



**Estudiantes:** Javier Quilumba y Jonathan Tipan

**Examen-01: El arte del detective de Software**

**Fecha:** 14/12/2025

**Aplicación:** Notion

## Ingeniería Inversa Aplicada a Notion

### Tabla de contenidos

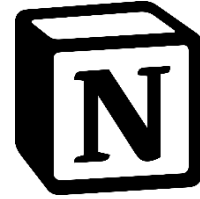
1. Aplicación seleccionada: Notion (Notion Labs).....	2
¿Por qué Notion? .....	2
1.1 Tabla Resumen de la Aplicación .....	2
2. Metodología de Análisis.....	2
3. Requisitos Funcionales Identificados .....	3
4. Arquitectura Hipotética del Sistema.....	4
4.1 Diagrama de Arquitectura de Alto Nivel .....	4
4.2 Descripción de Componentes .....	5
4.3 Flujo de Datos - Creación de un Bloque .....	6
5. Hallazgos adicionales .....	8
5.1 Reglas de Negocio Identificadas .....	8
5.2 Lógica de Sincronización en Tiempo Real.....	9
5.3 Características Técnicas Deducidas.....	9
Tecnologías del Frontend (Probables) .....	9
Tecnologías del Backend (Probables) .....	9
Optimizaciones Observadas .....	9
5.4 Análisis de flujos críticos .....	10
.....	10
5.5 Decisiones de Diseño Observadas .....	10
5.6 Modelo de negocio: Freemium .....	11
5.7 Análisis Competitivo (Contexto) .....	11
6. Conclusiones y Reflexión final .....	11
7. Referencias.....	12



## 1. Aplicación seleccionada: Notion (Notion Labs)

### ¿Por qué Notion?

Notion es una plataforma todo-en-uno para tomar notas, gestionar proyectos, crear bases de datos y colaborar en equipo. Su arquitectura basada en bloques modulares y su capacidad de colaboración en tiempo real la convierten en un caso de estudio excepcional para ingeniería inversa.



### 1.1 Tabla Resumen de la Aplicación

Aspecto	Detalle
<b>Tipo de Aplicación</b>	SaaS (Software as a Service) - Multiplataforma
<b>Plataformas Disponibles</b>	Web (navegador), iOS, Android, macOS, Windows
<b>Categoría</b>	Productividad / Gestión del Conocimiento / Colaboración
<b>Año de Lanzamiento</b>	2016
<b>Usuarios Principales</b>	Individuos, equipos pequeños, empresas, estudiantes
<b>Modelo de Negocio</b>	Freemium (gratis con limitaciones + planes de pago)
<b>Planes de Pago</b>	Personal (\$0), Plus (\$10/mes), Business (\$15/mes), Enterprise (Custom)
<b>Tecnologías Frontend</b>	React, TypeScript, Slate.js/ProseMirror (editor), WebSocket
<b>Tecnologías Backend (estimadas)</b>	Node.js/Go, PostgreSQL, Redis, Elasticsearch, AWS S3
<b>Arquitectura</b>	Microservicios, API Gateway, WebSocket para tiempo real
<b>Características Clave</b>	Bloques modulares, bases de datos relacionales, colaboración en tiempo real, plantillas, API pública
<b>Integraciones</b>	Slack, Google Drive, GitHub, Figma, Zapier, Trello, Asana, IA
<b>Almacenamiento</b>	Cloud (sincronización automática) + Cache local (IndexedDB)
<b>Sincronización</b>	Tiempo real con CRDT/Operational Transformation
<b>Seguridad</b>	Encriptación en tránsito (TLS), permisos granulares, SAML SSO (Enterprise)

## 2. Metodología de Análisis

Aplicamos **Ingeniería Inversa de Caja Negra (Black-Box)**, observando el comportamiento de la aplicación sin acceso al código fuente. Nuestra estrategia:

1. **Exploración Interactiva:** Creamos workspaces, páginas, bases de datos y probamos todas las funcionalidades principales
2. **Pruebas de Comportamiento:** Probamos casos límite (datos inválidos, concurrencia, offline)
3. **Análisis de Flujos:** Documentamos las interacciones usuario-sistema
4. **Deducción Arquitectónica:** Inferimos la estructura backend basándonos en el comportamiento observado



### 3. Requisitos Funcionales Identificados

A continuación, se presentan 24 historias de usuario extraídas del análisis de comportamiento de Notion, organizadas por categorías funcionales:

