debe ser una secuencia de dígitos y puntos.



Implementación de Conventional Commits

Conventional Commits es una convención basada en los mensajes de commit en git. En nuestro contexto nos proporciona un conjunto de reglas para crear un historial de commits, ya que describe las características, correcciones y cambios importantes realizados en los mensajes de commit.

Aplicaremos Conventional Commits para los textos de mensajes de cada commit en todas las ramas de nuestros repositorio.

Los mensajes de commit deben tener la siguiente estructura:

```
<Tipo>[ámbito opcional]: <Descripción>
[Cuerpo opcional]
[pie de página opcional]
```

Durante el desarrollo de los productos asociados a nuestra solución establecemos que cada commit debe contener los siguientes elementos.

fix: Indica una corrección en el código base.

feat: Introduce una nueva característica en el código base.

Otros tipos basados en la convención Angular como `build:`, `chore:`, `ci:`, `docs:`, `style:`, `refactor:`, `perf:`, `test:` y otros.

Establecemos estas convenciones con el propósito de comunicar de manera clara y precisa la intención de todo lo realizado durante el desarrollo de nuestra solución, tanto a los consumidores de los repositorios como a los miembros del equipo.

6.1.3. Source Code Style Guide & Conventions

Para el desarrollo de nuestra propuesta de solución, el equipo utilizará las convenciones estándar para cada lenguaje durante todo el ciclo de vida del proyecto y en todos los repositorios de trabajo. A continuación, se presentan las referencias para la nomenclatura de los elementos en cada lenguaje, procurando utilizar nomenclatura en inglés y seguir buenas prácticas de programación.

Lenguaje Referencias y Convenciones

	<ul style="list-style-type: none"> - Utilizar la estructura de documento propia de HTML, especificando el "DOCTYPE html". - Cerrar siempre las etiquetas de autocierre con />.
HTML	<ul style="list-style-type: none"> - Utilizar minúsculas "lowercase" para los nombres de las etiquetas y atributos. - Incluir atributos ALT en las imágenes para describir su contenido. - Mantener una indentación consistente en el código.
CSS	<ul style="list-style-type: none"> - Utilizar unidades relativas para definir el tamaño de las imágenes. - Separar las palabras con guiones en lugar de espacios o subrayados. - Preferir el uso de propiedades abreviadas cuando sea posible. - Emplear nombres descriptivos para las clases y los identificadores.
JavaScript	<ul style="list-style-type: none"> - Documentar el código de manera descriptiva. - Utilizar nombres descriptivos para variables y funciones. - Organizar el código en módulos y componentes.
Java	<ul style="list-style-type: none"> - Las clases e interfaces deben nombrarse con sustantivos en CamelCase. - Organizar el código en módulos y componentes. - Limitar la longitud de los métodos para que realicen una única función clara. - Manejar adecuadamente las excepciones con mensajes o acciones concretas.
Dart	<ul style="list-style-type: none"> - Los nombres de variables, métodos y funciones deben comenzar con una letra minúscula y seguir la convención lowerCamelCase. - Los nombres de clases, enumeraciones y tipos deben usar UpperCamelCase. - Evitar crear objetos o instancias innecesarias; preferir constantes <code>const</code> cuando sea posible. - Usar <code>late</code> o inicialización diferida solo cuando sea realmente necesario para optimizar recursos. - Los nombres de archivos, bibliotecas y paquetes deben escribirse en minúsculas, usando guiones bajos <code>snake_case</code> para separar palabras.
Gherkin	<p>Se usará la estructura "Given", "When", "Then", "And" para escribir casos de prueba.</p> <ul style="list-style-type: none"> - Uso de archivos .feature para cada caso de prueba. - Utilizar un Feature por funcionalidad del sistema. - Describir escenarios de un uso único, lo más independiente posible de otras funcionalidades.

6.1.4. Software Deployment Configuration

Se describe el proceso de configuración y despliegue de los productos digitales que conforman la solución **BibFlip**, incluyendo la **Landing Page** y los **Backend Services**.

El objetivo es detallar los pasos necesarios para lograr la publicación satisfactoria de ambos componentes a partir de sus respectivos repositorios en GitHub.

Entornos

Producto	Entorno	Dominio/URL	Notas
Landing	Prod	https://usuario.github.io/bibflip	GitHub Pages
Web (Vue)	Prod	https://bibflip-web.web.app	Firebase Hosting
Mobile	QA/Interno	Firebase App Distribution	APK/AAB para testers
Web Services	QA/Prod	Azure App Service	Pendiente primer release

Variables y secretos (estándar)

- `VITE_API_BASE_URL` (web), `API_BASE_URL` (mobile).
- Claves/IDs de Firebase (hosting/app distribution).
- Nunca en el repo: se configuran en **GitHub Secrets / Firebase CLI**.

1. Landing Page (GitHub Pages)

Repository

- **URL del repositorio:** <https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/LandingPage>
- **Tecnologías utilizadas:**
 - HTML
 - CSS
 - JavaScript

Pasos de despliegue

1. Clonar el repositorio:

```
git clone https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/LandingPage.git
cd LandingPage
```

2. Verificar la estructura del proyecto:

- Asegurarse de que el archivo principal `index.html` se encuentre en la raíz del repositorio, junto con las carpetas `/css` y `/js`.

3. Configurar GitHub Pages:

- Acceder a la sección `Settings` del repositorio.
- Ir a la pestaña `Pages`.
- En el apartado `Source`, seleccionar la rama `main` y la carpeta `/` (root).
- Guardar los cambios.

4. Esperar la publicación:

- GitHub Pages realizará automáticamente el build del sitio y lo publicará en la URL generada.

Resultado:

La Landing Page quedó publicada y accesible en el siguiente enlace:

<https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>

2. Frontend Web (Firebase Hosting)

```
# Requisitos
npm i -g firebase-tools
firebase login
firebase init hosting # seleccionar proyecto; setea build dir: dist

# Build y Deploy
npm ci
npm run build
firebase deploy
```

3. Mobile (Flutter/Dart)

```
flutter clean
flutter pub get
flutter build apk --release
# Distribuir con Firebase App Distribution
firebase appdistribution:distribute build/app/outputs/flutter-apk/app-release.apk --app <APP_ID> --groups testers
```

4. Web Services (Azure) — plan de despliegue

El despliegue de los servicios Backend de **BibFlip** se realizó a partir del código fuente alojado en el repositorio oficial de la organización en GitHub. Este proceso permitió publicar la API REST desarrollada en **Spring Boot** dentro de la plataforma **Microsoft Azure**, garantizando su disponibilidad pública.

Repository:

- **Nombre:** Backend
- **Plataforma:** GitHub
- **URL:** <https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Backend>

Pasos para el despliegue

1. Clonar el repositorio

```
git clone https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Backend.git
cd Backend
```

2. Compilar y empaquetar la aplicación

```
git clone https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Backend.git
cd Backend
```

3. Configurar el servicio en Azure

- Ingresar al portal de Microsoft Azure .
- Crear un App Service con las siguientes configuraciones:
 - Runtime Stack: Java 17
 - Web Server: Java SE
 - Sistema Operativo: Linux
- Asignar un nombre único para el servicio **bibflip-api-platform**.

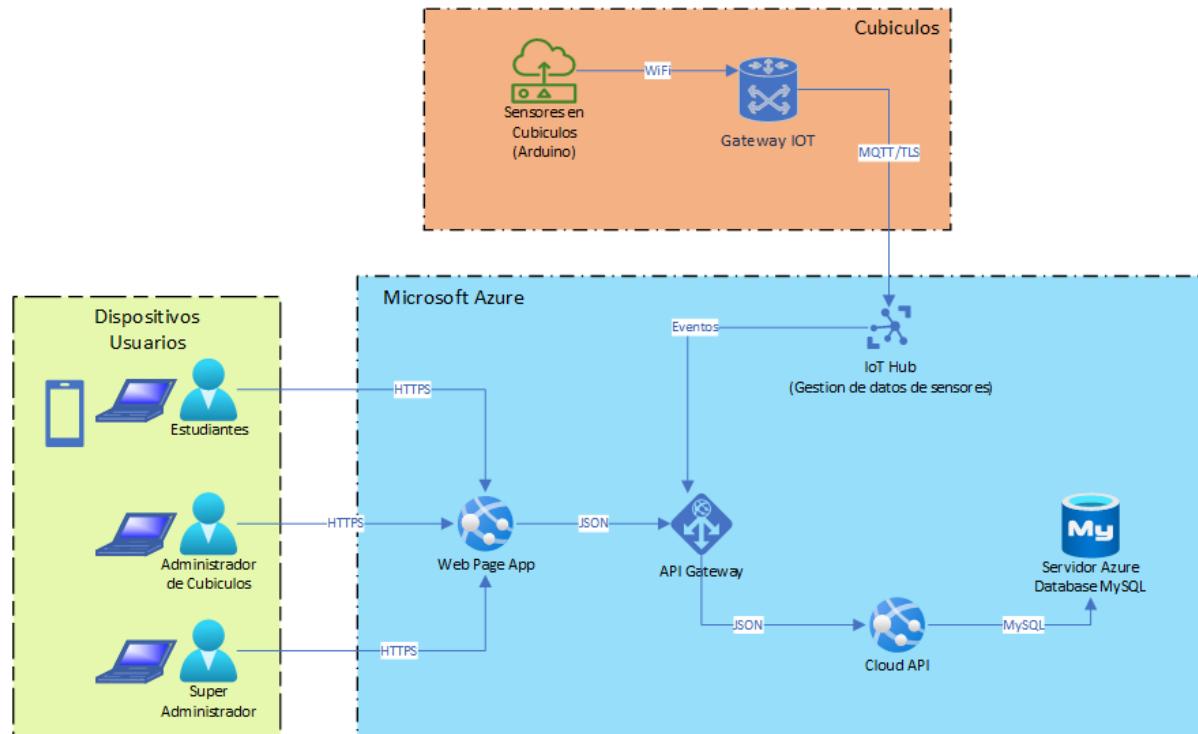
4. Conectar el repositorio con Azure

- En el App Service, acceder al apartado Deployment Center.
- Seleccionar GitHub como fuente de despliegue.
- Autorizar la cuenta de GitHub y elegir:
 - Organización: **upc-pre-202520-1asi0572-3355-BibFlip**
 - Repositorio: **Backend**
 - Rama: **main**

5. Verificar el despliegue

- Azure ejecutará automáticamente el build y desplegará la aplicación.
- Una vez completado, acceder al enlace generado por Azure para comprobar el funcionamiento de la API.
- URL de despliegue: <https://bibflip-api-platform.azurewebsites.net/swagger-ui/index.html>

Diagrama de despliegue de nuestras soluciones:



6.2. Landing Page, Services & Applications Implementation

6.2.1. Sprint 1

6.2.1.1. Sprint Planning 1

Introducción

Este sprint se orientó a entregar la primera versión navegable de la **Landing Page (HTML/CSS/JS)**, el **Frontend Web (Vue 3 + PrimeVue)** y la **App Mobile (Flutter/Dart)**, cumpliendo con los entregables solicitados para TP1.

Sprint #	Sprint 1	
Sprint Planning Background		
Date	YYYY-MM-DD	2025-10-05
Time	HH:MM	20:30
Location	(física/virtual)	Virtual
Prepared By	(Responsable del acta)	Aranda Vallejos, Oscar Gabriel
Attendees (to planning meeting)	(Equipo)	Aranda Vallejos, Oscar Gabriel / Bernaola Pérez, André Arturo / Gutierrez Garcia, Jose Eduardo / Oliveira Paucar, Mauricio / Pedraza Maldonado, Joaquin / Soriano Medrano, Diego / Velarde Luyo, Piero Alberto
Sprint 1 – 1 Review Summary		
<i>No aplica (proyecto inicia en Sprint 1).</i>		
Sprint 1 – 1 Retrospective Summary		
<i>No aplica (proyecto inicia en Sprint 1).</i>		
Sprint Goal & User Stories		
Sprint 1 Goal	(SMART)	Entregar la v1 navegable de: (1) Landing Page (GitHub Pages), (2) Frontend Web (Vue 3 + PrimeVue) en Firebase Hosting, (3) App Mobile (Flutter) con navegación base y pantallas Home/Reservas; evidenciar navegación y despliegues.
Sprint 1 Velocity	(Story Points)	30 SP
Sum of Story Points	(stories incluidos)	30 SP (US001=2, US002=2, US003=2, US004=3, US005=3, US006=5, US007=8, US009=5)

6.2.1.2. Aspect Leaders and Collaborators

Miembro	GitHub Username	Landing	Frontend Web (Vue + PrimeVue)	Mobile (Flutter/Dart)
Aranda Vallejos, Oscar Gabriel	OscarGAV	L	C	C
Bernaola Pérez, André Arturo	Andorla0	C	L	C
Gutierrez Garcia, Jose Eduardo	Elcrackje	C	C	L
Oliveira Paucar, Mauricio	Mauricio-nn	C	C	C
Pedraza Maldonado, Joaquin	JoaquinPedraza1	C	C	C
Soriano Medrano, Diego	diego5m	C	L	C
Velarde Luyo, Piero Alberto	P1er0VL	C	C	C

6.2.1.3. Sprint Backlog 1

Durante este sprint inicial, nos enfocamos en desarrollar las user stories que definen las características clave de nuestro software. Esto nos permitió, por un lado, asegurar que los usuarios tengan acceso a información relevante y explicativa sobre los productos y, por otro lado, completar la primera fase de desarrollo de nuestra aplicación web y los servicios centrales de nuestro dominio.

The screenshot shows a Trello board titled "BibFlip". It features four main lists:

- Capítulo VI**: Contains cards for "Capítulo VI: Product Implementation, Validation & Deployment", "6.1. Software Configuration Management.", "6.1.1. Software Development Environment Configuration.", "6.1.2. Source Code Management.", and "6.1.3. Source Code Style Guide & Conventions".
- Nombres**: Contains cards for Oscar, Jose, Mauricio, Joaquin, Andre, Piero, and Diego.
- Sprint 1**: Contains cards for US001 through US010, each with a checklist and a progress bar.

At the bottom, there are navigation buttons for "Inbox", "Planner", "Board" (which is selected), and "Switch boards".

URL publico de trello: <https://trello.com/b/htxHDgrX/bibflip>

Sprint #		Sprint 1					
User Story		Work-Item / Task					
Id	Title	Id	Title	Description	Estimation (Hours)	Assigned To	Status (To-do / In-Process / To-Review / Done)
US001	Ver sección "¿Qué es BibFlip?"	TA1	Diseñar contenido explicativo	Definir texto e imágenes que describan la funcionalidad principal de BibFlip	02	Mauricio Oliveira	Done
			TA2	Implementar sección en Landing Page	02	Mauricio Oliveira	Done
US002	Ver "Características clave"	TA1	Diseño visual de características	Diseñar una sección que resuma las principales características de la aplicación	03	Mauricio Oliveira	Done
			TA2	Implementar tarjetas informativas	03	Mauricio Oliveira	Done
US003	Ver galería virtual	TA1	Diseño de la galería	Definir estructura visual de la galería (grilla de imágenes o carrusel)	03	Mauricio Oliveira	Done
			TA2	Implementar galería en Landing	03	Mauricio Oliveira	Done
US004	Formulario de contacto	TA1	Diseñar formulario de contacto	Definir campos y validaciones visuales básicas del formulario	03	Mauricio Oliveira	Done
			TA2	Implementar validaciones y estilo	03	Piero Velarde	Done
US005	Ver servicios de una sede	TA1	Diseñar lista de servicios	Crear diseño del listado de servicios ofrecidos por cada sede	03	Oscar Aranda Vallejos	In Process
			TA2	Filtrado y visualización	03	Oscar Aranda Vallejos	To Review

Sprint # Sprint 1

User Story		Work-Item / Task					
US006	Visualizar reserva activa	TA1	Diseño interfaz de reserva activa	Diseñar vista principal donde se muestre información de la reserva actual	03	Oscar Aranda Vallejos	In Process
		TA2	Integrar datos de reserva activa	Simular datos desde UI (sin backend) y mostrarlos dinámicamente	03	Jose Gutierrez	To Review
US007	Disponibilidad de cubículos en tiempo real	TA1	Diseñar vista de disponibilidad	Crear prototipo de interfaz para visualizar cubículos disponibles (sin integración)	04	Jose Gutierrez	In Process
		TA2	Simular actualización de disponibilidad	Implementar actualización de estado visual con datos simulados en frontend	04	Jose Gutierrez	To Review
US008	Reservar un cubículo	TA1	Diseñar flujo de reserva	Diseñar las pantallas para seleccionar cubículo y confirmar reserva	04	Jose Gutierrez	In Process
		TA2	Implementar formulario de reserva	Programar formulario con validaciones básicas (mock sin backend)	04	André Bernaola	To Review
US009	Ver sedes en un mapa	TA1	Diseñar componente de mapa	Definir estructura visual con marcador de sedes en mock de mapa	03	Joaquin Pedraza	In Process
		TA2	Implementar popup de información	Agregar pop-up informativo al seleccionar una sede	03	Diego Soriano	In Process

Nota: SP (Story Points) y Velocity del Sprint pendientes de estimación formal del equipo.

6.2.1.4. Development Evidence for Sprint Review

Organización del github: <https://github.com/upc-pre-202520-1asi0572-3355-BibFlip>

Repository	Branch	Commit Id	Commit Message	Commit Message Body	Committed on (Date)
LandingPage	main	e65b846	edit: footer(style)	-	8/10/25
Web-Frontend	production	af43059	hotfix: update .env production	-	6/10/25
Mobile-Frontend	production	8f3b6f6	feat: update google maps integration	-	8/10/25

6.2.1.5. Testing Suite Evidence for Sprint Review

Para Sprint 1 no se incluyen Web Services; se planifica test unitario/UI para front y widget tests en Flutter en el Sprint 2.

Repo (Testing)	Tipo	Relación con historias	Estado
org/web-bibflip	UI/Component tests	US005, US007, US009	Planificado
org/mobile-bibflip	Flutter widget tests	US006, US008	Planificado

6.2.1.6. Execution Evidence for Sprint Review

Screenshots (avances reales):

- **Web (Frontend):**

The screenshot shows a web browser window for the Bibflip Supervisor Panel at the URL <https://bib-flip-web-app-2025-02.web.app/home>. The interface is divided into several sections:

- Header:** Includes a logo, the title "Bibflip", and navigation links for "Inicio", "Gestión de Cubículos", "Ver Reservas", and "SUPERVISOR".
- Panel de Supervisor:** A header section with a brown rounded rectangle containing a small icon and the text "Panel de Supervisor" and "Tu centro de gestión de reservas y cubículos".
- Mi Sede Asignada:** A section titled "San Miguel" showing details: "La Marina 2810, Lima, 1508, Perú", "6303333", "962222408", and "06:00 - 23:00".
- Acciones Rápidas:** A section with two buttons: "Gestionar Reservas" and "Gestionar Cubículos".

Firebase

Descripción general ▾

Accesos directos a proyectos

Hosting

Novedades

AI Logic NUEVO

Categorías de producto

Compilación ▾

Ejecución ▾

Analytics ▾

IA ▾

Herramientas de desarrollo relacionadas

Firebase Studio

Spark

Sin costo (USD 0 al mes) Actualizar

bib-flip-web-app-2025-02 ▾

+ Agregar app

Compilación

Hosting

Downloads 7 días en total)

9.62MB

9.54M

4.77M

0 oct 2 oct 8

— Esta semana — La semana pasada

Historial de implementaciones

Implementado

6 oct 2025 2:25 a.m.

Implementado por

github-action-1068621079@bib-flip-web-app-2025-02.iam.gserviceaccount.com

Próximos pasos con Gemini

Cuéntanos acerca de tu aplicación

Describe tu app y Gemini te sugerirá productos para ayudarte a comenzar

Ver los documentos

Agrega estadísticas y supervisión

Más información sobre los productos de supervisión de Google Analytics y Firebase

