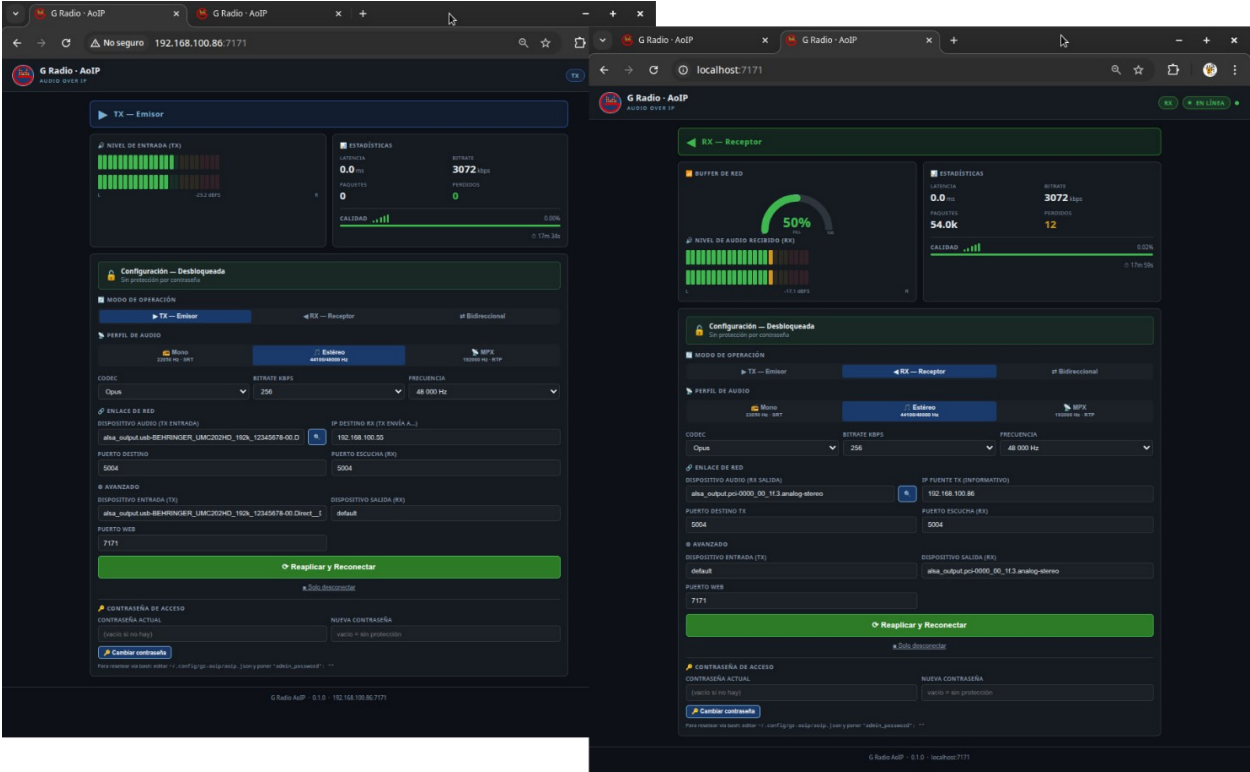


# G Radio AoIP — Manual de Usuario



Versión 0.1.0

Charles Escobar - 2026

---

# Índice

1. [¿Qué es G Radio AoIP?](#)
2. [Requisitos del sistema](#)
3. [Instalación](#)
4. [Inicio rápido](#)
5. [Interfaz web](#)
6. [Perfiles de audio](#)
7. [Modos de operación](#)
8. [Configuración de dispositivos de audio](#)
9. [Configuración de red](#)
10. [Protección por contraseña](#)
11. [Operación headless \(sin entorno gráfico\)](#)
12. [Integración con G Radio Player](#)
13. [Diagnóstico y solución de problemas](#)
14. [Referencia de la API REST](#)

---

# 1. ¿Qué es G Radio AoIP?

**G Radio AoIP** es una aplicación de Audio sobre IP (AoIP) diseñada para estaciones de radio. Permite transportar audio entre dos puntos remotos mediante una red IP (LAN o internet).

