

Scene Scheduler - Manual de Usuario

Versión: 0.4 **Fecha:** 29 de octubre de 2025 **Aplicación:** Scene Scheduler para OBS Studio

Tabla de Contenidos

1. [Primeros Pasos](#)
 2. [Comprendiendo Scene Scheduler](#)
 3. [Descripción General de la Interfaz Web](#)
 4. [Gestión de Horarios](#)
 5. [Configuración de Eventos](#)
 6. [Configuración del Sistema](#)
 7. [Cómo Funciona Internamente](#)
 8. [Casos de Uso y Ejemplos](#)
 9. [Mejores Prácticas](#)
 10. [Resolución de Problemas](#)
 11. [Referencia Técnica](#)
 12. [Tarjeta de Referencia Rápida](#)
-

1. Primeros Pasos

1.1 ¿Qué es Scene Scheduler?

Scene Scheduler es una **herramienta de automatización externa** para OBS Studio que automatiza su programación de transmisión como una estación de televisión. Gestiona la reproducción de contenido programado basándose en horarios de tiempo precisos, permitiendo transmisiones completamente automatizadas 24/7 sin intervención manual.

Propósito Principal: Scene Scheduler está diseñado para **automatizar horarios de transmisión** - permitiéndole planificar con anticipación qué contenido se transmitirá y cuándo, y luego dejar que el sistema ejecute esas transiciones automáticamente. Piense en ello como crear una guía de programación de canal de TV que OBS sigue automáticamente.

Cómo funciona: - Usted crea un **horario** (grilla de programación) definiendo qué contenido se reproduce en momentos específicos - Scene Scheduler monitorea el reloj y **activa automáticamente cambios de escena/fuente** cuando llega el

momento de cada evento - El sistema funciona continuamente, ejecutando su horario programado 24/7 sin intervención humana - Una **Vista de Monitor** basada en web le permite observar el horario y el estado actual desde cualquier lugar - Una **Vista de Editor** opcional proporciona una interfaz de calendario visual para modificar el horario

¿Por qué usar Scene Scheduler? - **Automatización 24/7:** Perfecto para canales de streaming, señalización digital, servicios religiosos, conferencias o cualquier transmisión programada - **Cero intervención manual:** Una vez programado, el horario se ejecuta automáticamente - **Accesible desde la red:** Monitoree y edite desde cualquier dispositivo en su red (laptop, tablet, teléfono) - **Carga reducida del servidor:** La interfaz web se ejecuta en dispositivos cliente, no en la máquina OBS - **Arquitectura externa segura:** Se ejecuta fuera de OBS, por lo que los fallos no afectan su transmisión

1.2 Requisitos Previos

Antes de instalar Scene Scheduler, asegúrese de tener:

1. **OBS Studio** (versión 28.0 o superior recomendada) - Descargue desde: <https://obsproject.com/>
2. **OBS WebSocket Plugin** (versión 5.x) - OBS Studio 28+ incluye esto por defecto - Para versiones anteriores, instale desde: <https://github.com/obsproject/obs-websocket>
3. **Sistema Operativo:** - **Linux:** Probado en Ubuntu 20.04+, otras distribuciones deberían funcionar - **Windows:** Windows 10/11 (64-bit)
4. **Red:** OBS y Scene Scheduler deben estar en la misma máquina o accesibles vía red

1.3 Instalación de Inicio Rápido

Instalación en Linux

Paso 1: Descargar Scene Scheduler

```
bash
# Extraer el archivo descargado
tar -xzf scenescheduler-linux-amd64.tar.gz
cd scenescheduler
```

Instalación en Windows

Paso 1: Descargar Scene Scheduler 1. Extraiga el archivo scenescheduler-windows-amd64.zip descargado 2. Extraiga el archivo ZIP a una carpeta (ej., C:\scenescheduler\) 3. Abra Command Prompt o PowerShell en esa carpeta

Paso 2: Configurar OBS WebSocket

1. Abra OBS Studio
2. Vaya a **Tools → WebSocket Server Settings**
3. Habilite "Enable WebSocket server"
4. Establezca una contraseña (recomendado) o déjela en blanco para acceso solo local
5. Note el puerto (predeterminado: 4455)
6. Haga clic en **OK**

Paso 3: Configurar Scene Scheduler

Edite config.json:

```
json
{
  "obsWebSocket": {
    "host": "localhost",
    "port": 4455,
    "password": "your-obs-password"
  },
  "webServer": {
    "host": "0.0.0.0",
    "port": 8080,
    "hlsPath": "hls"
  },
  "schedule": {
    "jsonPath": "schedule.json",
    "scheduleSceneAux": "scheduleSceneAux"
  },
  "paths": {
    "hlsGenerator": "./hls-generator"
  }
}
```

Notas críticas de configuración: - obsWebSocket.password: Debe coincidir con su contraseña de OBS WebSocket - webServer.hlsPath: Directorio para archivos de vista previa HLS (relativo al ejecutable) - schedule.scheduleSceneAux: Nombre de la escena auxiliar de OBS (debe existir en OBS)

Paso 4: Iniciar Scene Scheduler

Linux:

```
bash
# Hacer ejecutable
chmod +x scenescheduler

# Ejecutar
./scenescheduler
```

Windows:

```
cmd
REM Ejecutar en Command Prompt
scenescheduler.exe

REM O hacer doble clic en scenescheduler.exe en File Explorer
```

Debería ver una salida como:

```
2025/10/28 10:30:15 INFO Scene Scheduler starting version=1.6
2025/10/28 10:30:15 INFO WebSocket connecting host=localhost port=4455
2025/10/28 10:30:15 INFO Connected to OBS Studio version=30.0.0
2025/10/28 10:30:15 INFO Web server listening address=http://0.0.0.0:8080
2025/10/28 10:30:15 INFO Schedule loaded events=0
```

Paso 5: Acceder a la Interfaz Web

Abra su navegador y navegue a Scene Scheduler. Puede acceder desde:

- **Misma máquina:** <http://localhost:8080>
- **Otros dispositivos en la red:** <http://<server-ip>:8080>
- Ejemplo: <http://192.168.1.100:8080>
- Reemplace <server-ip> con la dirección IP real de la máquina que ejecuta Scene Scheduler

Encontrando la dirección IP de su servidor:

Linux:

```
bash
ip addr show | grep inet
```

Windows:

```
cmd  
ipconfig
```

Busque la dirección IPv4 en su interfaz de red activa (usualmente comienza con 192.168.x.x o 10.x.x.x).

¿Por qué acceso remoto? El beneficio clave de Scene Scheduler es que puede controlar y monitorear OBS desde **cualquier dispositivo en su red** (laptop, tablet, teléfono), reduciendo la carga en la máquina que ejecuta OBS y permitiendo que múltiples personas monitorean el horario simultáneamente.

Debería ver la interfaz web de Scene Scheduler con dos vistas principales: - **Monitor View:** Muestra eventos actuales y próximos (solo lectura) - **Editor View:** Editor visual para schedule.json

1.4 Su Primer Evento de Horario

Creemos un evento simple que cambia a una escena en un momento específico:

1. **Abra Editor View** (haga clic en el botón "Editor" en la navegación superior)
2. **Agregue un nuevo evento** (haga clic en el botón "+ Add Event")
3. **Configure el evento** en el diálogo modal: - **Time:** Establezca a unos minutos desde ahora (ej., si son las 10:30, establezca 10:35) - **OBS Scene:** Seleccione una escena existente de su OBS (ej., "Scene 1") - **Duration:** Deje en predeterminado (00:05:00 = 5 minutos) - **Sources Tab:** Deje vacío por ahora (solo cambio de escena)
4. **Guarde** el evento (haga clic en "Save Event")
5. **Observe:** - El evento aparece en su lista de horarios - Cuando llega el momento programado, OBS automáticamente cambia a la escena seleccionada - La vista de monitor muestra el resultado "CURRENT EVENT"

Felicitaciones! Ha creado su primera transición de escena automatizada.

1.5 Comprendiendo la Vista de Monitor

La Vista de Monitor está diseñada para **observación pasiva**. Es perfecta para: - Mostrar en un monitor secundario en una sala de control - Compartir con miembros

del equipo que necesitan visibilidad pero no acceso de edición - Verificar el estado actual sin riesgo de cambios accidentales

Lo que ve: - **Hora actual** (se actualiza cada segundo) - **Evento activo** (resaltado con temporizador de cuenta regresiva) - **Próximos eventos** (vista previa del horario próximo) - **Codificación por colores:** - Verde: Evento activo actual - Amarillo: Siguiente evento (comienza pronto) - Blanco: Eventos futuros

1.6 Comprendiendo la Vista de Editor

La Vista de Editor proporciona **control completo del horario**. Úsela para: - Agregar, editar y eliminar eventos - Reordenar entradas del horario - Configurar configuraciones complejas de fuentes - Previsualizar fuentes antes de confirmar

Elementos clave de la interfaz: - + **Add Event:** Crea nueva entrada de horario - **Lista de eventos:** Muestra todos los eventos programados con controles - **Botón Edit** (ícono de lápiz): Abre el modal de configuración de evento - **Botón Delete** (ícono de papelera): Elimina evento - **Mango de arrastre:** Reordene eventos arrastrando

2. Comprendiendo Scene Scheduler

2.1 Conceptos Fundamentales

Antes de profundizar en características avanzadas, entendamos cómo Scene Scheduler piensa sobre **automatización basada en tiempo**:

Eventos (Programas Programados)

Un **evento** es una **instrucción programada por tiempo** que le dice a OBS que: 1. **En un momento específico** (ej., 14:30:00): Cambie a una escena específica de OBS 2. **Opcionalmente**: Agregue/configure fuentes específicas de la escena (archivos multimedia, transmisiones, fuentes de navegador) 3. **Por una duración** (ej., 30 minutos): Mantenga esa configuración activa 4. **Luego limpíe**: Elimine las fuentes agregadas cuando termine el evento

Piense en los eventos como los "shows" o "segmentos" individuales en su horario de transmisión. Los eventos son los bloques fundamentales de construcción de su grilla de programación.

Escenas

Una **escena** en OBS es una colección de fuentes (video, audio, imágenes, etc.) organizadas en un diseño específico. Scene Scheduler no crea escenas—usa sus escenas existentes de OBS y las mejora mediante:

- Agregar/eliminar dinámicamente fuentes basándose en el horario
- Preparar fuentes en segundo plano antes de que sean visibles
- Limpiar después de que termine un evento

Fuentes

Una **fuente** es cualquier elemento de contenido en OBS:

- Archivos multimedia (videos, audio)
- Fuentes de navegador (páginas web, superposiciones HTML)
- Entradas de streaming (RTMP, RTSP, RTP, SRT)
- Imágenes
- Listas de reproducción VLC

Scene Scheduler puede configurar estas fuentes automáticamente por evento.

La Escena Auxiliar (`scheduleSceneAux`)

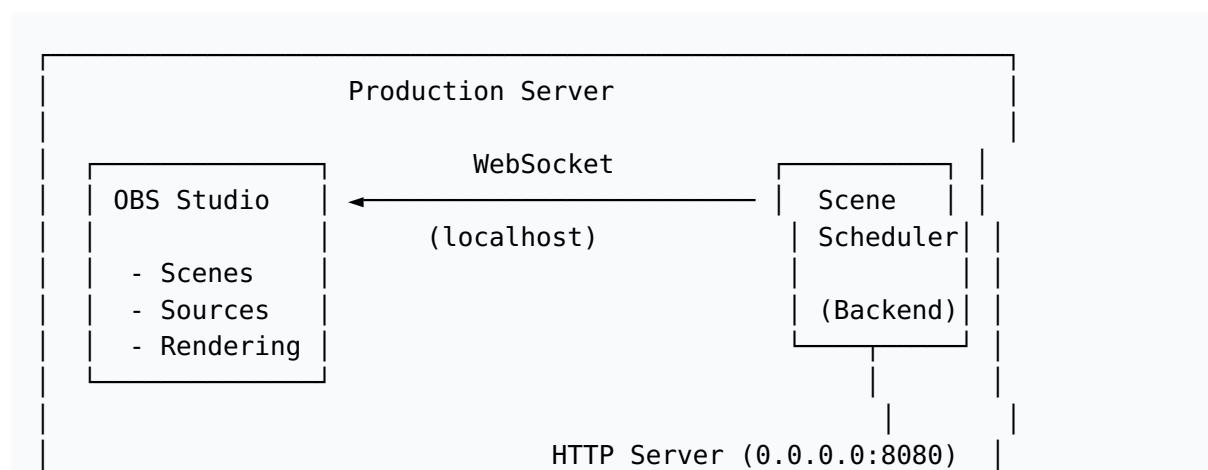
Esta es el "área detrás de escena" de Scene Scheduler—una escena oculta donde:

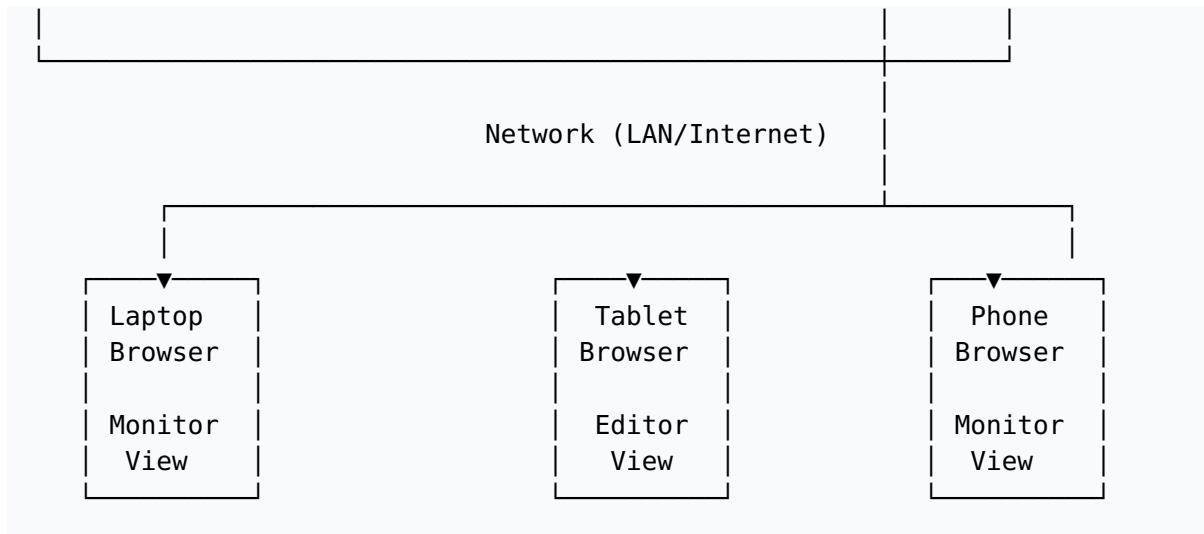
- Las fuentes se cargan y preparan antes de que sean necesarias
- Las entradas de streaming se prueban para conectividad
- Los archivos multimedia se precargan para evitar retrasos visibles

Nunca ve esta escena durante la transmisión, pero es crítica para una operación suave.

2.2 Descripción General de la Arquitectura

Scene Scheduler usa una **arquitectura cliente-servidor distribuida** que permite acceso y operación remota:





Flujo de comunicación: 1. **Backend ↔ OBS:** Conexión WebSocket para control de escena/fuente (localhost) 2. **Backend → Internet:** Servidor HTTP se vincula a 0.0.0.0 (accesible desde la red) 3. **Clientes Remotos → Backend:** HTTP/WebSocket desde cualquier dispositivo en la red 4. **Backend → Todos los Clientes:** Transmisiones en tiempo real de actualizaciones de horario

Beneficios arquitectónicos clave: - **Acceso distribuido:** Control desde cualquier lugar en la red (o internet si se expone) - **Carga reducida del servidor:** La IU web se ejecuta en dispositivos cliente, no en la máquina OBS - **Monitoreo multiusuario:** Múltiples personas pueden ver Monitor View simultáneamente - **Despliegue flexible:** El servidor no necesita pantalla, teclado o GUI - **Escalabilidad:** Agregue tantos clientes de monitoreo como necesite sin afectar el rendimiento

2.3 Cómo Funcionan las Transiciones de Escena

Scene Scheduler usa un **sistema de preparación sofisticado** para asegurar transiciones suaves sin artefactos visuales. El sistema opera cuando llega el momento de un evento programado.

El Proceso de Preparación (5 Pasos):

Paso 1: STAGING (Preparación en Segundo Plano)

- Nueva fuente creada en `scheduleSceneAux` (escena auxiliar/temporal)
- Fuente completamente configurada pero permanece invisible para los espectadores
- Todas las transformaciones aplicadas (posición, escala, recorte, etc.)
- Los archivos multimedia y transmisiones comienzan a cargarse
- Si este paso falla: El proceso se detiene, la transmisión actual no se afecta

Paso 2: ACTIVATION (Mover a Escena Visible)

- Fuente movida desde `scheduleSceneAux` a la escena objetivo de OBS
- Fuente hecha visible para la audiencia

- La transición ocurre instantáneamente (fuente ya preparada)
- Si este paso falla: Retroceso al contenido anterior

Paso 3: SCENE SWITCH (Cambio de Escena OBS)

- OBS hace la transición a la escena objetivo
- La audiencia ve el nuevo contenido inmediatamente
- Sin retrasos de almacenamiento en búfer o carga (gracias a la preparación)
- Si este paso falla: Reversión, contenido anterior mantenido

Paso 4: CLEANUP (Eliminar Elementos Temporales)

- Elemento temporal eliminado de scheduleSceneAux
- Recursos liberados para el siguiente evento
- Escena auxiliar lista para la siguiente operación de preparación

Paso 5: MONITOR (Gestión Continua)

- La escena permanece activa durante la duración programada
- Cuando termina el evento: Fuente eliminada automáticamente
- Sistema listo para el siguiente evento programado

Beneficios Clave:

- **Sin carga visible:** Las fuentes se preparan antes de ser mostradas
- **Transiciones atómicas:** O éxito completo o retroceso seguro
- **Eficiencia de recursos:** La limpieza previene fugas de memoria
- **Operación continua:** El sistema maneja programación automatizada 24/7

2.4 Modelo de Ejecución de Horario

Scene Scheduler usa un **sistema de activación basado en tiempo**:

- 1. Carga de Horario:** Al inicio, `schedule.json` se carga y analiza
- 2. Cola de Eventos:** Los eventos se ordenan por tiempo y se monitorean continuamente
- 3. Detección de Activación:** Cada segundo, el programador verifica si ha llegado el momento de algún evento
- 4. Ejecución:** Cuando llega el momento del evento, comienza el proceso de preparación de 5 pasos (ver Sección 2.3)
- 5. Limpieza:** Despues de que expira la duración del evento, los recursos se limpian

Importante: Los eventos son **activados por tiempo**, no secuenciales. Si se pierde el momento de un evento (ej., Scene Scheduler estaba detenido), no se ejecutará cuando se reinicie—solo se ejecutan los eventos próximos.

2.5 Sincronización en Tiempo Real

Todos los clientes conectados (vistas de Monitor y Editor) reciben **actualizaciones instantáneas** vía WebSocket:

- **Cambios de horario:** Agregar/editar/eliminar eventos actualiza todos los clientes inmediatamente
- **Seguimiento de evento actual:** Todas las vistas resaltan el evento activo
- **Cambios de estado de OBS:** Si cambia escenas manualmente en OBS, los clientes son notificados
- **Estado de conexión:** Indicadores visuales muestran el estado de conexión de OBS

Esto permite **operación colaborativa**: múltiples miembros del equipo pueden monitorear el mismo horario desde diferentes dispositivos.