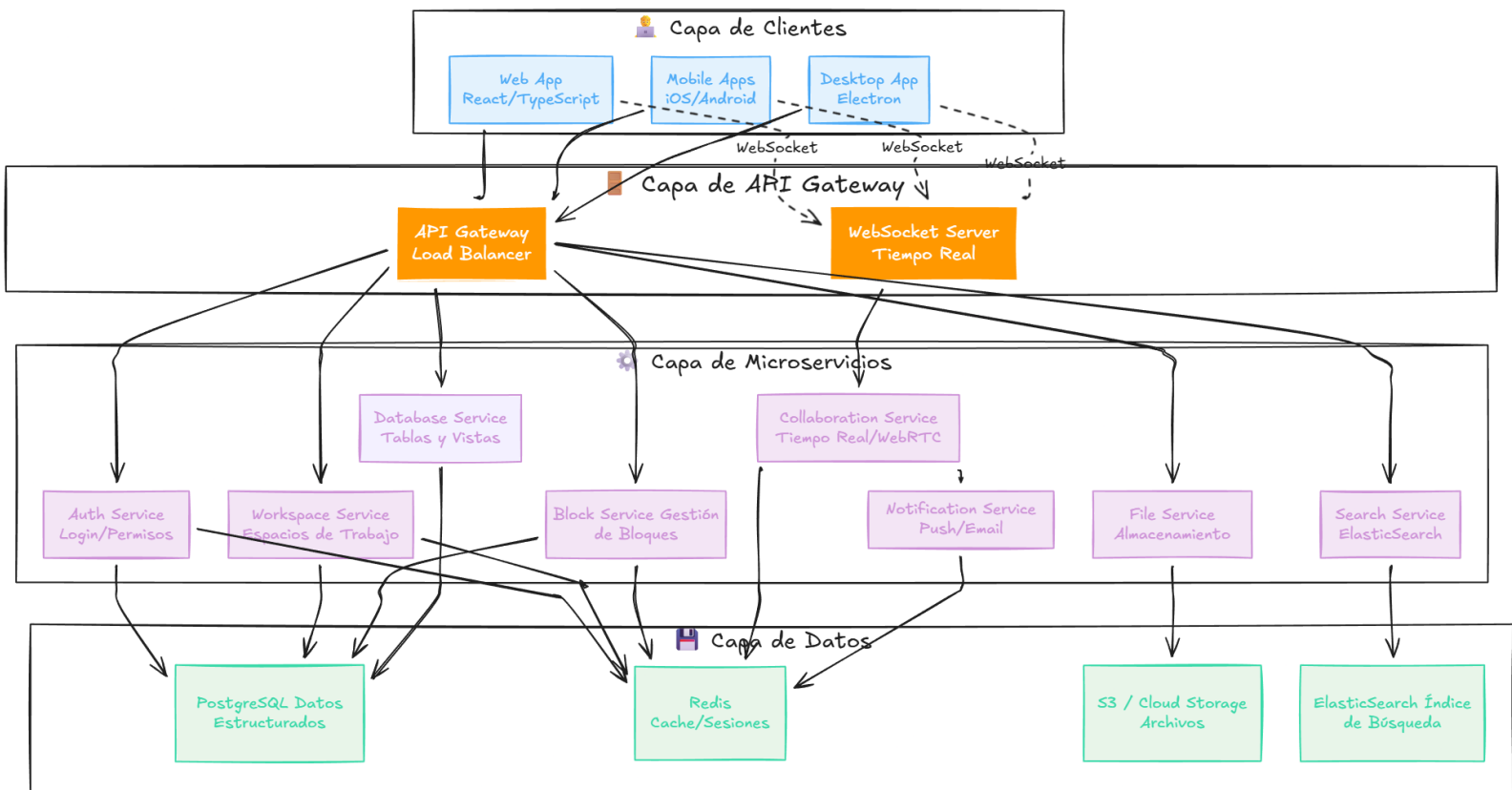
ID	Historia de Usuario	Cat
HU-01	Como usuario, quiero <b>crear diferentes tipos de bloques</b> (texto, título, lista, checklist, toggle, código), para estructurar mi contenido de manera flexible.	Gestión de Bloques
HU-02	Como usuario, quiero <b>transformar un bloque</b> de un tipo a otro usando "/" o menú contextual, para reorganizar contenido sin reescribir.	
HU-03	Como usuario, quiero <b>arrastrar y soltar bloques</b> para reordenarlos, cambiando la estructura de forma visual e intuitiva.	
HU-04	Como usuario, quiero <b>anidar bloques</b> dentro de otros (jerarquías), para organizar información compleja en estructuras tipo árbol.	
HU-05	Como usuario, quiero <b>usar markdown</b> (ej: # para título, - [ ] para checkbox), para escribir más rápido sin mouse.	
HU-06	Como usuario, quiero <b>crear bases de datos</b> (tabla, kanban, calendario, galería, lista, timeline) dentro de una página, para gestionar información estructurada con diferentes vistas.	Bases de Datos
HU-07	Como usuario, quiero <b>añadir propiedades personalizadas</b> (texto, número, select, fecha, persona, fórmula, relación, etc.), para capturar información relevante de mis registros.	
HU-08	Como usuario, quiero <b>aplicar filtros y ordenamientos</b> a las vistas, para encontrar información específica sin recorrer todos los registros.	
HU-09	Como usuario, quiero <b>crear múltiples vistas</b> de la misma base de datos con diferentes filtros, para visualizar datos desde perspectivas distintas.	
HU-10	Como usuario, quiero <b>usar fórmulas</b> en propiedades (suma, promedios, concatenación), para automatizar cálculos.	
HU-11	Como propietario de workspace, quiero <b>invitar usuarios y asignar roles</b> (ver/comentar/editar/admin), para controlar acceso y permisos.	Colaboración
HU-12	Como usuario colaborador, quiero <b>ver cambios en tiempo real</b> (cursores y ediciones simultáneas), para trabajar coordinadamente.	
HU-13	Como usuario, quiero <b>mencionar colaboradores</b> usando @nombre, para notificarles sobre contenido específico.	
HU-14	Como usuario, quiero <b>dejar comentarios</b> en bloques específicos, para dar feedback sin modificar el original.	
HU-15	Como usuario, quiero <b>recibir notificaciones</b> cuando me mencionan o hay cambios relevantes, para estar al tanto de la actividad.	
HU-16	Como usuario, quiero <b>crear plantillas personalizadas</b> , para reutilizar estructuras comunes (reuniones, tareas, proyectos).	Plantillas
HU-17	Como usuario, quiero <b>duplicar páginas o bases de datos</b> , para crear copias rápidas sin afectar el original.	
HU-18	Como usuario, quiero <b>acceder a la galería de plantillas públicas</b> , para inspirarme y usar estructuras pre-diseñadas.	
HU-19	Como usuario, quiero <b>guardado automático en tiempo real</b> , para no perder trabajo si se cierra el navegador.	Si no