## Casos de uso

Caso	Perfil	Descripción
Enlace en condiciones adversas	<b>Mono</b>	Voz/audio monofónico a 22050 Hz con Opus y SRT. Robusto ante pérdida de paquetes.
STL (Studio-to-Transmitter Link)	<b>Estéreo</b>	Audio estéreo a 44100/48000 Hz con Opus, AAC o PCM sobre SRT.
Enlace MPX para FM	<b>MPX</b>	Señal compuesta FM (MPX) a 192000 Hz con PCM L16 sobre RTP/UDP. Sin compresión.
Reportaje desde campo	<b>Mono</b>	El periodista transmite desde calle con un teléfono o RPi al estudio.

## Arquitectura

[Fuente de audio] | [TX — Emisor] — red IP — [RX — Receptor] |  
| UMC202HD, hw:x,y Tarjeta de audio local o default (PipeWire) o  
altavoz/transmisor

Cada extremo corre `gr-aoip` de forma independiente. La configuración y el monitoreo se hacen a través de un navegador web apuntando al equipo correspondiente.

---

## 2. Requisitos del sistema

### Mínimos (headless / RPi3+)

- Linux (Debian/Ubuntu/Raspberry Pi OS)
- GStreamer 1.20 o superior con plugins: `base`, `good`, `bad`
- Tarjeta de audio compatible con ALSA
- Para SRT (perfiles Mono y Estéreo): `gststreamer1.0-plugins-bad` con soporte SRT

### Con entorno gráfico (PipeWire)

- PipeWire o PulseAudio activo
- Los mismos plugins de GStreamer

### Para perfil MPX (192 kHz)

- Interfaz de audio que soporte 192000 Hz (ej. Behringer UMC202HD)
- Red LAN gigabit recomendada (bitrate ~3 Mbps en PCM L16 mono)

## Verificar GStreamer

```
gst-inspect-1.0 srtsink # para perfiles Mono y Estéreo
gst-inspect-1.0 rtpL16pay # para perfil MPX
```

---

## 3. Instalación

### Desde el paquete ZIP (recomendado)

```
# Descomprimir
unzip GR-aqip-v0.1.0-arm64.zip
cd GR-aqip-v0.1.0-arm64/
# Instalar
chmod +x install.sh
./install.sh
```

El instalador copia el binario a `~/.local/bin/gr-aqip`, crea la configuración inicial en `~/.config/gr-aqip/aqip.json` y registra el servicio `systemd` de usuario.

### Desde el paquete .deb

```
sudo dpkg -i gr-aqip_0.1.0_arm64.deb
```

### Dependencias del sistema (Debian/Ubuntu)

```
sudo apt install -y \
  libgstreamer1.0-0 \
  gstreamer1.0-plugins-base \
  gstreamer1.0-plugins-good \
  gstreamer1.0-plugins-bad \
  gstreamer1.0-alsa
```

Para perfiles Opus/AAC/MP3 (Mono y Estéreo) añadir también:

```
sudo apt install -y \
  gstreamer1.0-plugins-ugly \
  gstreamer1.0-libav
```

---

## 4. Inicio rápido

### Arrancar el servicio

```
# Como servicio systemd (recomendado — arranca automáticamente)
systemctl --user enable --now gr-aqip
# O manualmente
gr-aqip
```


### Acceder a la interfaz web

Abre un navegador y apunta a:

```
http://<ip-del-equipo>:7171
```

El puerto por defecto es **7171** (configurable).

### Primer uso

1. Abre la interfaz web en **TX** (el equipo emisor)
  2. En la sección de configuración, selecciona el **Perfil de audio** (MPX, Estéreo o Mono)
  3. Configura la **IP destino** con la IP del equipo RX
  4. Haz clic en  **Aplicar y Conectar**
  5. En el equipo **RX**, la interfaz mostrará los paquetes recibidos y el nivel de audio
- 

## 5. Interfaz web

La interfaz está organizada en dos zonas:

## Zona siempre visible (sin contraseña)

**Banner de modo** — indica si el equipo opera como TX (azul), RX (verde) o Bidireccional (naranja).

**Monitor — Buffer de red** — gauge semicircular que muestra el porcentaje de llenado del buffer de red (solo RX y modo bidireccional). Colores:


- Verde: 30%–70% — operación normal
- Naranja: <30% o >70% — buffer fuera del rango óptimo
- Rojo: <10% o >90% — riesgo de corte

**Monitor — Nivel de audio (VU)** — barras que muestran el nivel en dBFS:

- En TX: nivel de la señal capturada (lo que se está enviando)
- En RX: nivel del audio recibido y reproducido


**Estadísticas** — latencia, bitrate, paquetes recibidos, paquetes perdidos y calidad del enlace en barras.

## Zona de configuración (protegida opcionalmente)


Se accede haciendo clic en la barra  **Configuración**. Si hay contraseña configurada, pide autenticación.

