

Dispatcharr: Gestión y proxy de IPTV self-hosted

Introducción

Este artículo documenta **Dispatcharr**, un servicio self-hosted orientado a la **gestión, normalización y proxy de fuentes IPTV**, integrable dentro del stack de streaming doméstico. Se sitúa como capa intermedia entre **fuentes IPTV (M3U, Xtream, XMLTV)** y clientes finales como Jellyfin o Emby, ofreciendo una única entrada estable y controlada para canales en directo y VOD.

El enfoque se basa en centralizar playlists, EPG y streams en un único punto, reduciendo complejidad en los clientes y permitiendo reutilizar las mismas fuentes desde distintos servicios sin duplicar configuraciones.

Enfoque general / Arquitectura

Dispatcharr actúa como **middleware de streaming**, realizando las siguientes funciones dentro del stack:

- Importación de listas IPTV (M3U/M3U8, Xtream Codes).
- Gestión y normalización de EPG (XMLTV).
- Proxy de streams con control de conexiones y estado.
- Exposición de los canales mediante **endpoints compatibles con HDHomeRun y M3U**, consumibles por distintos clientes.

El despliegue documentado utiliza el **modo All-In-One (AIO)**, donde backend, Redis, Celery y frontend conviven en un único contenedor, simplificando el mantenimiento y reduciendo dependencias externas. Esta aproximación encaja bien en entornos homelab donde se prioriza estabilidad y bajo coste operativo frente a escalado horizontal.

Requisitos previos

- Entorno Docker operativo.
 - Red Docker externa definida previamente para integración con el resto del stack.
 - Cliente compatible con **M3U o HDHomeRun**.
-

Desarrollo

Qué se hizo y por qué

Se optó por el **modo AIO** de Dispatcharr para minimizar complejidad y acelerar la validación funcional. Esta elección elimina la necesidad de desplegar Redis o workers de forma separada y es suficiente para escenarios de consumo doméstico o pruebas prolongadas.

El servicio se utiliza como **único punto de entrada IPTV**, evitando configuraciones duplicadas en cada cliente multimedia y permitiendo cambiar o ajustar las fuentes sin impacto directo en los sistemas consumidores.

Configuración utilizada (solo enlaces)

La configuración completa se mantiene versionada en Gitea:

- [docker-compose.yml — Dispatcharr](#)
-

Validación

La validación realizada se ha centrado en **funcionamiento real**, no en pruebas sintéticas:

- Importación y reproducción correcta de **canales IPTV públicos**.
- Carga y exposición de canales mediante:
 - Endpoint **HDHomeRun**.
 - Playlist **M3U** generada por Dispatcharr.
- Integración verificada en Jellyfin y Emby.
- Reproducción estable desde ambos clientes sin errores de carga ni problemas de compatibilidad.
- Persistencia correcta de datos y configuración mediante volumen dedicado.

El comportamiento observado ha sido estable, sin incidencias relevantes durante las pruebas.

Decisiones importantes o problemas detectados

- El modo AIO simplifica el despliegue, pero limita la capacidad de separar cargas (streaming frente a tareas en segundo plano). Para un entorno homelab no supone una limitación práctica.
 - La exposición mediante HDHomeRun facilita la integración con clientes, evitando configuraciones manuales complejas.
 - El uso de playlists públicas permite validar funcionalidad sin depender de proveedores externos.
-

Resumen breve

Dispatcharr funciona como una capa intermedia eficaz para centralizar IPTV, EPG y streaming en un entorno self-hosted. El despliegue en modo AIO resulta estable y suficiente para uso doméstico, integrándose sin problemas con Jellyfin y Emby mediante M3U y HDHomeRun.

Referencias

- [Dispatcharr — Repositorio oficial en GitHub](#)
-

Revision #1

Created 2026-01-14 08:54:36 UTC by Juan Francisco

Updated 2026-01-14 08:59:04 UTC by Juan Francisco