HU-20	Como usuario, quiero <b>acceder a páginas en modo offline</b> , para consultar información sin conexión a internet.	Navegación
HU-21	Como usuario móvil, quiero <b>sincronización automática multi-dispositivo</b> , para acceder a la versión actualizada desde cualquier lugar.	
HU-22	Como usuario, quiero <b>marcar páginas como "Favoritas"</b> , para acceder rápidamente desde el sidebar.	
HU-23	Como usuario, quiero <b>búsqueda global</b> (Cmd/Ctrl + K), para encontrar cualquier contenido sin navegar manualmente.	
HU-24	Como usuario, quiero <b>crear páginas anidadas</b> hasta N niveles, para organizar información jerárquicamente.	

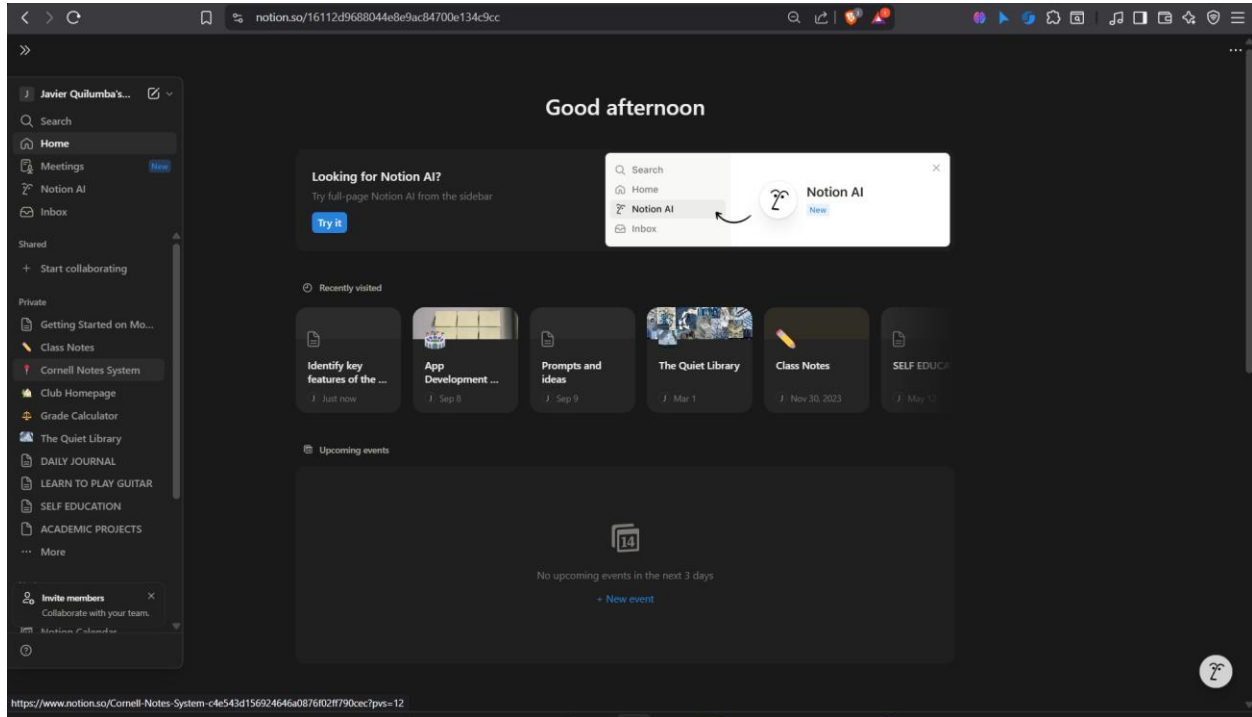
## 4. Arquitectura Hipotética del Sistema

### 4.1 Diagrama de Arquitectura de Alto Nivel





ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERIA DE SISTEMAS  
(ISWD633) CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE  
VICENTE EGUEZ



## 4.2 Descripción de Componentes

### Capa de Clientes

- **Web App:** Aplicación React/TypeScript single-page application (SPA)
- **Mobile Apps:** Apps nativas con componentes compartidos
- **Desktop App:** Electron wrapper de la web app con capacidades offline extendidas