**Modo de operación** — TX / RX / Bidireccional

**Perfil de audio** — MPX / Estéreo / Mono

**Dispositivo de audio** — campo de texto con botón  para abrir el selector de dispositivos

**IP destino y puertos** — configuración de red

 **Aplicar y Conectar** — guarda la configuración y reinicia el pipeline de audio. Si ya estaba conectado, reconecta con la nueva configuración.

■ **Solo desconectar** — aparece cuando el enlace está activo. Detiene el pipeline sin guardar cambios.

---

## 6. Perfiles de audio

### MPX — 192 kHz (Enlace FM)

Para transportar la señal compuesta MPX del procesador de audio (ej. Orban, Omnia) al transmisor.

- **Frecuencia de muestreo:** 192000 Hz
- **Formato:** PCM L16 (sin compresión — la señal MPX no puede comprimirse)
- **Transporte:** RTP sobre UDP — baja latencia, sin corrección de errores
- **Bitrate:** ~3.072 Mbps (PCM L16 mono)
- **Red recomendada:** LAN dedicada o VLAN con QoS

**Latencia de procesamiento** — seleccionable entre 128 y 4096 muestras:

Muestras	Latencia	Uso recomendado
128	0.67 ms	LAN con switch dedicado
256	1.33 ms	LAN estándar (recomendado)
512	2.67 ms	LAN con algo de carga
1024	5.33 ms	Red con jitter moderado
2048	10.67 ms	WAN de baja latencia
4096	21.33 ms	WAN con jitter alto

**Buffer de red** — adicional a la latencia de procesamiento. Configurable (ms). El clock recovery adaptativo ajusta el buffer automáticamente para mantener el fill cerca del 50%.

## Estéreo — 44100/48000 Hz (STL)

Para enlace estudio-transmisor con audio estéreo de alta calidad.

- **Frecuencia de muestreo:** 44100 Hz o 48000 Hz
- **Codecs disponibles:** Opus (recomendado), AAC-LC, MP3, FLAC, PCM L16
- **Transporte:** SRT — tolera pérdida de paquetes, reconexión automática
- **Bitrate Opus:** 64–320 kbps configurable

**Opus a 256 kbps** es indistinguible de audio sin comprimir para contenido de radio. Para archivos de audio perfectamente fieles usar FLAC o PCM L16.

## Mono — 22050 Hz (Ambiente hostil)

Para transmisiones desde campo (reportajes, corresponsalías) donde la red puede ser inestable.

- **Frecuencia de muestreo:** 22050 Hz → internamente resampled a 24000 Hz (Opus)
- **Codec:** Opus (incluye corrección de errores FEC)
- **Transporte:** SRT (ARQ + FEC)
- **Bitrate:** 16–48 kbps configurable
- **Bitrate recomendado:** 32 kbps para campo adverso

---

## 7. Modos de operación

### TX — Emisor

Captura audio del dispositivo de entrada y lo envía al RX configurado.

- El campo **Dispositivo audio** corresponde a la **entrada** (micrófono, interfaz de audio, monitor de G Radio)
- La **IP destino** es la dirección del equipo receptor
- El **Puerto destino** debe coincidir con el **Puerto escucha** del RX

## RX — Receptor

Escucha en el puerto configurado y reproduce el audio recibido.

- El campo **Dispositivo audio** corresponde a la **salida** (altavoz, entrada del transmisor)
- La **IP destino** muestra la IP del TX (informativo, no se usa para RTP/SRT listener)
- El **Puerto escucha** es el puerto UDP/TCP en que escucha

## Bidireccional (TX + RX)




Ambos extremos pueden configurarse en modo bidireccional para comunicación de dos vías. En este caso ambos extremos deben usar puertos distintos para TX y RX para evitar conflictos.