3. Descripción General de la Interfaz Web

3.1 Modos de Interfaz

Scene Scheduler proporciona dos interfaces web distintas optimizadas para diferentes casos de uso:

Monitor View (/)

Propósito: Observación pasiva y monitoreo de estado

Casos de uso: - Pantallas montadas en la pared en salas de control de transmisión - Monitores secundarios para operadores - Tableros de estado de cara al público - Dispositivos móviles para verificaciones rápidas de estado

Características: - Tipografía grande y legible - Evento actual mostrado prominentemente - Temporizador de cuenta regresiva al siguiente evento - Sin controles de edición (previene cambios accidentales) - Actualización automática cada segundo

URLs de acceso: - Misma máquina: <http://localhost:8080/> - Acceso de red: <http://<server-ip>:8080/>

Editor View (/editor.html)

Propósito: Editor visual para `schedule.json`

Qué hace: - Editar el archivo schedule.json a través de una interfaz web - Agregar, modificar o eliminar eventos - Configurar fuentes de eventos (multimedia, navegador, transmisiones) - Guardar cambios de vuelta a schedule.json

Características: - Lista de eventos con botones de agregar/editar/eliminar - Diálogo modal para configuración de eventos - Vista previa opcional de fuente (herramienta de prueba) - Selectores visuales de tiempo/duración

URLs de acceso: - Misma máquina: `http://localhost:8080/editor.html` - Acceso de red: `http://<server-ip>:8080/editor.html`

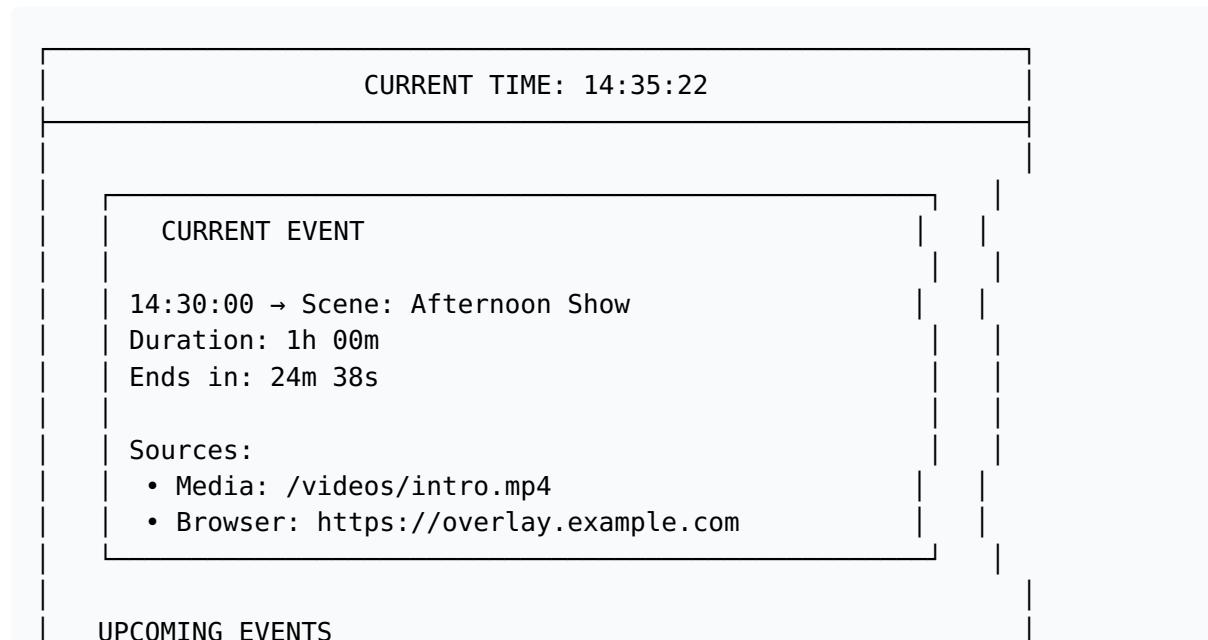
3.2 Navegación

Ambas vistas incluyen una **barra de navegación superior** con:



- **Scene Scheduler:** Título/logo de la aplicación
- **[Monitor]:** Botón para cambiar a Monitor view
- **[Editor]:** Botón para cambiar a Editor view
- **Indicador de conexión:**
 - Punto verde: Conectado a OBS y backend
 - Punto rojo: Desconectado (verifique el estado de OBS y backend)

3.3 Diseño de Monitor View



15:30:00 → Scene: News Segment
Starts in: 54m 38s

16:00:00 → Scene: Weather Report
Starts in: 1h 24m 38s

Codificación por colores: - **Fondo verde:** Evento actualmente activo - **Fondo amarillo:** Siguiente evento (comienza dentro de 30 minutos) - **Fondo blanco:** Eventos futuros

Visualizaciones de tiempo: - **Tiempo absoluto:** Hora de inicio del evento (formato HH:MM:SS) - **Tiempo relativo:** Cuenta regresiva o tiempo restante - **Duración:** Cuánto tiempo dura el evento

3.4 Diseño de Editor View

The screenshot shows the 'Editor View' interface. At the top, there is a header with '[+ Add Event]' on the left and 'Current: 14:35' on the right. Below the header, three events are listed vertically, each enclosed in a dashed border:

- 14:30:00 → Afternoon Show** [Edit] [Delete]
Duration: 1h 00m | Ends: 15:30:00
Sources: 2 configured
- 15:30:00 → News Segment** [Edit] [Delete]
Duration: 30m | Ends: 16:00:00
Sources: 1 configured
- 16:00:00 → Weather Report** [Edit] [Delete]
Duration: 15m | Ends: 16:15:00
Sources: 0 configured

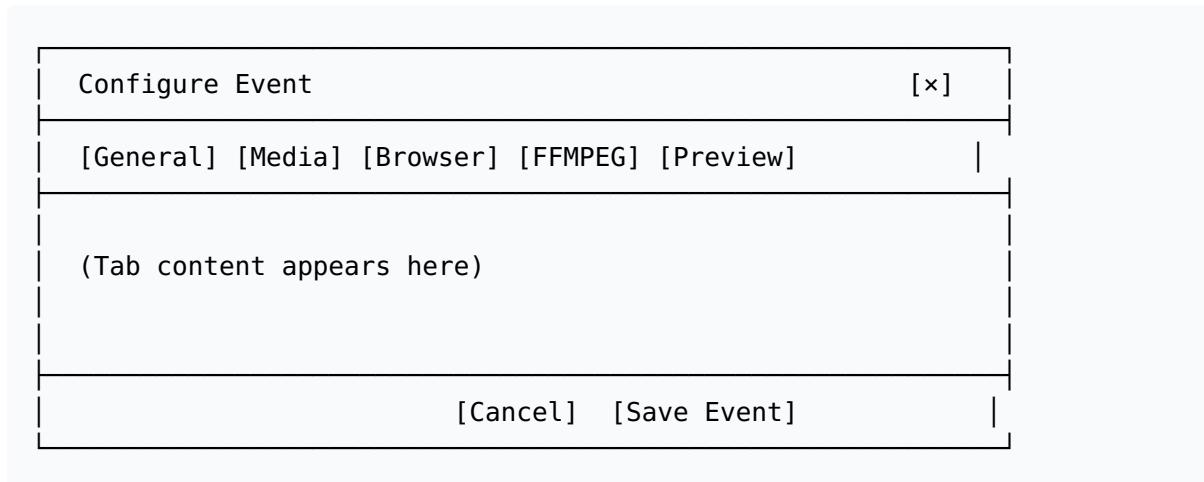
Elementos de la interfaz: - **Botón + Add Event:** Crea nueva entrada de horario (abre modal) - **Tarjetas de evento:** Cada evento mostrado como una tarjeta con

información resumida - **Botón Edit** (ícono de lápiz): Abre modal de configuración para ese evento - **Botón Delete** (ícono de papelera): Elimina evento (con confirmación) - **Mango de arrastre** (ícono :): Reordene eventos arrastrando

Información de la tarjeta de evento: - **Hora y nombre de escena:** Identificación primaria - **Duración y hora de finalización:** Calculado automáticamente - **Conteo de fuentes:** Número de fuentes configuradas - **Indicador de evento actual:** Fondo verde para evento activo

3.5 Modal de Configuración de Evento

Cuando hace clic en "Add Event" o "Edit" en un evento existente, se abre un diálogo modal con cinco pestañas:



Cinco pestañas: 1. **General:** Tiempo, selección de escena, duración 2. **Media:** Fuentes de archivos de video/audio 3. **Browser:** Fuentes de páginas web y superposiciones HTML 4. **FFMPEG:** Entradas de streaming de red (RTMP, RTSP, RTP, SRT, NDI) 5. **Preview:** Vista previa de fuente en tiempo real (probar antes de guardar)

Exploraremos cada pestaña en detalle en la Sección 5.

4. Gestión de Horarios

4.1 Comprendiendo Su Horario

Su horario se almacena en `schedule.json` como una lista ordenada por tiempo de eventos. Cada evento define: - **Cuándo:** El tiempo exacto de ejecución (formato HH:MM:SS) - **Qué:** Qué escena de OBS activar - **Cuánto tiempo:** Duración que la escena permanece activa - **Contenido:** Fuentes opcionales para agregar a la escena

Principios clave: 1. **Los eventos son activados por tiempo:** Se ejecutan en su tiempo programado sin importar los eventos anteriores 2. **Los eventos pueden superponerse:** Múltiples eventos pueden configurarse para el mismo tiempo (aunque esto puede causar conflictos) 3. **Los eventos no se repiten:** Cada evento se ejecuta una vez por día a menos que se repita explícitamente 4. **Los cambios son inmediatos:** Editar el horario actualiza OBS en tiempo real

4.2 Creando Su Primer Evento

Pasemos por la creación de un evento completo paso a paso:

Paso 1: Abrir Vista de Editor - Navegue a Editor View: - Misma máquina:<http://localhost:8080/editor.html> - Acceso de red: <http://<server-ip>:8080/editor.html> - Haga clic en el botón "+ Add Event" en la barra superior - Se abre el modal de configuración de evento

Paso 2: Configurar Ajustes Generales (Pestaña General)

La pestaña General contiene los parámetros esenciales del evento:

1. **Time** (requerido) - Formato: HH:MM:SS (reloj de 24 horas) - Ejemplo: 14:30:00 para 2:30 PM - Debe ser un tiempo válido (00:00:00 a 23:59:59) - **Consejo:** Establezca tiempos unos minutos en el futuro para pruebas
2. **OBS Scene** (requerido) - El menú desplegable muestra todas las escenas actualmente configuradas en OBS - Seleccione la escena que desea activar - **Importante:** La escena debe existir en OBS antes de programar - Si el menú desplegable está vacío, verifique su conexión a OBS
3. **Duration** (requerido) - Formato: HH:MM:SS (horas:minutos:segundos) - Predeterminado: 00:05:00 (5 minutos) - Ejemplos:
 - 00:30:00 = 30 minutos
 - 01:00:00 = 1 hora
 - 00:00:30 = 30 segundos
 - **Consejo:** La duración determina cuándo se limpian las fuentes
4. **Event Name** (opcional) - Una etiqueta descriptiva para su evento - Se muestra en las vistas de monitor y editor - Ejemplo: "Morning News", "Afternoon Show", "Commercial Break" - Si está vacío, se usa el nombre de la escena

Configuración de ejemplo:

Time: 14:30:00
Scene: Afternoon Show
Duration: 01:00:00
Name: Daily Afternoon Broadcast

Paso 3: Agregar Fuentes (Opcional)

Si su evento necesita contenido específico (videos, superposiciones, transmisiones), configúrelas en las pestañas de fuentes: - **Pestaña Media:** Agregar archivos de video o audio - **Pestaña Browser:** Agregar páginas web o superposiciones HTML - **Pestaña FFMPEG:** Agregar entradas de streaming de red

Cubriremos la configuración de fuentes en detalle en la Sección 5.

Paso 4: Previsualizar Fuentes (Opcional)

Antes de guardar, puede probar sus fuentes usando la **Pestaña Preview:** - Genera una transmisión HLS en tiempo real de cada fuente - Se reproduce en el navegador para verificación - Se detiene automáticamente después de 30 segundos - Consulte la Sección 5.6 para documentación completa de la vista previa

Paso 5: Guardar el Evento

Haga clic en "**Save Event**" en la parte inferior del modal. El sistema: 1. Valida todos los campos 2. Agrega el evento a `schedule.json` 3. Transmite la actualización a todos los clientes conectados 4. Cierra el modal 5. Muestra el nuevo evento en la lista del editor

Paso 6: Verificar

Después de guardar: - El evento aparece en la lista de eventos de Editor View - Monitor View lo muestra en eventos próximos - OBS cambiará automáticamente a la escena en el tiempo programado

4.3 Editando Eventos Existentes

Para modificar un evento:

1. Localice el evento en Editor View
2. Haga clic en el botón **Edit** (ícono de lápiz) en la tarjeta del evento
3. Realice cambios en cualquiera de las cinco pestañas
4. Guarde para aplicar cambios inmediatamente

Notas importantes: - Los cambios a eventos pasados no tienen efecto (los eventos son activados por tiempo) - Editar un evento activo actualiza OBS inmediatamente - Los cambios de fuentes toman efecto en el siguiente activador de evento

Ediciones comunes: - **Ajustar tiempo:** Cambie el campo de tiempo para reprogramar - **Cambiar duración:** Extender o acortar el evento - **Cambiar escenas:** Seleccione una escena diferente de OBS - **Actualizar fuentes:** Agregar, eliminar o modificar configuraciones de fuentes - **Corregir errores:** Corregir rutas de archivo o URLs inválidas

4.4 Eliminando Eventos

Para eliminar un evento:

1. **Haga clic en el botón Delete** (ícono de papelera) en la tarjeta del evento
2. **Confirme la eliminación** en el diálogo (si está habilitado)
3. El evento se elimina inmediatamente del horario

Qué sucede: - El evento desaparece de todas las vistas (Monitor y Editor) - schedule.json se actualiza - Si el evento está actualmente activo: - Las fuentes se limpian inmediatamente - OBS permanece en la escena actual (sin cambio automático) - El siguiente evento programado se activará normalmente

Consejo: Para deshabilitar temporalmente un evento sin eliminarlo, puede: - Cambiar su tiempo a algo muy futuro (ej., 23:59:59) - O eliminar y volver a agregar más tarde usando la función deshacer de su navegador

4.5 Reordenando Eventos

Los eventos en Editor View pueden reordenarse para organización visual:

1. **Pase el cursor sobre el mango de arrastre** (ícono : :) en una tarjeta de evento
2. **Haga clic y arrastre** el evento a una nueva posición
3. **Suelte** para dejarlo en su lugar

Importante: El reordenamiento en la interfaz es puramente visual—los eventos aún se ejecutan según su campo **time**, no su posición en la lista. El reordenamiento es útil para: - Agrupar eventos relacionados juntos - Separar diferentes "shows" o bloques de tiempo - Coincidir con una hoja de programación física

4.6 Validación de Horario

Scene Scheduler realiza validación cuando guarda eventos:

- Validación de campos:** - **Formato de tiempo:** Debe ser HH:MM:SS (ej., 14:30:00)
- **La escena existe:** La escena seleccionada debe estar presente en OBS
- Formato de duración:** Debe ser HH:MM:SS y mayor que cero
- **Rutas de fuentes:** Las rutas de archivo deben existir (verificado en tiempo de preparación)
- **URLs:** Las URLs de Browser y FFMPEG deben tener formato válido

Errores de validación comunes:

Mensaje de Error	Causa	Solución
"Invalid time format"	Tiempo no en HH:MM:SS	Use formato de 24 horas: 14:30:00
"Scene not found"	Escena eliminada de OBS	Cree la escena en OBS primero
"Invalid duration"	La duración es cero o negativa	Establezca duración positiva
"Invalid URL"	URL mal formada en browser/ FFMPEG	Verifique sintaxis de URL
"File not found"	La ruta de multimedia no existe	Verifique la ruta del archivo en disco

Cuándo ocurre la validación: - **Del lado del cliente:** Los campos del formulario se validan mientras escribe - **Del lado del servidor:** Validación completa al guardar evento - **Tiempo de activación del evento:** Se verifica existencia de archivo y conectividad cuando el evento se ejecuta

4.7 Persistencia del Horario

Su horario se almacena en `schedule.json` en el directorio de la aplicación:

Comportamiento de autoguardado: - Cada cambio (agregar/editar/eliminar) escribe inmediatamente al disco - No se requiere "guardar" manual - Los cambios persisten a través de reinicios de la aplicación

Recomendaciones de respaldo: 1. **Respaldos manuales:** Copie `schedule.json` periódicamente 2. **Control de versiones:** Almacene en git para seguimiento de cambios 3. **Respaldos automatizados:** Use herramientas de respaldo del sistema para incluir el directorio de la aplicación

Restaurando desde respaldo:

```
bash
# Detener Scene Scheduler
pkill scenescheduler

# Restaurar respaldo
cp schedule.json.backup schedule.json

# Reiniciar Scene Scheduler
./scenescheduler
```

4.8 Horarios Multi-Día

Scene Scheduler opera en un **reloj de 24 horas** (00:00:00 a 23:59:59). Para operaciones multi-día:

Enfoque 1: Repetición diaria (manual) - Configure eventos para un solo día - Copie y ajuste tiempos para días subsiguientes - Use nombres descriptivos para rastrear días (ej., "Monday Morning Show")

Enfoque 2: Horario continuo 24/7 - Cree eventos que abarquen el día completo - Use duraciones largas para períodos nocturnos - Ejemplo: Un evento a las 23:00:00 con duración de 8 horas cubre la noche

Enfoque 3: Patrones semanales - Cree una plantilla para cada día de la semana - Cambie horarios manualmente diariamente (reemplace `schedule.json`) - Considere scripting para automatización

Nota: Scene Scheduler v1.6 no admite repetición diaria automática o condiciones de día de la semana. Cada evento se ejecuta una vez por día en su tiempo programado.

4.9 Manejando Conflictos de Horario

¿Qué es un conflicto de horario? Dos o más eventos programados para exactamente el mismo tiempo.

Cómo Scene Scheduler maneja conflictos: - Los eventos se procesan en el orden en que aparecen en `schedule.json` - Los eventos posteriores **sobrescriben** a los eventos anteriores - Solo las fuentes del último evento son visibles

Ejemplo de conflicto:

```
json
[
  {
```

```
        "time": "14:00:00",
        "scene": "Scene A",
        "duration": "01:00:00"
    },
    {
        "time": "14:00:00",
        "scene": "Scene B",
        "duration": "00:30:00"
    }
]
```

Resultado: A las 14:00:00, ambos eventos se activan, pero Scene B se activa (es el último). Las fuentes de Scene A pueden prepararse pero nunca mostrarse.

Mejores prácticas para evitar conflictos: 1. **Escalonar tiempos:** Use desplazamientos de minutos o segundos (14:00:00, 14:01:00) 2. **Revisar horario visualmente:** Editor View muestra todos los eventos cronológicamente 3. **Use tiempos únicos:** Evite duplicar tiempos a menos que sea intencional 4. **Planear transiciones:** Permita tiempo de amortiguación entre eventos (ej., 30 segundos)

4.10 Probando Su Horario

Antes de confiar en su horario para producción:

Lista de verificación de pruebas:

- Crear eventos de prueba** - Establezca tiempos 2-3 minutos en el futuro - Use duraciones cortas (1-2 minutos) - Pruebe primero con escenas simples (sin fuentes)
- Verificar transiciones de escena** - Observe OBS en el tiempo programado - Confirme que la escena cambia automáticamente - Verifique que Monitor View resalta el evento correcto
- Probar carga de fuentes** - Agregue una fuente multimedia a un evento - Verifique que aparece en OBS en el tiempo programado - Confirme que la reproducción comienza automáticamente
- Probar edición de eventos** - Edite un evento próximo - Cambie el tiempo ligeramente - Verifique que OBS responde al nuevo tiempo
- Probar limpieza** - Espere a que expire la duración del evento - Confirme que las fuentes se eliminan de OBS - Verifique que scheduleSceneAux se limpia

6. Probar múltiples eventos - Programe 3-4 eventos en secuencia - Verifique que cada uno hace transición suavemente - Observe problemas de superposición

Solución de problemas de fallas de prueba: - Consulte la Sección 10 (Resolución de Problemas) para pasos de diagnóstico detallados

5. Configuración de Eventos

Esta sección cubre las cinco pestañas de configuración en el diálogo modal de eventos. Cada pestaña gestiona un aspecto diferente de las fuentes y comportamiento de su evento.

5.1 Pestaña General

La pestaña General contiene los parámetros principales del evento:

Campo Time

Formato: HH:MM:SS (reloj de 24 horas)

Ejemplos: - 00:00:00 - Medianoche - 09:30:00 - 9:30 AM - 14:45:30 - 2:45 PM y 30 segundos - 23:59:59 - Un segundo antes de medianoche

Validación: - Horas: 00-23 - Minutos: 00-59 - Segundos: 00-59 - Ceros iniciales requeridos (use 09:00:00, no 9:0:0)

Consejos: - Para pruebas, establezca tiempos 2-5 minutos en el futuro - Use segundos para tiempo preciso (ej., sincronización de pausas comerciales) - Recuerde: los eventos se activan una vez por día en este tiempo

Campo OBS Scene

Propósito: Seleccione a qué escena cambia OBS cuando este evento se activa.

Cómo funciona: 1. El menú desplegable se llena desde la lista de escenas actual de OBS Studio 2. La lista se actualiza automáticamente cuando agrega/elimina escenas en OBS 3. La escena seleccionada debe existir cuando el evento se activa (o el evento falla)

Solución de problemas: - **Menú desplegable vacío:** Se perdió la conexión WebSocket de OBS (verifique el indicador de conexión) - **Escena faltante:** La escena

fue eliminada en OBS (recree o seleccione una escena diferente) - **Escena atenuada:** scheduleSceneAux no puede seleccionarse (reservada para preparación)

Campo Duration

Formato: HH:MM:SS (horas:minutos:segundos)

Qué controla: - Cuánto tiempo permanecen activas las fuentes - Cuándo ocurre la limpieza - "Hora de finalización" implícita del evento (hora de inicio + duración)

Ejemplos: - 00:00:30 - Evento corto de 30 segundos (bumpers, stingers) - 00:05:00 - Evento de 5 minutos (segmentos de noticias, comerciales) - 00:30:00 - Evento de 30 minutos (shows, programas) - 01:00:00 - Evento de 1 hora (contenido de formato largo) - 02:30:00 - Evento de 2.5 horas (películas, programación extendida)

Notas importantes: - La duración determina cuándo **se limpian las fuentes**, no cuándo cambia la escena - OBS permanece en la escena hasta que el siguiente evento se activa - Duración mínima: 1 segundo (00:00:01) - Duración máxima: 23:59:59 (justo bajo 24 horas)

Campo Event Name (Opcional)

Propósito: Una etiqueta legible por humanos para el evento.