### API Gateway

- **Función:** Punto único de entrada, balanceo de carga, autenticación inicial
- **Tecnología Probable:** NGINX o AWS API Gateway
- **WebSocket Server:** Maneja conexiones persistentes para colaboración en tiempo real

### Microservicios Principales

1. **Auth Service:**
  - Autenticación (email/password, Google, Apple, SAML)
  - Gestión de sesiones y tokens JWT
  - Control de permisos a nivel de workspace
2. **Workspace Service:**
  - CRUD de workspaces
  - Gestión de miembros y roles
  - Configuración de workspace (nombre, icono, dominio)
3. **Block Service:**
  - Creación, actualización, eliminación de bloques



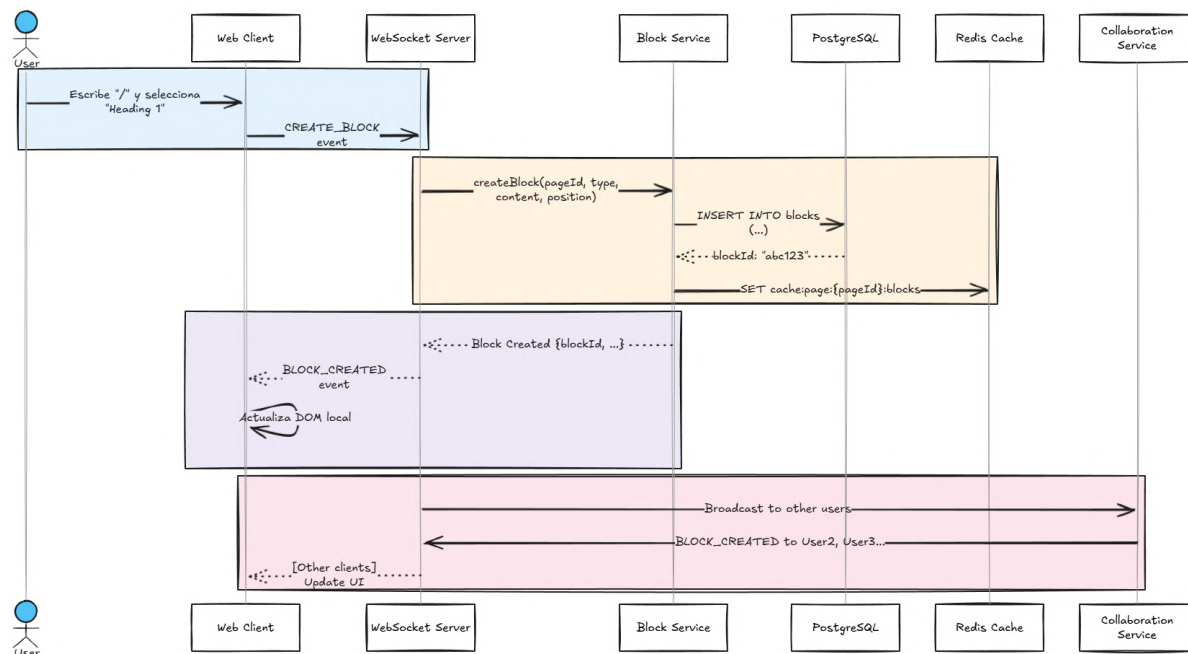
ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERIA DE SISTEMAS  
(ISWD633) CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE  
VICENTE EGUEZ



- Transformación de tipos de bloque
- Ordenamiento y anidamiento (árbol de bloques)
- 4. **Database Service:**
  - Gestión de bases de datos inline y full-page
  - Propiedades, vistas, filtros, ordenamientos
  - Evaluación de fórmulas
- 5. **Collaboration Service:**
  - Edición colaborativa en tiempo real (CRDT o Operational Transformation)
  - Presencia de usuarios (cursores, selecciones)
  - Comentarios y menciones
- 6. **Search Service:**
  - Indexación de contenido
  - Búsqueda full-text
  - Ranking y relevancia
- 7. **File Service:**
  - Upload/download de archivos e imágenes
  - Procesamiento de imágenes (thumbnails, compresión)
  - CDN para entrega rápida
- 8. **Notification Service:**
  - Push notifications
  - Email notifications
  - In-app notifications

#### 4.3 Flujo de Datos - Creación de un Bloque

En base a la experiencia de usuario, extraíamos el siguiente flujo de trabajo para crear un bloque dentro de notion:

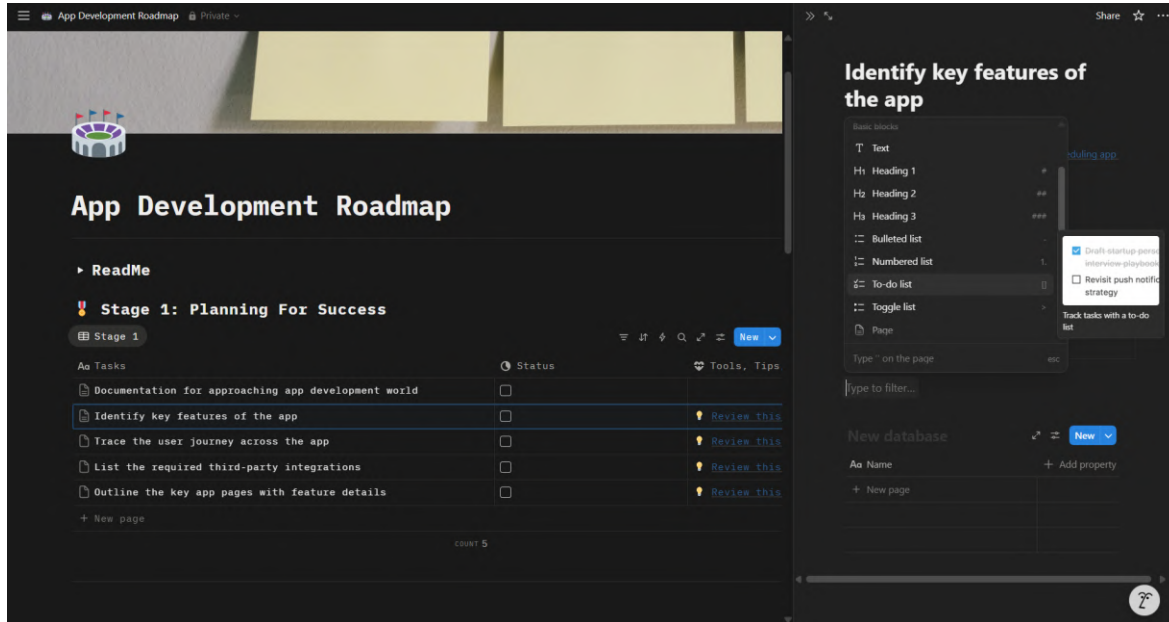


Visualmente la interfaz es bastante intuitiva y agradable:

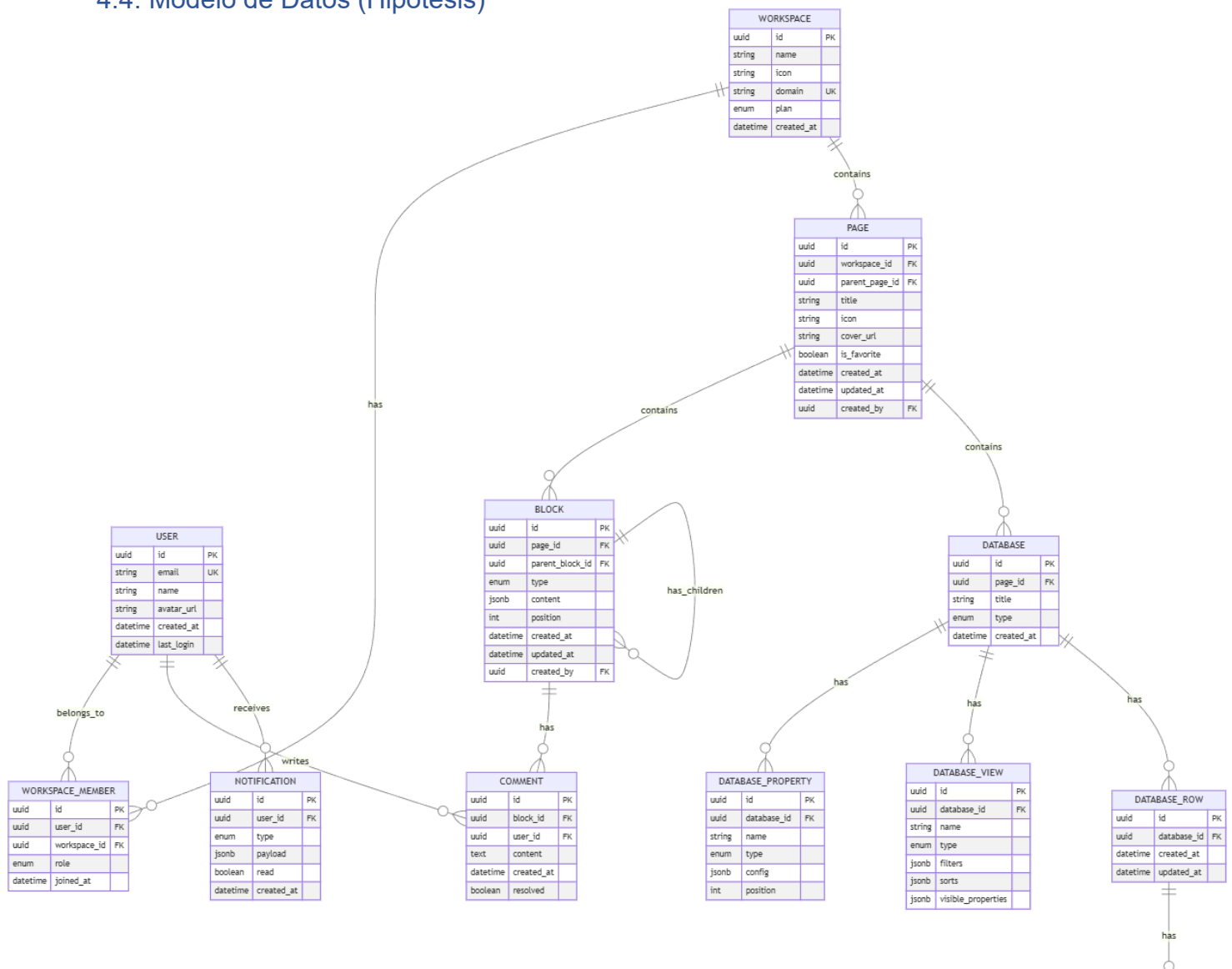




ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERIA DE SISTEMAS  
(ISWD633) CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE  
VICENTE EGUEZ