---

## 8. Configuración de dispositivos de audio


### Selector de dispositivos (botón )

Al hacer clic se abre un modal con tres pestañas:

-  **Entradas** — fuentes de audio (micrófonos, líneas de entrada)
-  **Monitores** — loopback de salidas (lo que G Radio Player está reproduciendo)
-  **Salidas** — destinos de audio (altavoces, líneas de salida)

Cada dispositivo muestra:

- Nombre descriptivo (modelo del hardware)
- Especificación: formato, canales, frecuencia
- Nombre técnico completo (node.name de PipeWire o **hw:x,y** ALSA)

La opción  **Default del sistema** usa el dispositivo configurado como predeterminado en el sistema. Es la opción más segura y no requiere conocer el nombre técnico.

### Opciones de dispositivo

Valor	Descripción	Entorno
<b>default</b>	Dispositivo del sistema	PipeWire/ALSA
<b>alsa_input.usb-...</b>	Nombre completo PipeWire	Gráfico
<b>alsa_output....monitor</b>	Monitor (loopback de salida)	Gráfico
<b>hw:2,0</b>	ALSA directo (tarjeta 2, dispositivo 0)	Headless y gráfico
<b>plughw:2,0</b>	ALSA con conversión automática	Headless

### Encontrar dispositivos desde terminal

```
# Con PipeWire/PulseAudiopactl list short sources # entradas y monitorespactl list short
```

```
sinks # salidas# Con ALSA directo (headless)arecord -l # tarjetas de
capturaaplay -l # tarjetas de reproducción
```

## Para capturar lo que G Radio Player reproduce

Usar el **Monitor** de la tarjeta de audio. En PipeWire su nombre termina en `.monitor`. Esto captura el audio de salida de G Radio (la señal procesada que va al aire) sin necesidad de conexión física.

## 9. Configuración de red

### Puertos

Por defecto se usa el **puerto 5004** para el audio y **7171** para la interfaz web.

Puerto	Protocolo	Uso
7171	TCP	Interfaz web y API REST
5004	UDP	Perfil MPX (RTP)
5004	TCP/UDP	Perfiles Mono y Estéreo (SRT)

### Esquema de conexión

Para **MPX** (RTP/UDP):

```
TX: udpsink host=<IP_RX> port=5004RX: udpsrc port=5004 (listener)
```

Para **Mono/Estéreo** (SRT):

```
TX: srtsink uri=srt://<IP_RX>:5004 ← TX como caller, conecta al RXRX: srtsrc uri=srt://:5004
← RX como listener, espera conexión
```

El TX debe iniciar o reiniciar cuando el RX ya está escuchando.


### Firewall

Abrir los puertos necesarios en el RX:


```
# Para MPX (UDP)sudo ufw allow 5004/udp# Para Mono/Estéreo (SRT usa UDP)sudo ufw
allow 5004/udp# Interfaz websudo ufw allow 7171/tcp
```

## 10. Protección por contraseña


### Configurar contraseña

1. Accede a la interfaz web
2. La sección de configuración está abierta (sin contraseña aún)
3. Al final de la sección busca  **Contraseña de acceso**
4. Deja "Contraseña actual" vacía (primera vez)
5. Escribe la nueva contraseña en "Nueva contraseña"

## 6. Haz clic en **Cambiar contraseña**



A partir de ese momento, cada vez que se cargue la página mostrará  **Configuración — Bloqueada**. Las estadísticas y el VU metro siguen siendo visibles sin contraseña.

## Desbloquear configuración

1. Haz clic en la barra  **Configuración — Bloqueada**
2. Ingresa la contraseña en el modal
3. La configuración queda desbloqueada durante la sesión del navegador
4. Al cerrar la pestaña o recargar la página vuelve a bloquearse

## Cambiar contraseña

Con la configuración desbloqueada:

1. Ir a la sección  **Contraseña de acceso**
2. Escribir la contraseña actual en "Contraseña actual"
3. Escribir la nueva en "Nueva contraseña" (vacío = quitar protección)
4. Clic en  **Cambiar contraseña**

