



Estudiante: Javier Quilumba

Taller-011: Ingeniería Inversa

Fecha: 9/11/2025

## 1. Aplicaciones Analizadas

Aplicación	Empresa	Tipo de Software	Documento Legal Base
Notion (v3.0) (Escritorio/Web)	Notion Labs, Inc.	Comercial (Código Cerrado/SaaS)	Políticas de Uso
Brave Browser (Núcleo) (v1.83.118)	Brave Software, Inc.	Código Abierto (Open Source)	Licencia Pública de Mozilla 2.0 (MPL 2.0)

## 2. Resumen de Hallazgos Legales y Citas

### Aplicación 1: Notion (Postura Restrictiva)



Notion adopta una postura de **prohibición total** de cualquier actividad relacionada con la ingeniería inversa o la descompilación, buscando proteger la integridad de su código fuente y sus derechos de propiedad intelectual. La restricción es directa y no presenta ambigüedades en su lenguaje.

**Protección de la Propiedad Intelectual:** *El usuario se compromete a **no realizar ingeniería inversa, descompilar, desensamblar ni intentar descifrar el código fuente del plugin.***

Fuente: [Políticas de uso de Bim Technology & Design \(Documentación vinculada al uso de software/plugins de Notion\)](#).

### Aplicación 2: Brave Browser (Postura Permisiva)



Brave, al ser un navegador web que promueve la privacidad, basa su código fuente principal en **código abierto**, específicamente la **Licencia Pública de Mozilla 2.0 (MPL 2.0)**. Esta licencia establece un estándar opuesto al código cerrado.

**Hallazgo Legal Clave (Licencia):** El código de cliente de Brave se halla publicado bajo la **Licencia pública de Mozilla 2.0**.

Fuente: [Página oficial de Brave](#) / [Repositorio de GitHub de Brave Browser \(MPL-2.0 license\)](#).

**Implicación Legal de la MPL 2.0:** La MPL 2.0 es una licencia permisiva que **permite** explícitamente a los usuarios **recibir, modificar, descompilar y usar el código fuente** (es decir, realizar ingeniería inversa), siempre que cualquier modificación significativa al código fuente también se publique bajo la MPL 2.0.

## 3. Análisis Comparativo y Reflexión

Aspecto	Notion (Cerrado)	Brave (Abierto - MPL 2.0)
Permite I. Inversa	<b>No, Prohibición Absoluta.</b> Es un incumplimiento de contrato.	<b>Sí, Permitida</b> (por los términos de la licencia).
Excepciones Legales	No hay excepciones explícitas mencionadas; la prohibición es total.	La I. Inversa está permitida por la licencia MPL 2.0. Las restricciones solo aplican a servicios propietarios asociados (si los hay).



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



<b>Riesgo Legal</b>	<b>Alto</b> (Demanda por incumplimiento del EULA y violación de derechos de autor).	<b>Bajo</b> (El uso y estudio del código está protegido por la licencia).
<b>Impacto en Seguridad</b>	<b>Negativo</b> , ya que limita la auditoría externa por investigadores de seguridad.	<b>Positivo</b> , ya que la comunidad puede auditar el código y encontrar vulnerabilidades.

### Postura Empresarial

- **Notion:** Su postura es la de una empresa de software propietario que busca proteger sus secretos comerciales y su modelo de negocio (SaaS). La I. Inversa es vista como una amenaza directa a su propiedad intelectual.
- **Brave:** Su postura es la de una empresa que basa su valor en la confianza y la transparencia. Al usar código abierto, permiten la I. Inversa para que los usuarios y la comunidad verifiquen que el navegador hace lo que promete (proteger la privacidad).

### Reflexión sobre Innovación y Educación

Creo que la **restricción total** de la ingeniería inversa, como la implementada por Notion, puede afectar la **innovación** al limitar la creación de productos complementarios o la mejora de la **seguridad** al impedir la auditoría independiente.

Sin embargo, el caso de Brave ilustra que la **ingeniería inversa es una herramienta de seguridad y educación** cuando se hace bajo licencias abiertas. Esto nos enseña que la ética y la legalidad de la I. Inversa no dependen de la técnica en sí, sino de la **naturaleza de la licencia** bajo la cual se distribuye el software.

## 4. Conclusiones para Ingenieros de Sistemas

Los futuros ingenieros de sistemas debemos entender que:

1. La **Licencia lo es todo**: La legalidad de la ingeniería inversa se reduce a lo que el EULA o la licencia de código abierto permitan.
2. **Riesgo vs. Transparencia**: Trabajar con software de código cerrado (Notion) sin permiso conlleva un alto riesgo legal. Trabajar con software abierto (Brave) permite un desarrollo más transparente y auditable.
3. **Ética**: Debemos priorizar la comprensión de la ley antes de aplicar la técnica. Si el objetivo es la interoperabilidad, es fundamental buscar las "excepciones legales" o utilizar APIs públicas.

## 5. Referencias Legales Consultadas

- [1] Notion Labs, Inc. "Políticas de uso." [Artículo en Línea]. <https://btddocs.notion.site/Pol-ticas-de-uso-c2e97bb6c74c45bc9d98860c400fb453?pvs=74>.
- [2] Brave Software, Inc. "Términos de uso." [Artículo en Línea]. <https://brave.com/terms-of-use/>.
- [3] Brave Software, Inc. "Brave browser for Android, iOS, Linux, macOS, Windows." [Online Repository]. <https://github.com/brave/brave-browser>.
- [4] Mozilla Foundation. "Mozilla Public License, version 2.0 (MPL-2.0)." [Documento de Licencia]. <https://www.mozilla.org/en-US/MPL/2.0/>.
- [7] Eguez Adrian. Presentations.ai. (2025). *La Ingeniería Inversa en el Desarrollo de Software*. [Diapositivas del curso]. <https://app.presentations.ai/view/ekPbpi>