Firebase

Descripción general ▾

Accesos directos a proyectos

Hosting

Novedades

AI Logic NUEVO

Categorías de producto

Compilación ▾

Ejecución ▾

Analytics ▾

IA ▾

Herramientas de desarrollo relacionadas

Firebase Studio

Spark

Sin costo (USD 0 al mes) Actualizar

bib-flip-web-app-2025-02 ▾

Hosting >

Administrar sitio

◆ Preguntarle a Gemini cómo configurar Hosting para tu app

Panel Uso

Versión actual

github-action-1068621079@bib-flip-web-app-2025-02.iam.gserviceaccount.com

6/10/25, 2:25 a.m. c6f337

Versión anterior

github-action-1068621079@bib-flip-web-app-2025-02.iam.gserviceaccount.com

6/10/25, 2:25 a.m. c6f337

github-action-1068621079@bib-flip-web-app-2025-02.iam.gserviceaccount.com

6/10/25, 1:33 a.m. 919a16

github-action-1068621079@bib-flip-web-app-2025-02.iam.gserviceaccount.com

2/10/25, 10:53 p.m. 8188fa

oscararandavallejos@gmail.com

2/10/25, 10:44 p.m. e4ff6f3

Dominios

bib-flip-web-app-2025-02.web.app ▾

Predeterminado

bib-flip-web-app-2025-02.firebaseioapp.com ▾

Predeterminado

Configuración de almacenamiento de actualizaciones

Ver detalles →

Agregar un dominio personalizado

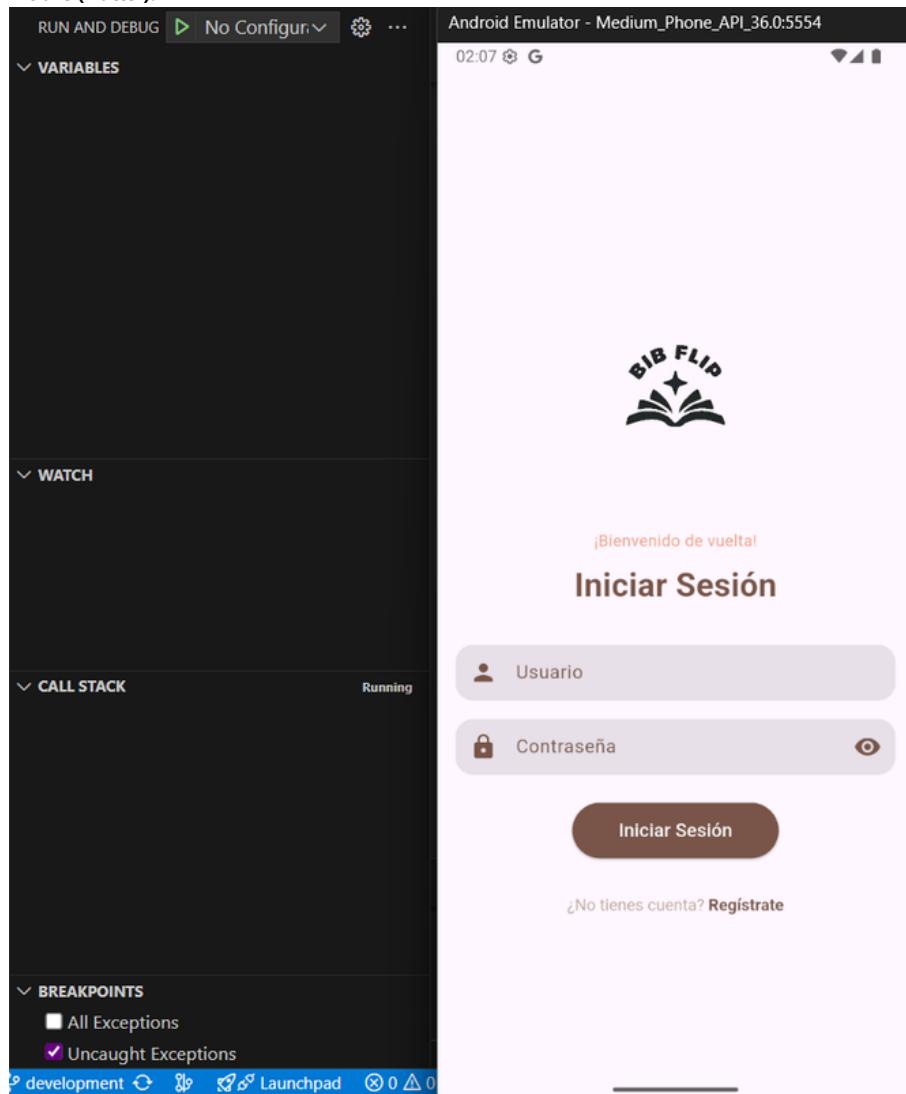
Ver los 2 dominios →

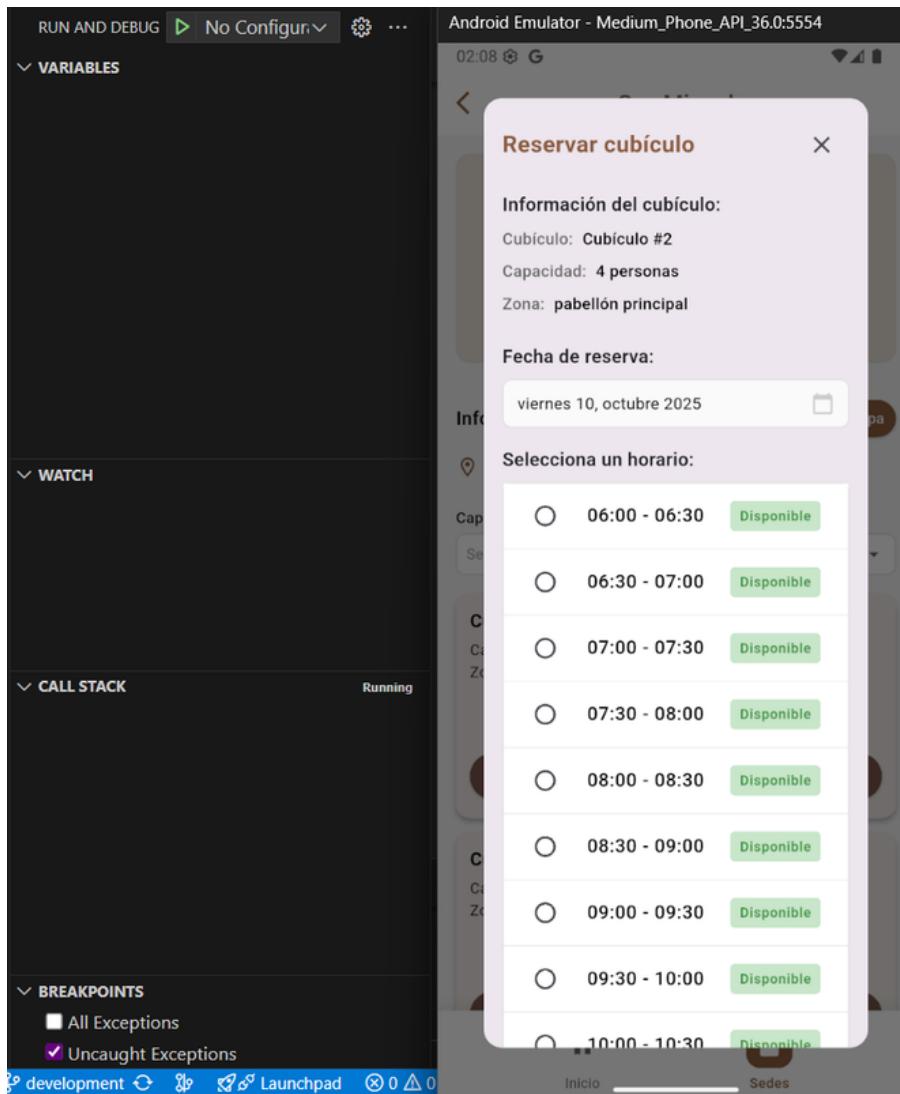
- **Landing:**



The screenshot shows the GitHub Pages settings for the repository 'upc-pre-202520-1asi0572-3355-BibFlip'. The left sidebar shows navigation options like 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Security', 'Insights', and 'Settings'. The 'Settings' tab is selected. Under 'GitHub Pages', it says 'Your site is live at <https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>'. It also shows the last deployment by Mauricio-nn yesterday. Buttons for 'Visit site' and 'Unpublish site' are present. Below this, under 'Build and deployment', it says 'Source GitHub Actions'. It notes the site was last deployed to the 'github-pages' environment by the 'Deploy to GitHub Pages' workflow. A link to learn more about deploying to GitHub Pages using custom workflows is provided. At the bottom, there is a section for 'Custom domain' with a note that custom domains allow serving from a different domain than the default. A link to learn more about configuring custom domains is also provided.

- **Mobile (Flutter):**





6.2.1.7. Services Documentation Evidence for Sprint Review

Sin Web Services en Sprint 1. Se documentarán endpoints con OpenAPI a partir de Sprint 2 (TS001–TS003).

Acción Verbo Endpoint Parámetros Ejemplo Response

6.2.1.8. Software Deployment Evidence for Sprint Review

- **Landing:** GitHub Pages — URL: <https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>
- **Frontend Web:** Firebase Hosting — URL: <https://bib-flip-web-app-2025-02.web.app/>
- **Mobile:** Build APK de pruebas internas.

Adjuntar capturas de consola y/o logs de despliegue (GitHub Actions / Firebase CLI).

6.2.1.9. Team Collaboration Insights during Sprint

- **Gestión en Trello:** tablero con columnas por cada Sprint.
 - URL del Board: <https://trello.com/b/htxHDgrX/bibflip>.
- **Git/Colaboración:** GitFlow (feature → develop → release → main); PRs con revisión cruzada; Conventional Commits y SemVer.
- **Observaciones:**
 - Picos de actividad: 08–10 oct (Landing/Web), 08–09 oct (Mobile).
 - Próximo Sprint: integrar API (TS001–TS003, TS006) y pruebas e2e con datos reales.

6.2.2. Sprint 2

6.2.2.1. Sprint Planning 2

Introducción

Este sprint se orientó a entregar la primera versión navegable de la **Landing Page (HTML/CSS/JS)**, el **Frontend Web (Vue 3 + PrimeVue)** y la **App Mobile (Flutter/Dart)**, cumpliendo con los entregables solicitados para la TB2.

Sprint #	Sprint 2			
Sprint Planning				
Background				
Date	YYYY-MM-DD	2025-10-31		
Time	HH:MM	20:30		
Location	(física/virtual)	Virtual		
Prepared By	(Responsable del acta)	Aranda Vallejos, Oscar Gabriel		
Attendees (to planning meeting)	(Equipo)	Aranda Vallejos, Oscar Gabriel / Bernaola Pérez, André Arturo / Gutierrez Garcia, Jose Eduardo / Oliveira Paucar, Mauricio / Pedraza Maldonado, Joaquin / Soriano Medrano, Diego / Velarde Luyo, Piero Alberto		
Sprint 2 – 2 Review Summary				
<i>El equipo completó la creación de la interfaz pública de la aplicación. Este trabajo de base establece una sólida presencia visual para los usuarios. Simultáneamente, se hizo un progreso significativo en las funcionalidades internas del usuario, incluyendo el diseño de las vistas para listar los servicios de las sedes, mostrar la reserva activa actual, y presentar la disponibilidad de los espacios de trabajo en tiempo real. Aunque la mayoría de las tareas de la página inicial fueron finalizadas, la implementación de estas funcionalidades más complejas, que requieren simulación de datos, se encuentra en fase de desarrollo avanzado y revisión.</i>				
Sprint 2 – 2 Retrospective Summary				
<i>El equipo logró alcanzar los objetivos relacionados con los elementos visibles de presentación. Sin embargo, se identificó una concentración de las tareas de mayor complejidad funcional las que gestionan la información dinámica y las interacciones del usuario cerca del final del ciclo de trabajo. Esta acumulación resultó en que varios entregables cruciales para la experiencia de reserva y visualización de servicios quedaron en proceso de desarrollo o pendientes de la verificación final de calidad. La lección aprendida es la necesidad de mejorar la planificación y distribución del esfuerzo para las integraciones simuladas y las revisiones de los productos desplegados.</i>				
Sprint Goal & User Stories				
Sprint 2 Goal	(SMART)	El objetivo del sprint es implementar, en un periodo no mayor a una semana, un conjunto funcional de herramientas que permitan a usuarios y administradores gestionar eficazmente los cubículos de la biblioteca: los usuarios podrán visualizar su reserva activa y realizar nuevas reservas de manera clara y confiable, mientras que los administradores podrán consultar las reservas realizadas, agregar nuevos cubículos y eliminar aquellos que ya no estén en uso. Con esto, se busca mejorar la organización, disponibilidad y precisión del sistema, asegurando una experiencia más eficiente y ordenada para todos los involucrados.		
Sprint 2 Velocity	(Story Points)	30 SP		
Sum of Story Points	(stories incluidos)	30 SP		

6.2.2.2. Aspect Leaders and Collaborators

6.2.2.3. Sprint Backlog 2

Durante este sprint, se desarrollarán capacidades clave como la visualización de reservas activas, la realización de nuevas reservas y la consulta de reservas existentes, además de herramientas administrativas para agregar y eliminar cubículos. Estas historias de usuario permiten estructurar un ciclo de desarrollo orientado a mejorar la eficiencia, claridad y control en la operación diaria de la biblioteca.

URL publico de trello: <https://trello.com/b/htxHDgrX/bibflip>

Sprint #	Sprint 2						
Technical Story		Work-Item / Task					
Id	Title	Id	Title	Description	Estimation (Hours)	Assigned To	Status
TS004	Implementar Firmware para Dispositivo IoT	TA1	Desarrollar lógica de lectura de sensores	Implementar la funcionalidad que lee datos del sensor de peso y detecta presencia en el cubículo.	04	Piero Velarde, Oscar Aranda	To-do
		TA2	Implementar transmisión de datos al Edge API	Programar la funcionalidad que genera paquetes de datos y los transmite al Edge API.	04	Piero Velarde, Oscar Aranda	To-do
		TA3	Implementar manejo de errores de hardware	Crear el sistema de registro de errores cuando falla la lectura del sensor.	02	Piero Velarde, Oscar Aranda	To-do
TS005	Integrar Firmware con Plataforma Embebida	TA1	Implementar inicialización del sistema	Desarrollar la rutina de arranque que inicializa sensores y configuración de red.	03	Jose Gutierrez, Mauricio Oliveira	To-do
		TA2	Configurar comunicación de red	Implementar la lógica de conexión WiFi/Ethernet para comunicarse con el Edge API.	03	Jose Gutierrez, Mauricio Oliveira	To-do
		TA3	Implementar recuperación ante fallos de red	Crear el sistema de reintentos automáticos cuando falla la conexión de red.	02	Jose Gutierrez, Mauricio Oliveira	To-do
TS006	Implementar Recepción de Datos en Edge API	TA1	Crear endpoint de recepción de datos IoT	Desarrollar el endpoint REST que recibe los datos transmitidos por los dispositivos IoT.	03	André Bernaola, Joaquín Pedraza	To-do
		TA2	Implementar validación de formato de datos	Crear la lógica que valida el formato y estructura de los paquetes recibidos.	02	André Bernaola, Joaquín Pedraza	To-do

Sprint #		Sprint 2							
Technical Story		Work-Item / Task							
Id	Title	Id	Title	Description		Estimation (Hours)	Assigned To	Status	
		TA3	Implementar reenvío al backend	Programar la funcionalidad que procesa y reenvía datos válidos al backend principal.		02	André Bernaola, Joaquín Pedraza	To-do	
TS007	Simular Sensores en Wokwi	TA1	Diseñar circuito de simulación en Wokwi	Crear el circuito virtual que simula el sensor de peso y su conexión con el microcontrolador.		02	Jose Gutierrez, Diego Soriano	To-do	
		TA2	Implementar lógica de detección simulada	Programar el comportamiento del sensor virtual para simular detección de ocupación.		02	Jose Gutierrez, Diego Soriano	To-do	
		TA3	Validar comunicación con Edge API simulado	Verificar que los datos simulados se transmiten correctamente al Edge API de prueba.		01	Jose Gutierrez, Diego Soriano	To-do	

6.2.2.4. Development Evidence for Sprint Review

Organización del github: <https://github.com/upc-pre-202520-1asi0572-3355-BibFlip>

Repository	Branch	Commit Id	Commit Message	Commit Message Body	Committed on (Date)
LandingPage	main	e65b846	edit: footer(style)	-	8/10/25
Web-Frontend	production	af43059	hotfix: update .env production	-	6/10/25
Mobile-Frontend	production	8f3b6f6	feat: update google maps integration	-	8/10/25

6.2.2.5. Testing Suite Evidence for Sprint Review

En esta sección se presenta el conjunto de **Acceptance Tests automatizados** para los **Web Services** relacionados con los **User Stories** especificados en el **Sprint 2**. Bajo el enfoque **BDD (Behavior-Driven Development)**, se elaboraron los archivos **.feature** utilizando el lenguaje **Gherkin** y los archivos **Steps** en **Java** para la automatización de los escenarios de prueba.

Para cada User Story trabajado en este Sprint, se incluye:

- **Archivos .feature:** Contienen los escenarios de aceptación escritos en formato *Gherkin (Given-When-Then)*, describiendo el comportamiento esperado del sistema desde la perspectiva del usuario.
- **Archivos Steps.java:** Implementan la lógica de automatización que mapea cada paso del archivo **.feature** a código ejecutable, utilizando **Cucumber** como framework de testing BDD.

Los tests de aceptación cubren las funcionalidades principales del sistema **Bib Flip**, incluyendo:

- Gestión de usuarios (registro e inicio de sesión).
- Administración de cubículos.
- Visualización y creación de reservas.
- Funcionalidades administrativas.

Cada archivo de prueba está vinculado a su **User Story** correspondiente y ha sido versionado en el **repositorio de control de versiones**, asegurando la trazabilidad completa del desarrollo.

Tabla de Commits - Acceptance Tests

Repository	Branch	Commit Id	Commit Message	Committed on (Date)
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	6fa67fc	test: add US001 acceptance test feature	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	9ad3d2b	test: implement US001 step definitions	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	bc22955	test: add US004 acceptance test feature	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	1a5c39d	test: implement US004 step definitions	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	8fdcb20	test: add US006 acceptance test feature	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	f87d2dd	test: implement US006 step definitions	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	b0c2381	test: add US008 acceptance test feature	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	36b4da5	test: implement US008 step definitions	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	010363a	test: add US010 acceptance test feature	05/11/2025

Repository	Branch	Commit Id	Commit Message	Committed on (Date)
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	a3e5836	test: implement US010 step definitions	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	59ac32a	test: add US011 acceptance test feature	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	9fe14c0	test: implement US011 step definitions	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	c056515	test: add US012 acceptance test feature	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	0fa4270	test: implement US012 step definitions	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	469eafe	test: add US013 acceptance test feature	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	cb6d64f	test: implement US013 step definitions	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	3500e16	test: add US014 acceptance test feature	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	4a4f03f	test: implement US014 step definitions	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	642fbfb	test: add US015 acceptance test feature	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	6bc36f2	test: implement US015 step definitions	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	a09bcd0	test: add US016 acceptance test feature	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	cb58299	test: implement US016 step definitions	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	6fdb3ae	test: add US017 acceptance test feature	05/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests	sprint-2	945257b	test: implement US017 step definitions	05/11/2025

A su vez hemos realizado una serie de **Unit Tests** automatizados para los **Web Services** que están relacionado también con los **User Stories** especificados en el **Sprint 2**

Repository	Branch	Commit Id	Commit Message	Committed on (Date)
upc-pre-202520-1asi0572-3355-BibFlip/Backend-Testing-Server	develop	72f2a32	test: add unit tests for CubicleQueryServiceImpl	12/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/Backend-Testing-Server	develop	9df10de	test: add unit tests for CubicleCommandServiceImpl create functionality	12/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/Backend-Testing-Server	develop	45afde0	test: add unit tests for CubicleCommandServiceImpl delete functionality	12/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/Backend-Testing-Server	develop	308f3e2	test: add unit tests for BookingQueryServiceImpl	12/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/Backend-Testing-Server	develop	162934c	test: add unit tests for BookingCommandServiceImpl	12/11/2025
upc-pre-202520-1asi0572-3355-BibFlip/Backend-Testing-Server	develop	9bd2b36	test: add unit tests for booking deletion functionality	12/11/2025

6.2.2.6. Execution Evidence for Sprint Review

Durante este sprint se lograron implementar las funcionalidades esenciales para optimizar la gestión de cubículos dentro del sistema. Los usuarios ahora pueden visualizar sus reservas activas y realizar nuevas reservas de manera clara y eficiente, garantizando un proceso más organizado al llegar a la biblioteca. Asimismo, los administradores cuentan con herramientas para consultar las reservas registradas, agregar nuevos cubículos y eliminar aquellos que ya no están en uso, manteniendo así la plataforma actualizada y alineada con las necesidades operativas. Estos avances fortalecen la usabilidad y la organización general del sistema, consolidando una base sólida para futuros incrementos.

- **Landing Page:**

The screenshot shows a web browser displaying the Bibflip landing page at <https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>. The landing page features a banner with the text "Gestiona tus cubículos fácilmente con tecnología IoT" and a subtitle about real-time monitoring of study spaces at the Universidad Peruana de Ciencias Aplicadas (UPC). Below the banner is a "Conocer más" button. At the top of the page is a navigation bar with links to "¿Qué es?", "Características", "Cómo usar", "Testimonios", and "Sobre nosotros". The GitHub repository settings page is also visible, showing the "Pages" section under the "General" tab. It displays the live site URL at <https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>, last deployed by Mauricio-nn yesterday, with "Visit site" and "Unpublish site" buttons. Other sections like "Build and deployment" and "Custom domain" are also present.

- **Web Application:**

Bibflip | Sign In

<https://bib-flip-web-app-2025-02.web.app/sign-in>

Iniciar Sesión

Nombre
Coloca tu nombre de usuario

Contraseña
Coloca tu contraseña

Iniciar sesión

¿No tienes una cuenta todavía? [Crear cuenta](#)

Bibflip | Home

<https://bib-flip-web-app-2025-02.web.app/home>

Mis Reservas

Viernes, 14 de noviembre de 2025

Cubiculo 3

Sede: San Miguel

18:00 - 19:00

19:00 - 20:00

Viernes, 14 de noviembre de 2025

Cubiculo 3

Sede: San Miguel

19:00 - 20:00

The screenshot shows a web browser window for the Bibflip website. The title bar says "Bibflip | Reservas". The URL is "https://bib-flip-web-app-2025-02.web.app/headquarters". The page header includes the Bibflip logo, a search icon, a star icon, a profile icon for "Oscar", and a "..." button. The main heading is "Conoce nuestras sedes y reserva!". Below it is a card for "San Miguel" featuring a coffee cup and map icon, address "La Marina 2810, Lima, 15088, Perú", operating hours (Monday-Friday 9:00 am - 8 pm, Saturday 10:00 am - 6 pm, Sunday 10:00 am - 5 pm), and two buttons: "Ver cubículos" and "Ver en Mapa".

The screenshot shows a web browser window for the Bibflip website. The title bar says "Bibflip | Cubículos de la Sede". The URL is "https://bib-flip-web-app-2025-02.web.app/headquarters/1/cubicles". The page header includes the Bibflip logo, a search icon, a star icon, a profile icon for "Oscar", and a "..." button. A back button "Volver a sedes" is visible. The main heading is "San Miguel - Cubículos". It shows the address "La Marina 2810, Lima, 15088, Perú". On the left is a "Filtros" sidebar with a dropdown for "Capacidad" set to "Todas las capacidades" and a "Limpiar filtros" button. To the right are three cards for "Cubículo 2" (Capacity: 4 people, Available), "Cubículo 1" (Capacity: 4 people, Available), and "Cubículo 3" (Capacity: 5 people, Available). Each card has a "Reservar" button.

The screenshot shows a web browser window for the Bibflip website. The main page displays a search bar and a list of cubicles. A modal dialog box titled "Reservar Cubiculo" is open, showing details for Cubiculo 3 (Capacity: 5 persons) on the date 14/11/2025. A grid of time slots from 06:00 to 22:00 is shown, with 18:00 highlighted in red. Buttons for "Cancelar" and "Confirmar (0 horarios)" are at the bottom.

Reservar Cubiculo

Cubiculo: Cubiculo 3
Capacidad: 5 personas
Fecha: 14/11/2025
Horarios disponibles:

06:00	07:00	08:00
09:00	10:00	11:00
12:00	13:00	14:00
15:00	16:00	17:00
18:00	19:00	20:00
21:00	22:00	

The screenshot shows a modal dialog box titled "Reserva Exitosa" confirming a successful reservation. It includes icons for a person, calendar, and clock, and lists the booking details: Cubiculo: Cubiculo 3, Capacity: 5 personas, Date: 2025-11-14, and Time: 19:00. A green "Aceptar" button is at the bottom.

