

# Jellyfin: Servidor multimedia para películas, series y música

---

## Introducción

Este artículo documenta el uso de Jellyfin como servidor multimedia dentro del stack self-hosted. Jellyfin es una plataforma totalmente libre y de código abierto orientada a la gestión y distribución de contenido multimedia (películas, series, música, fotografías y otros formatos) mediante un modelo cliente-servidor, sin dependencias externas, licencias propietarias ni funcionalidades bloqueadas.

Se integra como servicio de streaming interno, accesible desde navegadores web, aplicaciones móviles, clientes de escritorio, Smart TVs y dispositivos compatibles con DLNA, actuando como alternativa directa a soluciones privativas como Plex o Emby, pero manteniendo control total sobre los datos, el acceso y la infraestructura.

---

## Enfoque general / Arquitectura

El despliegue se realiza mediante contenedor Docker utilizando la imagen de LinuxServer.io, priorizando estabilidad, mantenimiento sencillo y compatibilidad con aceleración por hardware. El servicio se ejecuta de forma persistente, con volúmenes separados para configuración, caché de transcodificación y bibliotecas multimedia.

El acceso se limita a red local, sin exposición directa a Internet ni integración con sistemas de autenticación externos, ya que los clientes utilizados son mayoritariamente aplicaciones nativas dentro de la red doméstica. Para funcionalidades como DLNA, se opta por un modo de red que permita el descubrimiento automático de dispositivos sin fricción.

---

## Desarrollo

# Qué se hizo y por qué

- Se eligió Jellyfin por ser 100 % libre, sin telemetría ni dependencias en servicios cloud externos.
- Se utiliza aceleración por GPU para reducir carga de CPU durante transcodificación, especialmente en reproducción simultánea o dispositivos con codecs limitados.
- El uso de DLNA se considera prioritario para compatibilidad con televisores y dispositivos que no disponen de cliente nativo de Jellyfin.
- Para garantizar el correcto funcionamiento de DLNA, se ejecuta el contenedor en `network_mode: host`, evitando problemas de descubrimiento que aparecen al mapear puertos UDP en modo bridge.

## Configuración utilizada (solo enlaces)

- [docker-compose.yml](#)

Toda la configuración activa del servicio se mantiene versionada en Gitea.

---

## Validación

- Acceso correcto a la interfaz web de Jellyfin desde red local.
  - Reproducción directa y transcodificada funcionando según el cliente.
  - Uso efectivo de aceleración por GPU durante transcodificación.
  - Descubrimiento DLNA operativo en televisores y dispositivos compatibles.
  - Persistencia correcta de configuración y bibliotecas tras reinicios del contenedor.
- 

## Decisiones importantes o problemas detectados

- El modo `bridge` con puertos mapeados no permitió el descubrimiento DLNA en este entorno; el cambio a `network_mode: host` fue necesario.
  - No se integra con Authentik ni otros sistemas SSO al no existir exposición pública ni necesidad real de autenticación federada.
  - La carpeta de transcodificación se separa explícitamente para evitar impacto en volúmenes de datos principales.
- 

## Resumen breve

Jellyfin se utiliza como servidor multimedia centralizado, libre y sin dependencias externas, desplegado en Docker con aceleración por hardware y soporte DLNA completo. La configuración prioriza simplicidad, rendimiento y control total del acceso dentro de red local, manteniendo toda la definición del servicio versionada en Gitea.

---

## Referencias

- [Repositorio de Github de Jellyfin](#)
- 

Revision #4

Created 2024-11-21 16:23:11 UTC by Juan Francisco

Updated 2026-01-13 22:18:13 UTC by Juan Francisco