## Resetear contraseña desde bash

Si se olvidó la contraseña, se puede resetear directamente editando el archivo de configuración:

```
# Método 1 — Con Python (recomendado)python3 -c "import jsonpath =
'/home/$USER/.config/gr-aoip/aoip.json'with open(path) as f: c =
json.load(f)c['admin_password'] = 'with open(path, 'w') as f: json.dump(c, f,
indent=2)print('Contraseña eliminada — reinicia gr-aoip')"# Método 2 — Con sedsed -i
's/"admin_password": "[^"]*" /"admin_password": ""/' \ ~/.config/gr-aoip/aoip.json#
Reiniciar el servicio para que tome efecto:systemctl --user restart gr-aoip# O si lo corres
manualmente: matar el proceso y volver a iniciarlo
```

Después del reset, la interfaz web mostrará la configuración desbloqueada y podrás establecer una nueva contraseña.

## Verificar el estado del archivo de configuración

```
python3 -m json.tool ~/.config/gr-aoip/aoip.json | grep admin_password# Debe mostrar:
"admin_password": "" (vacío = sin protección)
```

---

# 11. Operación headless (sin entorno gráfico)

gr-aoip funciona completamente sin interfaz gráfica. La UI web es la única interfaz de usuario — un navegador en cualquier equipo de la red puede controlar el sistema.

## Configurar para headless

Usar dispositivos ALSA directos (**hw:x,y**) en vez de PipeWire:

1. Encontrar la tarjeta:

```
arecord -l # entradas aplay -l # salidas # Anotar el número de tarjeta y dispositivo (card N, device D → hw:N,D)
```

2. En la interfaz web, botón 🔍 → pestaña 🎤 **Entradas** o 🎧 **Salidas** → seleccionar el dispositivo con etiqueta (ALSA directo — headless)

O escribir directamente `hw:2,0` en el campo de dispositivo.

## Ventajas de ALSA directo vs PipeWire

	ALSA directo (hw:x,y)	PipeWire/PulseAudio
Entorno gráfico	No requerido	Requerido (sesión de usuario)
Latencia	Mínima	Algo mayor
Compatibilidad	Universal	Depende de la sesión
Para MPX	<b>Recomendado</b>	Puede añadir latencia
Para Mono/Estéreo	Compatible	Compatible

## Servicio systemd (recomendado para headless)

```
# Habilitar inicio automáticosystemctl --user enable gr-aoip# Ver estadossystemctl --user status gr-aoip# Ver logs en tiempo realjournalctl --user -u gr-aoip -f# Reiniciar después de cambio de configuraciónsystemctl --user restart gr-aoip
```

Para que el servicio de usuario arranque sin sesión activa (headless):

```
sudo loginctl enable-linger $USER
```

## 12. Integración con G Radio Player

### Caso 1: TX incorporado (STL desde estudio)

El RPi o PC del estudio corre G Radio Player y gr-aoip simultáneamente. gr-aoip captura el audio de salida de G Radio usando el **Monitor** de la tarjeta de audio.

```
G Radio Player → UMC202HD → [Monitor PipeWire] |
gr-aoip TX → red → gr-aoip RX → Transmisor
```

Configuración del TX:

- Dispositivo audio: `alsa_output.usb-<modelo>-sink.monitor` (visible en 📡 Monitores del picker)
- Perfil: Estéreo (para audio de programa) o MPX (si el procesador ya genera MPX)

## Caso 2: RX incorporado (Reportaje desde campo)

El periodista usa gr-aqip en un teléfono Android (APK) o laptop en modo TX. La estación corre gr-aqip en RX y alimenta la señal al mezclador.

### Arrancar ambos servicios juntos

Añadir gr-aqip al script de inicio de G Radio:

```
# En ~/.local/share/radio-player/gradio.sh, antes de iniciar radio-player:gr-aqip
&AOIP_PID=$!# ...# Al finalizar:kill $AOIP_PID 2>/dev/null
```

## 13. Diagnóstico y solución de problemas

### La página web no carga

```
# Verificar que el proceso está corriendops aux | grep gr-aqip# Ver en qué puerto está escuchandoss -ltnp | grep gr-aqip# Ver logs de iniciojournalctl --user -u gr-aqip -n 50# O si manual:cat /tmp/aqip-tx.log
```

### El enlace no conecta (packets\_rx = 0)

1. **Verificar conectividad:** `ping <IP_del_otro_equipo>`
2. **Verificar puertos:** en el RX, confirmar que el puerto de escucha está abierto
3. **Firewall:** `sudo ufw status` — asegurar que el puerto 5004 está permitido
4. **Orden de inicio:** para SRT, el RX debe estar escuchando antes que el TX intente conectar. Haz clic en "Aplicar y Conectar" en el RX primero, luego en el TX.