Uso: - Se muestra en Monitor View y Editor View - Ayuda a identificar eventos en listas - No afecta a OBS (puramente para organización)

Mejores prácticas: - Use nombres descriptivos: "Morning News", "Afternoon Show", "Commercial Block 1" - Incluya el día si programa multi-día: "Monday Morning Show" - Manténgalo conciso (se muestra mejor en la interfaz)

Comportamiento predeterminado: Si se deja vacío, se usa el nombre de la escena como el nombre del evento.

5.2 Pestaña Media

La pestaña Media configura tipos **media_source** y **vlc_source**—archivos de video y audio que se reproducen durante el evento.

Cuándo Usar Fuentes de Multimedia

- Reproducir archivos de video pregrabados
- Música de fondo o pistas de audio

- Búcles de video (con opción de bucle habilitada)
- Videos de intro/outro para shows

Agregando una Fuente de Multimedia

Paso 1: Tipo de Fuente Seleccione el tipo de fuente: - **media_source**: Fuente de multimedia nativa de OBS (recomendada) - Soporta: MP4, MOV, AVI, MKV, FLV - Soporte de decodificación por hardware - Mejor rendimiento para la mayoría de archivos

- **vlc_source**: Fuente de multimedia basada en VLC
- Soporta: Todos los formatos compatibles con VLC
- Soporte de lista de reproducción (múltiples archivos)
- Mejor compatibilidad de codec

Paso 2: Nombre de Fuente Ingrese un nombre único para esta fuente en OBS.

Reglas: - Debe ser único dentro del evento - Solo alfanumérico, espacios, guiones, guiones bajos - Ejemplo: IntroVideo, Background Music, Main Content

Consejo: Use nombres descriptivos que indiquen el contenido (ej., MondayIntro, CommercialBlock1)

Paso 3: Ruta de Archivo Ingrese la ruta absoluta al archivo multimedia en disco.

Formato: - **Linux**: /home/user/videos/intro.mp4 - **Windows**: C:\Videos\intro.mp4 o C:/Videos/intro.mp4 (ambos funcionan) - Debe ser legible por el proceso de Scene Scheduler - El archivo debe existir cuando el evento se activa (verificado durante la preparación)

Consejos: - Use el explorador de archivos si está disponible en su interfaz - Evite espacios en rutas de archivo (use guiones bajos: my_video.mp4) - Pruebe la vista previa antes de guardar evento

Paso 4: Configuraciones Adicionales

Loop (casilla de verificación): - Habilitado: El video se repite continuamente durante la duración del evento - Deshabilitado: El video se reproduce una vez y se detiene

Caso de uso para bucle: - Videos de fondo (ej., fondos animados) - Videos cortos que necesitan llenar duraciones largas - Pistas de música que se repiten

Caso de uso para no bucle: - Videos de intro únicos - Segmentos de noticias - Contenido específico del evento que no debe repetirse

Restart on activate (casilla de verificación): - Habilitado: El video se reinicia desde el principio cuando la escena se activa - Deshabilitado: El video continúa desde donde estaba

Hardware decoding (casilla de verificación, solo media_source): - Habilitado: Usa GPU para decodificación de video (mejor rendimiento) - Deshabilitado: Usa decodificación por CPU

Recomendado: Habilite para videos de alta resolución (1080p, 4K)

Paso 5: Vista Previa Antes de guardar, haga clic en la pestaña **Preview** para probar el archivo multimedia (consulte Sección 5.6).

Múltiples Fuentes de Multimedia

Puede agregar múltiples fuentes de multimedia a un solo evento:

Ejemplo de caso de uso:

```
Event: Morning Show
└─ BackgroundVideo (media_source, /videos/bg.mp4, loop=true)
└─ IntroMusic (media_source, /audio/intro.mp3, loop=false)
└─ Overlay (browser_source, https://overlay.com)
```

Todas las fuentes se agregan a la escena simultáneamente cuando el evento se activa.

Limitaciones de Fuentes de Multimedia

- **Tamaño de archivo:** Archivos grandes (>2GB) pueden tener tiempos de carga lentos
- **Soporte de codec:** Depende de los codecs de OBS y del sistema (H.264 recomendado)
- **Rutas de red:** Montajes SMB/NFS pueden causar retrasos (use copias locales)

5.3 Pestaña Browser

La pestaña Browser configura tipos **browser_source**—páginas web, superposiciones HTML y contenido web interactivo renderizado en OBS.

Cuándo Usar Fuentes de Navegador

- Superposiciones dinámicas (chat, alertas, temporizadores)
- Gráficos basados en web (HTML/CSS/JavaScript)
- Paneles de streaming
- Visualizaciones interactivas
- Contenido remoto (APIs, feeds de datos)

Agregando una Fuente de Navegador

Paso 1: Nombre de Fuente Ingrese un nombre único para la fuente de navegador.

Reglas: Igual que las fuentes de multimedia (alfanumérico, único dentro del evento)

Paso 2: URL Ingrese la URL completa a renderizar.

Protocolos soportados: - `https://` - Páginas web seguras (recomendado) - `http://` - Páginas web no seguras - `file:///` - Archivos HTML locales

Ejemplos: - `https://example.com/overlay.html` - Superposición alojada remotamente - **Linux:** `file:///home/user/overlays/timer.html` - **Windows:** `file:///C:/overlays/timer.html` - `http://localhost:3000` - Servidor de desarrollo local

Importante: - La URL debe ser accesible desde la máquina que ejecuta OBS - Se recomienda HTTPS por seguridad - Pruebe la URL en el navegador antes de agregar al evento

Paso 3: Dimensiones

Width y Height (píxeles): - Define el tamaño del viewport del navegador - Tamaños comunes: - 1920 x 1080 - Superposición Full HD - 1280 x 720 - Superposición HD - 400 x 300 - Widget pequeño - 800 x 100 - Ticker/banner

Por qué importan las dimensiones: - Afecta cómo se renderiza la página - Los diseños responsivos se adaptan a estas dimensiones - Tamaños más grandes consumen más recursos

Paso 4: CSS (Opcional) CSS personalizado para inyectar en la página.

Casos de uso: - Ocultar elementos específicos: `#ads { display: none; }` - Sobrescribir colores: `body { background: transparent; }` - Ajustar posicionamiento: `.widget { margin-top: 50px; }`

Ejemplo de CSS:

```
css
body {
    background-color: transparent;
    margin: 0;
    padding: 0;
}
#header {
    display: none;
}
```

Paso 5: Configuraciones Adicionales

Shutdown when not visible (casilla de verificación): - Habilitado: El navegador deja de renderizar cuando la fuente está oculta (ahorra CPU/GPU) - Deshabilitado: El navegador continúa renderizando (use para animaciones que necesitan ejecutarse continuamente)

Refresh when scene becomes active (casilla de verificación): - Habilitado: La página se recarga cada vez que la escena se activa (resetea el estado) - Deshabilitado: La página persiste a través de cambios de escena (mantiene el estado)

FPS (cuadros por segundo): - Predeterminado: 30 - Rango: 1-60 - FPS más alto = animaciones más suaves, más uso de CPU - Recomendado: 30 para la mayoría de superposiciones, 60 para animaciones suaves

Paso 6: Vista Previa Use la pestaña **Preview** para probar la fuente de navegador antes de guardar (consulte Sección 5.6).

Consejos de Rendimiento de Fuentes de Navegador

- **Optimizar páginas web:** Minimice JavaScript, comprima recursos
- **Use fondos transparentes:** Establezca `background: transparent` en CSS
- **Límite animaciones:** Las animaciones excesivas pueden causar caídas de cuadros
- **Shutdown when not visible:** Habilite para ahorrar recursos

Seguridad de Fuentes de Navegador

- **Solo HTTPS:** Evite HTTP para datos sensibles
- **Confíe en la fuente:** Solo use URLs que controle o en las que confíe
- **Archivos locales:** Use `file:///` para superposiciones HTML controladas

- **Sin entrada de usuario:** Las fuentes de navegador en OBS no manejan entrada de usuario

5.4 Pestaña FFMPEG

La pestaña FFMPEG configura tipos **ffmpeg_source**—entradas de streaming de red desde RTMP, RTSP, RTP, SRT, HLS y otros protocolos.

Cuándo Usar Fuentes FFMPEG

- Cámaras IP (transmisiones RTSP)
- Feeds de contribución remota (SRT, RTMP)
- Fuentes de video de red (NDI vía FFMPEG)
- Entradas de streaming en vivo (HLS, RTMP pull)
- Salidas de equipo de transmisión profesional

Agregando una Fuente FFMPEG

Paso 1: Nombre de Fuente Ingrese un nombre único para la fuente FFMPEG.

Ejemplos: Camera1, RemoteFeed, IP_Camera_Front, SRT_Input

Paso 2: URL de Entrada Ingrese la URL de streaming.

Protocolos soportados:

RTSP (Cámaras IP):

```
rtsp://192.168.1.100:554/stream  
rtsp://username:password@camera.local/live
```

RTMP (Servidores de streaming):

```
rtmp://server.example.com:1935/live/stream  
rtmp://192.168.1.50/live/feed
```

SRT (Secure Reliable Transport):

```
srt://192.168.1.200:9000?mode=caller  
srt://remote.server.com:9000?passphrase=secret
```

RTP (Real-time Protocol):

```
rtp://239.0.0.1:5004
```

HTTP/HLS:

```
https://stream.example.com/live/playlist.m3u8  
http://192.168.1.100/stream.m3u8
```

File (para pruebas):

```
file:///home/user/test.mp4
```

Paso 3: Input Format (Opcional) Especifique el formato de contenedor si FFMPEG no puede autodetectar.

Formatos comunes: - rtsp - Transmisiones RTSP - mpegs - MPEG Transport Stream - flv - Flash Video - mp4 - Contenedor MP4 - Dejar vacío para autodetección (recomendado)

Paso 4: Configuraciones Adicionales

Buffering (MB): - Predeterminado: 2 MB - Rango: 1-10 MB - Mayor almacenamiento en búfer = más latencia, reproducción más estable - Menor almacenamiento en búfer = menos latencia, posible entrecortado

Reconnect delay (segundos): - Cuánto tiempo esperar antes de reintentar la conexión después de desconectar - Predeterminado: 5 segundos - Útil para transmisiones de red inestables

Hardware decoding (casilla de verificación): - Habilitado: Usar decodificación por GPU (menor uso de CPU) - Deshabilitado: Usar decodificación por CPU - Recomendado para transmisiones de alto bitrate

Paso 5: Vista Previa Use la pestaña **Preview** para probar la conectividad antes de guardar (consulte Sección 5.6).

Importante para fuentes FFMPEG: La vista previa verifica que la transmisión sea alcanzable y se decodifique correctamente. Esto detecta problemas de conexión antes de que su evento esté en vivo.

Solución de Problemas de Fuentes FFMPEG

Problema	Causa	Solución
Connection timeout	Red inalcanzable	Verifique IP, firewall, enrutamiento
Authentication failed	Credenciales incorrectas	Verifique nombre de usuario/contraseña en URL
Protocol not supported	Codec faltante	Instale bibliotecas FFMPEG requeridas
Choppy playback	Ancho de banda de red	Aumente almacenamiento en búfer, verifique calidad de red
High latency	Búfer grande	Reduzca valor de almacenamiento en búfer

Mejores Prácticas de Fuentes FFMPEG

- 1. Pruebe antes de producción:** Use la pestaña Preview para verificar conectividad
- 2. Use IPs estáticas:** Evite DHCP para fuentes críticas
- 3. Monitoree ancho de banda:** Las transmisiones de alto bitrate necesitan capacidad de red adecuada
- 4. Habilite reconexión:** Las transmisiones de red pueden caerse; la reconexión automática es esencial
- 5. Asegure credenciales:** Use variables de entorno o archivos de configuración (no codifique contraseñas)

5.5 Patrones Comunes de Configuración de Fuentes

Aquí hay ejemplos del mundo real de combinación de tipos de fuentes:

Patrón 1: Show de Noticias

```
Event: Morning News (08:00:00, duration 01:00:00)
└─ Media: IntroVideo (/videos/news_intro.mp4, loop=false)
└─ FFMPEG: LiveFeed (rtsp://camera.local/stream)
└─ Browser: LowerThird (https://graphics.local/lowerthird.html)
└─ Browser: TickerBar (https://graphics.local/ticker.html)
```

Patrón 2: Lista de Reproducción Automatizada

```
Event: Music Videos (14:00:00, duration 02:00:00)
└─ VLC: Playlist (/playlists/afternoon.xspf)
└─ Browser: SongInfo (https://overlay.local/nowplaying.html)
```

Patrón 3: Relé de Transmisión en Vivo

```
Event: Remote Event (19:00:00, duration 03:00:00)
└─ FFMPEG: MainFeed (srt://remote.server:9000?mode=caller)
└─ FFMPEG: BackupFeed (rtmp://backup.server/live)
└─ Browser: EventInfo (https://overlay.local/event_details.html)
```

Patrón 4: Fondo en Buckle

```
Event: Holding Screen (23:00:00, duration 08:00:00)
└─ Media: BackgroundLoop (/videos/holding.mp4, loop=true)
└─ Browser: Clock (file:///overlays/clock.html)
```

5.6 Pestaña Preview - Herramienta de Prueba Opcional

La pestaña Preview es una **característica auxiliar opcional** que le permite probar configuraciones de fuentes antes de comprometerlas a su horario. Aunque el propósito principal de Scene Scheduler es la programación automatizada basada en tiempo, esta herramienta de vista previa ayuda a detectar errores de configuración durante la configuración.

Importante: La vista previa NO es requerida para la operación de Scene Scheduler. Es una característica de conveniencia para Editor View - el programador backend opera independientemente y no usa funcionalidad de vista previa.

5.6.1 ¿Por Qué Usar Vista Previa?

Beneficios de esta herramienta opcional: 1. **Verificar conectividad:** Probar transmisiones de red antes de programar 2. **Verificar rutas de archivo:** Asegurar que los archivos multimedia existen y son legibles 3. **Probar apariencia visual:** Ver cómo se renderizan las fuentes antes de ir en vivo 4. **Detectar errores temprano:** Identificar problemas durante la configuración, no durante la transmisión programada 5. **Ahorrar tiempo:** No hay necesidad de esperar al tiempo del evento para verificar la configuración

Problemas comunes detectados por la vista previa: - Rutas de archivo inválidas (errores tipográficos, archivos faltantes) - Transmisiones de red inalcanzables (firewall, IP incorrecta) - URLs mal formadas (errores de sintaxis) - Fuentes de navegador rotas (errores 404, problemas de CORS) - Problemas de codec (formatos no soportados)

5.6.2 Cómo Funciona la Vista Previa

El sistema de vista previa de Scene Scheduler usa **HLS (HTTP Live Streaming)** para generar una transmisión reproducible en navegador de su fuente:

Flujo técnico: 1. El usuario hace clic en el botón "► Preview Source" 2. El backend genera el proceso `hls-generator` con la configuración de fuente 3. `hls-generator` usa bibliotecas de OBS para: - Crear una escena temporal de OBS - Agregar la fuente a la escena - Codificar a H.264 - Segmentar en fragmentos HLS (archivos .ts) - Generar manifiesto de lista de reproducción (.m3u8) 4. El frontend sondea la disponibilidad de la lista de reproducción (máximo 30 segundos) 5. Una vez listo, el reproductor HLS.js carga y reproduce la transmisión 6. La vista previa se detiene automáticamente después de 30 segundos (limpieza de recursos)

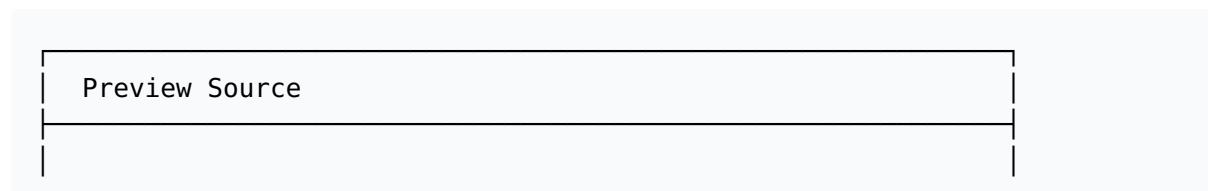
¿Por qué HLS? - **Nativo del navegador:** Funciona en todos los navegadores modernos sin plugins - **Adaptativo:** Maneja condiciones de red variables - **Estándar:** Protocolo de streaming estándar de la industria - **Eficiente:** Baja latencia, pequeña sobrecarga

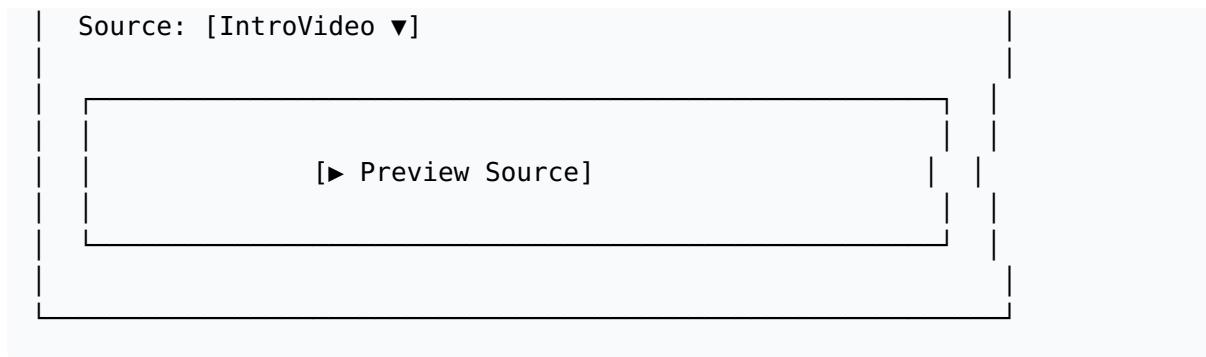
5.6.3 Usando la Pestaña Preview

Paso 1: Configure su fuente Antes de previsualizar, complete la configuración de la fuente en la pestaña apropiada: - Pestaña Media: Establezca nombre de fuente y ruta de archivo - Pestaña Browser: Establezca nombre de fuente, URL y dimensiones - Pestaña FFMPEG: Establezca nombre de fuente y URL de entrada

Paso 2: Cambie a la Pestaña Preview Haga clic en la pestaña "**Preview**" en el modal de evento.

Paso 3: Seleccione Fuente La pestaña Preview muestra un menú desplegable con todas las fuentes configuradas para este evento:





Seleccione la fuente que desea previsualizar del menú desplegable.

Paso 4: Iniciar Vista Previa Haga clic en el botón "**▶ Preview Source**".

Qué sucede: 1. El texto del botón cambia para mostrar el estado: - " Starting preview..." (solicitando del backend) - " Waiting for stream..." (esperando lista de reproducción HLS) - Aparece el reproductor de video y comienza la reproducción 2. El video se reproduce en el modal 3. La vista previa se detiene automáticamente después de 30 segundos

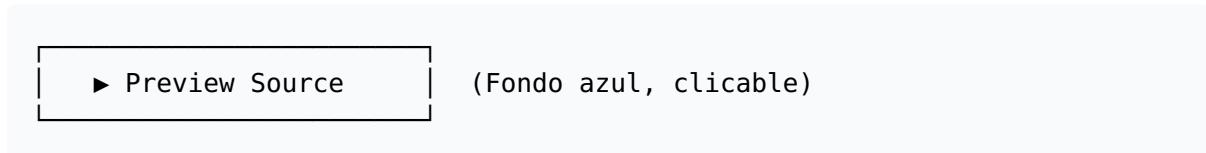
Paso 5: Observar Mire el video para verificar: - Contenido correcto (archivo/transmisión correcta) - Calidad visual (resolución, bitrate) - Audio (si aplica) - Sin errores o artefactos

Paso 6: Detener Vista Previa (Opcional) Puede detener manualmente la vista previa antes del timeout de 30 segundos: - Haga clic en el botón "**■ Stop Preview**" - O cierre el modal (la limpieza ocurre automáticamente)

5.6.4 Estados y Mensajes del Botón de Vista Previa

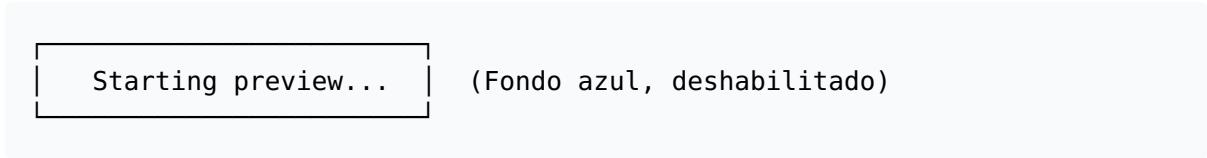
El botón de vista previa cambia apariencia y texto para indicar el estado:

Estado 1: Idle (Predeterminado)



Significado: Listo para iniciar vista previa. Haga clic para comenzar.