Reserva Exitosa

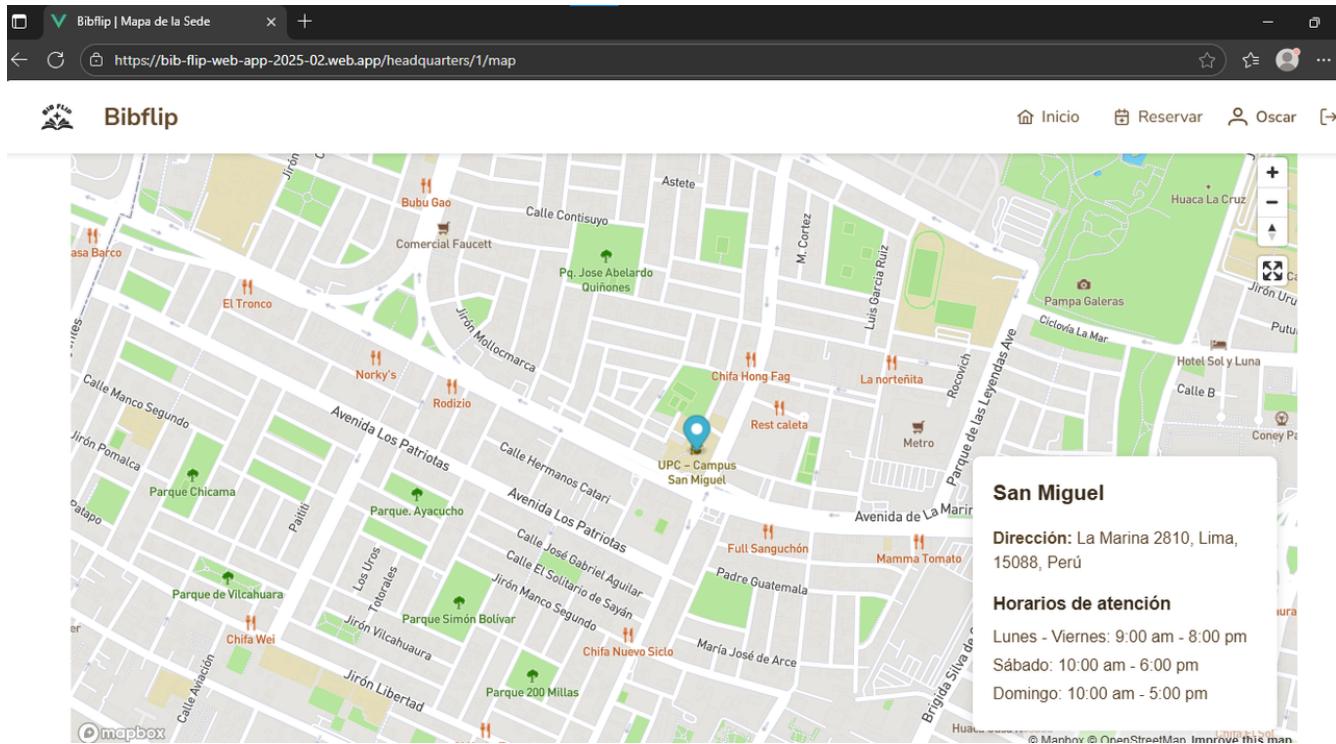
¡Reserva confirmada!

CUBICULO:
Cubiculo 3

CAPACIDAD:
5 personas

FECHA:
2025-11-14

HORARIOS:
19:00



- **Web Services:**

Swagger UI

BibFlip API Platform (V1) (GAS)

APIs

Swagger REST API Documentation

Contact Bibflip Help Team

Apache 2.0

Servers

<https://bibflip-api-platform.azurewebsites.net> - Generated server url

Authorize

Headquarter Supervisors Headquarter Supervisor Management Endpoints

- POST** /api/v1/headquarters/{headquarterId}/supervisors/{supervisorId} Assign supervisor to headquarter
- DELETE** /api/v1/headquarters/{headquarterId}/supervisors/{supervisorId} Remove supervisor from headquarter
- GET** /api/v1/headquarters/{headquarterId}/supervisors Get all supervisors for a headquarter
- GET** /api/v1/headquarters/supervisors/{supervisorId} Get headquarter by supervisor ID

Authentication Authentication Endpoints

- POST** /api/v1/authentication/sign-up
- POST** /api/v1/authentication/sign-in

Password Recovery Password Recovery Endpoints

- POST** /api/v1/password-recovery/request
- POST** /api/v1/password-recovery/confirm

Booking Booking Management Endpoints

- GET** /api/v1/bookings
- POST** /api/v1/bookings
- GET** /api/v1/bookings/{id}
- DELETE** /api/v1/bookings/{id}
- GET** /api/v1/bookings/client/{clientId}

Cubicle Cubicle Management Endpoints

- GET** /api/v1/cubicles
- POST** /api/v1/cubicles
- PATCH** /api/v1/cubicles/{cubicleId}/status
- PATCH** /api/v1/cubicles/{cubicleId}/availability-alert/status
- GET** /api/v1/cubicles/{cubicleId}
- DELETE** /api/v1/cubicles/{cubicleId}
- GET** /api/v1/cubicles/{cubicleId}/schedule
- GET** /api/v1/cubicles/headquarter/{headquarterId}

Headquarters Headquarter Management Endpoints

- GET** /api/v1/headquarters
- POST** /api/v1/headquarters
- GET** /api/v1/headquarters/{headquarterId}

Roles Role Management Endpoints

- GET** /api/v1/roles

Users User Management Endpoints

- GET** /api/v1/users
- GET** /api/v1/users/{userId}

- **Mobile (Flutter):**

RUN AND DEBUG ▶ No Configur.▼ ⚙ ... Android Emulator - Medium_Phone_API_36:0:5554
02:07 ⓘ G

VARIABLES

WATCH

CALL STACK Running

BREAKPOINTS

- All Exceptions
- Uncaught Exceptions

development ⌂ ⌂ ⌂ Launchpad ⌂ 0 △ 0

RUN AND DEBUG ▶ No Configur.▼ ⚙ ... Android Emulator - Medium_Phone_API_36:0:5554
02:08 ⓘ G

VARIABLES

WATCH

CALL STACK Running

RESERVA DE CUBÍCULOS

Reservar cubículo

Información del cubículo:

Cubículo: Cubículo #2
Capacidad: 4 personas
Zona: pabellón principal

Fecha de reserva:

viernes 10, octubre 2025

Selección de horario:

Hora	Estado
06:00 - 06:30	Disponible
06:30 - 07:00	Disponible
07:00 - 07:30	Disponible
07:30 - 08:00	Disponible
08:00 - 08:30	Disponible
08:30 - 09:00	Disponible
09:00 - 09:30	Disponible

166 / 210

The image shows two screenshots of a mobile application interface. The top screenshot displays a login screen with fields for 'Usuario' and 'Contraseña', a 'Iniciar Sesión' button, and a 'Regístrate' link. The bottom screenshot shows a reservation overlay titled 'Reservar cubículo' with information about the cubicle (Cubículo #2, 4 people, pabellón principal), a date selector set to Friday, October 10, 2025, and a list of time slots from 06:00 to 09:30, all marked as 'Disponible' (Available). The application is running on an Android emulator with API level 36.



- **Link video de visualización y navegación logrado en este Sprint:**

https://upcdeupe-my.sharepoint.com/:v/g/personal/u202114793_upc_edu_pe/IQBQVqiknnEQJsNC2!Ewg3qAc-pBdSmVQ8qelqnrx_a_ty0?e=HGXPdd&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWx8C8HAI0jTdhJIYW1XZWJBcHAiLCJyZWZlcnJhbFZpZXciOjTaGFyZURpYWxvZy1MaW5rliwicmVmZXJyYWx8C8HBQbGF0Zm9ybSI6IldlYilslnJlZmVycmFsTW9kZSI6InZpZXcifX0%3D

6.2.2.7. Services Documentation Evidence for Sprint Review

En esta sección se presenta los principales endpoints de la API del sistema, desarrollados para este Sprint, organizados por bounded context como autenticación, recuperación de contraseña, reservas, cubículos, sedes, usuarios y la gestión de supervisores. Su documentación busca ofrecer una referencia clara para comprender las acciones disponibles, los parámetros requeridos y las respuestas esperadas en cada operación, facilitando así el desarrollo, la integración y el mantenimiento de nuestra solución.

Sedes y Supervisores

Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Asignar supervisor	POST	/api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}	Path: headquarterId, supervisorId	POST /api/v1/headquarters/1/supervisors/5	Mensaje de éxito
Remover supervisor	DELETE	/api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}	Path: headquarterId, supervisorId	DELETE /api/v1/headquarters/1/supervisors/5	Mensaje de éxito
Obtener supervisores	GET	/api/v1/headquarters/{headquarterId}/supervisors	Path: headquarterId	GET /api/v1/headquarters/1/supervisors	Lista de supervisores
Obtener sede por supervisor	GET	/api/v1/headquarters/supervisors/{supervisorId}	Path: supervisorId	GET /api/v1/headquarters/supervisors/5	Datos de la sede asignada

Autenticación

Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Registro	POST	/api/v1/authentication/sign-up	Body: { name, email, password }	POST /api/v1/authentication/sign-up con JSON en el cuerpo	Usuario creado con token de autenticación
Inicio de sesión	POST	/api/v1/authentication/sign-in	Body: { email, password }	POST /api/v1/authentication/sign-in	Devuelve token y datos del usuario

Recuperación de Contraseña

Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Solicitar recuperación	POST	/api/v1/password-recovery/request	Body: { email }	POST /api/v1/password-recovery/request	Mensaje de confirmación de envío de correo
Confirmar recuperación	POST	/api/v1/password-recovery/confirm	Body: { token, newPassword }	POST /api/v1/password-recovery/confirm	Mensaje de éxito o error

Reservas

Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Listar reservas	GET	/api/v1/bookings	Ninguno	GET /api/v1/bookings	Lista de reservas
Crear reserva	POST	/api/v1/bookings	Body: { clientId, cubicleId, date, time }	POST /api/v1/bookings	Reserva creada
Obtener reserva	GET	/api/v1/bookings/{id}	Path: id	GET /api/v1/bookings/10	Detalles de la reserva
Eliminar reserva	DELETE	/api/v1/bookings/{id}	Path: id	DELETE /api/v1/bookings/10	Mensaje de éxito
Reservas por cliente	GET	/api/v1/bookings/client/{clientId}	Path: clientId	GET /api/v1/bookings/client/3	Lista de reservas del cliente

Cubículos

Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Listar cubículos	GET	/api/v1/cubicles	Ninguno	GET /api/v1/cubicles	Lista de cubículos
Crear cubículo	POST	/api/v1/cubicles	Body: { name, headquarterId, availability }	POST /api/v1/cubicles	Cubículo creado
Actualizar estado	PATCH	/api/v1/cubicles/{cubicleId}/status	Body: { status }	PATCH /api/v1/cubicles/4/status	Estado actualizado
Actualizar disponibilidad	PATCH	/api/v1/cubicles/{cubicleId}/availability-slot/status	Body: { slotId, status }	PATCH /api/v1/cubicles/4/availability-slot/status	Slot actualizado
Obtener cubículo	GET	/api/v1/cubicles/{cubicleId}	Path: cubicleId	GET /api/v1/cubicles/4	Detalles del cubículo
Eliminar cubículo	DELETE	/api/v1/cubicles/{cubicleId}	Path: cubicleId	DELETE /api/v1/cubicles/4	Mensaje de éxito
Horario del cubículo	GET	/api/v1/cubicles/{cubicleId}/schedule	Path: cubicleId	GET /api/v1/cubicles/4/schedule	Lista de horarios
Cubículos por sede	GET	/api/v1/cubicles/headquarter/{headquarterId}	Path: headquarterId	GET /api/v1/cubicles/headquarter/1	Lista de cubículos en la sede

Sedes

Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Listar sedes	GET	/api/v1/headquarters	Ninguno	GET /api/v1/headquarters	Lista de sedes
Crear sede	POST	/api/v1/headquarters	Body: { name, address, openingTime, closingTime }	POST /api/v1/headquarters	Sede creada
Obtener sede	GET	/api/v1/headquarters/{headquarterId}	Path: headquarterId	GET /api/v1/headquarters/1	Detalles de la sede

Roles y Usuarios

Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Listar roles	GET	/api/v1/roles	Ninguno	GET /api/v1/roles	Lista de roles
Listar usuarios	GET	/api/v1/users	Ninguno	GET /api/v1/users	Lista de usuarios
Obtener usuario	GET	/api/v1/users/{userId}	Path: userId	GET /api/v1/users/7	Detalles del usuario

IoT Edge

Acción	Verbo HTTP	Endpoint	Parámetros	Ejemplo	Response
Registrar dispositivo	POST	/api/v1/devices/register	Body: { "device_id": "string", "location": "string", "type": "string" }	POST /api/v1/devices/register Body: { "device_id": "chair-001", "location": "Sala A", "type": "sensor" }	Device registered successfully
Actualizar lectura	POST	/api/v1/devices/{device_id}/readings	Path: device_id Body: { "temperature": number, "humidity": number, "occupied": boolean }	POST /api/v1/devices/chair-001/readings Body: { "temperature": 22.5, "humidity": 45, "occupied": true }	Reading updated
Obtener dispositivos disponibles	GET	/api/v1/devices/status/available	Ninguno	GET /api/v1/devices/status/available	Available
Obtener dispositivos ocupados	GET	/api/v1/devices/status/occupied	Ninguno	GET /api/v1/devices/status/occupied	Sala C, Occupied
Verificar salud del backend	GET	/api/v1/devices/health/backend	Ninguno	GET /api/v1/devices/health/backend	Status: ok, Timestamp: 2025-11-14T18:00:00Z

Acción	Verbo HTTP	Endpoint	Parámetros	Ejemplo	Response
Verificar dispositivos offline	POST	/api/v1/devices/maintenance/check-offline	Ninguno	POST /api/v1/devices/maintenance/check-offline	Status: "offline"
Obtener dispositivo por ID	GET	/api/v1/devices/{device_id}	Path: device_id	GET /api/v1/devices/chair-001	device_id: "chair-001"
Eliminar dispositivo	DELETE	/api/v1/devices/{device_id}	Path: device_id	DELETE /api/v1/devices/chair-001	Device deleted
Obtener todos los dispositivos	GET	/api/v1/devices/	Ninguno	GET /api/v1/devices/	device_id: "chair-001", ...
Verificar salud general	GET	/health	Ninguno	GET /health	status: "ok"

Imagen de documentación con OpenAPI:

The screenshot shows the BibFlip API Platform's Swagger UI interface. At the top, there are tabs for 'Swagger' and 'OpenAPI'. Below that, a search bar contains 'https://bibflip-api-platform.azurewebsites.net - Generated server URL' and an 'Explore' button. The main area is organized into sections corresponding to different API modules:

- Headquarter Supervisors**: Headquarter Supervisor Management Endpoints. Includes POST /api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}, DELETE /api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}, GET /api/v1/headquarters/{headquarterId}/supervisors, and GET /api/v1/headquarters/supervisors/{supervisorId}.
- Authentication**: Authentication Endpoints. Includes POST /api/v1/authentication/sign-up and POST /api/v1/authentication/sign-in.
- Password Recovery**: Password Recovery Endpoints. Includes POST /api/v1/password-recovery/request and POST /api/v1/password-recovery/confirm.
- Booking**: Booking Management Endpoints. Includes GET /api/v1/bookings, POST /api/v1/bookings, GET /api/v1/bookings/{id}, DELETE /api/v1/bookings/{id}, and GET /api/v1/bookings/client/{clientId}.
- Cubicle**: Cubicle Management Endpoints. Includes GET /api/v1/cubicles, POST /api/v1/cubicles, PATCH /api/v1/cubicles/{cubicleId}/status, PATCH /api/v1/cubicles/{cubicleId}/availability-alert/status, GET /api/v1/cubicles/{cubicleId}, DELETE /api/v1/cubicles/{cubicleId}, GET /api/v1/cubicles/{cubicleId}/schedule, and GET /api/v1/cubicles/headquarter/{headquarterId}.
- Headquarters**: Headquarter Management Endpoints. Includes GET /api/v1/headquarters, POST /api/v1/headquarters, and GET /api/v1/headquarters/{headquarterId}.
- Roles**: Role Management Endpoints. Includes GET /api/v1/roles.
- Users**: User Management Endpoints. Includes GET /api/v1/users and GET /api/v1/users/{userId}.

BibFlip IoT Edge API 2.0.0 OAS 3.1

[/openapi.json](#)

Edge API for IoT chair sensors with backend synchronization

IoT Device Monitoring

POST	/api/v1/devices/register	Register Device	⌄
POST	/api/v1/devices/{device_id}/readings	Update Device Reading	⌄
GET	/api/v1/devices/status/available	Get Available Devices	⌄
GET	/api/v1/devices/status/occupied	Get Occupied Devices	⌄
GET	/api/v1/devices/health/backend	Check Backend Health	⌄
POST	/api/v1/devices/maintenance/check-offline	Check Offline Devices	⌄
GET	/api/v1/devices/{device_id}	Get Device	⌄
DELETE	/api/v1/devices/{device_id}	Delete Device	⌄
GET	/api/v1/devices/	Get All Devices	⌄

URL de documentación desplegada de Web Services:

<https://bibflip-api-platform.azurewebsites.net/swagger-ui/index.html>

URL de documentación desplegada de IoT Edge:

<https://bibflip-edge-api-platform.azurewebsites.net/api/docs>

URL del repositorio de Web Services:

<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Backend>

URL del repositorio de IoT Edge:

<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Edge-Api-Platform>

6.2.2.8. Software Deployment Evidence for Sprint Review

- Landing Page:

GitHub Pages — URL: <https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>

The screenshot shows the GitHub Pages settings for the repository `upc-pre-202520-1asi0572-3355-BibFlip`. The main heading is "GitHub Pages". Below it, a message states: "GitHub Pages is designed to host your personal, organization, or project pages from a GitHub repository." A box highlights the live site URL: <https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>, last deployed by Mauricio-nn yesterday. Buttons for "Visit site" and "Unpublish site" are visible. The left sidebar includes sections for General, Access, Collaborators and teams, Moderation options, Code and automation, Branches, Tags, Rules, Actions, Models, Webhooks, Copilot, Environments, Pages (which is selected), and Custom properties.

- **Web Application:**

Firebase Hosting — URL: <https://bib-flip-web-app-2025-02.web.app/>



1. Construir aplicación Vue

Genera la carpeta dist lista para producción:

```
npm run build
```

2. Inicializar Firebase en el proyecto

Dentro de la carpeta del proyecto ejecuta:

```
firebase init
```

3. Desplegar la aplicación

```
Desplegar la aplicación
```



Mis Reservas

The screenshot shows a "Mis Reservas" section with two cards. Each card displays a date ("Viernes, 14 de noviembre de 2025"), a location ("Cubículo 3"), a venue ("Sede: San Miguel"), and a time range ("18:00 - 19:00" and "19:00 - 20:00"). Navigation arrows are visible between the cards.

- **Mobile Application:**

Build APK de pruebas internas.

RUN AND DEBUG ▶ No Configur.▼ ⚙ ... Android Emulator - Medium_Phone_API_36:0:5554
02:07 ⓘ G

VARIABLES

WATCH

CALL STACK Running

BREAKPOINTS

- All Exceptions
- Uncaught Exceptions

development ⌂ ⌂ ⌂ Launchpad ⌂ 0 ⌂ 0

RUN AND DEBUG ▶ No Configur.▼ ⚙ ... Android Emulator - Medium_Phone_API_36:0:5554
02:08 ⓘ G

VARIABLES

WATCH

CALL STACK Running

RESERVA DE CUBÍCULOS

Reservar cubículo

Información del cubículo:

Cubículo: Cubículo #2
Capacidad: 4 personas
Zona: pabellón principal

Fecha de reserva:

viernes 10, octubre 2025

Selección de horario:

Hora	Estado
06:00 - 06:30	Disponible
06:30 - 07:00	Disponible
07:00 - 07:30	Disponible
07:30 - 08:00	Disponible
08:00 - 08:30	Disponible
08:30 - 09:00	Disponible
09:00 - 09:30	Disponible

173 / 210

The image shows two screenshots of a mobile application interface. The top screenshot displays a login screen with fields for 'Usuario' and 'Contraseña', a 'Iniciar Sesión' button, and a 'Regístrate' link. The bottom screenshot shows a reservation overlay titled 'Reservar cubículo' with information about the cubicle (Cubículo #2, 4 people, pabellón principal), a date selector set to Friday, October 10, 2025, and a list of time slots from 06:00 to 09:30, all marked as 'Disponible' (Available). The application is running on an Android emulator with API level 36.



- **Web Services y IoT Edge:**

Web Services — URL: <https://bibflip-api-platform.azurewebsites.net/swagger-ui/index.html>

IoT Edge — URL: <https://bibflip-edge-api-platform.azurewebsites.net/api/docs>

The screenshot shows the Microsoft Azure App Services dashboard. On the left, under 'App Services', it lists several services: 'bean-detect-ai-api-platform', 'bibflip-api-platform' (which is selected and highlighted in grey), 'bibflip-edge-api-platform', and 'smart-band-edge-api-platform-gru'. A message box indicates a new version is coming. On the right, the details for 'bibflip-api-platform' are shown, including its icon (a blue cube), name, and status as an 'Aplicación web'. It shows an error icon and the message 'Error de red'. Below this is a 'Resumen' section with various metrics and a 'Detalles' section with the message: 'Se agotó el tiempo de espera de la solicitud o el explorador rechazó la conexión. Consulte el estado de la red.'

The screenshot shows the BibFlip API Platform Swagger UI documentation. At the top, it says 'BibFlip API Platform v1.0.0 OAS 3.1'. Below this is a 'Servers' dropdown set to 'https://bibflip-api-platform.azurewebsites.net - Generated server url' and an 'Authorize' button. The main content area is titled 'Headquarter Supervisors' and shows management endpoints for headquarter supervisors. It includes four methods: a green 'POST' button for '/api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}' (Assign supervisor to headquarter), a red 'DELETE' button for '/api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}' (Remove supervisor from headquarter), a blue 'GET' button for '/api/v1/headquarters/{headquarterId}/supervisors' (Get all supervisors for a headquarter), and another blue 'GET' button for '/api/v1/headquarters/supervisors/{supervisorId}' (Get headquarter by supervisor ID).

BibFlip IoT Edge API 2.0.0 OAS 3.1

[/openapi.json](#)

Edge API for IoT chair sensors with backend synchronization

IoT Device Monitoring

<code>POST</code>	<code>/api/v1/devices/register</code>	Register Device	
<code>POST</code>	<code>/api/v1/devices/{device_id}/readings</code>	Update Device Reading	
<code>GET</code>	<code>/api/v1/devices/status/available</code>	Get Available Devices	
<code>GET</code>	<code>/api/v1/devices/status/occupied</code>	Get Occupied Devices	
<code>GET</code>	<code>/api/v1/devices/health/backend</code>	Check Backend Health	
<code>POST</code>	<code>/api/v1/devices/maintenance/check-offline</code>	Check Offline Devices	
<code>GET</code>	<code>/api/v1/devices/{device_id}</code>	Get Device	
<code>DELETE</code>	<code>/api/v1/devices/{device_id}</code>	Delete Device	
<code>GET</code>	<code>/api/v1/devices/</code>	Get All Devices	

6.2.2.9. Team Collaboration Insights during Sprint

- **Gestión en Trello:** Tablero con columnas por cada Sprint.

URL del Board: <https://trello.com/b/htxHDgrX/bibflip>.

