



Estudiante: Jonathan Tipán

Fecha: 09/11/2025

## Taller-011: Ingeniería Inversa

### 1. Aplicaciones Analizadas

Aplicación	Empresa	Tipo de Software	Documento Legal Base
Spotify (v2025)	Spotify Technology S.A.	Comercial (Código Cerrado)	Acuerdo de Usuario Final (EULA)
Unity (v2023)	Unity Technologies	Comercial (Código Cerrado)	Acuerdo de Licencia de Usuario Final (EULA)

### 2. Resumen de Hallazgos Legales y Citas

#### Aplicación 1: Spotify (Postura Restrictiva)

Spotify, como servicio de música en streaming, opera bajo una licencia propietaria que restringe la ingeniería inversa de su código. La compañía busca proteger sus derechos de propiedad intelectual y evitar la modificación no autorizada de su software y servicios.

#### Protección de la Propiedad Intelectual:

El usuario se compromete a no realizar ingeniería inversa, descompilar, desensamblar ni intentar descifrar el código fuente de la aplicación o sus componentes.

Fuente: Acuerdo de Usuario Final (EULA) de Spotify.

#### Aplicación 2: Unity (Postura Restrictiva con Excepciones)

Unity es una plataforma de desarrollo de videojuegos que también opera bajo una licencia propietaria. Sin embargo, su EULA permite ciertas excepciones para realizar ingeniería inversa en contextos específicos, como la depuración de errores o la creación de interoperabilidad entre software.

#### Hallazgo Legal Clave (Licencia):

Unity prohíbe la ingeniería inversa en general, pero permite ciertos casos de descompilación con fines de depuración y compatibilidad.

Fuente: Acuerdo de Licencia de Usuario Final (EULA) de Unity.

#### Implicación Legal del EULA de Unity:

La licencia permite ingeniería inversa solo en situaciones limitadas, como cuando es necesario para la interoperabilidad con otras plataformas o cuando la depuración es esencial para corregir errores del sistema.

### 3. Análisis Comparativo y Reflexión

Aspecto	Spotify (Cerrado)	Unity (Cerrado - Excepciones)
Permite Ingeniería Inversa	No, Prohibición Absoluta. Es un incumplimiento de contrato.	Sí, Permitida en casos limitados (como depuración y compatibilidad).

<b>Excepciones Legales</b>	No hay excepciones explícitas; la prohibición es total.	La ingeniería inversa está permitida solo para depuración o interoperabilidad.
<b>Riesgo Legal</b>	Alto (Demanda por incumplimiento del EULA y violación de derechos de autor).	Moderado (Solo permitido bajo condiciones específicas).
<b>Impacto en Seguridad</b>	Negativo, ya que limita la auditoría externa por investigadores de seguridad.	Positivo, ya que permite realizar auditorías bajo ciertas condiciones.
<b>Postura Empresarial</b>	Spotify: Postura de una empresa propietaria que busca proteger sus derechos de autor y evitar la modificación no autorizada del software.	Unity: Postura de una empresa que restringe la ingeniería inversa, pero permite excepciones para mejorar la interoperabilidad y corregir errores.

### Reflexión sobre Innovación y Educación:

La restricción total de la ingeniería inversa en Spotify puede limitar la innovación, ya que impide a los desarrolladores crear herramientas complementarias o mejorar la seguridad mediante auditorías. En el caso de Unity, la licencia permite la ingeniería inversa solo en circunstancias específicas, lo que refleja un enfoque más flexible para fomentar la innovación en la comunidad de desarrollo de videojuegos, sin comprometer la protección de los derechos de autor.

## 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 depende totalmente de lo que el EULA permita, y en ambos casos (Spotify y Unity), las licencias restringen esta práctica, aunque con diferentes enfoques.
2. **Riesgo vs. Flexibilidad:** Trabajar con software de código cerrado, como Spotify y Unity, sin permiso conlleva riesgos legales. Unity ofrece más flexibilidad al permitir ingeniería inversa bajo ciertas circunstancias, lo cual facilita la innovación en proyectos específicos.
3. **Ética:** Es fundamental comprender las implicaciones legales antes de aplicar la ingeniería inversa, especialmente si el objetivo es mejorar la interoperabilidad o realizar depuración en proyectos tecnológicos.

## 5. Referencias Legales Consultadas

[1] Spotify Technology S.A. "Acuerdo de Usuario Final (EULA) de Spotify." [Artículo en Línea].  
<https://www.spotify.com/legal/end-user-license-agreement>

[2] Unity Technologies. "Acuerdo de Licencia de Usuario Final (EULA) de Unity." [Documento en Línea].  
<https://unity.com/legal/eula>