Estado 2: Starting



Starting preview...

(Fondo azul, deshabilitado)

Significado: Solicitud enviada al backend, esperando respuesta. **Duración:** 1-2 segundos típicamente.

Estado 3: Waiting for Stream

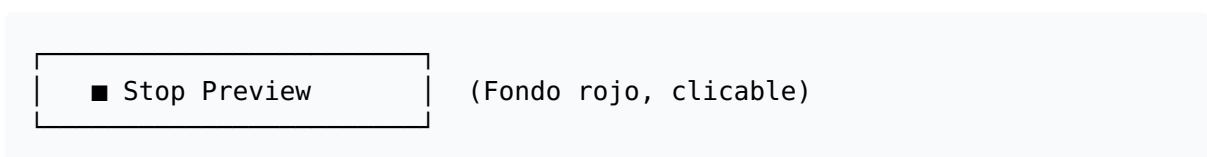


Waiting for stream...

(Fondo azul, deshabilitado)

Significado: El backend está generando la transmisión HLS, el frontend está sondeando la lista de reproducción. **Duración:** 5-30 segundos dependiendo del tipo de fuente. **Timeout:** 30 segundos máximo. Si la transmisión no comienza, vea solución de problemas abajo.

Estado 4: Playing



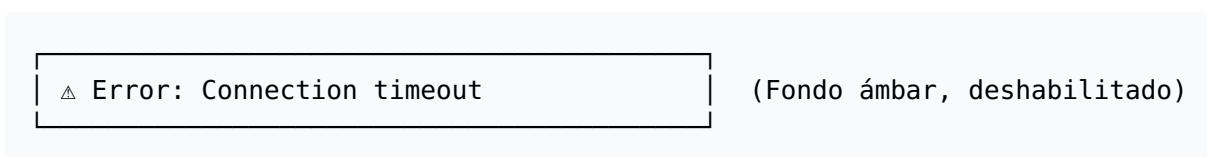
■ Stop Preview

(Fondo rojo, clicable)

Significado: La transmisión se está reproduciendo. Video visible debajo del botón.

Acción: Haga clic para detener temprano (o espere la parada automática de 30 segundos).

Estado 5: Error



⚠ Error: Connection timeout

(Fondo ámbar, deshabilitado)

Significado: La vista previa falló. El mensaje de error explica por qué. **Duración:** Mensaje mostrado por 5 segundos, luego se resetea al estado idle. **Acción:** Corrija el error (vea mensajes de error abajo) y reintente.

Estado 6: Stopped (Auto o Manual)

i Preview automatically stopped after 30 seconds

(Azul cielo, deshabilitado)

Significado: La vista previa se completó normalmente (se alcanzó el timeout de 30 segundos). **Duración:** Mensaje mostrado por 5 segundos, luego se resetea al estado idle. **Acción:** Ninguna requerida. Haga clic nuevamente para re-previsualizar.

5.6.5 Mensajes de Error de Vista Previa

Cuando la vista previa falla, el botón muestra un mensaje de error con detalles específicos:

Error: "Connection timeout"

Mensaje completo: `△ Error: Connection timeout`

Causa: El backend no pudo alcanzar la fuente dentro de 30 segundos.

Razones comunes: - Fuente FFMPEG: La transmisión de red es inalcanzable (IP incorrecta, firewall bloqueando) - Fuente Browser: La URL no responde (servidor caído, falla de DNS) - Fuente Media: El sistema de archivos es lento (retraso de montaje de red)

Soluciones: 1. **Transmisiones de red:** Verifique dirección IP, puerto y reglas de firewall 2. **Fuentes de navegador:** Pruebe la URL en un navegador regular 3. **Archivos multimedia:** Verifique ruta de archivo y permisos 4. **Problemas de red:** Verifique conectividad con ping o curl

Error: "File not found"

Mensaje completo: `△ Error: File not found: /path/to/file.mp4`

Causa: El archivo de fuente multimedia no existe en la ruta especificada.

Soluciones: 1. Verifique errores tipográficos en la ruta del archivo 2. Verifique que el archivo existe: `ls -la /path/to/file.mp4` 3. Verifique permisos: El archivo debe ser legible por el proceso de Scene Scheduler 4. Use rutas absolutas (no relativas)

Error: "Invalid URL"

Mensaje completo: `△ Error: Invalid URL format`

Causa: La URL de fuente Browser o FFmpeg está mal formada.

Problemas comunes: - Protocolo faltante: Use `https://example.com`, no `example.com` - Caracteres inválidos: Se necesita codificación de URL para caracteres especiales - Protocolo incorrecto: Use `rtsp://` para RTSP, no `http://`

Soluciones: 1. Verifique sintaxis de URL 2. Pruebe URL en navegador o reproductor multimedia 3. Codifique caracteres especiales en URL 4. Verifique que el protocolo coincida con el tipo de fuente

Error: "Stream failed to start"

Mensaje completo: `△ Error: Stream failed to start`

Causa: El proceso `hls-generator` falló o no pudo inicializarse.

Razones comunes: - Codec no soportado (la fuente usa un codec que OBS no puede decodificar) - Archivo multimedia corrupto - Bibliotecas de OBS faltantes o mal configuradas

Soluciones: 1. Verifique logs de `hls-generator` (si están disponibles) 2. Pruebe el archivo en OBS directamente 3. Re-codifique el archivo multimedia a H.264/AAC 4. Verifique que las bibliotecas de OBS están instaladas

Error: "Browser source load error"

Mensaje completo: `△ Error: Browser source failed to load`

Causa: La URL de fuente Browser devolvió un error (404, 500, etc.) o falló al renderizar.

Razones comunes: - 404 Not Found (ruta de URL incorrecta) - Errores de CORS (restricciones de origen cruzado) - Errores de JavaScript en la página - Errores de certificado SSL (HTTPS)

Soluciones: 1. Abra la URL en un navegador regular, verifique errores en la consola 2. Verifique que la URL es públicamente accesible (o localmente alcanzable) 3. Verifique logs del servidor para errores 4. Para HTTPS: Asegure certificado SSL válido

Error: "Preview already running"

Mensaje completo: `△ Error: Preview already in progress`

Causa: Se intentó iniciar una segunda vista previa mientras una ya está activa.

Solución: Espere a que termine la vista previa actual (30 segundos) o deténgala manualmente primero.

Error: "Authentication required"

Mensaje completo: ▲ Error: Authentication required

Causa: La transmisión de red (RTSP, RTMP) requiere credenciales que no se proporcionaron o son incorrectas.

Soluciones: 1. Incluya credenciales en la URL: - RTSP: rtsp://username:password@camera.local/stream - RTMP: rtmp://username:password@server.local/live 2. Verifique que las credenciales son correctas 3. Verifique configuración de autenticación de cámara/servidor

5.6.6 Timeout de Vista Previa (30 segundos)

Todas las vistas previas se detienen automáticamente después de **30 segundos** para prevenir agotamiento de recursos.

¿Por qué 30 segundos? - Gestión de recursos: Cada vista previa consume CPU/GPU (codificación) y espacio en disco (segmentos HLS) - **Suficiente para pruebas:** 30 segundos es suficiente para verificar funcionalidad de fuente - **Previene vistas previas olvidadas:** Los usuarios pueden dejar el modal abierto; la parada automática asegura limpieza

Qué sucede en el timeout: 1. El frontend recibe mensaje WebSocket previewStopped 2. El reproductor HLS.js se destruye graciosamente 3. El elemento de video se limpia 4. El botón muestra mensaje de información: "i Preview automatically stopped after 30 seconds" 5. El mensaje se limpia automáticamente después de 5 segundos 6. El botón vuelve al estado idle 7. El backend limpia: - Mata el proceso hls-generator - Elimina archivos HLS temporales - Libera recursos

¿Quiere previsualizar por más tiempo? Haga clic en "► Preview Source" nuevamente después del timeout para reiniciar la vista previa.

5.6.7 Previsualizando Múltiples Fuentes

Si su evento tiene múltiples fuentes (ej., video de fondo + superposición + feed de cámara), previsualice cada una individualmente:

Flujo de trabajo: 1. Configure todas las fuentes en sus pestañas respectivas 2. Cambie a la pestaña Preview 3. Seleccione la primera fuente del menú desplegable 4. Haga clic en "► Preview Source", observe cómo se reproduce por 30 segundos 5. Después del timeout (o parada manual), seleccione la siguiente fuente del menú desplegable 6. Repita hasta que todas las fuentes estén verificadas

¿Por qué vista previa individual? - **Aislamiento:** Pruebe cada fuente independientemente - **Solución de problemas:** Identifique qué fuente específica tiene problemas - **Rendimiento:** Generar múltiples vistas previas simultáneamente es intensivo en recursos

Nota: La vista previa muestra fuentes **individualmente**, no combinadas. Para ver todas las fuentes juntas como aparecerán en OBS, guarde el evento y actívelo manualmente (establezca tiempo a 2 minutos en el futuro).

5.6.8 Consideraciones de Rendimiento de Vista Previa

La generación de vista previa es **intensiva en recursos**:

Uso de CPU/GPU: - La codificación a H.264 requiere CPU o GPU - La renderización de fuentes de navegador usa GPU (CEF chromium) - Múltiples vistas previas aumentan el uso de recursos

Uso de disco: - Cada vista previa genera 30 segundos de segmentos HLS - Tamaño típico: 5-15 MB por vista previa - Limpieza automática después de que la vista previa se detiene

Uso de red: - Las fuentes FFMPEG descargan la transmisión de red - Las fuentes Browser obtienen contenido remoto - Los segmentos HLS se sirven sobre HTTP local

Mejores prácticas: 1. **Previsualice una fuente a la vez:** No ejecute múltiples vistas previas en paralelo 2. **Cierre el modal cuando termine:** Libera recursos inmediatamente 3. **Use vista previa con moderación:** Solo al configurar nuevas fuentes 4. **Confíe en configuraciones que funcionan:** Una vez que una fuente está verificada, no hay necesidad de previsualizar cada vez

5.6.9 Lista de Verificación de Solución de Problemas de Vista Previa

Si la vista previa falla o se comporta inesperadamente, trabaje a través de esta lista de verificación:

Conectividad del backend - ¿Está Scene Scheduler ejecutándose? (verifique logs) - ¿Está WebSocket conectado? (verifique indicador de conexión en la interfaz) - ¿Puede crear eventos y verlos en el horario? (verifique funcionalidad básica)

Configuración de fuente - ¿Está completado el nombre de fuente? - ¿Es correcta la ruta de archivo/URL? (sin errores tipográficos) - Para fuentes multimedia: ¿Existe el archivo? `ls -la /path/to/file.mp4` - Para fuentes de red: ¿Es alcanzable la fuente? `ping <ip>` o `curl <url>`

Binario hls-generator - ¿Existe `hls-generator` en la ubicación `paths.hlsGenerator`? - ¿Es ejecutable? `chmod +x hls-generator` - ¿Se ejecuta solo? `./hls-generator --help`

Directorio de salida HLS - ¿Existe el directorio `webServer.hlsPath`? - ¿Es escribible? `touch hls/test.txt && rm hls/test.txt` - Verifique espacio en disco: `df -h`

Navegador/red - Pruebe en un navegador diferente (descarte problemas específicos del navegador) - Verifique la consola del navegador para errores de JavaScript (F12 → Console) - Deshabilite extensiones del navegador que puedan bloquear reproducción de video - Verifique red: ¿Puede el navegador alcanzar el backend? (pruebe `http://localhost:8080`)

Logs - Verifique logs de Scene Scheduler para errores relacionados con vista previa - Busque mensajes que contengan "preview", "hls-generator" o "sourcepreview"

Si todas las verificaciones pasan y la vista previa aún falla, consulte la Sección 10 (Resolución de Problemas) para diagnósticos avanzados.

5.6.10 Comportamiento de Vista Previa vs. Producción

Importante: La vista previa muestra una **aproximación cercana** de cómo aparecerán las fuentes en OBS, pero hay diferencias sutiles:

Similitudes: - Mismo contenido de fuente (archivo, URL, transmisión) - Misma decodificación (bibliotecas de OBS) - Misma salida de video/audio - Verifica conectividad y existencia de archivo

Diferencias: - La vista previa usa instancia aislada de OBS (no su OBS principal) - La vista previa no muestra posicionamiento/recorte de fuente (estas son configuraciones a nivel de escena) - La vista previa no muestra filtros o efectos (aplicados en OBS, no a nivel de fuente) - La vista previa usa codificación HLS

(ligera pérdida de calidad vs. salida directa de OBS) - La latencia de vista previa es mayor (la segmentación HLS agrega 3-6 segundos)

Qué significa esto: - **Use vista previa para:** Verificar que la fuente funciona, el contenido es correcto, la conectividad es buena - **No confíe en la vista previa para:** Gradación de color exacta, tiempo preciso, efectos de filtro, posicionamiento final

Validación final: Despues de guardar su evento y antes del uso en producción, actívelo manualmente en OBS (establezca tiempo a 2 minutos en el futuro) y observe la composición completa de la escena.

5.6.11 Avanzado: Vista Previa de browser_source con CEF

Las fuentes de navegador requieren **CEF (Chromium Embedded Framework)** para renderizar:

Cómo funciona: 1. `hls-generator` inicializa CEF 2. CEF carga la URL especificada en un navegador sin interfaz 3. El JavaScript/CSS de la página se ejecuta 4. Los cuadros renderizados se capturan y codifican a H.264 5. Los segmentos HLS se generan de la transmisión codificada

Consideraciones especiales para vista previa de browser_source:

Tiempo de inicio más largo: - Inicialización de CEF: 2-5 segundos - Carga de página (JavaScript, recursos): 2-10 segundos - **Total:** 5-15 segundos antes de que comience la transmisión - Sea paciente con el estado "Waiting for stream..."

Transparencia: - La vista previa muestra transparencia como **fondo negro** - En OBS, se respeta la transparecia (la superposición se muestra sobre fuentes subyacentes) - No se preocupe si el fondo de la vista previa es negro

Elementos interactivos: - La entrada de mouse/teclado no funciona en vista previa (CEF es sin interfaz) - Las animaciones y temporizadores funcionan normalmente - Las llamadas WebSocket/API funcionan (si la página las usa)

Uso de recursos: - CEF es intensivo en memoria (200-500 MB por instancia) - El uso de GPU puede ser alto para páginas complejas - Limite las vistas previas de browser_source para evitar sobrecarga del sistema

Depuración de problemas de browser_source: 1. Abra la URL en Chrome/Chromium regular (verifique errores de JavaScript) 2. Use página de prueba simple primero (asegure que CEF funciona): `file:///home/user/test.html` 3. Verifique logs

de CEF (si está habilitado en config) 4. Verifique que el plugin de fuente de navegador de OBS está instalado (dependencia de CEF)

6. Configuración del Sistema

6.1 Descripción General del Archivo de Configuración

Scene Scheduler utiliza un único archivo de configuración: `config.json`, ubicado en el mismo directorio que el ejecutable principal.

Propósito: Define cómo Scene Scheduler se conecta a OBS, dónde sirve su interfaz web y dónde almacena archivos.

Formato: JSON estándar (JavaScript Object Notation)

Ubicación del archivo: - **Linux:** `./config.json` (mismo directorio que `scenescheduler`) - **Windows:** `config.json` (mismo directorio que `scenescheduler.exe`)

Estructura del archivo:

```
json
{
  "obsWebSocket": { ... },
  "webServer": { ... },
  "schedule": { ... },
  "paths": { ... },
  "logging": { ... }
}
```

Cinco secciones principales: 1. **obsWebSocket:** Configuración de conexión a OBS
2. **webServer:** Configuración del servidor web 3. **schedule:** Rutas de archivos de horario y escena auxiliar 4. **paths:** Ubicaciones de binarios externos 5. **logging:** Configuración de logs

Cada sección se detalla en las subsecciones siguientes.

6.2 Validación y Análisis de config.json

Sintaxis JSON: Scene Scheduler espera JSON válido y estricto. Errores sintácticos comunes: - Comas faltantes entre campos - Comillas faltantes alrededor de claves o valores de cadena - Comentarios (JSON no permite comentarios) - Comas finales (la última entrada en un objeto no debe tener coma)

Validando su config.json:

```
bash
# Linux - usar jq (instalar si es necesario)
jq . config.json

# Si válido: imprime JSON formateado
# Si inválido: muestra mensaje de error

# Windows - usar PowerShell
Get-Content config.json | ConvertFrom-Json

# O validadores JSON en línea: https://jsonlint.com
```

Comportamiento al iniciar: 1. Scene Scheduler lee config.json 2. Analiza JSON (falla si es inválido) 3. Valida campos requeridos 4. Aplica valores predeterminados para campos opcionales 5. Registra configuración cargada 6. Procede a iniciar servicios

Si config.json falta: Scene Scheduler usará valores predeterminados pero probablemente fallará la conexión a OBS si WebSocket requiere contraseña.

6.3 Opciones de Configuración Detalladas

6.3.1 Configuración de OBS WebSocket (`obsWebSocket`)

Controla cómo Scene Scheduler se conecta a OBS Studio.

host (string, requerido) - Dirección del servidor OBS WebSocket - **Predeterminado:** "localhost" - **Uso común:** - "localhost" o "127.0.0.1" - OBS en la misma máquina - "192.168.1.100" - OBS en una máquina diferente en LAN - **Nunca** use "0.0.0.0" (no es una dirección válida para conectar)

port (número, requerido) - Puerto del servidor OBS WebSocket - **Predeterminado:** 4455 (predeterminado de OBS v5.x) - **Rango:** 1-65535 - **Nota:** Debe coincidir con la configuración de WebSocket de OBS (Tools → WebSocket Server Settings)

password (string, opcional) - Contraseña para autenticación de WebSocket - **Predeterminado:** "" (sin contraseña) - **Seguridad:** - **Siempre** establezca una contraseña en producción - Use contraseñas fuertes y únicas (16+ caracteres) - **No** codifique contraseñas sensibles en archivos de configuración versionados - Use variables de entorno en su lugar (ver Sección 6.4)

Configuraciones de ejemplo:

Configuración local (sin contraseña):

```
json
"obsWebSocket": {
    "host": "localhost",
    "port": 4455,
    "password": ""
}
```

Configuración de producción (con contraseña):

```
json
"obsWebSocket": {
    "host": "localhost",
    "port": 4455,
    "password": "your_secure_password_here"
}
```

Control remoto de OBS (máquina diferente):

```
json
"obsWebSocket": {
    "host": "192.168.1.50",
    "port": 4455,
    "password": "remote_obs_password"
}
```

Solución de problemas de OBS WebSocket: - **Connection refused:** OBS no está ejecutándose o WebSocket no está habilitado - **Authentication failed:** La contraseña no coincide con la configuración de OBS - **Connection timeout:** Bloqueo de firewall o host inalcanzable - **Protocol error:** Incompatibilidad de versión (Scene Scheduler requiere OBS WebSocket v5.x)

6.3.2 Configuración del Servidor Web (webServer)

Controla dónde Scene Scheduler sirve su interfaz web.

host (string, requerido) - Dirección IP para vincular el servidor HTTP - **Predeterminado:** "0.0.0.0" (todas las interfaces) - **Opciones:** - "0.0.0.0" - Escucha en todas las interfaces de red (accesible desde cualquier lugar) - "localhost" o "127.0.0.1" - Solo accesible desde la misma máquina (más seguro) - "192.168.1.100" - Escucha en una interfaz específica

Cuándo usar cada opción: - **Producción (acceso de red):** "0.0.0.0" - Permite acceso remoto desde tablets/teléfonos - **Desarrollo (solo local):** "localhost" - Bloquea acceso externo - **Interfaz específica:** "192.168.1.100" - Vincular a una red LAN específica

port (número, requerido) - Puerto para el servidor HTTP - **Predeterminado:** 8080 - **Rango:** 1-65535 - **Notas:** - Debe estar libre (no usado por otra aplicación) - Puertos privilegiados (<1024) requieren permisos root/admin - Puertos comunes: 8080, 8000, 3000

hlsPath (string, requerido) - Ruta del directorio para archivos de vista previa HLS (relativa al ejecutable) - **Predeterminado:** "hls" - **Requisitos:** - Directorio debe existir y ser escribible - Se usa solo para la función de vista previa (no para operación del programador) - Limpiado automáticamente después de que terminan las vistas previas

Configuraciones de ejemplo:

Predeterminado (acceso de red):

```
json
"webServer": {
    "host": "0.0.0.0",
    "port": 8080,
    "hlsPath": "hls"
}
```

Solo local (más seguro):

```
json
"webServer": {
    "host": "localhost",
    "port": 8080,
    "hlsPath": "hls"
}
```

Puerto personalizado (evitar conflicto):

```
json
"webServer": {
    "host": "0.0.0.0",
    "port": 3000,
```

```
        "hlsPath": "hls"  
    }
```

