

Documentacion Sesiones

1) Diagnóstico rápido

Problema:

La sesión se crea correctamente (el API responde con `200` o `session_id` válido), pero **el video nunca se muestra en el iframe o canvas**.

Causa raíz (probable):

Hay una **desincronización o error silencioso** entre la sesión creada en backend (HeyGen) y el **canal LiveKit/WebRTC** que transporta el video.

No basta con que la sesión exista — el stream debe iniciarse, y a veces el evento `trackSubscribed` nunca se dispara.

Impacto:

El usuario ve una pantalla vacía o congelada sin error visible, aunque el log diga “session created successfully”.

2) Causas técnicas más comunes

Categoría	Causa	Síntoma típico
Timing	Llamas <code>connectToLiveKit()</code> antes de que el backend haya activado el avatar.	La conexión se establece pero no hay video.
Token	El <code>access_token</code> expira o fue generado para otro <code>session_id</code> .	El video no se muestra, pero consola indica “connected”.
CORS / HTTPS	Si corres el front local sin HTTPS o sin <code>localhost</code> , algunos navegadores bloquean el stream.	Solo falla en producción o ciertos navegadores.
Audio/Video Permissions	El navegador bloquea micrófono/cámara o auto-play.	No aparece el video o se corta al instante.
LiveKit issue	El servidor de LiveKit de HeyGen puede tardar en propagar el canal.	Intermitente: a veces funciona, otras no.
DOM / render	El contenedor del video no existe o está oculto en el momento del attach.	<code>trackSubscribed</code> ocurre, pero no se ve nada.

3) Plan de solución (estable y probado)

1. **Retrasar conexión unos milisegundos** después de crear la sesión:

```
const session = await createAvatarSession();
await new Promise(r => setTimeout(r, 500)); // pequeño delay
```

```
await connectToLiveKit(session);
```

◆ Este delay da tiempo a HeyGen para inicializar el avatar en su backend antes de la conexión WebRTC.

2. Validar token y session_id:

Asegúrate de que el token que envías a LiveKit se genere **después** del `session_id` correcto y **no sea cacheado**.

3. Verificar eventos WebRTC:

En consola:

```
room.on('trackSubscribed', (track) => console.log('Video track received'));
room.on('connectionStateChanged', (state) => console.log('Connection:', state));
```

Si nunca se dispara `trackSubscribed`, el problema es la sesión en el servidor.

4. Forzar reintento automático:

Si pasan >5 s sin recibir el stream:

```
setTimeout(() => {
  if (!videoTrack) reconnectToLiveKit();
}, 5000);
```

5. Comprobar origen HTTPS y permisos autoplay:



Chrome especialmente bloquea autoplay sin interacción o HTTPS.

Si estás en localhost, usa `https://localhost:5173` o similar.

4) Qué puedes hacer para depurar

- En consola: revisa si hay logs como `trackPublished` o `Disconnected`.
- Si el `session_id` cambia en cada prueba, asegúrate de no reusar tokens.
- Prueba en una **red distinta o VPN** (LiveKit a veces tiene latencia regional).
- Verifica en HeyGen Dashboard si la sesión aparece como "Live".

En resumen

-  **Tu sesión se crea bien, pero el stream no siempre inicia.**
-  **Agrega un pequeño retraso** y controla los eventos LiveKit.

- 🛡️ Asegura **token fresco** + HTTPS + contenedor visible.
- 🔄 Si tras 5 s no hay `trackSubscribed`, **reintenta conectar**.

¿Podrías pegar aquí el fragmento donde llamas a `connectToLiveKit(session)` y creas la sesión (`createSession` o `initAvatarSession`)?

Así te indico exactamente **dónde colocar el delay o reintento**, para dejarlo 100 % estable.

Dijiste:

pregunta, la sesion de heygen interpreta textos y audios? o solo textos

ChatGPT dijo:

Excelente pregunta 📌

Depende **del tipo de sesión** que crees en HeyGen (porque hay *dos grandes modos de operación*).

Te lo detallo claramente:

🧩 1) Tipos de sesión en HeyGen

Tipo de sesión	Endpoint / uso típico	Entrada aceptada	Salida	Ejemplo de uso
Video API (asíncrona)	<code>/v1/video/generate</code> o <code>/v1/talks</code>	Texto (prompt o script)	Video MP4 generado	Crear videos con un avatar leyendo un texto.
Live Session (en tiempo real)	<code>/v1/streaming.create_session</code> + LiveKit	Texto (via <code>sendTextMessage</code>) y Audio (via <code>sendAudioData</code>)	Video en tiempo real (streaming)	Chat con avatar que responde en vivo o habla en respuesta a tu voz.

🗣️ 2) En sesiones **Live** (Realtime API)

Sí: **interpreta tanto texto como audio** 📌

✅ Texto

Puedes enviar texto directamente para que el avatar lo hable:

```
await room.localParticipant.publishData({
  type: "text",
  text: "Hola, soy tu asistente virtual."
});
```

✓ Audio

También puedes enviar audio (por ejemplo, la voz del usuario grabada con el micrófono):

```
await room.localParticipant.publishData({
  type: "audio",
  data: audioBuffer
});
```

HeyGen internamente convierte el audio a texto (STT → Speech-to-Text), lo interpreta, y genera la respuesta del avatar con TTS y animación facial sincronizada.

👉 **Esto funciona solo si el modelo de la sesión está configurado para conversación multimodal** (`input_audio_enabled: true` o similar al crear la sesión).

🧠 3) En sesiones de video normal (no live)

Solo acepta **texto**.

No puede recibir ni interpretar audio — el audio de salida siempre será la voz sintética leyendo tu texto.

⚙️ 4) Cómo saber cuál tienes

Mira tu código:

- Si usas `create_talk` o `generate_video` : → **solo texto**.
- Si usas `create_session` o `connectToLiveKit` : → **texto y audio**.

✓ Conclusión

Caso	¿Interpreta texto?	¿Interpreta audio?
<code>generate_video</code> o <code>create_talk</code>	✓ Sí	✗ No
<code>create_session</code> (Live Streaming)	✓ Sí	✓ Sí (si lo habilitas)

¿Quieres que te muestre un ejemplo completo de cómo enviar audio desde el micrófono al avatar (para que te responda en tiempo real)?

Puedo darte el bloque JS listo para integrarlo.