The screenshot shows a Trello board titled "BibFlip". The board is organized into several vertical columns, each representing a different sprint or chapter. The columns are:

- Capítulo III - TB1:** Contains cards for "3.1. To-Be Scenario Mapping.", "3.2. User Stories.", "3.3. Impact Mapping.", and "3.4. Product Backlog.". There is a "+ Add a card" button at the bottom.
- Capítulo IV - TB1:** Contains cards for "4.1. Strategic-Level Domain-Driven Design.", "4.1.1. EventStorming.", "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.", and "4.1.3.3 Software Architecture Container Level Diagrams.". There is a "+ Add a card" button at the bottom.
- Capítulo V:** Contains cards for "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.", and "5.3. Landing Page UI Design.". There is a "+ Add a card" button at the bottom.
- Capítulo VI:** Contains cards for "6.1. Software Configuration Management.", "6.1.1. Software Development Environment Configuration.", "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.", and "6.2.1.2. Aspect Leaders and Contributors". There is a "+ Add a card" button at the bottom.
- Sprint 1:** Contains cards for "US001 - Ver sección ¿Qué es Bib Flip?", "US004 - Usar el formulario de contacto", "US005 - Eliminar cubículos", "US006 - Registro de cuenta de usuario", "US007 - Registro de cuenta de administrador", "US010 - Registro de cuenta de administrador", "US014 - Ver la biblioteca asignada", "US015 - Eliminar cubículos", "US016 - Inicio de sesión de administrador", and "US017 - Inicio de sesión de usuario". There is a "+ Add a card" button at the bottom.
- Sprint 2:** Contains cards for "US008 - Reservar un cubículo", "US009 - Visualizar reservas realizadas", "US011 - Inicio de sesión de usuario", "US012 - Agregar nuevos cubículos", and "US013 - Visualizar reservas realizadas". There is a "+ Add a card" button at the bottom.

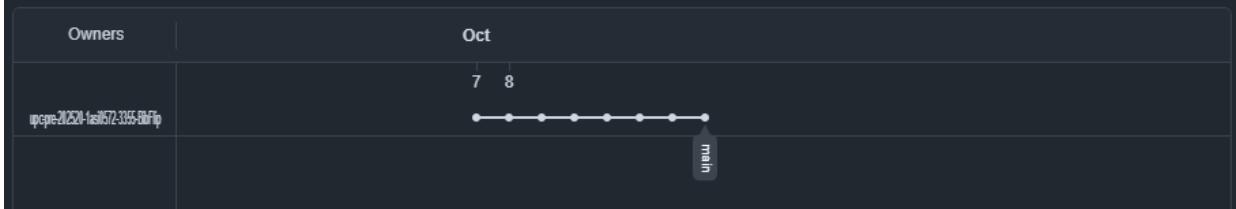
At the bottom of the board, there are navigation buttons for "Inbox", "Planner", "Board", and "Switch boards".

- Capturas de los analíticos de colaboración y commits en GitHub:

- **Landing Page:**

Network graph

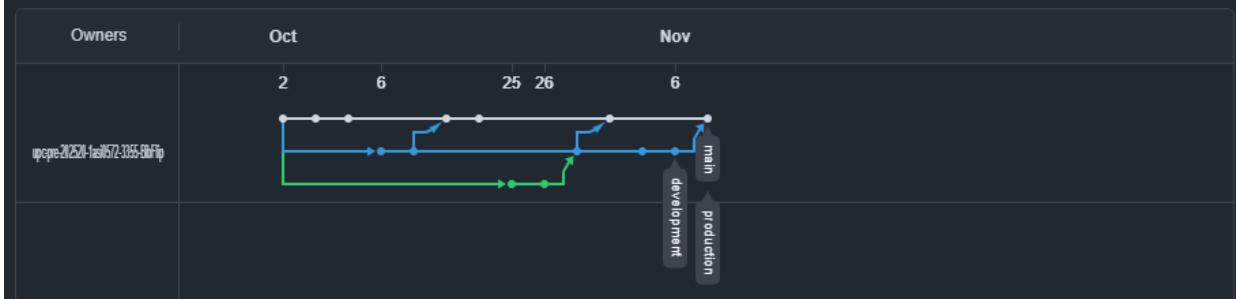
Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



- **Web Application:**

Network graph

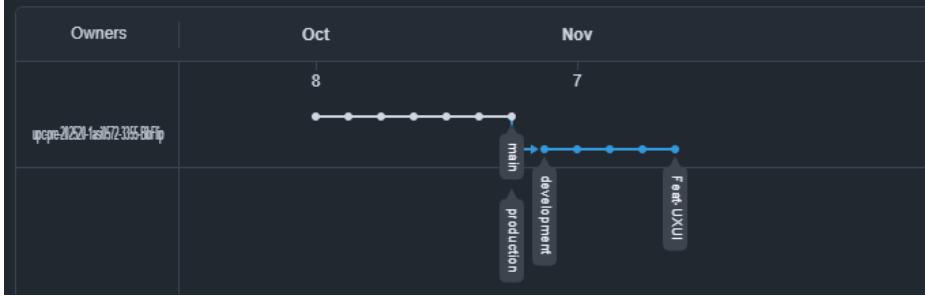
Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



- **Mobile Application:**

Network graph

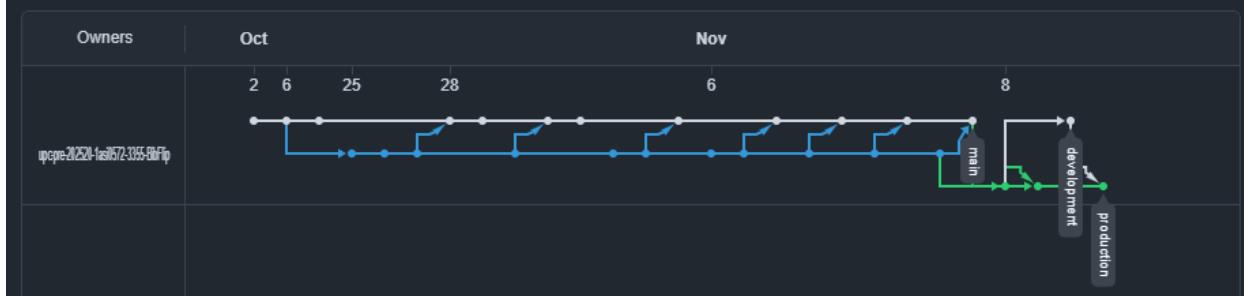
Timeline of the most recent commits to this repository and its network ordered by most recently pushed to.



- **Web Services & IoT Edge:**

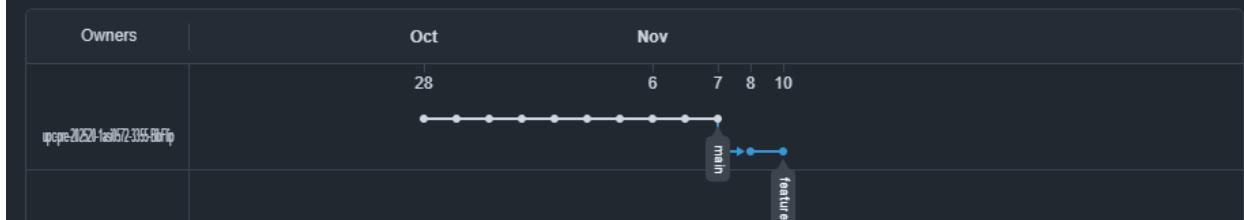
Network graph

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



Network graph

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



6.2.3. Sprint 3

6.2.3.1. Sprint Planning 3

Introducción

Este sprint se enfocó en el **Trabajo Final (TF)**: cerrar la integración extremo a extremo del flujo **IoT Device → Edge API → Backend → Frontends (Web + Mobile)**, estabilizar el producto y dejar lista la evidencia para la demo final (video, validaciones y despliegues en la nube).

Sprint #	Sprint 3			
Sprint Planning Background				
Date	YYYY-MM-DD	2025-11-20		
Time	HH:MM	20:30		
Location	(física/virtual)	Virtual		
Prepared By	(Responsable del acta)	Aranda Vallejos, Oscar Gabriel		
Attendees	(to planning meeting) (Equipo)			
	Aranda Vallejos, Oscar Gabriel / Bernaola Pérez, André Arturo / Gutierrez Garcia, Jose Eduardo / Oliveira Paucar, Mauricio / Pedraza Maldonado, Joaquin / Soriano Medrano, Diego / Velarde Luyo, Piero Alberto			
Sprint 3 – 3 Review Summary				
Al cierre del Sprint 3 se consolidó la integración entre el dispositivo IoT simulado en Wokwi, el Edge API desplegado en Azure, el Backend de reservas y los frontends Web/Mobile. Se estabilizaron los flujos de reserva y disponibilidad, se completaron los escenarios de prueba de extremo a extremo y se generaron las evidencias necesarias para el Trabajo Final (video, documentación y Student Outcomes).				
Sprint 3 – 3 Retrospective Summary				
El equipo identificó que la mayor complejidad se concentró en la integración IoT–Backend, por lo que se incrementó el uso de pair programming y sesiones de depuración conjunta. También se mejoró la disciplina de ramas y pull requests para evitar conflictos en la etapa final. Como mejora futura se propone adelantar actividades de integración y validación en sprints anteriores, para reducir el riesgo acumulado en el último sprint.				
Sprint Goal & User Stories / Technical Stories				
Sprint 3 Goal	(SMART)	En un período máximo de dos semanas, completar la integración en producción de la solución BibFlip, conectando los datos del dispositivo IoT (ocupación de cubículos) con el Edge API y el Backend, exponiendo dicha información en los frontends Web y Mobile y dejando un entorno estable para la demo del Trabajo Final, con pruebas automatizadas y documentación actualizada.		
Sprint 3 Velocity	(Story Points)	30 SP		

Sprint #	Sprint 3	
Sum of Story Points	(stories incluidos)	30 SP (TS008=8, TS009=10, TS010=12)

6.2.3.2. Aspect Leaders and Collaborators.

En este sprint se reforzó un **liderazgo distribuido** orientado a la integración y estabilización del producto.

Aspecto	Líder	Colaboradores
Planificación y coordinación del Sprint	Aranda Vallejos, Oscar Gabriel	Todo el equipo
Integración IoT (Wokwi + Firmware)	Velarde Luyo, Piero Alberto	Gutierrez Garcia, Jose Eduardo / Soriano Medrano, Diego
Edge API & Routing de datos	Soriano Medrano, Diego	Bernaola Pérez, André Arturo / Pedraza Maldonado, Joaquin
Backend de reservas & cubículos	Bernaola Pérez, André Arturo	Pedraza Maldonado, Joaquin / Oliveira Paucar, Mauricio
Frontend Web (estado en tiempo real)	Pedraza Maldonado, Joaquin	Aranda Vallejos, Oscar Gabriel / Velarde Luyo, Piero Alberto
App Mobile (visualización de reserva)	Gutierrez Garcia, Jose Eduardo	Pedraza Maldonado, Joaquin

6.2.3.3. Sprint Backlog 3.

6.2.3.3. Sprint Backlog 3

Durante este sprint se priorizaron **historias técnicas** relacionadas con la integración IoT, la observabilidad del sistema y el cierre de la experiencia de usuario para el demo final del TF.

URL público de Trello: <https://trello.com/b/htxHDgrX/bibflip>

Sprint #		Sprint 3					
Technical Story		Work-Item / Task					
ID	Title	ID	Title	Description	Estimation (Hours)	Assigned To	Status
TS008	Integrar dispositivo IoT con Edge API y Backend	TA1	Ajustar lectura de sensores y payload	Refinar la lógica de lectura de peso/ocupación y el formato del JSON enviado al Edge API.	04	Piero Velarde, Diego Soriano	Done
			TA2	Conectar Edge API con servicios de cubículos	04	Diego Soriano, André Bernaola	Done
		TA3	Validar flujo end-to-end	Verificar que los cambios de ocupación se reflejen en Web y Mobile en tiempo casi real.	02	Jose Gutierrez, Joaquin Pedraza	Done
TS009	Implementar monitoreo, health-checks y logging	TA1	Extender endpoints de salud	Agregar métricas y estados de salud para Edge API y Backend (latencia, conectividad, errores).	03	André Bernaola	Done
			TA2	Configurar logs de integración IoT	03	Diego Soriano	Done
		TA3	Panel básico de monitoreo	Preparar consultas y visualizaciones simples (logs, estados) para revisión en la demo.	02	Oscar Aranda	Done
TS010	Cierre de experiencia de usuario y demo TF	TA1	Ajustar vistas de disponibilidad	Refinar UI Web/Mobile para mostrar claramente cubículos libres/ocupados y reservas activas.	04	Joaquin Pedraza, Jose Gutierrez	Done
			TA2	Grabar video About-the-Product	03	Piero Velarde, Mauricio Oliveira	Done
		TA3	Actualizar documentación y Student Outcome	Incorporar evidencias finales en el informe, Student Outcome y anexos.	03	Mauricio Oliveira, Oscar Aranda	Done

6.2.3.4. Development Evidence for Sprint Review.

Organización del github: <https://github.com/upc-pre-202520-1asi0572-3355-BibFlip>

Repository	Branch	Commit Id	Commit Message	Commit Message Body	Committed on (Date)
LandingPage	main	82d64e	feat: update features section	-	02/12/25
Web-Frontend	develop	7d4676f	feat: update sign in request	-	6/11/25
Mobile-Frontend	develop	420d355	feat: update cubicle bookings hours	-	8/10/25
Edge-Api-Platform	main	e881e0d	Feat: Implement connection with backend and persistence to database	-	15/11/25
ESP32-BibFlip-Embedded-System	main	a11c3d0	Feat: Update project from wokwi	-	15/11/25
Backend	production	95baaea	Feat: Put time zone config commented	-	15/11/25
Backend	production	e63579	Feat: Reupdate the time zone for better synchronization with edge api	-	15/11/25

6.2.3.5. Testing Suite Evidence for Sprint Review.

US17- Registro de cuenta de administrador- Segmento SuperAdmin

The screenshot displays the Bibflip application interface and the Selenium IDE tool. The application window shows a map of Lima, Peru with a red marker indicating the location. Below the map, there are input fields for 'Hora de Apertura' (10:00), 'Hora de Cierre' (23:00), and 'Intervalo de Reservas (minutos)' (100). The Selenium IDE panel on the right lists the following test steps:

```

Project: Blipflip Test SuperAdmin VF*
Executing: https://bib-flip-web-app-2025-02.web.app
US 17 Superadmin*
Command Target Value
61 click #intervalMinutes
62 type #intervalMinutes 120
63 click #openingTime
64 run script window.scrollTo(0,0)
65 click #closingTime
66 click #intervalMinutes
67 click #openingTime
68 click #closingTime
69 click #intervalMinutes
70 click #openingTime
71 click #closingTime
72 click #intervalMinutes
73 click #openingTime
74 type #intervalMinutes 100
75 click #closingTime
76 type #openingTime 10.00
77 type #closingTime 23.00
78 click #bttn-primary
    
```

The Selenium log at the bottom shows the execution results:

```

Log Reference
74: type on #intervalMinutes with value 120 OK
75: click on #openingTime OK
76: type on #openingTime with value 10.00 OK
77: type on #closingTime with value 23.00 OK
78: click on #bttn-primary OK
"US 17 Superadmin" completed successfully
Activar Windows
Ve a Configuración para activar Windows.
21:38:15
21:38:15
21:38:15
21:38:16
21:38:16
    
```

US08 Reserva de cubículos -Segmento Estudiante

The screenshot shows a browser window with the URL <https://bib-flip-web-app-2025-02.web.app/headquarters/3/cubicles>. On the left, there's a sidebar with filters like 'Cubículos' and 'Todas las capacidades'. In the center, a modal titled 'Reservar Cubículo' is open, showing 'Cubiculo: Cubículo 2' and 'Capacidad: 4 personas'. It has a date input set to '02/12/2025' and a grid of time slots from 07:00 to 21:00. The slot from 07:00 to 11:00 is highlighted in red, indicating it's selected. At the bottom are 'Cancelar' and 'Confirmar (0 horarios)' buttons. On the right, the Selenium IDE log shows 16 commands for the reservation process.

Command	Target	Value
1	open	/sign-in
2	set window size	1936x1048
3	click	id=username
4	type	id=username Luis2025
5	type	id=password @Luis2025
6	click	css=p-button
7	click	css=p-button-outlined>.p-button-label
8	click	css=headquarter-card:nth-child(3) .btn-cubicles
9	click	id=capacityFilter
10	click	id=capacityFilter
11	click	id=capacityFilter
12	click	css=main-content:nth-child(2)
13	click	css=cubicule-card:nth-child(2)> .btn-reserve
14	click	css=time-slot:nth-child(3)> .slot-time
15	click	css=btn-reserve:nth-child(2)
16	click	css=btn-success

6.2.3.6. Execution Evidence for Sprint Review.

- **Landing Page:**

The screenshot shows a web browser displaying the Bibflip landing page at <https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>. The landing page features a banner with the text "Gestiona tus cubículos fácilmente con tecnología IoT" and a subtitle about real-time monitoring of study spaces at the Universidad Peruana de Ciencias Aplicadas (UPC). Below the banner is a "Conocer más" button. At the top of the page, there is a navigation bar with links to "¿Qué es?", "Características", "Cómo usar", "Testimonios", and "Sobre nosotros".

Below the landing page, the browser's address bar shows the URL `upc-pre-202520-1asi0572-3355-BibFlip / LandingPage`. The GitHub repository interface is visible, with tabs for "Code", "Issues", "Pull requests", "Actions", "Projects", "Security", "Insights", and "Settings". The "Settings" tab is currently selected.

In the "General" section of the Settings page, under "GitHub Pages", it is indicated that the site is live at <https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>, last deployed by Mauricio-nn yesterday. There are "Visit site" and "Unpublish site" buttons. The "Source" dropdown is set to "GitHub Actions". Under "Build and deployment", it notes that the site was last deployed to the `github-pages` environment by the [Deploy to GitHub Pages workflow](#). A link to "Learn more about deploying to GitHub Pages using custom workflows" is provided. In the "Custom domain" section, it says that custom domains allow serving the site from a different domain than `upc-pre-202520-1asi0572-3355-bibflip.github.io`, with a link to "Learn more about configuring custom domains".

- **Web Application:**

Bibflip | Sign In

<https://bib-flip-web-app-2025-02.web.app/sign-in>

Iniciar Sesión

Nombre
Coloca tu nombre de usuario

Contraseña
Coloca tu contraseña

Iniciar sesión

¿No tienes una cuenta todavía? [Crear cuenta](#)

Bibflip | Home

<https://bib-flip-web-app-2025-02.web.app/home>

Mis Reservas

Viernes, 14 de noviembre de 2025

Cubiculo 3

Sede: San Miguel

18:00 - 19:00

19:00 - 20:00

Viernes, 14 de noviembre de 2025

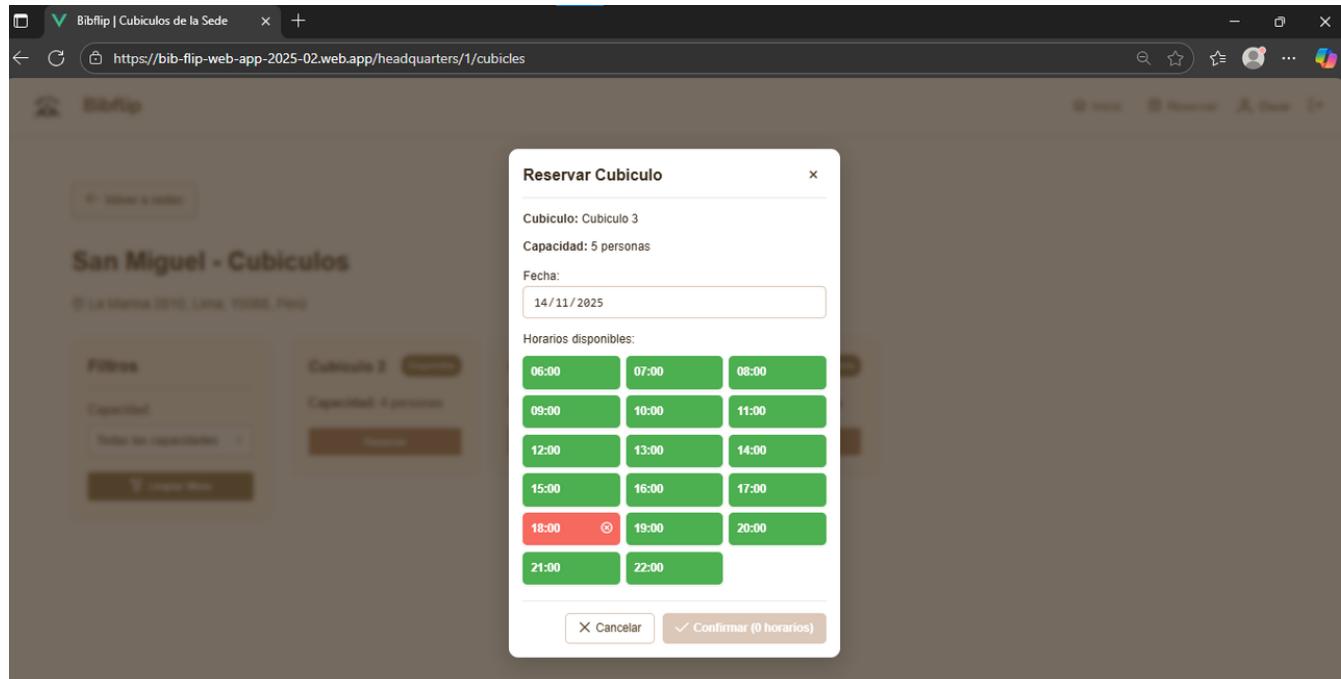
Cubiculo 3

Sede: San Miguel

19:00 - 20:00

The screenshot shows a web browser window for the Bibflip website. The title bar says "Bibflip | Reservas". The URL is "https://bib-flip-web-app-2025-02.web.app/headquarters". The page header includes the Bibflip logo, a search icon, a star icon, a profile icon for "Oscar", and a menu icon. Below the header, there's a navigation bar with links for "Inicio", "Reservar", and "Oscar". A main heading "Conoce nuestras sedes y reserva!" is displayed. A section for "San Miguel" is shown, featuring a coffee cup and map icon, address "La Marina 2810, Lima, 15088, Perú", operating hours (Monday-Friday 9:00 am - 8 pm, Saturday 10:00 am - 6 pm, Sunday 10:00 am - 5 pm), and two buttons: "Ver cubículos" and "Ver en Mapa".

The screenshot shows a web browser window for the Bibflip website. The title bar says "Bibflip | Cubículos de la Sede". The URL is "https://bib-flip-web-app-2025-02.web.app/headquarters/1/cubicles". The page header and navigation bar are identical to the previous screenshot. A back button "Volver a sedes" is visible. The main content is titled "San Miguel - Cubículos" and shows the address "La Marina 2810, Lima, 15088, Perú". On the left, a "Filtros" sidebar allows filtering by capacity (with a dropdown menu showing "Todas las capacidades" and a "Limpiar filtros" button). Three cubicles are listed: "Cubículo 2" (Capacity: 4 people, Available, Reservar button), "Cubículo 1" (Capacity: 4 people, Available, Reservar button), and "Cubículo 3" (Capacity: 5 people, Available, Reservar button).