Ruta absoluta para HLS (para despliegues Docker):

```
json  
"webServer": {  
    "host": "0.0.0.0",  
    "port": 8080,  
    "hlsPath": "/var/lib/scenescheduler/hls"  
}
```

Solución de problemas del servidor web: - **Port already in use:** Otra aplicación está usando el puerto (cambie el puerto o detenga la aplicación conflictiva) - **Permission denied:** Puerto <1024 requiere permisos elevados (use un puerto más alto o ejecute como root/admin) - **Cannot bind to address:** Dirección host no válida o interfaz de red no disponible - **Firewall blocking:** Abra el puerto en el firewall para acceso de red

6.3.3 Configuración de Horario (schedule)

Controla ubicaciones de archivos de horario y escena auxiliar.

jsonPath (string, requerido) - Ruta al archivo de horario (relativa al ejecutable o ruta absoluta) - **Predeterminado:** "schedule.json" - **Formato:** Array JSON de eventos (ver Sección 11.1 para el esquema) - **Permisos:** Debe ser legible y escribible - **Respaldo:** Recomendado mantener respaldos de este archivo

scheduleSceneAux (string, requerido) - Nombre de la escena auxiliar de OBS utilizada para preparación - **Predeterminado:** "scheduleSceneAux" - **Creación automática:** Scene Scheduler crea automáticamente esta escena si no existe - **Propósito:** Escena oculta donde las fuentes se precargan antes de las transiciones

Notas importantes: - El nombre de la escena auxiliar distingue mayúsculas de minúsculas - La escena debe permanecer vacía (Scene Scheduler gestiona su contenido automáticamente) - No elimine esta escena mientras Scene Scheduler esté ejecutándose - Si cambia el nombre de la escena en config.json, Scene Scheduler creará una nueva escena con ese nombre

Configuraciones de ejemplo:

Predeterminado:

```
json
"schedule": {
    "jsonPath": "schedule.json",
    "scheduleSceneAux": "scheduleSceneAux"
}
```

Ubicación personalizada del archivo de horario:

```
json
"schedule": {
    "jsonPath": "/var/lib/scenescheduler/production_schedule.json",
    "scheduleSceneAux": "scheduleSceneAux"
}
```

Nombre personalizado de escena auxiliar:

```
json
"schedule": {
    "jsonPath": "schedule.json",
    "scheduleSceneAux": "staging_scene"
}
```

Solución de problemas de horario: - **Schedule not loading:** Verifique que el archivo jsonPath existe y es JSON válido - **Scene not found error:** Verifique que el nombre de scheduleSceneAux en config.json coincide (Scene Scheduler lo crea automáticamente) - **Permission denied:** Asegure que Scene Scheduler puede leer/ escribir el archivo de horario

6.3.4 Configuración de Rutas (paths)

Controla ubicaciones de binarios externos y herramientas.

hlsGenerator (string, requerido) - Ruta al ejecutable hls-generator (relativa al ejecutable principal) - **Predeterminado:** "./hls-generator" - **Propósito:** Genera transmisiones de vista previa HLS - **Requisitos:** - Debe existir y ser ejecutable (chmod +x hls-generator) - Debe ser compatible con su sistema (Linux x86_64) - Debe tener bibliotecas de OBS disponibles

Configuraciones de ejemplo:

Predeterminado (mismo directorio):

```
json
"paths": {
    "hlsGenerator": "./hls-generator"
}
```

Ruta absoluta:

```
json
"paths": {
    "hlsGenerator": "/usr/local/bin/hls-generator"
}
```

Subdirectorio:

```
json
"paths": {
    "hlsGenerator": "./bin/hls-generator"
}
```

Solución de problemas de hls-generator: - **File not found:** Verifique que el archivo existe en la ruta especificada - **Permission denied:** Ejecute `chmod +x hls-generator` - **Exec format error:** Binario no compatible con su sistema (arquitectura incorrecta)

6.3.5 Configuración de Logging (logging)

Controla el comportamiento de los logs de la aplicación.

level (string, opcional) - Nivel de verbosidad de los logs - **Opciones:** "debug", "info", "warn", "error" - **Predeterminado:** "info" - **Recomendación:** - Producción: "info" o "warn" - Depuración: "debug" - Solo críticos: "error"

format (string, opcional) - Formato de salida de logs - **Opciones:** - "text" - Legible por humanos (predeterminado) - "json" - Analizable por máquina (para herramientas de agregación de logs) - **Predeterminado:** "text"

Configuraciones de ejemplo:

Producción (predeterminado):

```
json
"logging": {
    "level": "info",
```

```
        "format": "text"
    }
```

Depuración:

```
json
"logging": {
    "level": "debug",
    "format": "text"
}
```

Agregación de logs:

```
json
"logging": {
    "level": "info",
    "format": "json"
}
```

Logging mínimo:

```
json
"logging": {
    "level": "error",
    "format": "text"
}
```

Niveles de log explicados: - **debug**: Todos los mensajes (muy verboso, incluye cambios de estado interno) - **info**: Información general (inicio, conexiones, activadores de eventos) - **warn**: Advertencias (problemas no críticos, características obsoletas) - **error**: Solo errores (fallas, excepciones)

6.4 Variables de Entorno

Algunas configuraciones pueden sobrescribirse con variables de entorno (útil para Docker, systemd):

OBS_WS_HOST - Sobrescribe obsWebSocket.host

```
bash
export OBS_WS_HOST="192.168.1.50"
./scenescheduler
```

OBS_WS_PORT - Sobrescribe obsWebSocket.port

```
bash
export OBS_WS_PORT="4456"
./scenescheduler
```

OBS_WS_PASSWORD - Sobrescribe obsWebSocket.password (recomendado por seguridad)

```
bash
export OBS_WS_PASSWORD="s3cur3p@ss"
./scenescheduler
```

WEB_SERVER_PORT - Sobrescribe webServer.port

```
bash
export WEB_SERVER_PORT="3000"
./scenescheduler
```

Prioridad: Variables de entorno > config.json > predeterminados

6.5 Validación de Configuración

Scene Scheduler valida la configuración al iniciar:

Verificaciones de validación: 1. El archivo de configuración existe y es JSON válido 2. Los campos requeridos están presentes 3. Los números de puerto están en rango válido (1-65535) 4. Las rutas de archivo son accesibles 5. El directorio HLS existe y es escribible

Comportamiento al iniciar: - **Configuración válida:** La aplicación inicia normalmente - **Configuración inválida:** Error registrado y la aplicación sale - **Configuración faltante:** Usa predeterminados (puede fallar si OBS requiere contraseña)

Ejemplos de errores de validación:

JSON inválido:

```
FATAL: Failed to parse config.json: invalid character '}' looking for beginning of object
```

Solución: Corrija la sintaxis JSON (verifique comas faltantes, comillas)

Campo requerido faltante:

```
FATAL: Missing required config field: obsWebSocket.host
```

Solución: Agregue el campo faltante a config.json

Puerto inválido:

```
FATAL: Invalid port number: 99999 (must be 1-65535)
```

Solución: Use un número de puerto válido

6.6 Ejemplos de Configuración para Escenarios Comunes

Escenario 1: Configuración de Una Sola Computadora

Tanto OBS como Scene Scheduler en la misma máquina, acceso solo local.

```
json
{
  "obsWebSocket": {
    "host": "localhost",
    "port": 4455,
    "password": ""
  },
  "webServer": {
    "host": "localhost",
    "port": 8080,
    "hlsPath": "hls"
  },
  "schedule": {
    "jsonPath": "schedule.json",
    "scheduleSceneAux": "scheduleSceneAux"
  },
  "paths": {
    "hlsGenerator": "./hls-generator"
  }
}
```

Acceso: <http://localhost:8080>

Escenario 2: Servidor de Producción (Acceso de Red)

Scene Scheduler accesible desde múltiples dispositivos en la red.

```

json
{
  "obsWebSocket": {
    "host": "localhost",
    "port": 4455,
    "password": "production_password_123"
  },
  "webServer": {
    "host": "0.0.0.0",
    "port": 8080,
    "hlsPath": "hls"
  },
  "schedule": {
    "jsonPath": "/var/lib/scenescheduler/schedule.json",
    "scheduleSceneAux": "scheduleSceneAux"
  },
  "paths": {
    "hlsGenerator": "/opt/scenescheduler/hls-generator"
  },
  "logging": {
    "level": "info",
    "format": "text"
  }
}

```

Acceso: <http://192.168.1.100:8080> (use la IP del servidor)

Escenario 3: Control Remoto de OBS

Scene Scheduler en una máquina diferente a OBS.

```

json
{
  "obsWebSocket": {
    "host": "192.168.1.50",
    "port": 4455,
    "password": "obs_remote_password"
  },
  "webServer": {
    "host": "0.0.0.0",
    "port": 8080,
    "hlsPath": "hls"
  },
  "schedule": {
    "jsonPath": "schedule.json",
    "scheduleSceneAux": "scheduleSceneAux"
  },
  "paths": {

```

```
        "hlsGenerator": "./hls-generator"
    }
}
```

Requisitos: - La máquina OBS (192.168.1.50) debe tener WebSocket habilitado - El firewall debe permitir el puerto 4455 - Ambas máquinas en la misma red (o VPN)

Escenario 4: Despliegue Docker

Usando variables de entorno para configuración dinámica.

config.json (mínimo):

```
json
{
  "schedule": {
    "jsonPath": "/data/schedule.json",
    "scheduleSceneAux": "scheduleSceneAux"
  },
  "paths": {
    "hlsGenerator": "/app/hls-generator"
  }
}
```

Comando Docker run:

```
bash
docker run -d \
-e OBS_WS_HOST=192.168.1.50 \
-e OBS_WS_PORT=4455 \
-e OBS_WS_PASSWORD=secure_password \
-e WEB_SERVER_PORT=8080 \
-v /path/to/schedule.json:/data/schedule.json \
-p 8080:8080 \
scenescheduler:latest
```

6.7 Consideraciones de Seguridad

Contraseña de OBS WebSocket: - Siempre establezca una contraseña en producción - Use contraseñas fuertes y únicas (16+ caracteres) - No comprometa contraseñas al control de versiones - Use variables de entorno para valores sensibles

Acceso al Servidor Web: - Use host: "localhost" si no se necesita acceso de red - Configure firewall para restringir acceso al puerto del servidor web - No exponga a internet público sin autenticación - Considere proxy inverso (nginx) con HTTPS para producción

Permisos de Archivos: - Archivo de horario: chmod 600 schedule.json (solo lectura/escritura del propietario) - Archivo de configuración: chmod 600 config.json - Directorio HLS: chmod 700 hls/

Estrategia de Respaldo: - Respaldos regulares de schedule.json - Almacene respaldos de forma segura (encriptado si es sensible) - Pruebe el procedimiento de restauración periódicamente

6.8 Actualizando Configuración

Sin reiniciar: - Los cambios de horario (schedule.json) se aplican automáticamente (recarga en caliente) - Los cambios de configuración de fuente se aplican en el próximo activador de evento

Requiere reinicio: - Configuraciones de OBS WebSocket - Host/puerto del servidor web - Rutas de archivo - Configuración de logging

Cómo reiniciar:

Linux:

```
bash
# Detener
pkill scenescheduler

# Reiniciar
./scenescheduler
```

O con systemd:

```
bash
sudo systemctl restart scenescheduler
```

Windows:

```
cmd
REM Detener: Presione Ctrl+C en la ventana del símbolo del sistema
REM 0: Cierre la ventana del símbolo del sistema
```

```
REM 0: Use el Administrador de tareas para finalizar scenescheduler.exe  
  
REM Reiniciar  
scenescheduler.exe
```

Lista de verificación de cambio de configuración: 1. Edite config.json con un editor de texto 2. Valide la sintaxis JSON (use jq . config.json o validador en línea) 3. Respalde la configuración anterior (opcional pero recomendado) 4. Reinicie Scene Scheduler si es necesario 5. Verifique la conexión (verifique interfaz web, indicador de conexión de OBS) 6. Pruebe funcionalidad (cree evento de prueba)

7. Cómo Funciona Internamente

Comprender los mecanismos internos de Scene Scheduler ayuda a optimizar configuraciones, resolver problemas y predecir el comportamiento durante escenarios complejos.

7.1 Sistema de Preparación

Una de las características más importantes de Scene Scheduler es el **sistema de preparación**, que asegura transiciones suaves y sin interrupciones sin retrasos de carga visibles.

Por Qué Existe la Preparación

Sin preparación, las transiciones de escena mostrarían retrasos de carga:

```
Evento activa → OBS cambia escena → Fuentes cargan → Usuario ve buffering
```

Con preparación, las fuentes se preparan con anticipación:

```
Evento activa → Proceso de preparación (5 pasos de la Sección 2.3) → Transición instantánea
```

El sistema de preparación usa la escena auxiliar scheduleSceneAux como área detrás de escena donde las fuentes se crean e inicializan antes de moverse a la escena visible.

Descripción General del Proceso de Preparación

Cuando llega el momento de un evento, Scene Scheduler ejecuta el proceso de 5 pasos descrito en la Sección 2.3:

1. **STAGING**: Fuentes creadas en scheduleSceneAux (invisible para espectadores)
2. **ACTIVATION**: Fuentes movidas a la escena objetivo
3. **SCENE SWITCH**: OBS transiciona a la escena objetivo
4. **CLEANUP**: Elementos temporales eliminados de scheduleSceneAux
5. **MONITOR**: El evento se ejecuta durante su duración configurada

Beneficios

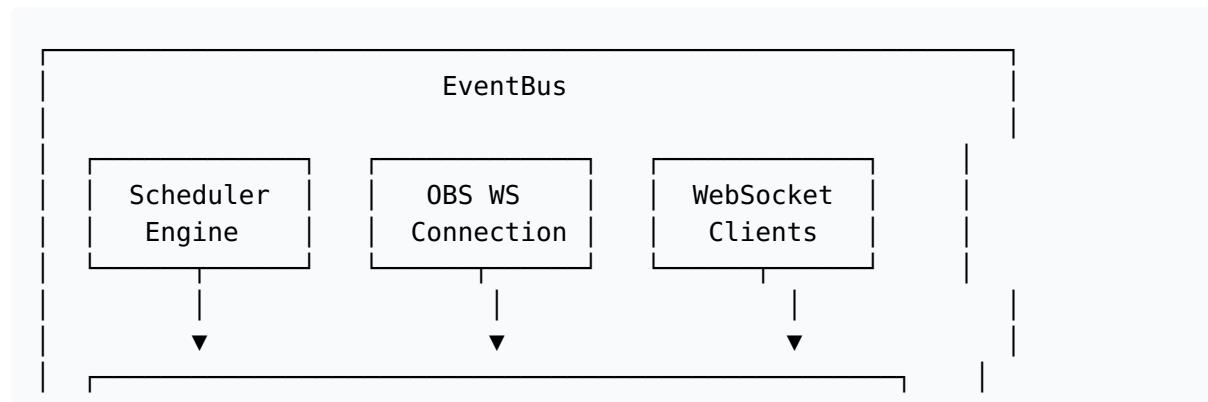
- **Sin carga visible**: Fuentes preparadas antes de mostrarlas a los espectadores
- **Transiciones atómicas**: O éxito completo o reversión segura
- **Eficiencia de recursos**: La limpieza previene fugas de memoria y fuentes huérfanas
- **Operación continua**: El sistema maneja programación automatizada 24/7

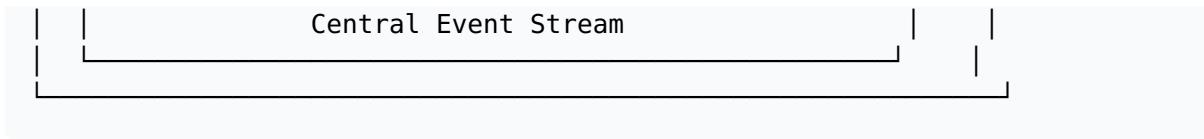
Consejos de Optimización de Preparación

1. **Use archivos locales cuando sea posible**: Archivos montados en red pueden agregar latencia
2. **Optimice fuentes de navegador**: Mantenga páginas simples, minimice JavaScript
3. **Pruebe transmisiones de red**: Verifique conectividad antes de programar
4. **Monitoree uso de recursos**: Verifique CPU/GPU durante transiciones

7.2 Sistema EventBus (Sincronización en Tiempo Real)

Scene Scheduler usa una arquitectura **EventBus** para sincronizar estado a través de todos los componentes:





Tipos de Eventos

Eventos de Horario: - scheduleUpdated: Activado cuando schedule.json cambia - eventTriggered: Ha llegado el momento de un evento - eventCompleted: Ha expirado la duración de un evento

Eventos de OBS: - obsConnected: Conexión WebSocket establecida - obsDisconnected: Conexión WebSocket perdida - sceneChanged: OBS cambió escenas (manual o automático) - sourceCreated: Una fuente fue agregada a OBS - sourceRemoved: Una fuente fue eliminada de OBS

Eventos de Frontend: - clientConnected: Navegador web conectado vía WebSocket - clientDisconnected: Navegador web cerró conexión - scheduleRequest: Cliente solicitó horario actual - previewRequest: Cliente solicitó vista previa de fuente

Flujo de Eventos Ejemplo: Agregando un Evento

1. Usuario hace clic en "Save Event" en interfaz web
 - ↳ Frontend envía mensaje WebSocket: { type: "addEvent", payload: {...} }
2. Backend recibe mensaje
 - ↳ Valida datos del evento
 - ↳ Agrega a schedule.json
 - ↳ EventBus emite: scheduleUpdated
3. Todos los componentes suscritos reaccionan:
 - ↳ Scheduler Engine: Recalcula próximo tiempo de activación
 - ↳ WebSocket Handler: Transmite actualización a todos los clientes
 - ↳ File Watcher: Actualiza horario en memoria
4. Todos los clientes web conectados reciben actualización
 - ↳ Monitor View: Actualiza lista de eventos
 - ↳ Editor View: Muestra nuevo evento en lista

Por Qué Importa el EventBus

Beneficios: - **Acoplamiento flexible:** Los componentes no dependen directamente entre sí - **Sincronización en tiempo real:** Todos los clientes reflejan instantáneamente los cambios - **Extensibilidad:** Nuevas características pueden

suscribirse a eventos sin modificar código existente - **Depuración:** El log de eventos muestra actividad completa del sistema

7.3 Comunicación OBS WebSocket

Scene Scheduler se comunica con OBS a través del **protocolo OBS WebSocket (v5.x)**.

Ciclo de Vida de Conexión

Fase 1: Conexión Inicial

1. Scene Scheduler inicia
2. Cliente WebSocket intenta conexión a OBS
3. Si se requiere contraseña, ocurre desafío de autenticación
4. Conexión establecida, mensaje identificado recibido
5. Scene Scheduler se suscribe a eventos de OBS

Fase 2: Estado Estacionario

- Mensajes de heartbeat cada 10 segundos (keepalive)
- Scene Scheduler envía solicitudes (GetSceneList, CreateInput, etc.)
- OBS envía respuestas y notificaciones de eventos

Fase 3: Reconexión

Si se pierde la conexión:

1. Scene Scheduler detecta desconexión
2. Comienzan intentos de reconexión automática (backoff exponencial)
3. Intervalos de reintentos: 1s, 2s, 4s, 8s, 16s, máx 30s
4. Al reconnectar: Re-sincroniza estado, re-suscribe a eventos

Operaciones Clave de OBS WebSocket

Gestión de Escenas: - GetSceneList: Recuperar todas las escenas (para población de desplegables) - SetCurrentProgramScene: Cambiar a una escena - GetCurrentProgramScene: Consultar escena activa

Gestión de Fuentes: - CreateInput: Agregar una nueva fuente a una escena - RemoveInput: Eliminar una fuente - SetInputSettings: Configurar propiedades de fuente - GetInputSettings: Consultar configuración de fuente

Gestión de Scene Items: - GetSceneItemList: Listar fuentes en una escena - SetSceneItemEnabled: Mostrar/ocultar una fuente - SetSceneItemTransform: Posicionar, escalar, recortar fuentes

Manejo de Errores

Errores de conexión: - **Authentication failure:** Registra error, sale (requiere corrección manual de configuración) - **Host unreachable:** Reintenta con backoff exponencial - **Protocol mismatch:** Registra error, sale (versión de OBS incompatible)

Errores de comando: - **Scene not found:** Registra advertencia, omite transición - **Source creation failed:** Registra error, continúa con otras fuentes - **Timeout:** Reintenta una vez, luego registra error y continúa

7.4 Recarga en Caliente de Horario

Scene Scheduler observa schedule.json en busca de cambios y recarga automáticamente sin reiniciar:

Cómo Funciona la Recarga en Caliente

Observación de Archivos:

1. Al iniciar, Scene Scheduler comienza a observar schedule.json
2. Eventos del sistema de archivos (write, modify) activan recarga
3. Debouncing previene múltiples recargas (espera 500ms después del último cambio)

Proceso de Recarga:

1. Detectar cambio de archivo
2. Leer schedule.json actualizado
3. Analizar y validar JSON
4. Si es válido:
 - ↳ Reemplazar horario en memoria
 - ↳ Recalcular próximo tiempo de activación de evento
 - ↳ Transmitir actualización a todos los clientes conectados
 - ↳ Registrar: "Schedule reloaded (N events)"
5. Si es inválido:
 - ↳ Mantener horario anterior (no romper sistema en ejecución)
 - ↳ Registrar error con detalles de análisis JSON
 - ↳ Notificar a clientes del error

Qué Activa la Recarga

Activadores automáticos: - Frontend guarda evento (agregar, editar, eliminar) - Editor externo modifica schedule.json (vim, nano, etc.) - Script escribe horario actualizado (automatización)

NO activa recarga: - Cambios en config.json (requiere reinicio) - Cambios en archivo de fuente (afecta próxima preparación)

Casos Extremos de Recarga

Durante preparación (activación de evento en progreso): - La recarga ocurre inmediatamente - El evento actualmente en preparación continúa con **configuración antigua** - El próximo evento usa nueva configuración

Durante evento activo (evento actualmente ejecutándose): - La recarga ocurre inmediatamente - El evento activo continúa con configuración antigua - Las fuentes permanecen como fueron configuradas originalmente - La limpieza ocurre en la duración original

Recomendación: Evite editar eventos que están ejecutándose activamente. Edite eventos futuros en su lugar.