#### 4.4. Modelo de Datos (Hipótesis)

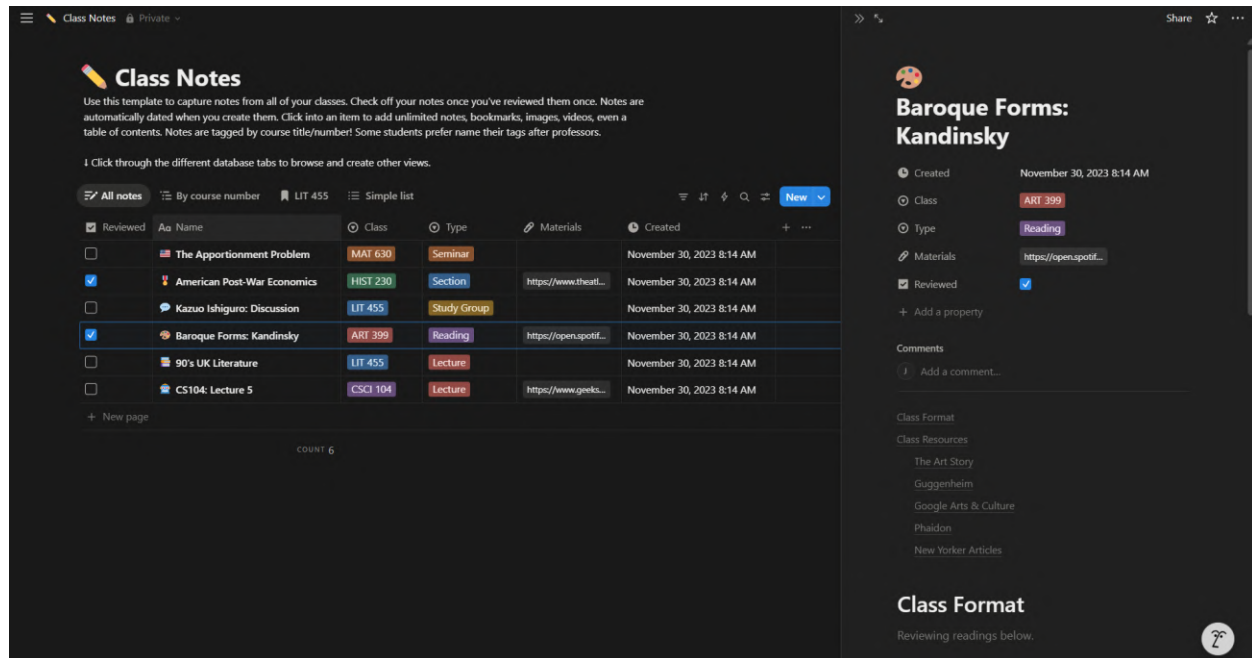




ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERIA DE SISTEMAS  
(ISWD633) CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE  
VICENTE EGUEZ



Screenshot de una base de datos de Notion mostrando diferentes propiedades y vistas:



## 5. Hallazgos adicionales

### 5.1 Reglas de Negocio Identificadas

**RN-01:** Una página no puede ser su propia página padre (prevención de ciclos en la jerarquía)

**RN-02:** Un usuario con rol "puede ver" no puede crear, editar ni eliminar bloques

**RN-03:** Solo el propietario del workspace o los admins pueden cambiar roles de otros miembros

**RN-04:** Un bloque de tipo "database\_inline" debe estar contenido dentro de una página, no puede ser top-level

**RN-05:** Las propiedades de tipo "title" en una base de datos son obligatorias (toda DB debe tener exactamente una propiedad title)

**RN-06:** Los usuarios de plan gratuito tienen límite de:  
Máximo 1000 bloques

Sin historial de versiones ilimitado (solo últimos 7 días)

Sin acceso a features avanzados (sync blocks, advanced permissions)