Reservar Cubiculo

Cubículo: Cubículo 3
Capacidad: 5 personas
Fecha: 14/11/2025
Horarios disponibles:

06:00	07:00	08:00
09:00	10:00	11:00
12:00	13:00	14:00
15:00	16:00	17:00
18:00	19:00	20:00
21:00	22:00	



Reserva Exitosa

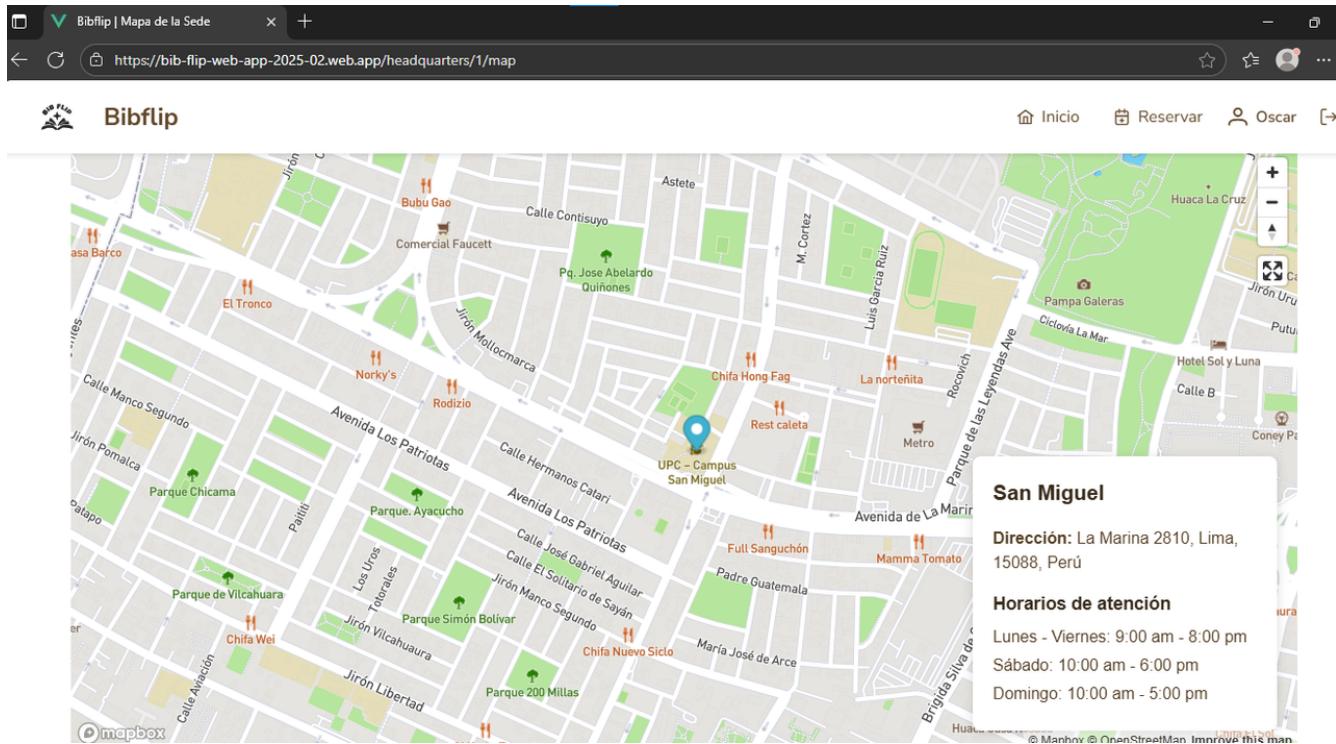
✓ Reserva confirmada!

CUBÍCULO:
Cubículo 3

CAPACIDAD:
5 personas

FECHA:
2025-11-14

HORARIOS:
19:00



- **Web Services:**

<https://bibflip-api-platform.azurewebsites.net/swagger/docs/v3/index.html>

BibFlip API Platform REST API

Swagger UI Documentation | Contact | Help | Home | Apache 2.0

Servers: https://bibflip-api-platform.azurewebsites.net - Generated server url [Authorize](#)

Headquarter Supervisors Headquarter Supervisor Management Endpoints

- POST** /api/v1/headquarters/{headquarterId}/supervisors/{supervisorId} Assign supervisor to headquarter
- DELETE** /api/v1/headquarters/{headquarterId}/supervisors/{supervisorId} Remove supervisor from headquarter
- GET** /api/v1/headquarters/{headquarterId}/supervisors Get all supervisors for a headquarter
- GET** /api/v1/headquarters/supervisors/{supervisorId} Get headquarter by supervisor ID

Authentication Authentication Endpoints

- POST** /api/v1/authentication/sign-up
- POST** /api/v1/authentication/sign-in

Password Recovery Password Recovery Endpoints

- POST** /api/v1/password-recovery/request
- POST** /api/v1/password-recovery/confirm

Booking Booking Management Endpoints

- GET** /api/v1/bookings
- POST** /api/v1/bookings
- GET** /api/v1/bookings/{id}
- DELETE** /api/v1/bookings/{id}
- GET** /api/v1/bookings/client/{clientId}

Cubicle Cubicle Management Endpoints

- GET** /api/v1/cubicles
- POST** /api/v1/cubicles
- PATCH** /api/v1/cubicles/{cubicleId}/status
- PATCH** /api/v1/cubicles/{cubicleId}/availability-alert/status
- GET** /api/v1/cubicles/{cubicleId}
- DELETE** /api/v1/cubicles/{cubicleId}
- GET** /api/v1/cubicles/{cubicleId}/schedule
- GET** /api/v1/cubicles/headquarter/{headquarterId}

Headquarters Headquarter Management Endpoints

- GET** /api/v1/headquarters
- POST** /api/v1/headquarters
- GET** /api/v1/headquarters/{headquarterId}

Roles Role Management Endpoints

- GET** /api/v1/roles

Users User Management Endpoints

- GET** /api/v1/users
- GET** /api/v1/users/{userId}

- **Mobile (Flutter):**

RUN AND DEBUG ▶ No Configur.▼ ⚙ ... Android Emulator - Medium_Phone_API_36:0:5554
02:07 ⓘ G

VARIABLES

WATCH

CALL STACK Running

BREAKPOINTS

- All Exceptions
- Uncaught Exceptions

development ⌂ ⌂ ⌂ Launchpad ⌂ 0 △ 0

RUN AND DEBUG ▶ No Configur.▼ ⚙ ... Android Emulator - Medium_Phone_API_36:0:5554
02:08 ⓘ G

VARIABLES

WATCH

CALL STACK Running

RESERVA DE CUBÍCULOS

Reservar cubículo

Información del cubículo:

Cubículo: Cubículo #2
Capacidad: 4 personas
Zona: pabellón principal

Fecha de reserva:

viernes 10, octubre 2025

Selección de horario:

Hora	Estado
06:00 - 06:30	Disponible
06:30 - 07:00	Disponible
07:00 - 07:30	Disponible
07:30 - 08:00	Disponible
08:00 - 08:30	Disponible
08:30 - 09:00	Disponible
09:00 - 09:30	Disponible

187 / 210

The image shows two screenshots of a mobile application interface. The top screenshot displays a login screen with fields for 'Usuario' and 'Contraseña', a 'Iniciar Sesión' button, and a 'Regístrate' link. The bottom screenshot shows a reservation overlay titled 'Reservar cubículo' with information about the cubicle (Cubículo #2, 4 people, pabellón principal), a date selector set to Friday, October 10, 2025, and a list of time slots from 06:00 to 09:30, all marked as 'Disponible' (Available). The application is running on an Android emulator with API level 36.



- Edge Api

<https://bibflip-edge-api-platform.azurewebsites.net/api/redoc>

BibFlip IoT Edge API (2.0.0)

Download OpenAPI specification: [Download](#)

Edge API for IoT chair sensors with PostgreSQL persistence

IoT Device Monitoring

Register Device

Register a new IoT device in Edge API and sync with backend. Device stored in PostgreSQL.

REQUEST BODY SCHEMA: application/json

required

device_id	string (Device Id) <code>>= 3 characters</code>	Unique device identifier
device_type	string (Device Type)	Type of device (chair_sensor, table_sensor, environmental)
branch_id	string (Branch Id)	Branch ID where device is located

API docs by Redocly

POST /api/v1/devices/register

Request samples

Payload

Content type
application/json

```
{
  "device_id": "string",
  "device_type": "string",
  "branch_id": "string",
  "zone": "string",
  "position": "string"
}
```

Copy

https://bibflip-edge-api-platform.azurewebsites.net/api/redoc#tag/IoT-Device-Monitoring/operation/register_device_api_v1_devices_register.post

Content type
application/json

```
{
  "device_id": "string",
  "device_type": "string",
  "branch_id": "string",
  "zone": "string",
  "position": "string"
}
```

Copy

Responses

201 Successful Response

422 Validation Error

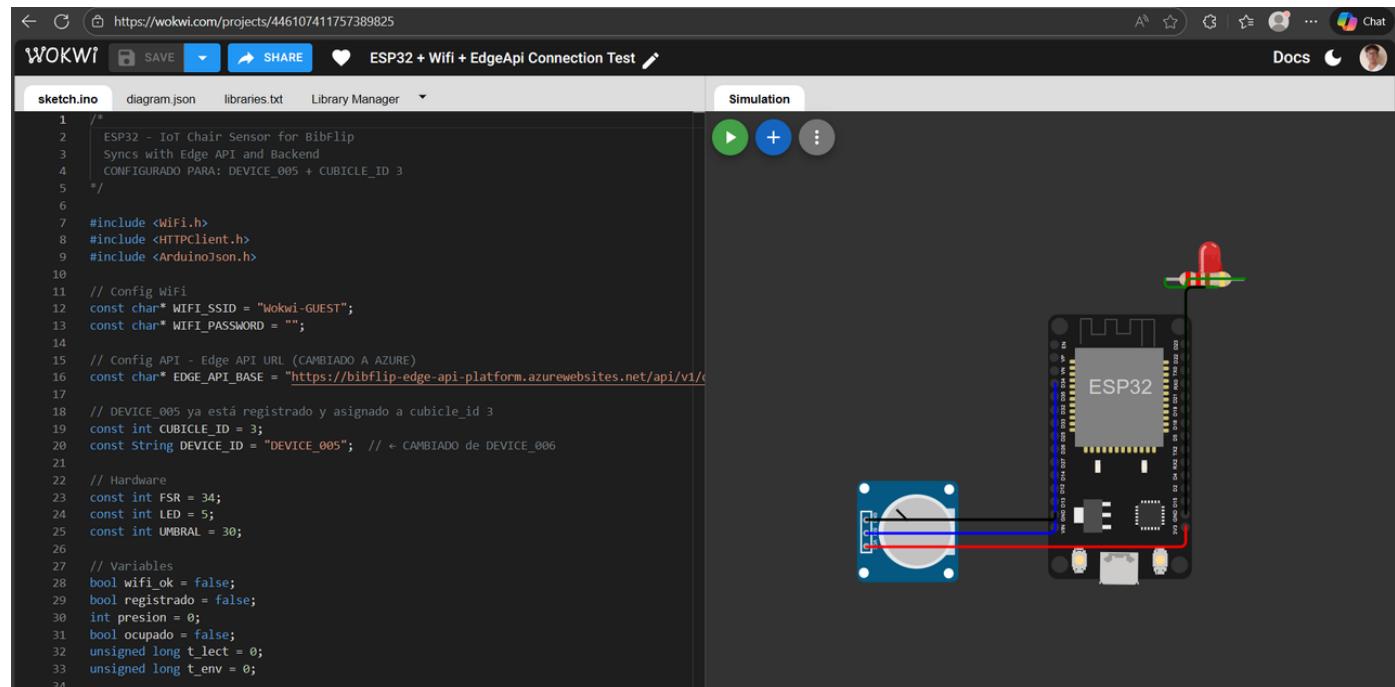
Content type
application/json

```
{
  "id": "string",
  "type": "string",
  "status": "string",
  "location": {},
  "cubicle_id": 0,
  "last_reading": {},
  "last_update": "string"
}
```

Copy Expand all Collapse all

API docs by Redocly

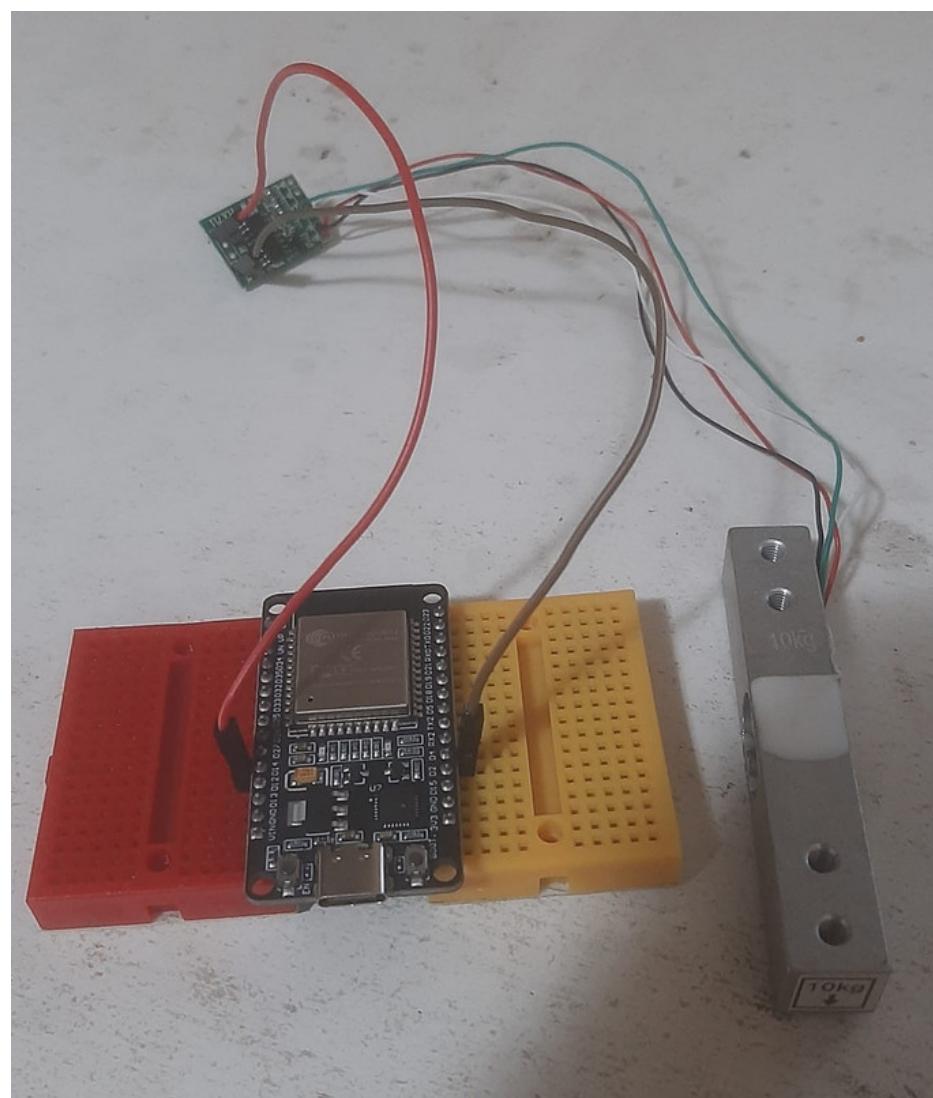
- Embedded System

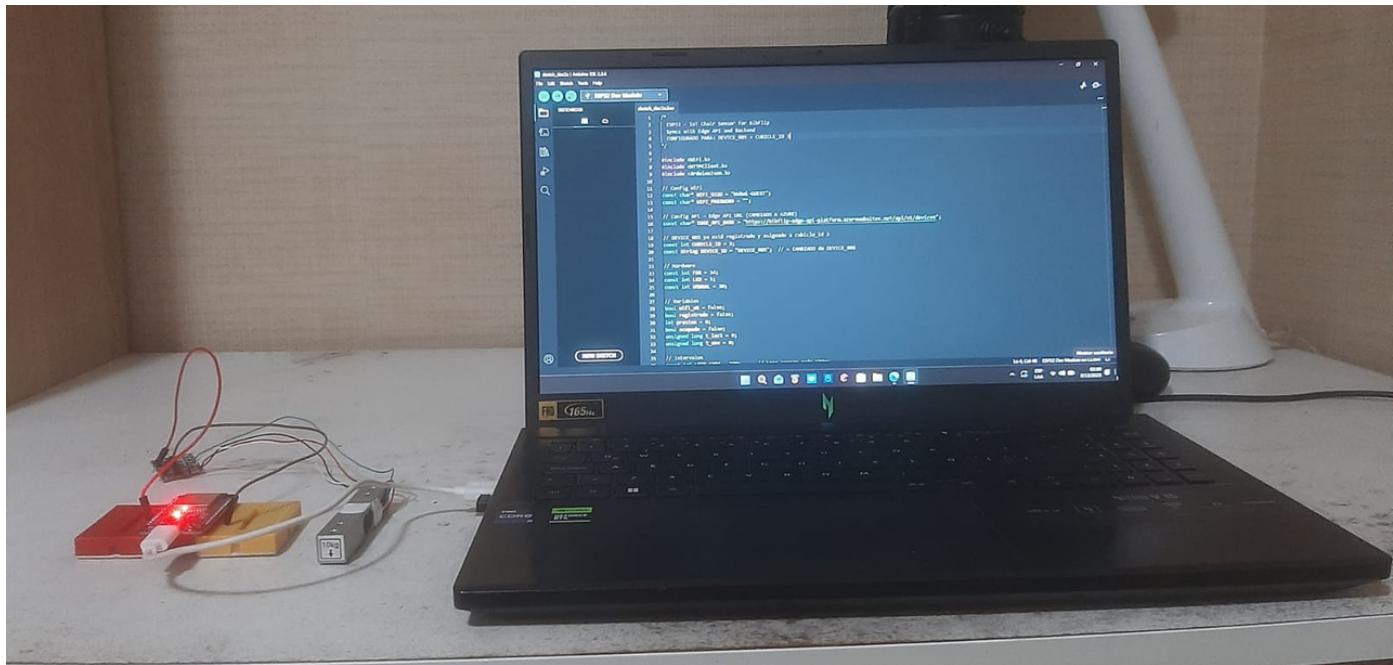


The screenshot shows the Wokwi simulation interface. On the left, the `sketch.ino` file contains the following Arduino-like code:

```
1 // IoT Chair Sensor for BibFlip
2 Syncs with Edge API and Backend
3 CONFIGURADO PARA: DEVICE_005 + CUBICLE_ID 3
4 */
5
6 #include <WiFi.h>
7 #include <HTTPClient.h>
8 #include <ArduinoJson.h>
9
10 // Config WiFi
11 const char* WIFI_SSID = "Wokwi-GUEST";
12 const char* WIFI_PASSWORD = "";
13
14 // Config API - Edge API URL (CAMBIADO A AZURE)
15 const char* EDGE_API_BASE = "https://bibflip-edge-api-platform.azurewebsites.net/api/v1/";
16
17 // DEVICE_005 ya está registrado y asignado a cubicle_id 3
18 const int CUBICLE_ID = 3;
19 const String DEVICE_ID = "DEVICE_005"; // + CAMBIADO de DEVICE_006
20
21 // Hardware
22 const int FSR = 34;
23 const int LED = 5;
24 const int UMBRAL = 30;
25
26 // Variables
27 bool wifi_ok = false;
28 bool registrado = false;
29 int presion = 0;
30 bool ocupado = false;
31 unsigned long t_lect = 0;
32 unsigned long t_env = 0;
33
```

The right side of the interface shows a digital circuit simulation. It features an ESP32 microcontroller connected to a breadboard. On the breadboard, there is a blue circular component (likely a pushbutton or sensor) connected to digital pins D1 and D2. A red LED is connected to digital pin D5 through a resistor. A green component (possibly a relay or driver) is also connected to the breadboard. The simulation includes a play button and other controls.





- **Link video de visualización y navegación logrado en este Sprint:**

https://upcdeupe-my.sharepoint.com/:v/g/personal/u202218167_upc_edu_pe/IQABQOS8HltySZW1F3XMy1nhARQ1f3P-irvGt6_ZWdZo9Po?#nav=eyJZWlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAIoJPbmVEcmI2ZUZvckJ1c2luZXNzliwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYilsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWlcnJhbFZpZXciOjNeUZpbGVzTGlua0NvcHkifX0&e=md2xAV

6.2.3.7. Services Documentation Evidence for Sprint Review.

En esta sección se presenta los principales endpoints de la API del sistema, desarrollados para este Sprint, organizados por bounded context como autenticación, recuperación de contraseña, reservas, cubículos, sedes, usuarios y la gestión de supervisores. Su documentación busca ofrecer una referencia clara para comprender las acciones disponibles, los parámetros requeridos y las respuestas esperadas en cada operación, facilitando así el desarrollo, la integración y el mantenimiento de nuestra solución.