7.5 Ciclo de Vida del Proceso de Vista Previa

Comprender el flujo interno del sistema de vista previa ayuda a resolver problemas:

Flujo de Solicitud de Vista Previa

Fase 1: Solicitud Iniciada (Frontend)

1. Usuario hace clic en botón "▶ Preview Source"
2. Frontend recopila configuración de fuente (nombre, tipo, URL/ruta, configuraciones)
3. Mensaje WebSocket enviado: { type: "startPreview", payload: {...} }
4. Estado del botón: " Starting preview..."

Fase 2: Procesamiento del Backend

1. Backend recibe mensaje startPreview
2. Valida configuración de fuente
3. Genera ID de vista previa único y ID de conexión
4. Crea sesión de vista previa en memoria

5. Genera subprocesso hls-generator con configuración de fuente
6. Respuesta enviada al frontend: { previewID, hlsURL }

Fase 3: Generación HLS (proceso hls-generator)

1. hls-generator inicializa bibliotecas de OBS
2. Crea escena temporal de OBS
3. Agrega fuente a escena (media/browser/ffmpeg)
4. Fuente carga e inicializa:
 - Media: Archivo abierto y buffered
 - Browser: CEF inicia, carga URL
 - FFMPEG: Conexión de red establecida
5. Comienza codificación (H.264 + AAC)
6. Segmentos HLS generados (archivos .ts)
7. Manifiesto de playlist escrito (.m3u8)
8. Verificación del primer segmento (busca etiqueta #EXTINF)

Fase 4: Sondeo del Frontend

1. Frontend sondea playlist: GET /hls/{previewID}/playlist.m3u8
2. Intervalos de reinicio: 500ms, 1s, 2s, 4s, 8s (backoff exponencial)
3. Timeout: 30 segundos máx
4. Estado del botón: " Waiting for stream..."

Fase 5: Comienza Reproducción

1. Playlist encontrado, HLS.js inicializa
2. HLS.js descarga segmentos y reproduce
3. Elemento de video muestra transmisión
4. Estado del botón: "■ Stop Preview"
5. Temporizador de timeout de 30 segundos comienza

Fase 6: Limpieza (Auto o Manual)

1. Timeout alcanzado (30s) o usuario hace clic en "Stop Preview"
2. Backend envía mensaje WebSocket previewStopped
3. Frontend destruye graciosamente HLS.js (previene errores 404)
4. Elemento de video limpiado
5. Limpieza del backend:
 - ↳ Mata proceso hls-generator (SIGTERM)
 - ↳ Espera 5s para apagado gracioso
 - ↳ Fuerza kill si aún ejecutándose (SIGKILL)
 - ↳ Elimina directorio HLS y todos los segmentos
 - ↳ Elimina sesión de vista previa de memoria

6. Estado del botón: "i Preview automatically stopped after 30 seconds"
7. Despues de 5s, botón se resetea al estado idle

Seguimiento de Conexión de Vista Previa

Scene Scheduler v1.6 usa **IDs de conexión** (no direcciones IP) para rastrear sesiones de vista previa:

¿Por qué IDs de conexión? - Seguro para NAT (múltiples clientes detrás del mismo NAT tienen IDs únicos) - Seguro (sin exposición de dirección IP en logs) - Confiable (sobrevive cambios de red)

Ciclo de vida del ID de conexión:

1. Conexión WebSocket establecida
2. ID único generado: "conn_<timestamp>_<random>"
3. ID asociado con sesión de vista previa
4. Al desconectar: Todas las vistas previas para esa conexión se limpian
5. Previene procesos de vista previa huérfanos

7.6 Limpieza y Gestión de Recursos

Scene Scheduler implementa limpieza exhaustiva para prevenir fugas de recursos:

Limpieza de Eventos (Después de Expirar Duración)

1. Temporizador de duración de evento se dispara
2. Comienza secuencia de limpieza:
 - Obtener lista de fuentes dinámicas creadas por este evento
 - Para cada fuente:
 - Verificar si aún existe en OBS
 - Eliminar de escena objetivo
 - Eliminar fuente de OBS
 - Limpiar escena scheduleSceneAux
 - Registrar: "Event cleanup completed"
3. Sistema listo para próximo evento

Limpieza idempotente: Todas las operaciones de limpieza son seguras de llamar múltiples veces: - `delete()` en clave de mapa no existente: no-op - `os.RemoveAll()` en directorio faltante: sin error - Eliminación de fuente de OBS de fuente faltante: ignorado

Limpieza de Vista Previa

1. Timeout de vista previa (30s) o parada manual
2. Secuencia de limpieza:
 - Detener temporizador de timeout (si está ejecutándose)
 - Enviar mensaje WebSocket previewStopped
 - Esperar 100ms (asegurar que mensaje fue entregado)
 - Matar proceso hls-generator:
 - Enviar SIGTERM (apagado gracioso)
 - Esperar 5 segundos
 - Enviar SIGKILL si aún ejecutándose (forzar)
 - Eliminar directorio HLS: rm -rf hls/{previewID}/
 - Eliminar sesión de vista previa del mapa en memoria
 - Registrar: "Preview cleanup completed"

Limpieza de Cliente WebSocket

Cuando cliente desconecta:

1. Manejador WebSocket detecta cierre de conexión
2. Obtener ID de conexión
3. Verificar vistas previas activas con este ID de conexión
4. Para cada vista previa:
 - ↳ Ejecutar limpieza completa de vista previa
5. Eliminar cliente de lista de transmisión
6. Registrar: "Client disconnected, cleaned up N previews"

Limpieza de Apagado de Aplicación

Cuando Scene Scheduler sale (SIGTERM/SIGINT):

1. Manejador de señal activado
2. Secuencia de apagado gracioso:
 - Detener aceptación de nuevas solicitudes
 - Detener todas las vistas previas activas
 - Limpiar todos los directorios HLS
 - Detener todos los procesos de preparación
 - Cerrar conexión OBS WebSocket
 - Cerrar servidor web
 - Vaciar logs
3. Salir con código 0

7.7 Sincronización de Estado

Scene Scheduler mantiene consistencia a través de múltiples componentes:

Sincronización de Estado Inicial (Conexión de Cliente)

Cuando cliente web conecta:

1. Conexión WebSocket establecida
2. Cliente envía: { type: "getInitialState" }
3. Backend responde con:
 - ↳ Horario actual (todos los eventos)
 - ↳ Evento activo (si hay alguno)
 - ↳ Estado de conexión de OBS
 - ↳ Escenas de OBS disponibles
 - ↳ Escena actual de OBS
4. Cliente renderiza UI con este estado

Sincronización de Estado Continua

Cambios de horario:

Usuario agrega evento → Backend actualiza schedule.json → EventBus emite scheduleUpdated

Cambios de estado de OBS:

Cambio manual de escena en OBS → Evento WebSocket recibido → EventBus emite sceneChanged

Estado de vista previa:

Vista previa inicia → Mensaje WebSocket solo al cliente solicitante → Estado del botón a 1
Vista previa detiene → Mensaje WebSocket solo al cliente solicitante → Botón se resetea a 0

Resolución de Conflictos

Múltiples clientes editando simultáneamente: - La última escritura gana (sin bloqueo optimista) - Todos los clientes reciben estado final vía transmisión - Condiciones de carrera son raras (velocidad de edición humana es lenta)

Cliente fuera de sincronización: - Cliente puede solicitar re-sincronización completa de estado en cualquier momento - Ocurre automáticamente en reconexión

8. Casos de Uso y Ejemplos

Esta sección proporciona escenarios del mundo real mostrando cómo usar Scene Scheduler efectivamente.

8.1 Canal Automatizado 24/7

Escenario: Un canal de TV comunitario que funciona 24 horas al día con programación programada.

Requisitos: - Cambio automático de escena - Intervención manual mínima - Programación nocturna - Pausas publicitarias entre shows

Diseño de Horario:

```
json
[
  {
    "time": "06:00:00",
    "scene": "Morning Show",
    "duration": "02:00:00",
    "sources": [
      {
        "type": "media_source",
        "name": "MorningIntro",
        "file": "/media/intros/morning.mp4",
        "loop": false
      },
      {
        "type": "browser_source",
        "name": "Clock",
        "url": "file:///overlays/clock.html",
        "width": 1920,
        "height": 1080
      }
    ],
    {
      "time": "08:00:00",
      "scene": "News Block",
      "duration": "01:00:00",
      "sources": [
        {
          "type": "ffmpeg_source",
          "name": "NewsFeed",
          "input": "rtsp://newscamera.local/live"
        },
        {

```

```
        "type": "browser_source",
        "name": "LowerThird",
        "url": "https://graphics.local/news_lower_third.html",
        "width": 1920,
        "height": 200
    }
]
},
{
    "time": "09:00:00",
    "scene": "Ad Break",
    "duration": "00:03:00",
    "sources": [
        {
            "type": "vlc_source",
            "name": "Commercials",
            "playlist": "/media/ads/morning_ads.xspf"
        }
    ]
},
{
    "time": "09:03:00",
    "scene": "Feature Film",
    "duration": "02:00:00",
    "sources": [
        {
            "type": "media_source",
            "name": "Movie",
            "file": "/media/films/morning_feature.mp4",
            "loop": false,
            "hw_decode": true
        }
    ]
},
{
    "time": "23:00:00",
    "scene": "Overnight Loop",
    "duration": "07:00:00",
    "sources": [
        {
            "type": "media_source",
            "name": "NightLoop",
            "file": "/media/overnight/holding_screen.mp4",
            "loop": true
        },
        {
            "type": "browser_source",
            "name": "Schedule",
            "url": "file:///overlays/tomorrow_schedule.html",
            "width": 400,
            "height": 800
        }
    ]
}
```

```
        }
    ]
}
]
```

Mejores prácticas para este caso de uso: - Pruebe todas las transiciones al menos una vez antes de transmitir en vivo - Mantenga contenido de respaldo listo (use bucles de larga duración) - Monitoree el sistema remotamente vía Monitor View - Configure alertas para fallas de OBS o pérdida de conexión

8.2 Automatización de Conferencia o Evento

Escenario: Conferencia de múltiples días con oradores programados, descansos y contenido de patrocinadores.

Requisitos: - Diapositivas de introducción de oradores - Feeds de cámara en vivo durante charlas - Anuncios de patrocinadores durante descansos - Temporizadores de cuenta regresiva

Ejemplo: Horario de Un Solo Día

```
json
[
  {
    "time": "08:00:00",
    "scene": "Welcome Screen",
    "duration": "01:00:00",
    "sources": [
      {
        "type": "media_source",
        "name": "WelcomeLoop",
        "file": "/conference/welcome_loop.mp4",
        "loop": true
      },
      {
        "type": "browser_source",
        "name": "Countdown",
        "url": "https://timer.local/countdown?target=09:00:00",
        "width": 400,
        "height": 200
      }
    ],
    {
      "time": "09:00:00",
      "scene": "Keynote",
      "duration": "01:00:00",
      "sources": [
        {
          "type": "media_source",
          "name": "KeynoteLoop",
          "file": "/conference/keynote_loop.mp4",
          "loop": true
        }
      ]
    }
]
```

```

"sources": [
    {
        "type": "ffmpeg_source",
        "name": "MainCamera",
        "input": "rtsp://camera1.local/stream"
    },
    {
        "type": "browser_source",
        "name": "SpeakerInfo",
        "url": "https://graphics.local/speaker?id=keynote",
        "width": 500,
        "height": 150
    }
],
{
    "time": "10:00:00",
    "scene": "Break",
    "duration": "00:15:00",
    "sources": [
        {
            "type": "vlc_source",
            "name": "SponsorAds",
            "playlist": "/conference/sponsor_ads.xspf"
        }
    ]
},
{
    "time": "10:15:00",
    "scene": "Talk 1",
    "duration": "00:30:00",
    "sources": [
        {
            "type": "ffmpeg_source",
            "name": "Speaker1Camera",
            "input": "rtsp://camera2.local/stream"
        },
        {
            "type": "browser_source",
            "name": "Slides",
            "url": "https://slides.local/talk1",
            "width": 1280,
            "height": 720
        }
    ]
}
]

```

Consejos: - Use la pestaña Preview para verificar todos los feeds de cámara antes del evento - Tenga escenas de respaldo listas para dificultades técnicas - Mantenga

duraciones de eventos ligeramente más largas de lo esperado (tiempo de amortiguación) - Ejecute un ensayo completo el día anterior

8.3 Señalización Digital

Escenario: Pantalla de tienda minorista mostrando promociones, videos de productos y anuncios.

Requisitos: - Contenido en bucle durante horario comercial - Promociones especiales en momentos específicos - Capacidad de anuncio de emergencia

Horario de Ejemplo:

```
json
[
  {
    "time": "09:00:00",
    "scene": "Store Opening",
    "duration": "00:05:00",
    "sources": [
      {
        "type": "media_source",
        "name": "OpeningVideo",
        "file": "/signage/store_opening.mp4",
        "loop": false
      }
    ],
    "time": "09:05:00",
    "scene": "General Promotions",
    "duration": "03:55:00",
    "sources": [
      {
        "type": "vlc_source",
        "name": "PromoPlaylist",
        "playlist": "/signage/general_promos.xspf"
      },
      {
        "type": "browser_source",
        "name": "Clock",
        "url": "file:///overlays/store_clock.html",
        "width": 300,
        "height": 100
      }
    ],
    "time": "12:55:00",
    "scene": "Emergency Announcements",
    "duration": "00:05:00",
    "sources": [
      {
        "type": "media_source",
        "name": "EmergencyAnnouncement",
        "file": "/signage/emergency_announcement.mp4",
        "loop": false
      }
    ]
  }
]
```

```
"time": "13:00:00",
"scene": "Lunch Special",
"duration": "01:00:00",
"sources": [
  {
    "type": "media_source",
    "name": "LunchPromo",
    "file": "/signage/lunch_special.mp4",
    "loop": true
  }
],
},
{
  "time": "14:00:00",
  "scene": "General Promotions",
  "duration": "07:00:00",
  "sources": [
    {
      "type": "vlc_source",
      "name": "PromoPlaylist",
      "playlist": "/signage/general_promos.xspf"
    }
  ]
},
{
  "time": "21:00:00",
  "scene": "Store Closing",
  "duration": "00:05:00",
  "sources": [
    {
      "type": "media_source",
      "name": "ClosingVideo",
      "file": "/signage/store_closing.mp4",
      "loop": false
    }
  ]
},
{
  "time": "21:05:00",
  "scene": "Closed",
  "duration": "11:55:00",
  "sources": [
    {
      "type": "media_source",
      "name": "ClosedScreen",
      "file": "/signage/closed_screen.mp4",
      "loop": true
    }
  ]
}
```

```
    }  
]
```

Anuncios de emergencia: - Active manualmente el cambio de escena en OBS - O: Edite el horario para insertar evento urgente con tiempo cercano al futuro - Después de la emergencia: El horario se reanuda automáticamente en el próximo evento

8.4 Producción de Transmisión en Vivo

Escenario: Transmisión en vivo semanal con pre-roll, contenido principal y post-roll.

Requisitos: - Video pre-roll automatizado antes de ir en vivo - Cambiar a cámara en vivo en momento exacto - Video post-roll después de que termina la transmisión

Ejemplo:

```
json  
[  
  {  
    "time": "19:55:00",  
    "scene": "Pre-Roll",  
    "duration": "00:05:00",  
    "sources": [  
      {  
        "type": "media_source",  
        "name": "PreRollVideo",  
        "file": "/stream/pre_roll.mp4",  
        "loop": false  
      },  
      {  
        "type": "browser_source",  
        "name": "StartingSoonOverlay",  
        "url": "https://overlay.local/starting_soon?time=20:00",  
        "width": 1920,  
        "height": 1080  
      }  
    ]  
  },  
  {  
    "time": "20:00:00",  
    "scene": "Live Stream",  
    "duration": "01:00:00",  
    "sources": [  
      {  
        "type": "ffmpeg_source",  
        "name": "MainCamera",
```

```
        "input": "rtsp://camera.local/main"
    },
    {
        "type": "ffmpeg_source",
        "name": "ScreenCapture",
        "input": "rtmp://localhost/screen"
    },
    {
        "type": "browser_source",
        "name": "ChatOverlay",
        "url": "https://chat.local/embed",
        "width": 400,
        "height": 600
    }
]
},
{
    "time": "21:00:00",
    "scene": "Post-Roll",
    "duration": "00:03:00",
    "sources": [
        {
            "type": "media_source",
            "name": "PostRollVideo",
            "file": "/stream/post_roll.mp4",
            "loop": false
        },
        {
            "type": "browser_source",
            "name": "ThankYouOverlay",
            "url": "https://overlay.local/thanks",
            "width": 1920,
            "height": 1080
        }
    ]
},
{
    "time": "21:03:00",
    "scene": "Offline Screen",
    "duration": "22:52:00",
    "sources": [
        {
            "type": "media_source",
            "name": "OfflineLoop",
            "file": "/stream/offline.mp4",
            "loop": true
        }
    ]
}
]
```

Consejos profesionales: - Inicie la grabación de OBS 5 minutos antes del tiempo programado - Use Scene Scheduler para tiempo, pero monitoree chat manualmente - Tenga escenas de respaldo para dificultades técnicas - Pruebe vista previa de todas las fuentes 30 minutos antes de ir en vivo

8.5 Automatización de Servicio Religioso

Escenario: Elementos automatizados de un servicio religioso mientras se permite control manual para elementos en vivo.

