



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
CONSTRUCCIÓN Y EVOLUCIÓN DE SOFTWARE



Nombre: Belén Cholango.

Curso: GR2SW.

Taller: Ingeniería Inversa y Sus Implicaciones Legales

1. Aplicaciones Analizadas

Spotify

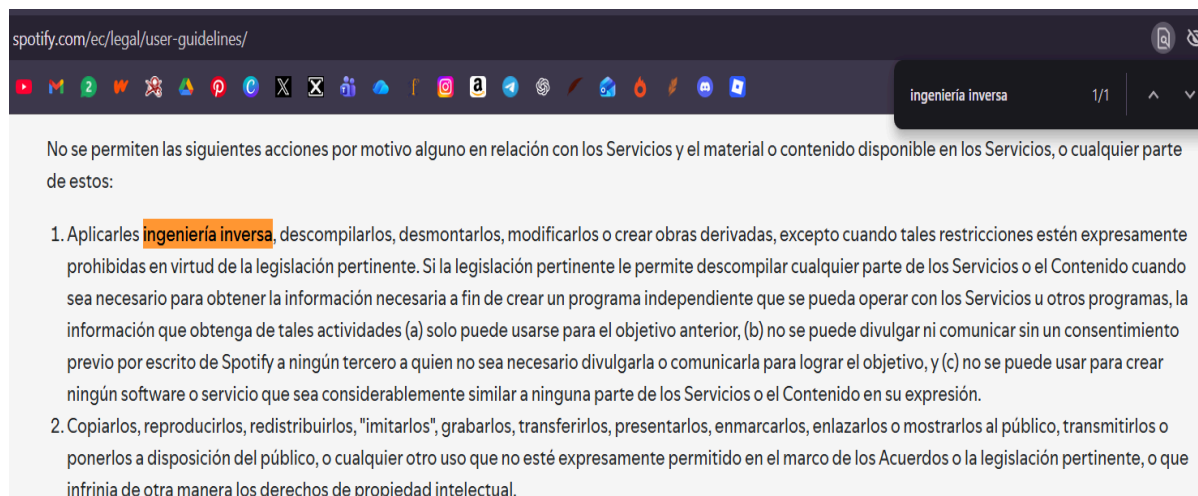
- **Empresa:** Spotify AB
- **Tipo:** Plataforma de streaming de música
- **Versión:** Aplicación de escritorio/móvil (versión pública más reciente)

Telegram

- **Empresa:** Telegram FZ-LLC
- **Tipo:** Aplicación de mensajería instantánea
- **Versión:** Cliente oficial (Android/iOS)

2. Resumen de Hallazgos Legales

Spotify: Los Términos de Uso de Spotify contienen prohibiciones claras y explícitas sobre ingeniería inversa.



Esto significa que cualquier intento de descompilar o analizar el funcionamiento interno del software está prohibido contractualmente, salvo que una ley específica obligue a lo contrario.

Telegram: La situación es diferente. Los clientes oficiales de Telegram (por ejemplo, Telegram Android) están publicados bajo licencia GPL, lo cual sí permite estudiar y modificar el código, siempre que se respeten los términos de la GPL.

Sin embargo, los Términos de Servicio prohíben actividades que afecten sus sistemas:

“No debe intentar hackear, desbordar o interferir con los sistemas o servicios.”

(Términos de uso de Telegram)

Compilaciones reproducibles para iOS y Android

Esta página contiene instrucciones para verificar que el código fuente abierto de Telegram es exactamente el mismo que el código que se utiliza para crear las aplicaciones que están disponibles en la [App Store](#) , [Google Play](#) y [directamente](#) en el sitio web de Telegram.

Advertencia: Telegram admite compilaciones reproducibles a partir de [la versión 5.13](#) . Tenga en cuenta que, en esta etapa, el proceso de verificación debe considerarse **experimental** . Actualizaremos nuestras aplicaciones y estas instrucciones para que este proceso sea lo más sencillo posible.

- [Telegram para Android](#)
- [Telegram para iOS](#)

Esto implica que el cliente open source puede analizarse legalmente, porque la licencia lo permite y el backend/servidor es propietario, y cualquier ingeniería inversa sobre él estaría restringida.

3. Análisis Comparativo

Aspecto	Spotify	Telegram
Ingeniería inversa permitida	Prohibida totalmente	Permitida solo en el cliente (por GPL). Restringida en el servidor
Claridad en términos	Muy clara: prohibición directa	Ambigua: depende de la parte del sistema
Tipo de licencia	Propietaria	Mixta: cliente GPL + backend cerrado
Riesgos legales	Altos (violación de contrato)	Moderados: depende del alcance del análisis
Postura general	Muy restrictiva	Parcialmente abierta

Spotify mantiene un modelo completamente cerrado, mientras que Telegram ofrece un enfoque híbrido, permitiendo estudio del cliente pero protegiendo la infraestructura.

4. Conclusiones

1. Spotify adopta una postura estrictamente restrictiva, prohibiendo la ingeniería inversa de forma explícita y sin excepciones prácticas significativas.
2. Telegram tiene un enfoque más flexible, ya que parte de su software es open source. Esto permite la ingeniería inversa del cliente, aunque el servicio en sí sigue estando protegido por términos legales.
3. Las diferencias muestran cómo las empresas equilibran seguridad, protección intelectual e interoperabilidad.

5. Referencias Legales Consultadas

- Spotify – User Guidelines y Términos de Uso: <https://www.spotify.com/ec/legal/user-guidelines/>
- Telegram – Términos de Servicio disponibles públicamente: <https://telegram.org/tos>
- Licencia GPL del cliente oficial de Telegram (repositorios open source): <https://core.telegram.org/reproducible-builds>