Sedes y Supervisores

Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Asignar supervisor	POST	/api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}	Path: headquarterId, supervisorId	POST /api/v1/headquarters/1/supervisors/5	Mensaje de éxito
Remover supervisor	DELETE	/api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}	Path: headquarterId, supervisorId	DELETE /api/v1/headquarters/1/supervisors/5	Mensaje de éxito
Obtener supervisores	GET	/api/v1/headquarters/{headquarterId}/supervisors	Path: headquarterId	GET /api/v1/headquarters/1/supervisors	Lista de supervisores
Obtener sede por supervisor	GET	/api/v1/headquarters/supervisors/{supervisorId}	Path: supervisorId	GET /api/v1/headquarters/supervisors/5	Datos de la sede asignada

Autenticación

Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Registro	POST	/api/v1/authentication/sign-up	Body: { name, email, password }	POST /api/v1/authentication/sign-up con JSON en el cuerpo	Usuario creado con token de autenticación
Inicio de sesión	POST	/api/v1/authentication/sign-in	Body: { email, password }	POST /api/v1/authentication/sign-in	Devuelve token y datos del usuario

Recuperación de Contraseña

Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Solicitar recuperación	POST	/api/v1/password-recovery/request	Body: { email }	POST /api/v1/password-recovery/request	Mensaje de confirmación de envío de correo

Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Confirmar recuperación	POST	/api/v1/password-recovery/confirm	Body: { token, newPassword }	POST /api/v1/password-recovery/confirm	Mensaje de éxito o error
Reservas					
Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Listar reservas	GET	/api/v1/bookings	Ninguno	GET /api/v1/bookings	Lista de reservas
Crear reserva	POST	/api/v1/bookings	Body: { clientId, cubicleId, date, time }	POST /api/v1/bookings	Reserva creada
Obtener reserva	GET	/api/v1/bookings/{id}	Path: id	GET /api/v1/bookings/10	Detalles de la reserva
Eliminar reserva	DELETE	/api/v1/bookings/{id}	Path: id	DELETE /api/v1/bookings/10	Mensaje de éxito
Reservas por cliente	GET	/api/v1/bookings/client/{clientId}	Path: clientId	GET /api/v1/bookings/client/3	Lista de reservas del cliente
Cubículos					
Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Listar cubículos	GET	/api/v1/cubicles	Ninguno	GET /api/v1/cubicles	Lista de cubículos
Crear cubículo	POST	/api/v1/cubicles	Body: { name, headquarterId, availability }	POST /api/v1/cubicles	Cubículo creado
Actualizar estado	PATCH	/api/v1/cubicles/{cubicleId}/status	Body: { status }	PATCH /api/v1/cubicles/4/status	Estado actualizado
Actualizar disponibilidad	PATCH	/api/v1/cubicles/{cubicleId}/availability-slot/status	Body: { slotId, status }	PATCH /api/v1/cubicles/4/availability-slot/status	Slot actualizado
Obtener cubículo	GET	/api/v1/cubicles/{cubicleId}	Path: cubicleId	GET /api/v1/cubicles/4	Detalles del cubículo
Eliminar cubículo	DELETE	/api/v1/cubicles/{cubicleId}	Path: cubicleId	DELETE /api/v1/cubicles/4	Mensaje de éxito
Horario del cubículo	GET	/api/v1/cubicles/{cubicleId}/schedule	Path: cubicleId	GET /api/v1/cubicles/4/schedule	Lista de horarios
Cubículos por sede	GET	/api/v1/cubicles/headquarter/{headquarterId}	Path: headquarterId	GET /api/v1/cubicles/headquarter/1	Lista de cubículos en la sede
Sedes					
Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Listar sedes	GET	/api/v1/headquarters	Ninguno	GET /api/v1/headquarters	Lista de sedes
Crear sede	POST	/api/v1/headquarters	Body: { name, address, openingTime, closingTime }	POST /api/v1/headquarters	Sede creada
Obtener sede	GET	/api/v1/headquarters/{headquarterId}	Path: headquarterId	GET /api/v1/headquarters/1	Detalles de la sede
Roles y Usuarios					
Acción	Verbo	Endpoint	Parámetros	Ejemplo	Response
Listar roles	GET	/api/v1/roles	Ninguno	GET /api/v1/roles	Lista de roles
Listar usuarios	GET	/api/v1/users	Ninguno	GET /api/v1/users	Lista de usuarios
Obtener usuario	GET	/api/v1/users/{userId}	Path: userId	GET /api/v1/users/7	Detalles del usuario
IoT Edge					
Acción	Verbo HTTP	Endpoint	Parámetros	Ejemplo	Response

Acción	Verbo HTTP	Endpoint	Parámetros	Ejemplo	Response
Registrar dispositivo	POST	/api/v1/devices/register	Body: { "device_id": "string", "location": "string", "type": "string" }	POST /api/v1/devices/register Body: { "device_id": "chair-001", "location": "Sala A", "type": "sensor" }	Device registered successfully
Actualizar lectura	POST	/api/v1/devices/{device_id}/readings	Path: device_id Body: { "temperature": number, "humidity": number, "occupied": boolean }	POST /api/v1/devices/chair-001/readings Body: { "temperature": 22.5, "humidity": 45, "occupied": true }	Reading updated
Obtener dispositivos disponibles	GET	/api/v1/devices/status/available	Ninguno	GET /api/v1/devices/status/available	Available
Obtener dispositivos ocupados	GET	/api/v1/devices/status/occupied	Ninguno	GET /api/v1/devices/status/occupied	Sala C, Occupied
Verificar salud del backend	GET	/api/v1/devices/health/backend	Ninguno	GET /api/v1/devices/health/backend	Status: ok, Timestamp: 2025-11-14T18:00:00Z
Verificar dispositivos offline	POST	/api/v1/devices/maintenance/check-offline	Ninguno	POST /api/v1/devices/maintenance/check-offline	Status: "offline"
Obtener dispositivo por ID	GET	/api/v1/devices/{device_id}	Path: device_id	GET /api/v1/devices/chair-001	device_id: "chair-001"
Eliminar dispositivo	DELETE	/api/v1/devices/{device_id}	Path: device_id	DELETE /api/v1/devices/chair-001	Device deleted
Obtener todos los dispositivos	GET	/api/v1/devices/	Ninguno	GET /api/v1/devices/	device_id: "chair-001", ...
Verificar salud general	GET	/health	Ninguno	GET /health	status: "ok"

Imagen de documentación con OpenAPI:

The screenshot shows the BibFlip API Platform's Swagger UI interface. At the top, it displays "BibFlip API Platform" with "VIEW" and "OAS 3.1" buttons. Below this is a navigation bar with links for "Swagger", "About", "Contact", "Apache 2.0", and "Logout". The main content area is titled "BibFlip API Platform" and contains sections for "Headquarter Supervisors", "Authentication", "Password Recovery", "Booking", "Cubicle", "Headquarters", "Roles", and "Users". Each section lists various HTTP methods (e.g., POST, GET, DELETE) and their corresponding URLs and descriptions. The "Headquarter Supervisors" section includes endpoints like "/api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}" for assigning supervisors and "/api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}" for removing them. The "Booking" section includes endpoints for creating, getting, and deleting bookings. The "Cubicle" section includes endpoints for managing vehicle status and availability. The "Headquarters" section includes endpoints for managing headquarters. The "Roles" section includes an endpoint for roles. The "Users" section includes endpoints for users.

BibFlip IoT Edge API 2.0.0 OAS 3.1[/openapi.json](#)

Edge API for IoT chair sensors with backend synchronization

IoT Device Monitoring ^

The screenshot shows the BibFlip IoT Edge API's Swagger UI interface, specifically the "IoT Device Monitoring" section. It lists several endpoints:

- POST /api/v1/devices/register** Register Device
- POST /api/v1/devices/{device_id}/readings** Update Device Reading
- GET /api/v1/devices/status/available** Get Available Devices
- GET /api/v1/devices/status/occupied** Get Occupied Devices
- GET /api/v1/devices/health/backend** Check Backend Health
- POST /api/v1/devices/maintenance/check-offline** Check Offline Devices
- GET /api/v1/devices/{device_id}** Get Device
- DELETE /api/v1/devices/{device_id}** Delete Device
- GET /api/v1/devices/** Get All Devices

URL de documentación desplegada de Web Services:<https://bibflip-api-platform.azurewebsites.net/swagger-ui/index.html>

URL de documentación desplegada de IoT Edge:<https://bibflip-edge-api-platform.azurewebsites.net/api/docs>**URL del repositorio de Web Services:**<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Backend>**URL del repositorio de IoT Edge:**<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Edge-Api-Platform>**6.2.3.8. Software Deployment Evidence for Sprint Review.**• **Landing Page:**GitHub Pages — URL: <https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>

The screenshot shows the GitHub Pages settings interface for a repository named "upc-pre-202520-1asi0572-3355-bibflip". The left sidebar has a "Pages" section selected. The main area displays the "GitHub Pages" configuration, which is active and points to the URL <https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>. It also shows the last deployment was made by Mauricio-nn yesterday. Below this, there's a "Build and deployment" section using GitHub Actions, and a "Custom domain" section.

• **Web Application:**Firebase Hosting — URL: <https://bib-flip-web-app-2025-02.web.app/>

The screenshot shows the Firebase console interface for creating a new project. The project name is "bib-flip-web-app-2025-02". The "Nombre del proyecto" field contains the same text. A button at the bottom right is labeled "bib-flip-web-app-2025-02-de7b5".

1. Construir aplicación Vue

Genera la carpeta dist lista para producción:

```
npm run build
```

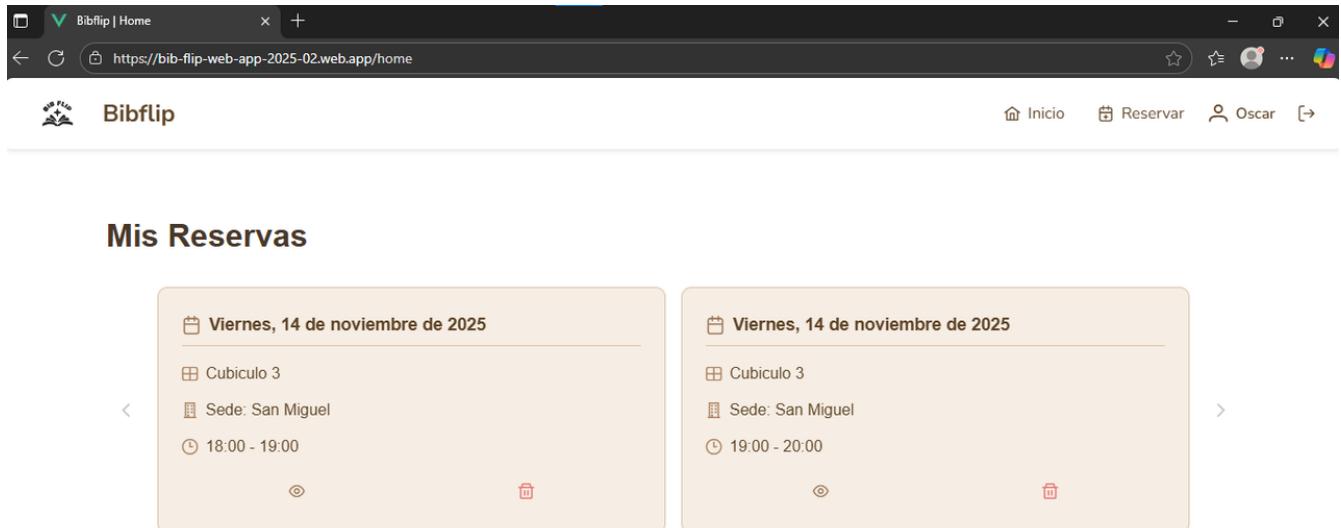
2. Inicializar Firebase en el proyecto

Dentro de la carpeta del proyecto ejecuta:

```
firebase init
```

3. Desplegar la aplicación

Desplegar la aplicación



- **Mobile Application:**

Build APK de pruebas internas.

RUN AND DEBUG ▶ No Configur.▼ ⚙ ... Android Emulator - Medium_Phone_API_36:0:5554
02:07 ⓘ G

VARIABLES

WATCH

CALL STACK Running

BREAKPOINTS

- All Exceptions
- Uncaught Exceptions

development ⌂ ⌂ ⌂ Launchpad ⌂ 0 △ 0

RUN AND DEBUG ▶ No Configur.▼ ⚙ ... Android Emulator - Medium_Phone_API_36:0:5554
02:08 ⓘ G

VARIABLES

WATCH

CALL STACK Running

RESERVA DE CUBÍCULOS

Reservar cubículo

Información del cubículo:

Cubículo: Cubículo #2
Capacidad: 4 personas
Zona: pabellón principal

Fecha de reserva:

viernes 10, octubre 2025

Selección de horario:

Hora	Estado
06:00 - 06:30	Disponible
06:30 - 07:00	Disponible
07:00 - 07:30	Disponible
07:30 - 08:00	Disponible
08:00 - 08:30	Disponible
08:30 - 09:00	Disponible
09:00 - 09:30	Disponible

196 / 210



- **Web Services y IoT Edge:**

Web Services — URL: <https://bibflip-api-platform.azurewebsites.net/swagger-ui/index.html>

IoT Edge — URL: <https://bibflip-edge-api-platform.azurewebsites.net/api/docs>

App Services Universidad Peruana de Ciencias (upc.edu.pe)

+ Crear ...

ⓘ [Está viendo una nueva versión de la experiencia de exploración. Haga clic aquí para acceder a la experiencia anterior.](#)

Nombre ↑
<input type="checkbox"/> bean-detect-ai-api-platform
<input checked="" type="checkbox"/> bibflip-api-platform
<input type="checkbox"/> bibflip-edge-api-platform
<input type="checkbox"/> smart-band-edge-api-platform-gru

Resumen

Id. de sesión
e2461d26a5f848cdb8067fbe5d577614

Identificador de recurso
No disponible

Extensión
HubsExtension

Contenido
ResourceMenuBlade

Código de error
0

Detalles
Se agotó el tiempo de espera de la solicitud o el explorador rechazó la conexión. Consulte el estado de la red,

BibFlip API Platform v1.0.0 OAS 3.1

/v3/api-docs

BibFlip REST API documentation.

Contact BibFlip Help Team

Apache 2.0

Servers: https://bibflip-api-platform.azurewebsites.net - Generated server url [Authorize](#)

Headquarter Supervisors Headquarter Supervisor Management Endpoints

POST	/api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}	Assign supervisor to headquarter	
DELETE	/api/v1/headquarters/{headquarterId}/supervisors/{supervisorId}	Remove supervisor from headquarter	
GET	/api/v1/headquarters/{headquarterId}/supervisors	Get all supervisors for a headquarter	
GET	/api/v1/headquarters/supervisors/{supervisorId}	Get headquarter by supervisor ID	

BibFlip IoT Edge API 2.0.0 OAS 3.1

[/openapi.json](#)

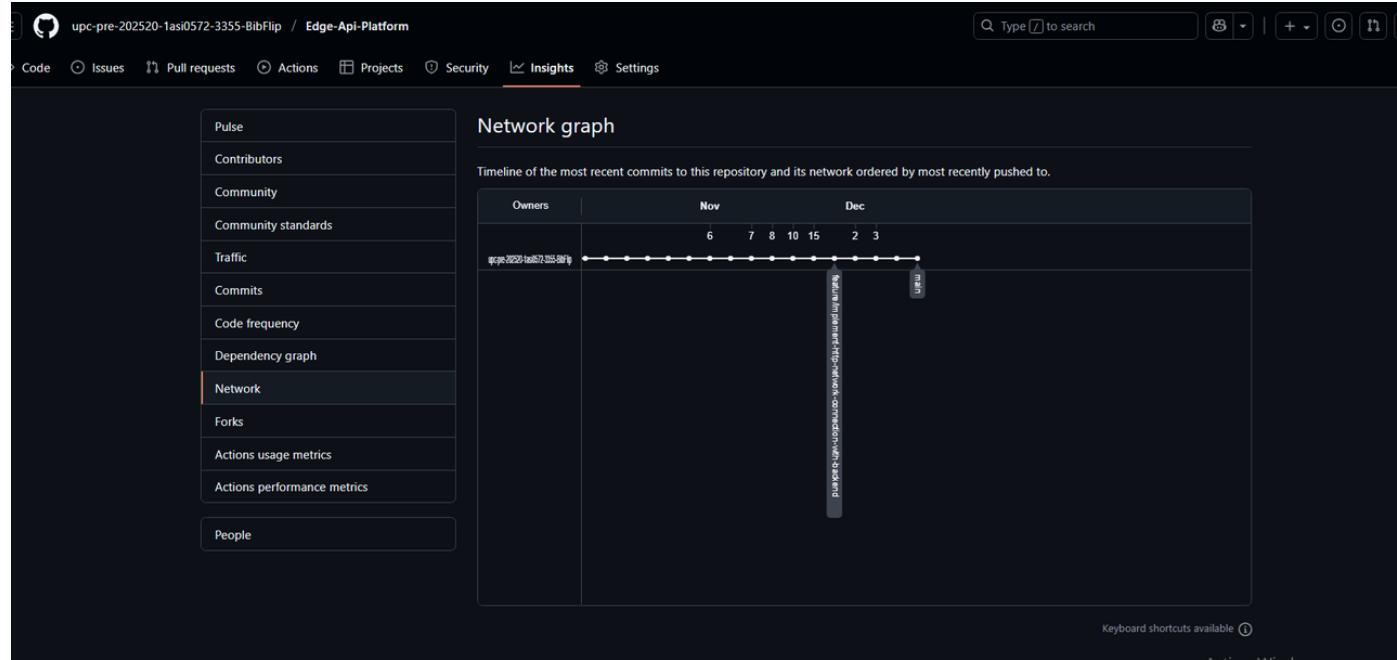
Edge API for IoT chair sensors with backend synchronization

IoT Device Monitoring

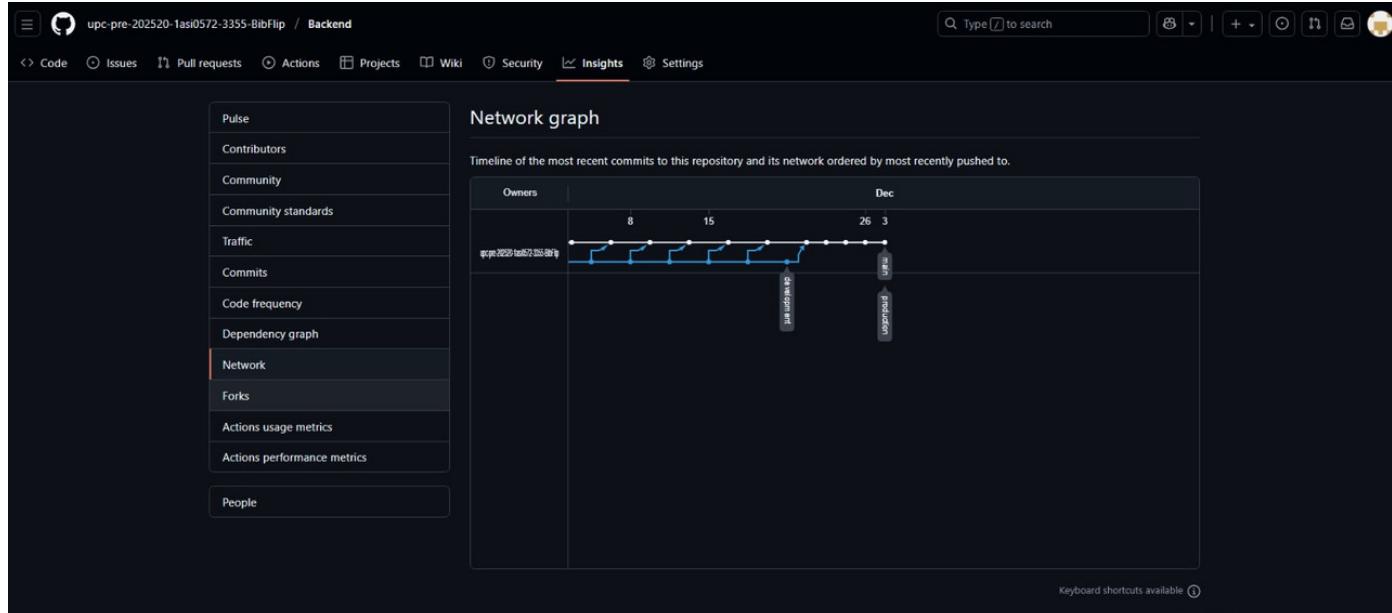
<code>POST</code>	<code>/api/v1/devices/register</code>	Register Device	⌄
<code>POST</code>	<code>/api/v1/devices/{device_id}/readings</code>	Update Device Reading	⌄
<code>GET</code>	<code>/api/v1/devices/status/available</code>	Get Available Devices	⌄
<code>GET</code>	<code>/api/v1/devices/status/occupied</code>	Get Occupied Devices	⌄
<code>GET</code>	<code>/api/v1/devices/health/backend</code>	Check Backend Health	⌄
<code>POST</code>	<code>/api/v1/devices/maintenance/check-offline</code>	Check Offline Devices	⌄
<code>GET</code>	<code>/api/v1/devices/{device_id}</code>	Get Device	⌄
<code>DELETE</code>	<code>/api/v1/devices/{device_id}</code>	Delete Device	⌄
<code>GET</code>	<code>/api/v1/devices/</code>	Get All Devices	⌄