Requisitos: - Anuncios pre-servicio y cuenta regresiva - Letras de himnos/canciones automatizadas - Control manual del sermón - Bucle post-servicio

Enfoque de Automatización Mixta:

```
json
[
  {
    "time": "09:30:00",
    "scene": "Pre-Service",
    "duration": "00:30:00",
    "sources": [
      {
        "type": "vlc_source",
        "name": "Announcements",
        "playlist": "/worship/announcements.xspf"
      },
      {
        "type": "browser_source",
        "name": "ServiceCountdown",
        "url": "https://timer.local/countdown?target=10:00:00",
        "width": 600,
        "height": 200
      }
    ]
  },
  {
    "time": "10:00:00",
    "scene": "Welcome",
    "duration": "00:05:00",
    "sources": [
      {
        "type": "ffmpeg_source",
        "name": "MainCamera",
        "input": "rtsp://camera.local/front"
      },
      {
        "type": "browser_source",
        "name": "ServiceCountdown"
      }
    ]
  }
]
```

```

        "name": "WelcomeSlide",
        "url": "file:///slides/welcome.html",
        "width": 1920,
        "height": 1080
    }
]
},
{
    "time": "10:05:00",
    "scene": "Worship Songs",
    "duration": "00:25:00",
    "sources": [
        {
            "type": "ffmpeg_source",
            "name": "WideShot",
            "input": "rtsp://camera.local/wide"
        },
        {
            "type": "browser_source",
            "name": "Lyrics",
            "url": "https://lyrics.local/service?set=1",
            "width": 1920,
            "height": 400
        }
    ]
}
]

```

Anulación manual: - El operador puede cambiar escenas manualmente en OBS durante el sermón - El horario se reanuda después del sermón con bucle post-servicio - Scene Scheduler maneja elementos repetitivos, el humano maneja partes dinámicas

9. Mejores Prácticas

9.1 Recomendaciones de Archivos Multimedia

Codificación de Video

Configuraciones de codec recomendadas: - **Codec:** H.264 (AVC) - **Perfil:** High - **Nivel:** 4.2 o superior - **Bitrate:** - 1080p: 8-12 Mbps (CBR para reproducción consistente) - 720p: 5-8 Mbps - 4K: 25-40 Mbps (pruebe rendimiento del sistema) - **Frame rate:** Coincida con salida de OBS (típicamente 30 o 60 fps) - **Intervalo de keyframe:** 2 segundos (60 cuadros a 30fps, 120 cuadros a 60fps)

Codificación de audio: - **Codec:** AAC - **Bitrate:** 192 kbps (estéreo) o 384 kbps (5.1) - **Sample rate:** 48 kHz (coincida con OBS) - **Canales:** Estéreo (2.0) para la mayoría de casos de uso

Formato de contenedor: - **Preferido:** MP4 (.mp4) - **Alternativo:** MKV (.mkv) para flexibilidad - **Evite:** AVI (antiquado), MOV (problemas de codec)

Ejemplo de codificación con ffmpeg:

```
bash
ffmpeg -i input.mov -c:v libx264 -preset medium -crf 20 \
-c:a aac -b:a 192k -ar 48000 \
-movflags +faststart output.mp4
```

¿Por qué estas configuraciones? - H.264: Compatibilidad universal, soporte de decodificación por hardware - CBR: Previene subejecución de buffer durante reproducción - Keyframes: Permite búsqueda, bucle suave - AAC: Codec de audio estándar, baja latencia - faststart: Metadatos al principio (carga más rápida)

Organización de Archivos

Estructura de directorio recomendada:

```
/media/
├── intros/
│   ├── morning_intro.mp4
│   ├── evening_intro.mp4
│   └── weekend_intro.mp4
├── content/
│   ├── show1/
│   │   ├── episode01.mp4
│   │   └── episode02.mp4
│   └── show2/
│       └── episode01.mp4
└── ads/
    ├── commercial_1.mp4
    └── commercial_2.mp4
└── overlays/
    ├── lower_third.html
    └── clock.html
└── backgrounds/
    ├── holding_screen.mp4
    └── offline_loop.mp4
```

Beneficios: - Fácil de localizar archivos al configurar eventos - Nombres claros previenen errores - Respaldos organizados

Convenciones de nomenclatura: - Use minúsculas con guiones bajos: morning_intro.mp4 - Incluya fecha/versión si aplica: news_2025_10_28.mp4 - Evite espacios (use my_file.mp4 no my file.mp4) - Sea descriptivo: commercial_acme_30s.mp4 no comm1.mp4

Almacenamiento de Archivos

Almacenamiento Local vs. Red:

Almacenamiento local (recomendado): - Acceso más rápido (sin latencia de red) - Más confiable (sin dependencias de red) - Mejor para archivos grandes (video 4K) - Limitado por tamaño de disco

Almacenamiento de red (NFS/SMB): - Gestión centralizada - Fácil de actualizar contenido remotamente - Latencia de red afecta tiempos de carga - Punto único de falla (red/servidor)

Enfoque híbrido: - Almacene archivos usados frecuentemente localmente (intros, bucles) - Almacene archivos grandes en red (episodios pasados) - Cache archivos de red localmente cuando sea posible

9.2 Optimización de Rendimiento

Requisitos del Sistema

Especificaciones mínimas: - CPU: 4 núcleos, 2.5 GHz - RAM: 8 GB - Disco: SSD (para acceso rápido a multimedia) - Red: 100 Mbps (para transmisiones de red)

Especificaciones recomendadas: - CPU: 6-8 núcleos, 3.0+ GHz - RAM: 16 GB - Disco: SSD NVMe - GPU: GPU dedicada para codificación/decodificación por hardware - Red: 1 Gbps

Para 4K o múltiples transmisiones de red: - CPU: 8+ núcleos o GPU dedicada para codificación - RAM: 32 GB - Disco: RAID para redundancia

Configuración de OBS

Reducir uso de CPU: - Habilite codificación por hardware (NVENC, QuickSync, o AMF) - Reduzca resolución de salida si es posible (1080p vs. 4K) - Deshabilite

fuentes/escenas sin usar - Use decodificación por hardware para fuentes de multimedia

Optimice fuentes: - Limite conteo de fuentes de navegador (alto uso de CPU/GPU) - Use imágenes estáticas en lugar de fuentes de navegador cuando sea posible - Deshabilite "Shutdown when not visible" para fuentes siempre encendidas

Específico de Scene Scheduler: - Pruebe transiciones de eventos con todas las fuentes configuradas - Monitoree uso de CPU/GPU durante activadores de eventos - Use duraciones más cortas si las fuentes son livianas (reduce sobrecarga de limpieza)

Optimización de Fuentes de Navegador

Optimización HTML/CSS/JS: - Minimice JavaScript (evite frameworks pesados) - Use animaciones CSS (no bucles JavaScript setTimeout) - Optimice imágenes (comprima, use SVG cuando sea posible) - Carga diferida de assets (no cargue todo al cargar la página)

Ejemplo: Superposición de reloj optimizada

```
html
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background: transparent;
    margin: 0;
    padding: 20px;
    font-family: Arial, sans-serif;
    color: white;
    font-size: 48px;
    text-shadow: 2px 2px 4px black;
}
</style>
</head>
<body>
<div id="clock"></div>
<script>
// Actualice solo una vez por segundo (no 60fps)
setInterval(() => {
    document.getElementById('clock').textContent =
        new Date().toLocaleTimeString();
}, 1000);
</script>
```

```
</body>
</html>
```

Configuración de FPS: - Superposiciones estáticas: 1-5 FPS - Superposiciones animadas: 30 FPS - Animaciones suaves: 60 FPS (solo si es necesario)

9.3 Confiabilidad y Tiempo de Actividad

Para Operación 24/7

A nivel de sistema: - **Linux:** Más estable para servicios de larga ejecución, use systemd para auto-reinicio - **Windows:** Use Task Scheduler o NSSM (Non-Sucking Service Manager) para instalación de servicio - Deshabilite actualizaciones automáticas (ventanas de mantenimiento manual) - Monitoree recursos del sistema (CPU, RAM, disco)

Configuración de OBS: - Deshabilite auto-actualizaciones - Configure auto-reconexión para streaming - Use colecciones de escenas (recuperación fácil) - Respaldos regulares de perfil

Scene Scheduler: - **Linux:** Ejecute como servicio systemd (auto-reinicio en falla) - **Windows:** Ejecute como Servicio de Windows usando NSSM o Task Scheduler - Configure rotación de logs (prevenir llenado de disco) - Monitoree logs en busca de errores - Alerte en desconexión de OBS

Ejemplo de servicio systemd de Linux:

```
ini
[Unit]
Description=Scene Scheduler
After=network.target

[Service]
Type=simple
User=obs
WorkingDirectory=/opt/scenescheduler
ExecStart=/opt/scenescheduler/scenescheduler
Restart=always
RestartSec=10
Environment="OBS_WS_PASSWORD=yourpassword"

[Install]
WantedBy=multi-user.target
```

Servicio de Windows con NSSM:

```
cmd
REM Instale NSSM desde https://nssm.cc/

REM Instale Scene Scheduler como servicio
nssm install SceneScheduler "C:\scenescheduler\scenescheduler.exe"
nssm set SceneScheduler AppDirectory "C:\scenescheduler"
nssm set SceneScheduler AppEnvironmentExtra OBS_WS_PASSWORD=yourpassword
nssm start SceneScheduler
```

Estrategia de Respaldo

Qué respaldar: 1. schedule.json (crítico) 2. config.json 3. Colecciones de escenas de OBS 4. Archivos multimedia (si no son fácilmente reemplazables)

Frecuencia de respaldo: - schedule.json: Después de cada cambio importante - config.json: Después de configuración inicial y cambios - Archivos multimedia: Semanalmente o después de agregar nuevo contenido

Métodos de respaldo:

Linux:

```
bash
# Script de respaldo simple
#!/bin/bash
DATE=$(date +%Y%m%d_%H%M%S)
BACKUP_DIR="/backups/scenescheduler"

# Respaldar configuración y horario
cp /opt/scenescheduler/config.json "$BACKUP_DIR/config_$DATE.json"
cp /opt/scenescheduler/schedule.json "$BACKUP_DIR/schedule_$DATE.json"

# Mantener solo los últimos 30 respaldos
ls -t $BACKUP_DIR/schedule_*.json | tail -n +31 | xargs rm -f
```

Ejecutar automáticamente con cron:

```
cron
# Respaldar cada día a las 3 AM
0 3 * * * /opt/scenescheduler/backup.sh
```

Windows:

```

batch
REM backup.bat
@echo off
set TIMESTAMP=%date:~-4%date:~3,2%%date:~0,2%_time:~0,2%%time:~3,2%%time:~6,2%
set BACKUP_DIR=C:\backups\scenescheduler

REM Respaldar configuración y horario
copy C:\scenescheduler\config.json "%BACKUP_DIR%\config_%TIMESTAMP%.json"
copy C:\scenescheduler\schedule.json "%BACKUP_DIR%\schedule_%TIMESTAMP%.json"

REM Eliminar respaldos mayores a 30 días
forfiles /p "%BACKUP_DIR%" /m schedule_*.json /d -30 /c "cmd /c del @path"

```

Ejecutar automáticamente con Task Scheduler: 1. Abra Task Scheduler 2. Crear Tarea Básica → Diaria a las 3:00 AM 3. Acción: Iniciar un programa → C:\scenescheduler\backup.bat

9.4 Pruebas y Validación

Pruebas Pre-Producción

Lista de verificación de validación de horario: 1. Todos los tiempos de evento están en el futuro 2. No hay eventos superpuestos (o anulaciones intencionales) 3. Todas las escenas existen en OBS 4. Todas las rutas de archivo multimedia son válidas 5. Todas las URLs de red son alcanzables 6. Las duraciones de eventos son apropiadas 7. Cobertura de 24 horas (o brechas intencionales)

Validación de fuente: 1. Previsualice cada fuente en la pestaña Preview 2. Verifique apariencia visual 3. Verifique niveles de audio 4. Pruebe comportamiento de bucle (si está habilitado) 5. Verifique que fuentes de navegador cargan completamente

Validación del sistema: 1. Pruebe horario de día completo en avance rápido (establezca tiempos con 1 minuto de separación) 2. Verifique que la preparación funciona (fuentes aparecen instantáneamente) 3. Verifique limpieza (fuentes eliminadas después de duración) 4. Pruebe reconexión (mate OBS, reinicie, verifique reconexión) 5. Pruebe interfaz web desde dispositivo remoto

Monitoreo Continuo

Verificaciones diarias: - Verifique que el horario se cargó correctamente (verifique interfaz web) - Confirme estado de conexión de OBS (indicador verde) - Revise logs en busca de errores

Verificaciones semanales: - Pruebe manualmente algunas transiciones de eventos - Verifique que archivos multimedia son accesibles - Verifique espacio en disco (vistas previas HLS, logs)

Verificaciones mensuales: - Revisión completa de horario (elimine eventos antiguos) - Actualice contenido multimedia - Pruebe restauración de respaldo - Revise uso de recursos del sistema (tendencias de CPU, RAM)

9.5 Mejores Prácticas de Seguridad

OBS WebSocket: - Siempre use una contraseña fuerte - Use variables de entorno (no codificado en configuración) - No exponga puerto WebSocket a internet público - Use reglas de firewall para restringir acceso

Servidor Web: - Vincule a localhost si no se necesita acceso de red - Use proxy inverso con autenticación para acceso remoto - No exponga sin autenticación - Use HTTPS si es accesible sobre redes no confiables

Permisos de Archivos:

Linux:

```
bash
chmod 600 config.json schedule.json
chmod 700 hls/
chown obs:obs /opt/scenescheduler -R
```

Windows: - Haga clic derecho en config.json → Propiedades → Seguridad - Asegure que solo su usuario y SYSTEM tengan acceso - Elimine grupo "Everyone" o "Users" si está presente

Control de Acceso: - Limite acceso SSH al servidor - Use claves SSH (no contraseñas) - Revise regularmente cuentas de usuario - Monitoree logs de acceso

P: ¿Puedo acceder a Scene Scheduler desde otra computadora/teléfono/tableta?

R: ¡Sí! Esta es una **característica clave** de Scene Scheduler: controlar OBS remotamente sin cargar la máquina de OBS.

Configuración para acceso de red:

1. Configure config.json para enlace de red:

```
json
"webServer": {
    "host": "0.0.0.0",
    "port": 8080,
    "hlsPath": "hls"
}
```

2. Encuentre la dirección IP de su servidor:

Linux:

```
bash
ip addr show | grep inet
```

Windows:

```
cmd
ipconfig
```

Busque la dirección IPv4 (ej., 192.168.1.100)

1. Acceda desde cualquier dispositivo en su red: - Vista Monitor: <http://192.168.1.100:8080/> - Vista Editor: <http://192.168.1.100:8080/editor.html>

2. Si la conexión falla, verifique el firewall:

Linux:

```
bash
sudo ufw allow 8080/tcp
```

Windows: - Abra Windows Defender Firewall - Agregue regla de entrada para puerto TCP 8080

Beneficios: Controle OBS desde su laptop/tableta mientras OBS se ejecuta en un servidor dedicado, reduciendo la carga y permitiendo que múltiples personas monitorean simultáneamente.

P: La interfaz web muestra estado "Disconnected"

R: El frontend no puede alcanzar el backend. Diagnóstico:

Verificación 1: El backend está ejecutándose

```
bash
ps aux | grep scenescheduler
# Debería mostrar proceso en ejecución
```

Verificación 2: El puerto del servidor web es accesible

```
bash
# Pruebe desde la misma máquina
curl http://localhost:8080
# Debería retornar HTML (no "connection refused")

# Pruebe desde red (si accede remotamente)
curl http://<server-ip>:8080
```

Verificación 3: Configuración de host - Verifique configuración `webServer.host` en `config.json` - Para acceso de red, debe ser "`0.0.0.0`" no "`localhost`"

Verificación 4: El firewall permite el puerto

```
bash
sudo ufw status
# Verifique que el puerto 8080 esté permitido
```

Verificación 4: El navegador puede alcanzar el backend - Abra consola del navegador (F12 → Console) - Busque errores de WebSocket - Verifique que la URL sea correcta (`http://localhost:8080`, no `https://`)

P: Las escenas transicionan pero las fuentes del evento anterior permanecen visibles

R: Esto indica fallo de limpieza. Posibles causas:

- 1. La duración del evento es demasiado larga:** - La limpieza de fuentes ocurre DESPUÉS de que expire la duración - Verifique campo de duración del evento - Reduzca si es necesario

2. Fuentes manuales agregadas en OBS: - Scene Scheduler solo remueve fuentes que ÉL creó - Las fuentes manuales permanecen - Solución: Remueva manualmente o use escenas dedicadas

3. Error de limpieza: - Verifique logs para mensajes "cleanup failed" - OBS podría no responder (reinicie OBS)

P: La vista previa funciona pero el staging del evento falla con la misma fuente

R: La vista previa y el staging son sistemas separados. Los fallos de staging pueden indicar:

1. Problema de tiempo (tiempo de staging insuficiente): - Archivos grandes pueden no cargar en 30 segundos - Use archivos más pequeños o copias locales - Pruebe con pestaña Preview (si carga en <30s, staging debería funcionar)

2. Condición de carrera (múltiples eventos haciendo staging simultáneamente): - Verifique programación para eventos <30 segundos aparte - Escalone tiempos de eventos

3. Agotamiento de recursos: - CPU/GPU al máximo durante staging - Monitoree recursos: htop durante ventana T-30s - Reduzca cuenta o complejidad de fuentes

P: ¿Cómo reinicio todo a un estado limpio?

R: Procedimiento de reinicio completo:

Linux:

```
bash
# 1. Detenga Scene Scheduler
pkill scenescheduler

# 2. Limpie programación
echo "[]" > schedule.json

# 3. Limpie archivos de preview HLS
rm -rf hls/*

# 4. Reinicie OBS (limpie todas las fuentes dinámicas)
killall obs
obs &
```

```
# 5. Inicie Scene Scheduler (la escena auxiliar se recreará automáticamente)
./scenescheduler
```

Windows:

```
cmd
REM 1. Detenga Scene Scheduler (Ctrl+C o cierre ventana)

REM 2. Limpie programación
echo [] > schedule.json

REM 3. Limpie archivos de preview HLS
rmdir /s /q hls
mkdir hls

REM 4. Reinicie OBS
taskkill /IM obs64.exe /F
start "" "C:\Program Files\obs-studio\bin\64bit\obs64.exe"

REM 5. Inicie Scene Scheduler (la escena auxiliar se recreará automáticamente)
scenescheduler.exe
```

P: ¿Puedo ejecutar múltiples instancias de Scene Scheduler?

R: Sí, pero cada una necesita: - Puerto de servidor web diferente (config.json: "port": 8081) - Directorio HLS diferente (config.json: "hlsPath": "hls2") - Archivo de programación diferente (config.json: "jsonPath": "schedule2.json") - Escena auxiliar diferente (config.json: "scheduleSceneAux": "scheduleSceneAux2")

Todas las instancias pueden conectarse al mismo OBS.

P: Mi archivo de programación está corrupto. ¿Cómo lo recupero?

R: Pasos de recuperación:

1. Verifique auto-backups (si está configurado):

```
bash
ls -la /backups/scenescheduler/schedule_*.json
# Restaure el último
cp /backups/scenescheduler/schedule_LATEST.json schedule.json
```

2. Repare JSON manualmente:

```
bash
# Valide archivo actual
jq . schedule.json
# Muestra línea/columna de error

# Edite con editor de texto
nano schedule.json
```

3. Comience desde cero:

```
json
[]
```

Guarde como `schedule.json`, luego reconstruya programación en interfaz web

P: El evento se dispara a la hora incorrecta (¿problemas de zona horaria?)

R: Scene Scheduler usa hora local del sistema, no UTC.

Verifique zona horaria del sistema:

```
bash
timedatectl
# Muestra: Time zone: America/New_York (EST, -0500)
```

Si está incorrecta, establezca zona horaria correcta:

```
bash
sudo timedatectl set-timezone America/Los_Angeles
```

Verifique hora del sistema:

```
bash
date
# Debería coincidir con su hora local
```

Los eventos se disparan cuando la hora del sistema coincide con el campo `time` del evento (HH:MM:SS).

P: ¿Cómo actualizo a una nueva versión de Scene Scheduler?

R: Procedimiento de actualización segura:

Linux:

```
bash
# 1. Haga backup de config y programación actuales
cp config.json config.json.backup
cp schedule.json schedule.json.backup

# 2. Detenga versión actual
pkill scenescheduler

# 3. Extraer nueva versión
tar -xzf scenescheduler-v0.4-linux.tar.gz

# 4. Restaure config y programación
cp config.json.backup config.json
cp schedule.json.backup schedule.json

# 5. Inicie nueva versión
./scenescheduler
```

Windows:

```
cmd
REM 1. Haga backup de config y programación actuales
copy config.json config.json.backup
copy schedule.json schedule.json.backup

REM 2. Detenga versión actual (Ctrl+C o cierre ventana)

REM 3. Descargue y extraiga nueva versión a la misma carpeta
REM (Sobrescriba scenescheduler.exe y hls-generator.exe)

REM 4. Restaure config y programación (si fue sobrescrito)
copy config.json.backup config.json
copy schedule.json.backup schedule.json

REM 5. Inicie nueva versión
scenescheduler.exe
```

Revise notas de lanzamiento para cambios incompatibles o actualizaciones de esquema de config.

10.2 Flujo de Diagnóstico

Cuando encuentre problemas, siga este proceso de diagnóstico sistemático:

Paso 1: Identifique el Síntoma

Categorice su problema: - **La aplicación no inicia:** Vea Sección 10.3 - **Problemas de conexión con OBS:** Vea Sección 10.4 - **Fallos de vista previa:** Vea Sección 10.5 - **Problemas de staging/transición de eventos:** Vea Sección 10.6 - **Problemas de interfaz web:** Vea Sección 10.7 - **Problemas de rendimiento:** Vea Sección 10.8

Paso 2: Recopile Información

Recolete datos de diagnóstico:

1. Logs de Scene Scheduler:

```
bash
./scenescheduler 2>&1 | tee scenescheduler.log
```

2. Logs de OBS:

Linux:

```
bash
tail -f ~/.config/obs-studio/logs/$(ls -t ~/.config/obs-studio/logs/ | head -1)
```

Windows:

```
Abra: %APPDATA%\obs-studio\logs\
Vea el archivo de log más reciente en Notepad
```

1. Uso de recursos del sistema:

Linux:

```
bash
htop
# Note uso de CPU, RAM, disco
```

Windows:

```
Abra Task Manager (Ctrl+Shift+Esc)
Revise pestaña Performance para uso de CPU, RAM, disco
```

1. Configuración:

```
bash
cat config.json
cat schedule.json
```

2. Conectividad de red (si usa OBS remoto o fuentes de red):

```
bash
ping <obs-host>
curl -I <stream-url>
```

Paso 3: Reproduzca el Problema

Cree un caso de prueba mínimo:

1. **Simplifique programación:** - Remueva todos los eventos excepto uno - Use fuente simple (archivo de video local) - Establezca tiempo 2 minutos en el futuro
2. **Pruebe aisladamente:** - ¿El evento simplificado funciona? - Si sí: Problema de complejidad (demasiadas fuentes/eventos) - Si no: Problema fundamental de configuración
3. **Documente pasos de reproducción:** - Secuencia exacta de acciones - Comportamiento esperado vs. actual - Cualquier mensaje de error

Paso 4: Aplique la Solución

Después de identificar la causa raíz (Secciones 10.3-10.8), aplique la solución y verifique:

1. Haga un cambio a la vez
2. Pruebe después de cada cambio
3. Documente qué solucionó el problema
4. Actualice su configuración/programación en consecuencia

10.3 Problemas de Inicio de Aplicación

Problema: Scene Scheduler falla al iniciar o sale inmediatamente.