**RN-07:** Los archivos subidos tienen un límite de tamaño por archivo (5MB en plan gratuito, ilimitado en pago)

**RN-08:** Las menciones (@) solo funcionan con usuarios que tienen acceso a esa página

**RN-09:** La búsqueda solo muestra resultados de páginas a las que el usuario tiene acceso

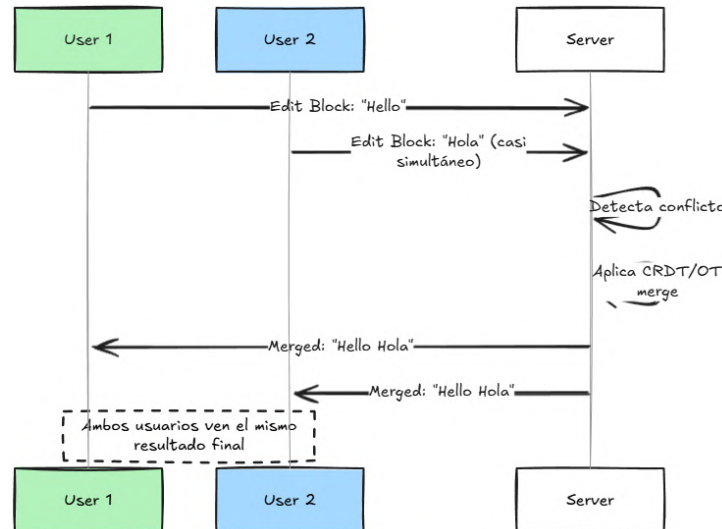
**RN-10:** Los cambios en modo offline se guardan localmente y se sincronizan cuando se recupera la conexión (conflict resolution: last-write-wins o manual merge)





## 5.2 Lógica de Sincronización en Tiempo Real

Notion utiliza probablemente CRDT (Conflict-free Replicated Data Types) u Operational Transformation para manejar edición concurrente:



## 5.3 Características Técnicas Deducidas

### Tecnologías del Frontend (Probables)

- **Framework:** React con TypeScript
- **State Management:** Redux o Zustand
- **Editor de Texto:** Slate.js o ProseMirror (editores extensibles)
- **WebSocket:** Socket.io o native WebSocket API
- **Offline Storage:** IndexedDB para cache local
- **Styling:** CSS-in-JS (Styled Components o Emotion)

### Tecnologías del Backend (Probables)

- **Lenguaje:** Node.js/TypeScript o Go
- **API:** GraphQL o REST
- **Base de Datos:** PostgreSQL (datos estructurados)
- **Cache:** Redis (sesiones, presencia en tiempo real)
- **Búsqueda:** Elasticsearch
- **Storage:** AWS S3 o Google Cloud Storage
- **CDN:** CloudFlare o AWS CloudFront
- **Infraestructure:** Kubernetes, Docker

### Optimizaciones Observadas

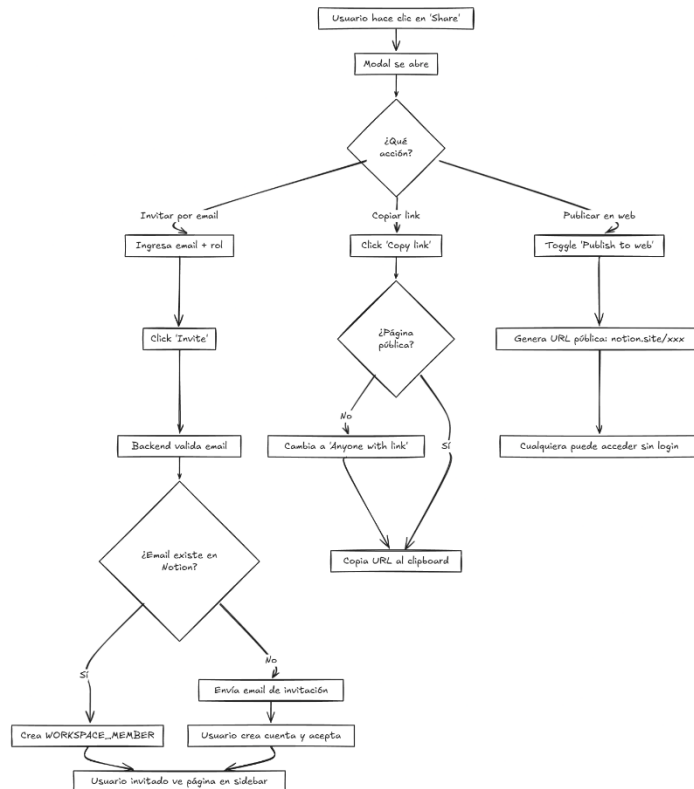
1. **Lazy Loading:** Las páginas solo cargan bloques visibles en viewport
2. **Debouncing:** Los cambios se agrupan antes de enviarlos al servidor (no envía cada keystroke)
3. **Optimistic UI:** La interfaz se actualiza inmediatamente antes de confirmar con el servidor



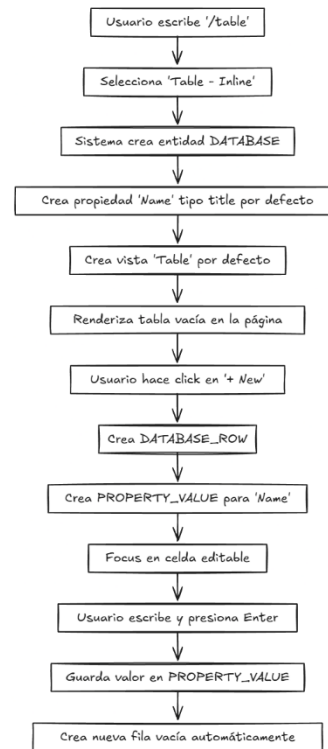
4. **Cache Agresivo:** Las páginas visitadas recientemente se guardan en IndexedDB
5. **Virtual Scrolling:** En bases de datos grandes, solo renderiza las filas visibles