6.2.3.9. Team Collaboration Insights during Sprint.

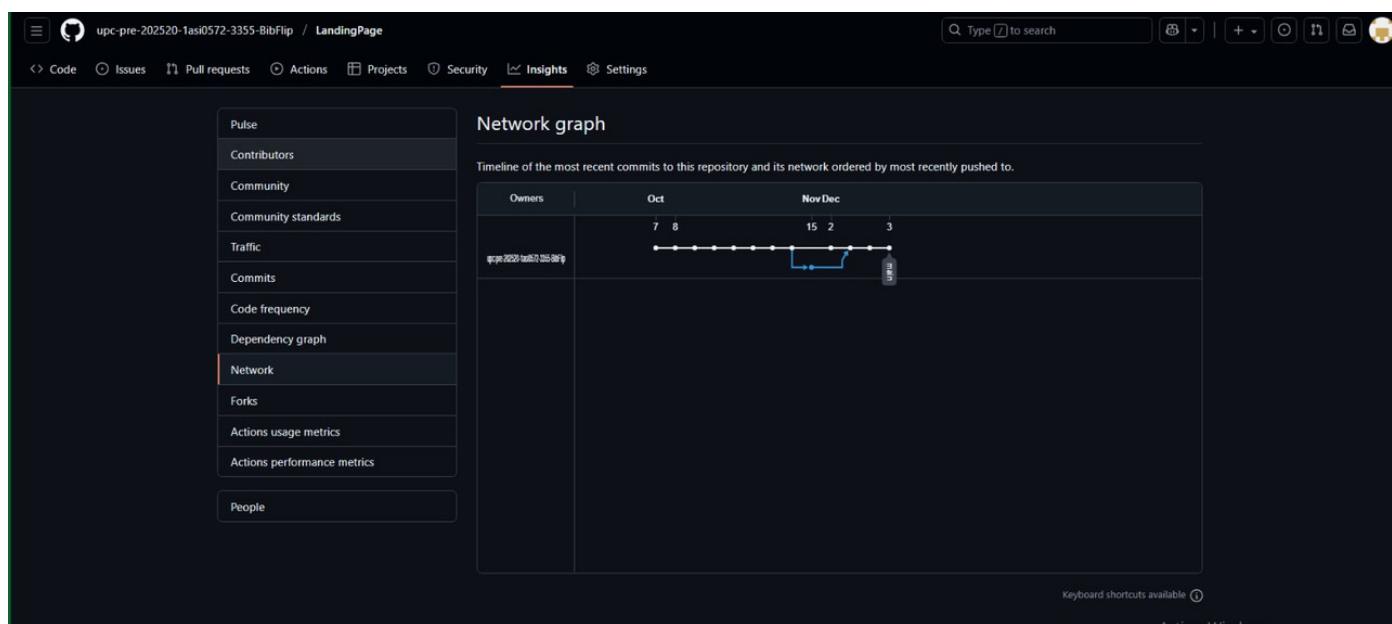
Edge API Platform Repository Insights



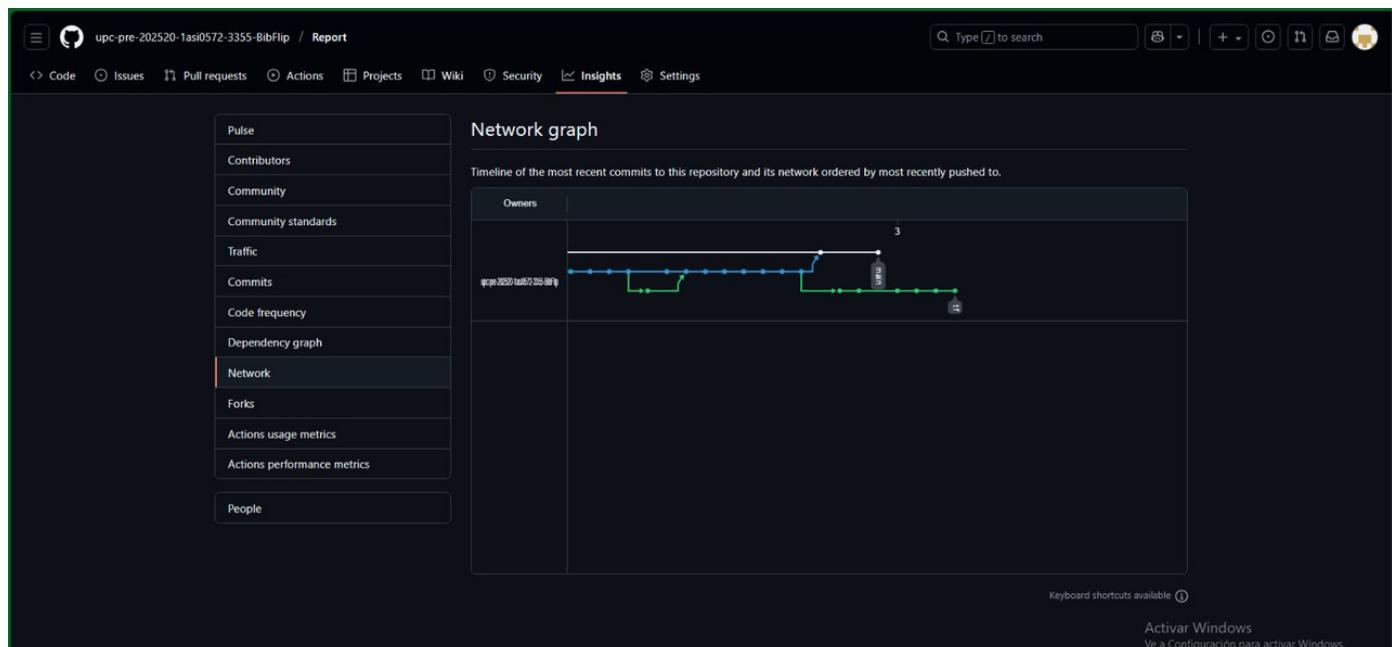
Web API Platform Repository Insights



Landing Page Repository Insights



Report Repository Insights



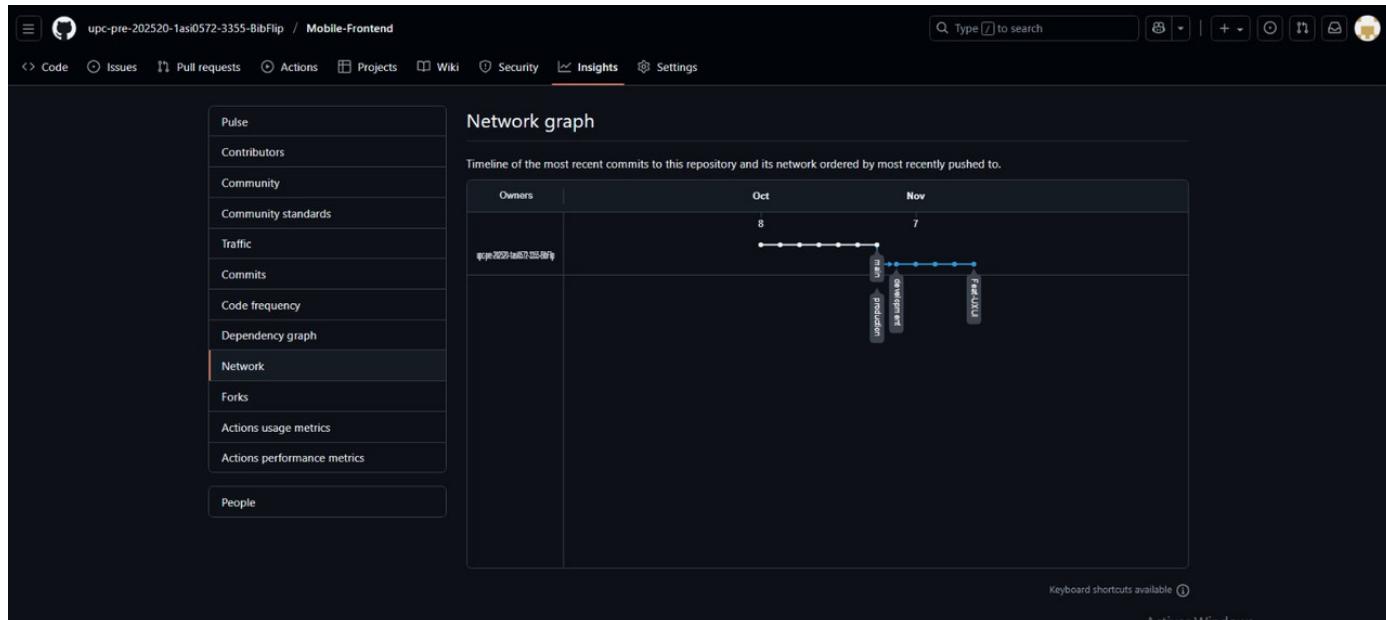
Embedded System Repository Insights

The screenshot shows the GitHub Insights interface for the repository 'upc-pre-202520-1asi0572-3355-BibFlip'. The 'Insights' tab is selected. On the left, a sidebar lists various metrics: Pulse, Contributors, Community, Community standards, Traffic, Commits, Code frequency, Dependency graph, Network (which is selected), Forks, Actions usage metrics, Actions performance metrics, and People. The main area is titled 'Network graph' and displays a timeline of commits. A horizontal bar at the top indicates the timeline from 'Sep 2020' to 'Nov 15'. Below this, a series of small dots represent individual commits, with the most recent ones clustered near the end of the timeline. A vertical scroll bar is visible on the right side of the main content area.

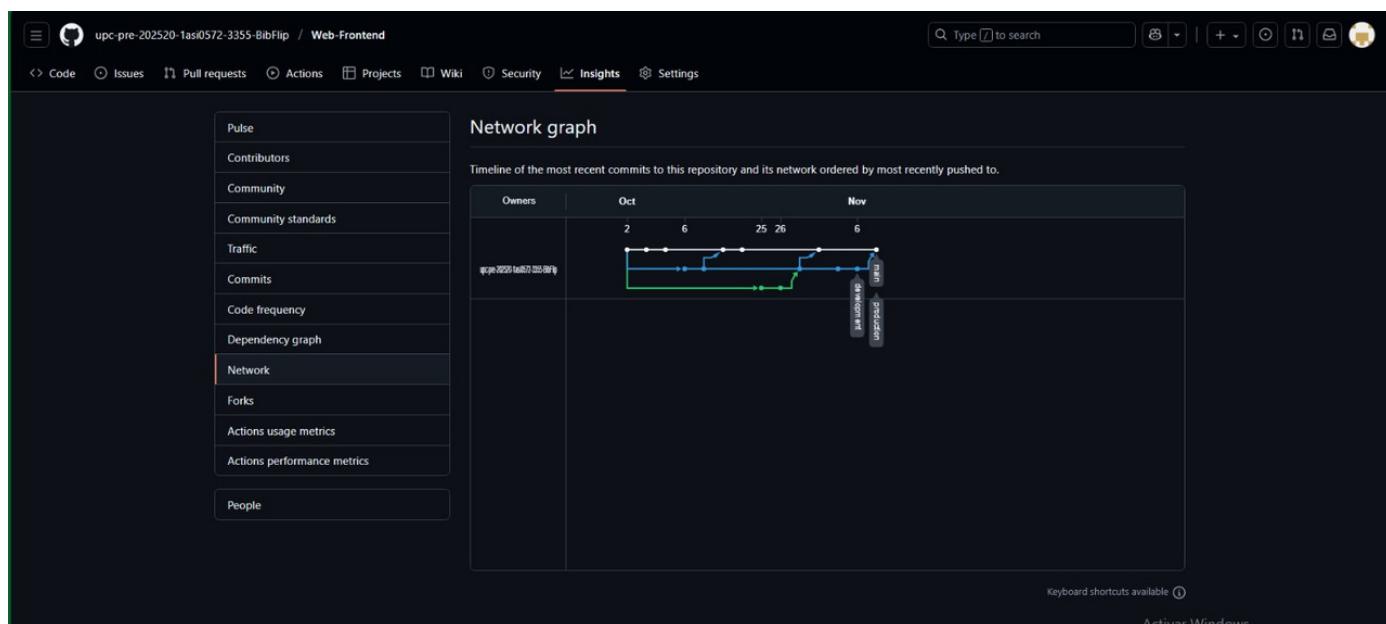
Acceptance Test Repository Insights

The screenshot shows the GitHub Insights interface for the repository 'BibFlip-Acceptance-Tests'. The 'Insights' tab is selected. The sidebar on the left is identical to the one in the previous screenshot, showing the same list of metrics. The main area is titled 'Network graph' and displays a timeline of commits. A horizontal bar at the top indicates the timeline from 'Sep 2020' to 'Nov 15'. Below this, a series of small dots represent individual commits, with a higher density of commits visible towards the end of the timeline. A vertical scroll bar is visible on the right side of the main content area.

Mobile Frontend Repository Insights



Web Page Frontend Repository Insights



6.3. Validation Interviews

6.3.1. Diseño de Entrevistas

Buenas tardes, gracias por asistir a esta entrevista

Preguntas Generales:

1. ¿Cuál es su nombre?
2. ¿Cuántos años tiene?
3. ¿Cuál es su ocupación actual?

Preguntas Sobre la Landing Page

1. ¿Qué tan accesible le pareció la página?
2. ¿Qué parte le resultó más útil?
3. ¿Alguna parte le resultó confusa? ¿Cuál?
4. ¿Qué aspecto le gustaría cambiar o mejorar al respecto?

Preguntas Específicas:

Personal de bibliotecas universitarias

Preguntas sobre la Aplicacion Web

1. ¿Cómo te pareció la aplicación web en general?

2. ¿Cómo le pareció el dashboard y gestión de cubículos?
 3. ¿Hay alguna parte del proceso que le haya parecido confusa? ¿Cuál?
 4. ¿Hay alguna característica que le gustaría ver en la aplicación?
-

Estudiantes universitarios

Preguntas sobre la Aplicacion Web

1. ¿Cómo te pareció la aplicación web en general?
2. ¿Cómo le pareció el dashboard y gestión de reservas de cubículos?
3. ¿Hay alguna parte del proceso que le haya parecido confusa? ¿Cuál?
4. ¿Hay alguna característica que le gustaría ver en la aplicación?

Preguntas sobre la Aplicacion Movil

1. ¿Cómo le pareció la aplicación mobile?
2. ¿Hay alguna parte del proceso que le haya parecido confusa? ¿Cuál?
3. ¿Hay alguna característica que le gustaría ver en la aplicación?

6.3.2. Registro de Entrevistas

Entrevistas para el Segmento Objetivo 1 - Personal Bibliotecario

Entrevista N°1:

Entrevistado: Jhon Galvez

Sexo: Masculino

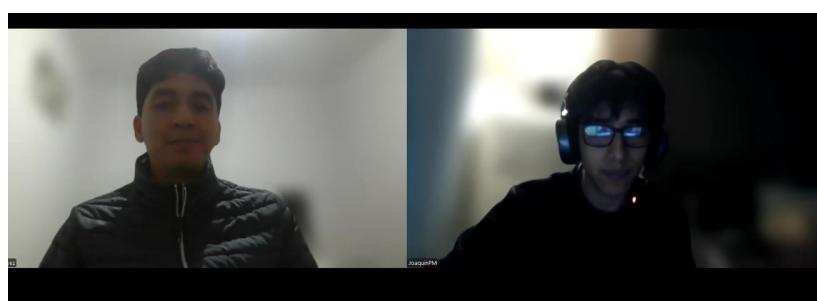
Edad: 30 años

Domicilio: Lima

Inicio de la Entrevista: 0:00

Fin de la Entrevista: 5:13

Link : [Link de la entrevista](#)



Resumen de la Entrevista:

El entrevistado, John Gálvez, supervisor de cubículos de 30 años, comentó que la landing page le pareció accesible, con textos legibles, botones visibles y navegación intuitiva, destacando como parte más útil la sección que explica la funcionalidad principal de la plataforma; sin embargo, notó que algunas secciones eran breves y sugirió incluir más detalles y ejemplos visuales como capturas del flujo de reserva o del dashboard. Sobre la aplicación web, indicó que está bien organizada, con una interfaz limpia y de fácil uso, y que el dashboard permite identificar rápidamente el estado de los cubículos al centralizar la información; no obstante, encontró cierta confusión en la transición entre estados de los cubículos, recomendando una guía más clara. Finalmente, propuso como mejora agregar herramientas de reportes más completas, como historiales de uso, estadísticas por períodos y un módulo para registrar incidentes dentro de los cubículos.

Entrevista N°2:

Entrevistado: Gerson Ambrosio

Sexo: Masculino

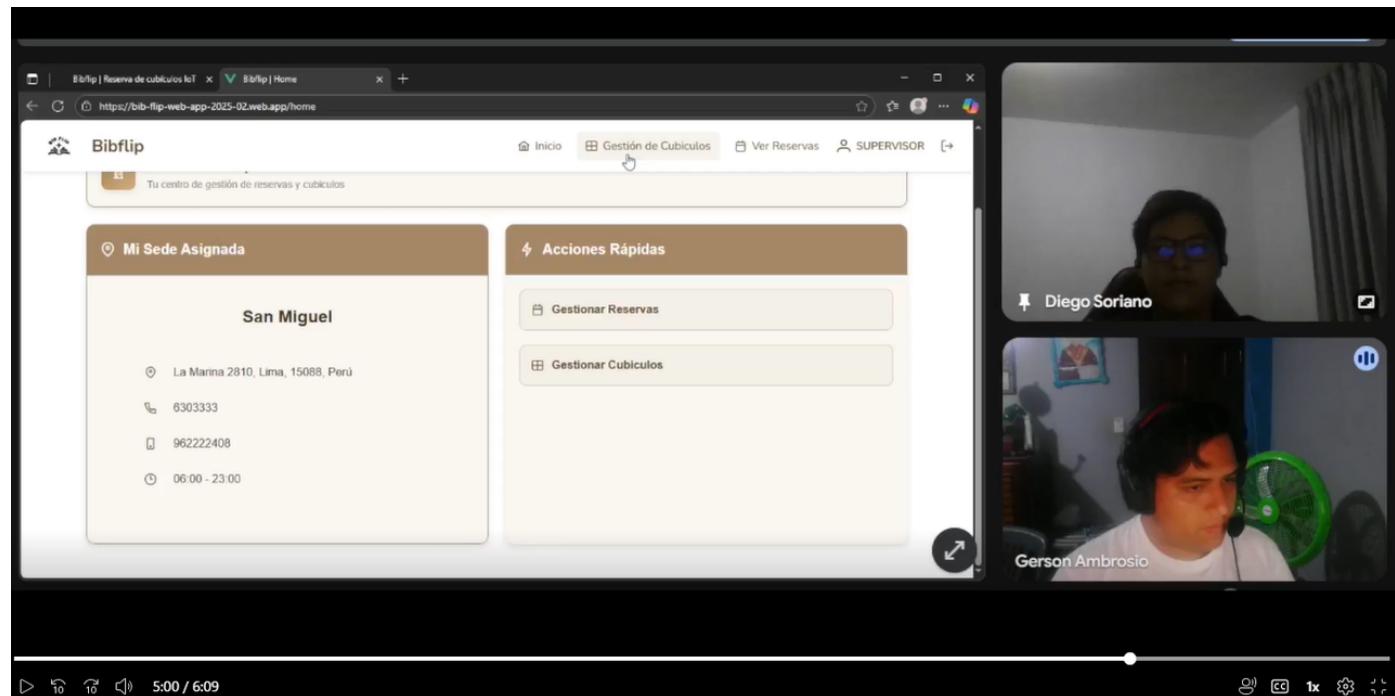
Edad: 20 años

Domicilio: Callao

Inicio de la Entrevista: 3:03

Fin de la Entrevista: 6:09

Link : [Link de la entrevista](#)



Resumen de la Entrevista:

El entrevistado, Gerson, quien trabaja como bibliotecario, compartió sus principales apreciaciones sobre las tareas y funcionalidades presentadas en la entrevista para la solución Bibflip. En cuanto a la landing page, la consideró muy práctica, destacando como la parte más útil la sección de "cómo usarla" por considerarla básica e importante. No obstante, sugirió que las descripciones de las características fuesen más detalladas, específicamente para el rol de bibliotecario. Respecto a la aplicación web para supervisores, la calificó como buena y que cumple con sus operativas, en particular para gestionar cubículos. Encontró el dashboard y la sección de gestión de cubículos simple pero práctica, ya que contiene los avisos y botones necesarios. Sin embargo, sugirió que la información de ubicación de los cubículos por sede fuese más descriptiva y que el proceso para agregar cubículos fuese más práctico con un acceso más sencillo al formulario. Finalmente, propuso la implementación de filtros en el dashboard, como por ejemplo de cubículos o sedes, para hacer la aplicación más eficiente.

Entrevista N°3:

Entrevistado: Hanna Pairazaman

Sexo: Femenino

Edad: 25 años

Domicilio: Callao

Inicio de la Entrevista: 0:14

Fin de la Entrevista: 6:19

Link : [Link de la entrevista](#)



Resumen de la Entrevista:

La entrevista post-demo con Hanna Pairazaman, coordinadora de biblioteca de 25 años en el Callao, permitió validar la solución desde la perspectiva del personal administrativo tras mostrarle la landing page y la aplicación web. Hanna consideró la landing page clara y funcional, destacando como más útil la sección que explica cómo la plataforma optimiza la gestión de espacios, aunque sugirió añadir testimonios de otras bibliotecas y datos concretos sobre la reducción de tiempos de gestión; no encontró partes confusas pero recomendó mejorar la visibilidad de los botones de llamada a acción. Respecto a la aplicación web, señaló que le pareció muy completa y fácil de navegar, valorando especialmente el dashboard que permite monitorear en tiempo real todos los cubículos y gestionar rápidamente cambios de estado; el módulo de gestión de cubículos le resultó eficiente para actualizar disponibilidad y consultar reservas activas de manera centralizada. Identificó como área de mejora la necesidad de agregar filtros más específicos para buscar cubículos por características (capacidad, ubicación, equipamiento), y propuso como características deseables: un sistema de notificaciones push para alertar sobre reservas próximas a vencer, herramientas de análisis predictivo para identificar patrones de uso y optimizar la asignación de recursos, y la posibilidad de exportar datos en diferentes formatos para reportes institucionales. En general, considera que BibFlip representaría una mejora significativa en la eficiencia operativa y reduciría considerablemente el tiempo dedicado a tareas administrativas repetitivas.

Entrevista para el Segmento Objetivo 2 - Estudiantes Universitarios

Entrevista N°1:

Entrevistado: Nasthya del Carpio

Sexo: Femenino

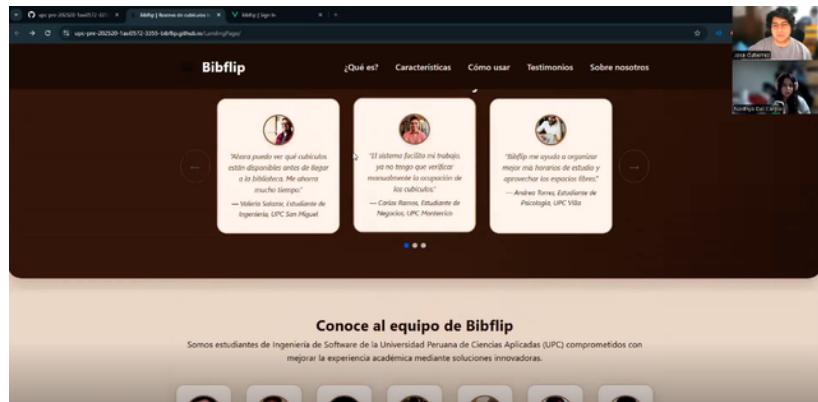
Edad: 21 años

Domicilio: La Perla

Inicio de la Entrevista: 0:14

Fin de la Entrevista: 0:00

Link : [Link de la entrevista](#)

*Resumen de la Entrevista:*

La entrevista post-demo con Nasthya del Carpio, estudiante de 21 años de Diseño de Interiores en la UPC, permitió validar la solución de reserva de cubículos tras mostrarle la landing page, la aplicación web y la app móvil. Nasthya percibió la landing como clara y accesible, aunque sugirió mejorar la diferenciación entre secciones y añadir más ejemplos visuales y un botón principal más destacado. La aplicación web le pareció intuitiva, especialmente el dashboard con estados de cubículos y gestión de reservas, proponiendo como mejora un resumen antes de confirmar la hora y funciones como favoritos, integración con calendario y notificaciones. La app móvil la consideró muy útil para el uso diario, con una interfaz simple, pero recomendó explicar mejor el uso del QR y añadir modo oscuro, mapa de cubículos y notificaciones personalizables. En general, considera que la solución responde a sus frustraciones previas al buscar espacios de estudio y que, bien implementada, sería usada por muchos estudiantes.