### Pérdida alta de paquetes (>2%)

- **Red con jitter:** aumentar el Buffer de red (ms) en la configuración
- **Red congestionada:** usar QoS/DSCP para priorizar el tráfico de audio
- **Para MPX:** la red LAN no debe tener switches con buffering excesivo

### Audio con artefactos / cortes

- **MPX:** aumentar la latencia de procesamiento (pasar de 256 a 512 o 1024 muestras)
- **Mono/Estéreo:** aumentar el buffer de jitter en el RX
- **Buffer fill muy bajo o muy alto:** el clock recovery ajusta automáticamente, pero puede tardar 1-2 minutos en estabilizarse

### El dispositivo de audio no funciona (headless)

```
# Verificar que la tarjeta existearecord -lplay -l# Probar captura directa (Ctrl+C para detener)arecord -D hw:2,0 -f S16_LE -r 48000 -c 2 test.wav# Probar reproducciónplay -D hw:2,0 test.wav
```

Si el dispositivo está ocupado por PipeWire:

```
# Ver qué proceso lo usa sof /dev/snd/pcmC2D0c # C2=card 2, D0=device 0, c=capture#
Opción 1: Usar PipeWire (pulsesrc/pulsesink) en vez de hw:# → Cambiar el dispositivo a 'default' en la configuración#
Opción 2: Detener PipeWire para acceso ALSA
```

```
exclusivosystemctl --user stop pipewire pipewire-pulse
```

## Restablecer configuración completa

```
# Backup del config actualcp ~/.config/gr-aqip/aoip.json ~/.config/gr-aqip/aoip.json.bak# Eliminar config para regenerar por defectorm ~/.config/gr-aqip/aoip.jsonsystemctl --user restart gr-aqip
```

## 14. Referencia de la API REST

Todos los endpoints son relativos a <http://<ip>:<puerto>>.

Método	Endpoint	Descripción
GET	<a href="#">/api/config</a>	Obtener configuración actual
POST	<a href="#">/api/config</a>	Guardar configuración (JSON)
POST	<a href="#">/api/apply</a>	Guardar config + reiniciar pipeline
POST	<a href="#">/api/start</a>	Iniciar pipeline (config actual)
POST	<a href="#">/api/stop</a>	Detener pipeline
GET	<a href="#">/api/stats</a>	Estadísticas en tiempo real
GET	<a href="#">/api/devices</a>	Listar dispositivos de audio
GET	<a href="#">/api/auth</a>	Consultar si hay password activo
POST	<a href="#">/api/auth</a>	Verificar password ({"password":"..."})
POST	<a href="#">/api/password</a>	Cambiar password ({"old":"...", "new":"..."})
WS	<a href="#">/ws</a>	WebSocket — push de stats cada 200ms

### Ejemplo: verificar estado desde bash

```
# Stats actualescurl -s http://192.168.100.86:7171/api/stats | python3 -m json.tool# Iniciar pipelinecurl -s -X POST http://192.168.100.86:7171/api/start# Aplicar nueva config y reconectarcurl -s -X POST http://192.168.100.86:7171/api/apply \ -H 'Content-Type: application/json' \ -d @~/.config/gr-aqip/aoip.json
```

# Notas de versión

## v0.1.0 (2026-05-03)

- Perfiles MPX (192 kHz RTP), Estéreo (SRT+Opus/AAC/MP3/FLAC/PCM) y Mono (SRT+Opus)
  - Interfaz web con VU metro en TX y RX
  - Clock recovery adaptativo para MPX
  - Soporte PipeWire y ALSA directo (headless)
  - Selector de dispositivos con detección automática
  - Protección por contraseña de la configuración
  - Autostart configurable — enlace prioritario
  - Persistencia local de IP y dispositivo (localStorage)
  - API REST + WebSocket para integración con otros sistemas
- 

*G Radio AoIP es parte del ecosistema [G Radio Player](#).*

*Para soporte y actualizaciones: [chescobar@gmail.com](mailto:chescobar@gmail.com)*