## 5.4 Análisis de flujos críticos

### Compartir una página:



### Crear una base de datos:



## 5.5 Decisiones de Diseño Observadas

### UX Patterns

1. **Comando Slash (/):** Acceso rápido a crear cualquier bloque sin usar mouse
2. **Drag Handles (::):** Manijas visuales a la izquierda de cada bloque para arrastrar
3. **Hover Actions:** Botones aparecen solo al pasar el mouse (comentar, más opciones)
4. **Breadcrumbs:** Navegación contextual siempre visible en la parte superior
5. **Empty States:** Mensajes guiados cuando no hay contenido ("Press space to continue...")

### Principios de Diseño Deducidos

- **Progressive Disclosure:** Funcionalidades avanzadas escondidas hasta que se necesitan
- **Keyboard-First:** Todo puede hacerse con teclado (atajos extensivos)



- **Consistency:** Misma interacción funciona igual en todos los tipos de bloques
- **Flexibility:** Múltiples formas de lograr lo mismo (mouse, teclado, comandos)

## 5.6 Modelo de negocio: Freemium

Feature	Plan Personal (Free)	Plan Plus	Plan Business	Plan Enterprise
<b>Bloques</b>	Ilimitados (antes 1000)	Ilimitados	Ilimitados	Ilimitados
<b>File uploads</b>	5MB/file	Ilimitado	Ilimitado	Ilimitado
<b>Invitados</b>	10	100	250	Ilimitado
<b>Version history</b>	7 días	30 días	90 días	Ilimitado
<b>API access</b>	☑	☑	☑	☑
<b>Advanced permissions</b>	✗	☑	☑	☑
<b>SAML SSO</b>	✗	✗	✗	☑
<b>Dedicated support</b>	✗	✗	✗	☑

## 5.7 Análisis Competitivo (Contexto)

¿Cómo se diferencia Notion?

Feature	Notion	Confluence	Google Docs	Evernote	Obsidian
<b>Bloques modulares</b>	☑	✗	✗	✗	Parcial
<b>Bases de datos</b>	☑	✗	Parcial	✗	Parcial
<b>Tiempo real</b>	☑	☑	☑	✗	✗
<b>Offline robusto</b>	Parcial	✗	Parcial	☑	☑
<b>Templates</b>	☑	☑	☑	✗	☑
<b>API pública</b>	☑	☑	Parcial	✗	Limitada

**Ventaja Competitiva:** Notion combina **notas + base de datos + wiki + gestión de proyectos** en una sola herramienta con UX excepcional.

## 6. Conclusiones y Reflexión final

- **Deconstrucción Analítica y Arquitectura Inferida:** La ingeniería inversa demostró ser una metodología crítica para descomponer sistemas complejos ("Caja Negra"); al observar el comportamiento del frontend, logramos inferir con precisión la arquitectura backend, el modelo de datos relacional y los requisitos funcionales sin necesidad de acceder al código fuente.
- **Sincronización y Patrones Modernos:** Identificamos que las aplicaciones SaaS contemporáneas como Notion dependen de arquitecturas distribuidas y microservicios; confirmamos que la "colaboración en tiempo real" es una orquestación técnica sofisticada



ESCUELA POLITÉCNICA NACIONAL  
FACULTAD DE INGENIERIA DE SISTEMAS  
(ISWD633) CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE  
VICENTE EGUEZ



que combina WebSockets persistentes con algoritmos complejos de resolución de conflictos (CRDT/OT).

- **Simbiosis entre UX e Ingeniería:** Comprendimos que decisiones de diseño como la "Optimistic UI" o la carga diferida (*lazy loading*) no son meramente estéticas, sino imperativos arquitectónicos; la experiencia de usuario fluida dicta requisitos técnicos estrictos sobre la latencia, el manejo de errores y la consistencia de datos.

### Reflexión Final

Este ejercicio de examen reafirma que la excelencia en la ingeniería de software trasciende la escritura de código; se trata fundamentalmente de la capacidad para **leer, comprender y diagnosticar sistemas vivos**. Al actuar como "detectives tecnológicos", aprendimos que un ingeniero completo no solo construye desde cero, sino que posee la agudeza analítica para aprender de las estructuras, patrones y complejidades de los sistemas existentes para diseñar soluciones futuras más robustas, escalables y eficientes.

## 7. Referencias

- [1] Notion Labs Inc., "Notion - The all-in-one workspace," Notion.so. [Online]. Available: <https://www.notion.so>. [Accessed: Dec. 14, 2024].
- [2] Notion Labs Inc., "Notion API Documentation," Notion Developers, 2024. [Online]. Available: <https://developers.notion.com>. [Accessed: Dec. 14, 2024].
- [3] M. Kleppmann, "Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems," O'Reilly Media, 2017.
- [4] Google, Gemini. [Prompt] "Ayúdame con una estructura completa y didáctica de informe para un análisis de ingeniería inversa..." 2025. Disponible: <https://gemini.google.com/>.