Entrevista N°2:

Entrevistado: Sebastián Pacheco

Sexo: Masculino

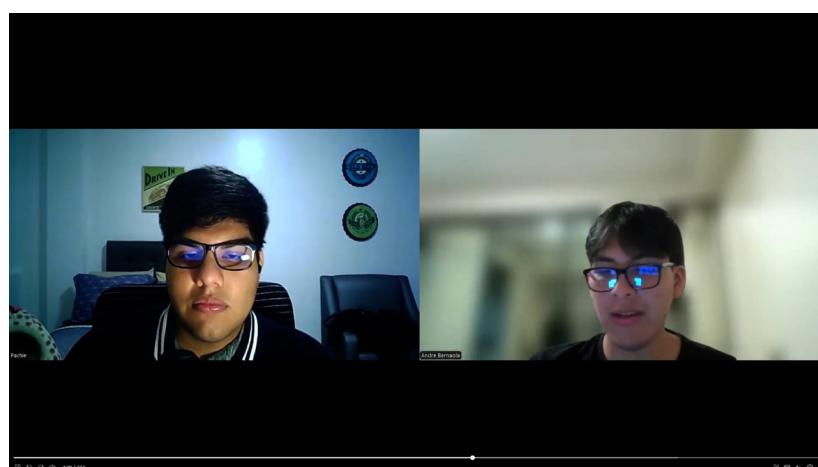
Edad: 22 años

Domicilio: [No especificado]

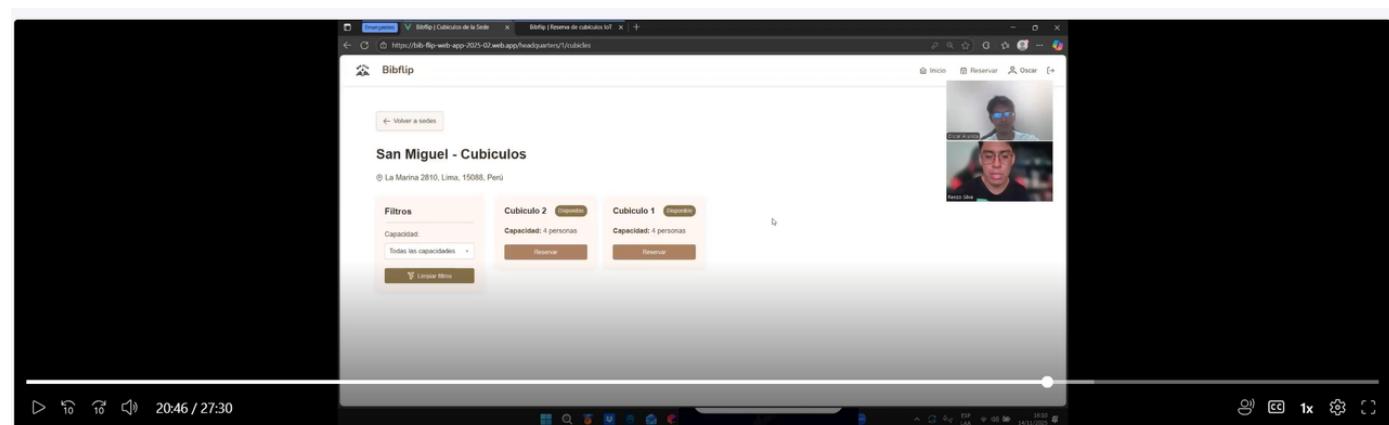
Inicio de la Entrevista: 0:00

Fin de la Entrevista: [Duración del video]

Link : [Link de la entrevista](#)

*Resumen de la Entrevista:*

La entrevista post-demo con Sebastián Pacheco, estudiante de 22 años de la carrera de Ingeniería de Software, permitió validar la solución de reserva de cubículos tras mostrarle la landing page, la aplicación web y la app móvil. Sebastián percibió la landing page como accesible y ordenada, destacando como parte más útil la sección que describe las funciones mediante tres secciones claras, además de apreciar la inclusión de reseñas y características; no encontró aspectos confusos ni oportunidades de mejora significativas. Respecto a la aplicación web, señaló que cumple exactamente con el objetivo principal de reservar cubículos y visualizar horarios disponibles; consideró que el dashboard y la gestión de reservas son sencillos y no resulta complicado hacer clic en los horarios disponibles, aunque identificó una oportunidad de mejora al sugerir implementar funcionalidad para validar y confirmar cuántos alumnos van a estar presentes en cada cubículo, así como una visualización más dinámica de horarios disponibles por día. La aplicación móvil le pareció más completa que la web, siendo honesto al afirmar que el flujo es más comprensible en móvil; no detectó flujos confusos y consideró que visualmente se ve bien y bonita, sin necesidad de características adicionales más allá de la validación de asistentes mencionada anteriormente.

Entrevista N°3:*Entrevistado: Renzo Silva**Sexo: Masculino**Edad: 20 años**Domicilio: [No especificado]**Inicio de la Entrevista: 18:11**Fin de la Entrevista: 22:13**Link : [Link de la entrevista](#)***Resumen de la Entrevista:**

La entrevista post-demo con Renzo Silva, estudiante de 20 años de la UPC, permitió percibir la opinión general de un estudiante ante una mejora en el sistema de reserva de cubículos. Durante la duración de esta, él entrevistado percibió la landing page como suficiente y concisa, destacando como parte más útil la sección de testimonios. Esto desmuestra que nuevos usuarios podrían presentar un mayor interés dependiendo de la recepción de la audiencia objetiva, tomando en cuenta las opiniones de las demás. Respecto a la aplicación web y móvil señaló que cumple exactamente con el objetivo principal de reservar cubículos y visualizar horarios disponibles, por lo cual decidió no indicar alguna funcionalidad extra que él añadiría de momento.

6.3.3. Evaluaciones según heurísticas**6.3.3. Evaluación según Heurísticas**

A continuación se presenta la evaluación de la Landing Page y la Aplicación Web del proyecto **BibFlip** utilizando las 10 heurísticas de usabilidad de Jakob Nielsen. Esta evaluación permite identificar fortalezas, problemas y oportunidades de mejora en la experiencia de usuario.

Tabla General de Evaluación

Heurística	Evaluación en BibFlip	Puntuación (1-5)	Oportunidades de Mejora
1. Visibilidad del estado del sistema	El sistema informa algunos cambios, pero acciones como reservación o actualización no muestran feedback inmediato.	3	Agregar loaders, mensajes de éxito/error y animaciones de carga.
2. Correspondencia con el mundo real	Usa lenguaje familiar para estudiantes y bibliotecarios. Iconografía adecuada.	4	Mejorar descripciones en mapa y etiquetas más naturales.
3. Control y libertad del usuario	La navegación es clara, pero algunas acciones no permiten deshacer.	3	Añadir confirmaciones, opción de cancelar acciones y botón "volver".
4. Consistencia y estándares	Se mantienen estilos, colores e íconos coherentes.	4	Unificar formularios y estandarizar todos los componentes UI.
5. Prevención de errores	Validaciones básicas, pero no completas.	3	Añadir validación en tiempo real y evitar entradas inválidas.
6. Reconocer antes que recordar	Los menús son visibles y los estados entendibles.	4	Usar más íconos o colores para reforzar el estado de cubículos.
7. Flexibilidad y eficiencia	Adecuado para usuarios nuevos; pocas funciones avanzadas.	3	Filtros, búsqueda de cubículos y atajos para administradores.
8. Estética y diseño minimalista	Interfaz limpia y moderna.	4	Mejorar espaciado y reducir texto extenso en la Landing Page.
9. Ayuda en recuperación de errores	Mensajes básicos de error.	3	Incluir mensajes explicativos y recomendaciones de solución.
10. Ayuda y documentación	No incluye tutoriales o sección de ayuda.	2	Crear guía rápida, FAQ y tutoriales para administradores.

Análisis Detallado

1. Visibilidad del estado del sistema

El sistema muestra algunos cambios, pero varias acciones no presentan retroalimentación clara, lo que puede generar incertidumbre en el usuario.

Mejoras sugeridas: loaders, notificaciones, mensajes claros.

2. Correspondencia con el mundo real

El uso de términos como "cubículo", "reserva" y "dashboard" coincide con el lenguaje del usuario y facilita la comprensión del sistema.

Mejoras sugeridas: etiquetas más descriptivas, mayor claridad en mapas.

3. Control y libertad del usuario

Aunque la navegación es fluida, algunas acciones no permiten deshacer o cancelar, lo cual afecta el control del usuario.

Mejoras sugeridas: agregar confirmaciones, opciones de cancelar y deshacer.

4. Consistencia y estándares

El estilo visual es coherente, los elementos UI son uniformes y se mantiene una estética institucional clara.

Mejoras sugeridas: estandarizar formularios y componentes.

5. Prevención de errores

Existen validaciones básicas pero no suficientes para evitar errores comunes o entradas inválidas.

Mejoras sugeridas: validaciones mejoradas en tiempo real.

6. Reconocer antes que recordar

Las acciones están visibles y los usuarios no deben memorizar rutas o estados.

Mejoras sugeridas: reforzar iconografía de estados del cubículo.

7. Flexibilidad y eficiencia de uso

Presenta facilidad para nuevos usuarios, pero es poco eficiente para usuarios expertos o administradores con tareas repetitivas.

Mejoras sugeridas: filtros, búsqueda, accesos rápidos.

8. Estética y diseño minimalista

La interfaz es limpia, moderna y simple, lo cual facilita la comprensión visual.

Mejoras sugeridas: mejorar espaciado, evitar saturación visual.

9. Ayuda al usuario a recuperarse de errores

Los mensajes actuales son básicos y no explican claramente qué ocurrió o cómo solucionarlo.

Mejoras sugeridas: mensajes con diagnóstico y pasos de solución.

10. Ayuda y documentación

El sistema aún no incluye secciones de ayuda, tutoriales ni documentación accesible desde la interfaz.

Mejoras sugeridas: añadir FAQ, guía rápida y tutorial interactivo.

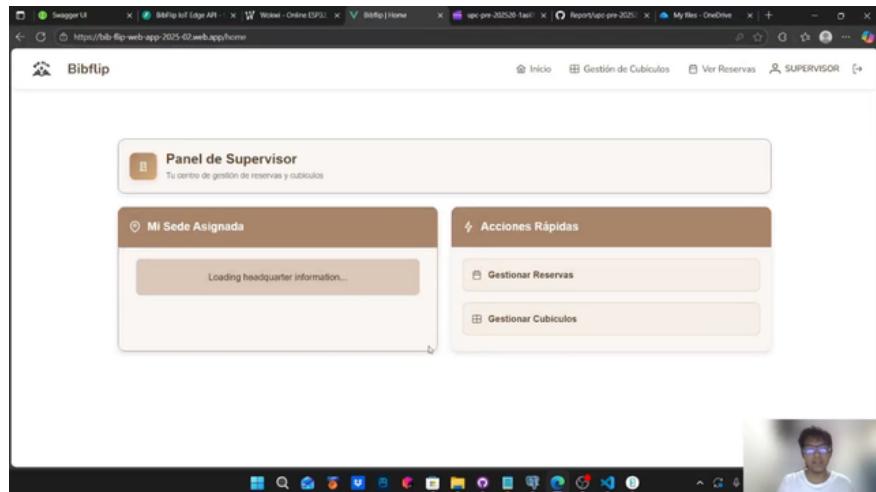
Conclusión General

La evaluación heurística revela que **BibFlip** presenta una interfaz consistente, moderna y clara, con buena comprensión del entorno bibliotecario. Sin embargo, deben reforzarse elementos clave como la retroalimentación visual, la prevención de errores y la documentación disponible para el usuario. Implementar estas mejoras permitirá una experiencia más segura, intuitiva y eficiente tanto para estudiantes como para el personal bibliotecario.

6.4. Video About-the-Product

Introducción

El Video About-the-Product de BibFlip presenta nuestra solución innovadora para la gestión inteligente de espacios de estudio en bibliotecas universitarias. Este contenido audiovisual comunica de manera efectiva el valor de nuestra plataforma tanto a visitantes del Landing Page que desean comprender nuestro modelo de negocio, como a usuarios actuales y potenciales que buscan conocer las funcionalidades clave del sistema.



Duración: 6:31

URL Microsoft Stream: https://upcedupe-my.sharepoint.com/:v/g/personal/u202218167_upc_edu_pe/IQCRUJH6EhUnSrTAUMgJSLGjAXp0_sITH5o-VyID7lnwlFw?#nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOijPbmVEcmI2ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6Illd1YiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFpzZxi0iJNeUZpbGVzTGlua0NvcHkifX0&e=26Qx50

URL YouTube: <https://youtu.be/EhF8cJ7II6Q>

Resumen del Contenido

El video muestra cómo BibFlip transforma la experiencia de reserva de cubículos mediante tecnología IoT con sensores de peso que permiten consultar disponibilidad en tiempo real desde cualquier dispositivo. Presentamos el flujo intuitivo de reserva para estudiantes y las herramientas de gestión administrativas que facilitan al personal bibliotecario la supervisión de ocupación y generación de reportes estadísticos. Además, incorporamos el testimonio de un estudiante participante en nuestras entrevistas de validación, quien destaca: "BibFlip cambió completamente mi rutina de estudio. Ahora puedo planificar mis sesiones con certeza y llegar sabiendo que tendrá un espacio disponible". El contenido mantiene coherencia con la identidad de BibFlip, empleando un tono profesional pero accesible, combinando elementos visuales modernos con una narrativa enfocada en beneficios tangibles para estudiantes, administradores y stakeholders educativos.

Conclusiones y Recomendaciones

Conclusiones

TB1

- Es importante realizar un análisis real de los costos para la instalación y mantenimiento de los dispositivos IOT para cada cubículo. Además, se debe considerar la cantidad general de los cubículos, y el gasto en el uso de tecnologías CLOUD
- El diseño de la arquitectura de software y el DDD, tanto a nivel táctico como estratégico, nos permite expresar la estructura del software de manera gráfica y con mayor profundidad. Esto permite que todos los integrantes puedan comprender la funcionalidad de cada bounded context y del sistema en general.

TP

- Como equipo, logramos integrar de manera efectiva el diseño UI/UX con la implementación, validación y despliegue del producto, trabajando bajo un liderazgo colaborativo y una comunicación constante. Cada integrante aportó desde su rol técnico y creativo, fortaleciendo la organización, la planificación y la toma de decisiones conjuntas. Este proceso nos permitió desarrollar un producto funcional y coherente, al mismo tiempo que consolidamos nuestras habilidades de trabajo en equipo, liderazgo compartido y compromiso con los objetivos del proyecto.

TB2

- Para este sprint utilizamos Wokwi como principal herramienta de simulación del sistema IoT del ESP32 conectado a sensores de peso. Esto fue fundamental para mapear su conexión con el Edge API y su integración con el backend que utiliza la aplicación tanto web como móvil.

TF1

- Para esta entrega final (TF1), la integración del modelo físico de BibFlip con el ecosistema IoT (ESP32, sensores de peso, Edge API y backend) permitió validar en un entorno real los flujos de reserva y ocupación diseñados en las etapas previas. El paso de la simulación en Wokwi al prototipo físico confirmó la viabilidad técnica de la solución propuesta para gestionar cubículos de estudio.

- En esta entrega se verificó, mediante pruebas end-to-end desde el dispositivo en el cubículo hasta la visualización en las aplicaciones web y móvil, que la arquitectura por bounded contexts y los Web Services diseñados soportan correctamente los casos de uso de estudiantes, administradores y superadministradores, manteniendo coherencia entre el modelo conceptual y la implementación.
- El proceso de calibración de sensores, manejo de ruido en las lecturas y depuración del firmware permitió identificar desafíos reales de confiabilidad, latencia y estabilidad del sistema. Esto aportó aprendizajes clave sobre la importancia del diseño robusto en la capa IoT y la necesidad de monitorear continuamente el estado de los dispositivos desplegados.
- En conjunto, la integración física lograda en esta entrega demostró que BibFlip no es solo un prototipo de software, sino una solución ciberfísica capaz de conectar infraestructura de biblioteca, dispositivos IoT y servicios en la nube, sentando una base sólida para escalamiento a más sedes y escenarios de uso.

Recomendaciones

- Implementar un flujo de **check-in mediante códigos QR** físicos en cada cubículo: el estudiante escanea el QR al llegar para confirmar su reserva, actualizar el estado de ocupación en tiempo real y reducir las "reservas fantasma". Este mismo mecanismo podría utilizarse para el check-out y para reportar incidencias del cubículo.
- Extender el módulo de **analítica para administradores y superadministradores**, incorporando indicadores como horas pico de uso, tasa de cancelaciones, porcentaje de ocupación por sede y por tipo de cubículo. Estos dashboards permitirían tomar decisiones de capacidad, mantenimiento y ampliación de infraestructura basadas en datos.
- Ampliar el ecosistema IoT incorporando **nuevos tipos de sensores** (ruido ambiental, temperatura, iluminación) que permitan ofrecer información de confort y detectar usos inadecuados de los cubículos, mejorando la experiencia del estudiante y la supervisión del personal de biblioteca.
- Fortalecer los aspectos de **seguridad y gobernanza** del sistema, reforzando controles de autenticación y autorización para cada rol (estudiante, administrador, superadmin), así como el registro de logs de auditoría sobre reservas, cambios de estado de cubículos y operaciones críticas en los Web Services.
- Planificar una estrategia de **escalabilidad y mantenibilidad** que incluya documentación técnica estandarizada, guías de instalación de dispositivos, scripts de despliegue automatizado en la nube y procedimientos para reemplazo de hardware en campo, con el fin de facilitar la adopción de BibFlip en múltiples campus o instituciones educativas.

Video About-the-team

Este video presenta el trabajo colaborativo del equipo responsable del desarrollo de BibFlip, una solución IoT-Web orientada a optimizar la gestión de cubículos de estudio en bibliotecas universitarias. A través de fragmentos de reuniones de planificación, sesiones de diseño de arquitectura (Landing, Frontend Web, Backend e IoT Edge), modelado de dominios y pruebas sobre el prototipo físico y las simulaciones en Wokwi, se evidencia cómo el equipo articuló hardware (sensores y microcontrolador), firmware embebido, APIs REST y frontends (web y mobile) para ofrecer una experiencia integrada de reserva y monitoreo de cubículos en tiempo casi real.

El video destaca la participación activa de cada integrante en diferentes etapas del ciclo de vida del sistema: desde la definición de requisitos y el diseño de las historias de usuario, hasta la implementación del backend con servicios de reservas y cubículos, la configuración del Edge API para recibir lecturas de los dispositivos IoT y la construcción de interfaces en Vue y Flutter para estudiantes, administradores y superadministradores. Se incluyen momentos de pair programming, debugging colaborativo, revisiones de código y sincronización de tareas en Trello y GitHub, así como breves testimonios en los que cada miembro comenta sus responsabilidades, retos técnicos y aprendizajes.

En conjunto, el material evidencia no solo el uso de metodologías ágiles y buenas prácticas de ingeniería de software e IoT (versionado, testing automatizado, despliegues en la nube), sino también el desarrollo de competencias de comunicación, trabajo en equipo y responsabilidad profesional alineadas con el Student Outcome 3 de ABET, centrado en la capacidad de funcionar eficazmente en equipos que establecen metas, planifican tareas y cumplen compromisos.

URL Microsoft Stream: https://upcedupe-my.sharepoint.com/:v/g/personal/u202218167_upc_edu_pe/IQABw7pheb1ETYdES6x5oRrjAaZnJymeXa-TXuDqbgVArIw?nav=eyJyZWZ1cnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcm12ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6Ild1YiIsInJ1ZmVycmFsTW9kZSI6InZpZXciLCJyZWZ1cnJhbFZpZxcioiJNeUZpbGVzTGlua0NvcHkifX0&e=siE5iq

URL YouTube: <https://youtu.be/xKDkc9CdFKo>

Bibliografía

Diario UNO. (2025, abril 2). *Alumnos presentan molestia e inconformidad por la caída del sistema de matrícula de la Universidad Peruana de Ciencias Aplicadas (UPC)*. <https://diariouno.pe/2025/04/02/alumnos-presentan-molestia-e-inconformidad-por-la-caida-del-sistema-de-matricula-de-la-universidad-peruana-de-ciencias-aplicadas-upc/>

Montoro, R. (2024). *Esta es la segunda biblioteca más grande del Perú y se encuentra en una prestigiosa universidad*. Infobae.

<https://www.infobae.com/peru/2024/12/10/esta-es-la-segunda-biblioteca-mas-grande-del-peru-y-se-encuentra-en-una-prestigiosa-universidad/>

SUNEDU. (2022). *III Informe bienal sobre la realidad universitaria en el Perú*. <https://repositorio.minedu.gob.pe/handle/20.500.12799/7913>

Anexos

Repositorios

<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Report>

<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Edge-Api-Platform>

<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Backend-Testing-Server>

<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/LandingPage>

<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Web-Frontend>

<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Mobile-Frontend>

<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/Backend-Testing-Server>

<https://github.com/upc-pre-202520-1asi0572-3355-BibFlip/BibFlip-Acceptance-Tests>

Deployments

Landing Page: <https://upc-pre-202520-1asi0572-3355-bibflip.github.io/LandingPage/>

Web Application: <https://bib-flip-web-app-2025-02.web.app/>

API Web Service: <https://bibflip-api-platform.azurewebsites.net/swagger-ui/index.html>

IoT Edge API: <https://bibflip-edge-api-platform.azurewebsites.net/api/docs>

Entrevistas

Needfinding: https://upcdeupe-my.sharepoint.com/:v/g/personal/u202218167_upc_edu_pe/EbipDl3neqpBkcsL2Ewh0tIBwZL3w3QbSfnV5e7A5g833w?#q=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAi0iJPbmVEcm12ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6Ild1YiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXci0iJNeUZpbGVzTGlua0NvcHkifX0&e=h1Nh0

Validation: https://upcdeupe-my.sharepoint.com/:v/r/personal/u202114192_upc_edu_pe/Documents/estudiante-bipflip.mp4?#q=1&web=1&nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAi0iJTdHJ1YW1XZWJBcHAiLCJyZWZlcnJhbFZpZXci0iJTaGFyZURpYWxvZy1MaW5rIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6Ild1YiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXci0iJNeUZpbGVzTGlua0NvcHkifX0%3D&e=bh04fe

TB1

Exposición: https://upcdeupe-my.sharepoint.com/:v/g/personal/u202218167_upc_edu_pe/EaOh8E4kvCZBse_K6G2QpFQB2tMN6ft_1cgIcr2lp7fNyw?#q=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAi0iJPbmVEcm12ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6Ild1YiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXci0iJNeUZpbGVzTGlua0NvcHkifX0&e=cFTTEB

TP1

Exposición: https://upcdeupe-my.sharepoint.com/:v/g/personal/u202218167_upc_edu_pe/EbaC_7b6-cp0r9fc8eRi2SIBC0CdyTv6xPxbdar4eIvDGQ?#q=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAi0iJPbmVEcm12ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6Ild1YiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXci0iJNeUZpbGVzTGlua0NvcHkifX0&e=v7hTif

TB2

Exposición: https://upcdeupe-my.sharepoint.com/:v/g/personal/u202218167_upc_edu_pe/IQBBWmbs1qG7RreZ6vZTN02ZARTiyZ6fi3mCC09zbDIXXTA?#q=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAi0iJPbmVEcm12ZUZvckJ1c2luZXNzIiwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6Ild1YiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXci0iJNeUZpbGVzTGlua0NvcHkifX0&e=CLGhp

Video About The Product

About The Product: [https://upcdeupe-my.sharepoint.com/:v/g/personal/u202218167_upc_edu_pe/IQCRUJH6EhUnSrTAUMgJSLGjAXp0_sITH5o-VyID7lnwlFw](https://upcdeupe-my.sharepoint.com/:v/g/personal/u202218167_upc_edu_pe/IQCRUJH6EhUnSrTAUMgJSLGjAXp0_sITH5o-VyID7lnwlFw?#q=eyID7lnwlFw)

nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcm12ZUZvckJ1c2luZXNzIiwi cmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6
InZpZXciLCJyZWZlcnJhbFZpZXciOijNeUZpbGVzTGlua0NvcHkifX0&e=26Qx50

Video About The Team

About The Team: https://upcedupe-my.sharepoint.com/:v/g/personal/u202218167_upc_edu_pe/IQABw7pheb1ETYdES6x5oRrjAaZnJymeXa-TXuDqbgVArIw

nav=eyJyZWZlcnJhbEluZm8iOnsicmVmZXJyYWxBcHAiOiJPbmVEcm12ZUZvckJ1c2luZXNzIiwi cmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6
InZpZXciLCJyZWZlcnJhbFZpZXciOijNeUZpbGVzTGlua0NvcHkifX0&e=siE5iq