Error: "Failed to parse config.json"

Síntoma:

```
FATAL: Failed to parse config.json: invalid character '}' looking for beginning of objec
```

Causa: Sintaxis JSON inválida en config.json.

Solución: 1. Valide JSON:

```
bash
jq . config.json
```

1. Problemas comunes: - Comas faltantes entre campos - Comas al final antes de llaves de cierre - Comillas faltantes alrededor de strings - Corchetes/llaves desparejados
2. Corrija sintaxis y reintente

Error: "OBS WebSocket connection failed"

Síntoma:

```
ERROR: Failed to connect to OBS WebSocket: dial tcp [::1]:4455: connect: connection refused
```

Causa: OBS no ejecutándose o servidor WebSocket deshabilitado.

Solución: 1. Inicie OBS Studio 2. Habilite WebSocket: Tools → WebSocket Server Settings → "Enable WebSocket server" 3. Verifique que el puerto coincida con config.json 4. Reinicie Scene Scheduler

Error: "Authentication failed"

Síntoma:

```
ERROR: OBS WebSocket authentication failed: invalid password
```

Causa: Desajuste de contraseña entre config.json y configuración de OBS.

Solución: 1. Verifique contraseña de OBS: Tools → WebSocket Server Settings 2. Actualice config.json para coincidir:

```
json
  "obsWebSocket": {
    "password": "correct-password-here"
  }
```

1. O use variable de entorno:

```
bash
export OBS_WS_PASSWORD="correct-password"
./scenescheduler
```

Error: "Schedule file not found"

Síntoma:

```
ERROR: Failed to load schedule: open schedule.json: no such file or directory
```

Causa: schedule.json no existe en ubicación esperada.

Solución: 1. Cree programación vacía:

```
bash
echo "[]" > schedule.json
```

1. O especifique ruta diferente en config.json:

```
json
  "schedule": {
    "jsonPath": "/path/to/your/schedule.json"
  }
```

Error: "Port already in use"

Síntoma:

```
FATAL: Failed to start web server: listen tcp :8080: bind: address already in use
```

Causa: Otro proceso está usando puerto 8080.

Solución: 1. Encuentre proceso usando puerto:

```
bash
sudo lsof -i :8080
```

1. Deténgalo (si es seguro):

```
bash
kill <PID>
```

2. O use puerto diferente en config.json:

```
json
"webServer": {
    "port": 8081
}
```

10.4 Problemas de Conexión con OBS

Problema: Scene Scheduler inicia pero no puede comunicarse con OBS.

Se Desconecta Inmediatamente Después de Conectar

Síntomas: - "Connected to OBS" seguido de "Disconnected" en logs - Interfaz web muestra punto rojo (desconectado)

Causas y soluciones:

1. Desajuste de versión de OBS WebSocket:

```
bash
# Verifique versión de OBS
obs --version

# Scene Scheduler v1.6 requiere OBS WebSocket 5.x
# OBS 28+ incluye WebSocket 5.x por defecto
# OBS más antiguo: Instale plugin obs-websocket 5.x
```

2. Inestabilidad de red (OBS remoto):

```
bash
# Pruebe estabilidad de conexión
ping -c 100 <obs-host>
# Busque pérdida de paquetes
```

```
# Verifique latencia de red  
ping <obs-host>  
# Debería ser <50ms para red local
```

3. Firewall bloqueando reconexión:

```
bash  
# Verifique reglas de firewall  
sudo ufw status  
  
# Permita puerto OBS WebSocket  
sudo ufw allow 4455/tcp
```

Desconexiones/Reconexiones Frecuentes

Síntomas: - Logs muestran ciclos repetidos de desconexión/reconexión - Eventos se disparan inconsistentemente

Causas y soluciones:

1. OBS crasheando o congelándose: - Verifique logs de OBS para crashes - Reduzca complejidad de escenas de OBS - Actualice OBS a última versión

2. Agotamiento de recursos del sistema:

```
bash  
# Monitoree durante desconexiones  
htop  
  
# Si CPU/RAM al máximo:  
# - Reduzca cuenta de fuentes de OBS  
# - Deshabilite preview en OBS  
# - Cierre otras aplicaciones
```

3. Problemas de red (OBS remoto): - Verifique logs de switch/router - Pruebe con conexión ethernet directa - Use cableado en lugar de inalámbrico

Comandos Timeout o Fallan

Síntomas: - Transiciones de escenas retrasadas - Fuentes no creadas - Logs muestran errores "request timeout"

Causas y soluciones:

1. OBS sobrecargado: - Reduzca complejidad de escenas - Habilite codificación por hardware - Cierre escenas no usadas

2. Scene Scheduler haciendo demasiadas peticiones: - Reduzca frecuencia de eventos - Simplifique configuraciones de fuentes - Aumente ventana de staging (requiere cambio de código)

10.5 Problemas del Sistema de Vista Previa

Problema: La vista previa falla al iniciar, muestra errores o se comporta inesperadamente.

Timeout de Vista Previa (30 Segundos)

Síntoma: "Waiting for stream..." nunca se resuelve, timeout después de 30s.

Pasos de diagnóstico:

1. Verifique que hls-generator existe y ejecuta:

```
bash
# Verifique existencia
ls -la ./hls-generator

# Intente ejecutar manualmente
./hls-generator --help
# Debería mostrar uso, no "command not found"
```

2. Verifique permisos del directorio HLS:

```
bash
# Verifique escribible
touch hls/test.txt
rm hls/test.txt

# Si permiso denegado:
chmod 755 hls
```

3. Pruebe accesibilidad de fuente:

Para fuentes de medios:

```
bash
# ¿Archivo existe?
```

```
ls -la /path/to/file.mp4

# ¿Archivo legible?
cat /path/to/file.mp4 > /dev/null
```

Para fuentes de navegador:

```
bash
# ¿URL alcanzable?
curl -I https://example.com/overlay.html

# ¿DNS funciona?
nslookup example.com
```

Para fuentes FFMPEG:

```
bash
# ¿Stream alcanzable?
ffprobe rtsp://camera.local/stream

# ¿Ruta de red existe?
traceroute camera.local
```

4. Verifique espacio en disco:

```
bash
df -h
# Asegure que partición con hls/ tenga espacio libre
```

Vista Previa Muestra Pantalla Negra

Síntoma: Vista previa inicia pero video está negro/en blanco.

Causas:

1. Fuente de navegador con fondo transparente: - Esto es normal para fuentes de navegador - Transparencia se muestra como negro en vista previa - En OBS, overlay funcionará correctamente

2. Codec de video no soportado:

```
bash
# Verifique codec
ffprobe /path/to/file.mp4
```

```
# Busque codec_name (debería ser h264)
# Si no es h264, re-codifique:
ffmpeg -i input.mp4 -c:v libx264 -c:a aac output.mp4
```

3. Stream de red no enviando video: - Pruebe stream en VLC u otro reproductor - Verifique configuración de cámara/codificador

Vista Previa con Audio Pero Sin Video (o viceversa)

Síntoma: Se puede escuchar audio pero no hay video, o video reproduce silenciosamente.

Causas:

1. Archivo de medios de pista única: - Archivo puede contener solo video o audio - Verifique con ffprobe:

```
bash
ffprobe file.mp4
# Busque tanto streams "Video:" como "Audio:"
```

2. Problema de codec: - Codec de video no soportado pero audio funciona - Recodifique con codecs estándar (H.264 + AAC)

3. Audio de fuente de navegador deshabilitado en OBS: - Esta es configuración a nivel de OBS - Vista previa usa instancia aislada de OBS - Audio debería funcionar en producción

10.6 Problemas de Staging y Transición de Eventos

Problema: Eventos se disparan pero escenas no cambian, o fuentes no aparecen.

La Escena No Cambia a la Hora del Evento

Síntoma: Llega hora del evento, pero OBS permanece en escena actual.

Diagnóstico:

1. Verifique formato de hora del evento:

```
json
// CORRECTO:
```

```
"time": "14:30:00"

// INCORRECTO:
"time": "2:30 PM"
"time": "14:30"
"time": "14:30:00:000"
```

2. Verifique hora del sistema:

```
bash
date
# Debería mostrar hora local correcta
```

3. Verifique programación cargada: - Abra interfaz web - Verifique que evento aparezca en lista - Verifique "Current time" coincide con hora del sistema

4. Verifique logs para errores:

```
bash
# Busque mensaje de disparo de evento
grep "event triggered" scenescheduler.log
```

La Escena Cambia Pero Las Fuentes No Aparecen

Síntoma: OBS cambia a escena correcta, pero fuentes faltan o están negras.

Causas comunes:

1. Problema de configuración de escena auxiliar:

Solución: Verifique nombre scheduleSceneAux en config.json (Scene Scheduler la crea automáticamente)

2. Rutas de archivos incorrectas:

```
bash
# Verifique rutas en schedule.json
# Las rutas deben ser absolutas:

# Linux:
"/home/user/video.mp4"
 "~/video.mp4"
 "./video.mp4"
 "video.mp4"
```

```
# Windows:  
"C:/Videos/video.mp4"  
"C:\Videos\video.mp4"  
"Videos\video.mp4"  
"video.mp4"
```

3. Fuentes fallaron al crearse durante staging:

```
# Verifique logs para errores durante disparo de evento  
# Busque mensajes "failed to create source"
```

4. Problema de permisos:

```
bash  
# Verifique que usuario de Scene Scheduler pueda leer archivos  
sudo -u obs cat /path/to/file.mp4 > /dev/null  
# No debería mostrar "permission denied"
```

Fuentes Aparecen Tarde (No Precargadas)

Síntoma: Escena cambia pero fuentes cargan visiblemente (buffering, pantalla negra por segundos).

Causas:

1. Staging no completó: - Ventana de 30 segundos insuficiente - Use archivos más pequeños - Mejore velocidad de red (para streams)

2. Archivo en almacenamiento lento: - Montaje de red con alta latencia - Copie archivos localmente:

```
bash  
cp /nfs/remote/file.mp4 /local/file.mp4
```

3. Demasiadas fuentes cargando simultáneamente: - Reduzca cuenta de fuentes por evento - Pruebe con menos fuentes para aislar el problema

10.7 Problemas de Interfaz Web

Problema: Interfaz web no carga, muestra errores, o actualizaciones no aparecen.

"Cannot GET /" o Connection Refused

Síntoma: Navegador muestra error al acceder interfaz web de Scene Scheduler.

Causas:

1. Backend no ejecutándose:

```
bash
ps aux | grep scenescheduler
# Si nada: Inicie backend
./scenescheduler
```

2. URL incorrecta: - Verifique protocolo: http:// no https:// - Verifique que puerto coincida con config.json - **Accediendo desde misma máquina:** Use http://localhost:8080 - **Accediendo desde red:** Use http://<server-ip>:8080 (ej., http://192.168.1.100:8080)

3. Configuración de host en config.json: - **Para acceso de red:** "host": "0.0.0.0" (enlaza a todas las interfaces) - **Solo local:** "host": "localhost" (bloquea acceso de red) - Si no puede conectar desde red pero necesita, cambie host a 0.0.0.0 y reinicie

4. Firewall bloqueando:

```
bash
# Pruebe localmente primero
curl http://localhost:8080

# Si funciona localmente pero no desde red:
sudo ufw allow 8080/tcp
```

Estado "Disconnected" de WebSocket

Síntoma: Interfaz carga pero muestra indicador rojo "Disconnected".

Causas:

1. Conexión WebSocket bloqueada: - Verifique consola del navegador (F12 → Console) - Busque errores de WebSocket - Algunas redes corporativas bloquean WebSockets

2. Backend reiniciado: - Actualice página (F5) - WebSocket debería reconectar automáticamente

3. Problema CORS/proxy: - Si accede a través de proxy/reverse proxy - Configure proxy para permitir actualizaciones WebSocket

Cambios No Aparecen en Tiempo Real

Síntoma: Edite evento en Vista Editor, pero Vista Monitor no actualiza.

Causas:

- 1. WebSocket desconectado:** - Verifique indicador de conexión - Actualice página
- 2. Caché del navegador:** - Actualización forzada: Ctrl+Shift+R (Linux) - O limpie caché
- 3. Múltiples instancias de backend:** - Cada backend tiene estado separado - Asegure que todos los clientes conecten al mismo backend

Botón de Vista Previa Atascado en Estado "Starting..."

Síntoma: Botón muestra "Starting preview..." indefinidamente.

Causas:

- 1. Mensaje WebSocket perdido:** - Verifique indicador de conexión - Detenga vista previa manualmente (recargue página)
- 2. Proceso de vista previa del backend crasheó:** - Verifique logs para errores "preview failed" - Reinicie backend si es necesario
- 3. Error de JavaScript del navegador:** - Verifique consola (F12 → Console) - Recargue página

10.8 Problemas de Rendimiento

Problema: Uso alto de CPU/RAM, lag, o tiempos de respuesta lentos.

Uso Alto de CPU

Síntoma: CPU constantemente alto (>80%) incluso sin eventos activos.

Causas:

- 1. Fuentes de navegador ejecutándose continuamente:** - Fuentes de navegador consumen CPU incluso cuando escena está oculta - Solución: Habilite "Shutdown when not visible" en configuración de fuente
- 2. Demasiadas escenas en OBS:** - Cada escena consume memoria - Elimine escenas no usadas
- 3. Múltiples vistas previas ejecutándose:** - Verifique ventanas de vista previa olvidadas - Cierre modal de vista previa después de probar
- 4. JavaScript de fuente de navegador ineficiente:** - Animaciones excesivas o polling - Optimice JavaScript (vea Sección 9.2)

Uso Alto de Memoria

Síntoma: Uso de RAM crece con el tiempo, eventualmente causa crashes.

Causas:

- 1. Fuga de memoria en fuentes de navegador:** - SPAs complejas con JavaScript con fugas - Actualice fuentes periódicamente (cambio de escena)
- 2. Archivos de medios grandes:** - Videos 4K consumen mucha RAM - Use 1080p cuando sea posible
- 3. Archivos de vista previa HLS huérfanos:** - Directorios de vista previa viejos no limpiados - Limpieza manual:

```
bash
rm -rf hls/*
```

Transiciones de Escena Lentas

Síntoma: Cambios de escena ocurren varios segundos tarde.

Causas:

- 1. OBS sobrecargado:** - Demasiadas fuentes renderizando - Reduzca complejidad de escenas
- 2. Latencia de red (OBS remoto):**

```
bash
ping <obs-host>
# Debería ser <10ms para red local
```

3. Contención de recursos del sistema: - Otras aplicaciones compitiendo por CPU/GPU - Cierre aplicaciones innecesarias

4. Cuello de botella de E/S de disco:

```
bash
iostat
# Verifique si disco al 100%
```

- Use SSD en lugar de HDD
- Reduzca acceso concurrente a archivos

11. Referencia Técnica

11.1 Esquema JSON de Programación

Referencia completa para estructura de schedule.json.

Estructura Raíz

```
json
[
  {
    "time": "HH:MM:SS",
    "scene": "SceneName",
    "duration": "HH:MM:SS",
    "name": "Optional Event Name",
    "sources": [...]
  }
]
```

Campos de Objeto Event

Campo	Tipo	Requerido	Descripción
time	string	Sí	Hora para disparar (formato HH:MM:SS, 24 horas)
scene	string	Sí	Nombre de escena de OBS a activar
duration	string	Sí	Cuánto tiempo mantener fuentes activas (HH:MM:SS)

Campo	Tipo	Requerido	Descripción
name	string	No	Nombre de evento legible (para mostrar en UI)
sources	array	No	Array de objetos de fuente a agregar a escena

Objeto Source: media_source

```
json
{
  "type": "media_source",
  "name": "SourceName",
  "file": "/absolute/path/to/file.mp4",
  "loop": true,
  "restart_on_activate": false,
  "hw_decode": true
}
```

Campo	Tipo	Requerido	Por Defecto	Descripción
type	string	Sí	-	Debe ser "media_source"
name	string	Sí	-	Nombre único de fuente en OBS
file	string	Sí	-	Ruta absoluta a archivo de medios
loop	boolean	No	false	Hacer loop de video continuamente
restart_on_activate	boolean	No	false	Reiniciar desde inicio al activar escena
hw_decode	boolean	No	false	Usar decodificación por hardware (GPU)

Objeto Source: browser_source

```
json
{
  "type": "browser_source",
  "name": "SourceName",
  "url": "https://example.com/overlay.html",
  "width": 1920,
  "height": 1080,
  "css": "body { background: transparent; }",
  "shutdown_when_hidden": true,
  "refresh_on_activate": false,
  "fps": 30
}
```

Campo	Tipo	Requerido	Por Defecto	Descripción
type	string	Sí	-	Debe ser "browser_source"
name	string	Sí	-	Nombre único de fuente
url	string	Sí	-	URL completa (https://, http://, file:///)
width	integer	Sí	-	Ancho de viewport (píxeles)
height	integer	Sí	-	Alto de viewport (píxeles)
css	string	No	""	CSS personalizado a inyectar
shutdown_when_hidden	boolean	No	false	Detener renderizado cuando oculto
refresh_on_activate	boolean	No	false	Recargar página al activar escena
fps	integer	No	30	Tasa de frames (1-60)

Objeto Source: ffmpeg_source

```
json
{
  "type": "ffmpeg_source",
  "name": "SourceName",
  "input": "rtsp://camera.local/stream",
  "input_format": "",
  "buffering_mb": 2,
  "reconnect_delay_sec": 5,
  "hw_decode": false
}
```

Campo	Tipo	Requerido	Por Defecto	Descripción
type	string	Sí	-	Debe ser "ffmpeg_source"
name	string	Sí	-	Nombre único de fuente
input	string	Sí	-	URL de stream (rtsp://, rtmp://, srt://, etc.)
input_format	string	No	""	Formato de contenedor (auto-detecta si vacío)
buffering_mb	integer	No	2	Tamaño de buffer en MB (1-10)
reconnect_delay_sec	integer	No	5	Segundos a esperar antes de intento de reconexión
hw_decode	boolean	No	false	Usar decodificación por hardware

Objeto Source: vlc_source

```
json
{
    "type": "vlc_source",
    "name": "SourceName",
    "playlist": "/path/to/playlist.xspf",
    "loop": true,
    "shuffle": false
}
```

Campo	Tipo	Requerido	Por Defecto	Descripción
type	string	Sí	-	Debe ser "vlc_source"
name	string	Sí	-	Nombre único de fuente
playlist	string	Sí	-	Ruta a archivo de playlist (.xspf, .m3u)
loop	boolean	No	false	Hacer loop de playlist
shuffle	boolean	No	false	Orden de reproducción aleatorio

Ejemplo Completo

```
json
[
    {
        "time": "14:30:00",
        "scene": "Afternoon Show",
        "duration": "01:00:00",
        "name": "Daily Afternoon Broadcast",
        "sources": [
            {
                "type": "media_source",
                "name": "IntroVideo",
                "file": "/media/intros/afternoon.mp4",
                "loop": false,
                "hw_decode": true
            },
            {
                "type": "browser_source",
                "name": "LowerThird",
                "url": "https://graphics.local/lowerthird.html",
                "width": 1920,
                "height": 200,
                "css": "body { background: transparent; }",
                "shutdown_when_hidden": true,
                "fps": 30
            }
        ]
    }
]
```

```

    {
      "type": "ffmpeg_source",
      "name": "LiveCamera",
      "input": "rtsp://camera.local:554/stream",
      "buffering_mb": 3,
      "hw_decode": true
    }
  ]
},
{
  "time": "15:30:00",
  "scene": "News Segment",
  "duration": "00:30:00",
  "sources": []
}
]

```

11.2 Protocolo WebSocket

Scene Scheduler usa WebSocket para comunicación en tiempo real entre backend y frontend.

Conexión

Cliente inicia conexión:

```

URL WebSocket: ws://localhost:8080/ws
Protocolo: WebSocket estándar (RFC 6455)

```

Handshake:

```

Cliente → Servidor: Petición de actualización WebSocket
Servidor → Cliente: 101 Switching Protocols
Conexión establecida

```

Formato de Mensaje

Todos los mensajes son JSON:

```

json
{
  "type": "messageType",
  "payload": { ... }
}

```

Mensajes Cliente → Servidor

Obtener programación actual:

```
json
{
  "type": "getSchedule"
}
```

Agregar evento:

```
json
{
  "type": "addEvent",
  "payload": {
    "time": "14:30:00",
    "scene": "SceneName",
    "duration": "01:00:00",
    "sources": [...]
  }
}
```

Editar evento:

```
json
{
  "type": "editEvent",
  "payload": {
    "index": 0,
    "event": { ... }
  }
}
```

Eliminar evento:

```
json
{
  "type": "deleteEvent",
  "payload": {
    "index": 0
  }
}
```

Iniciar vista previa:

```
json
{
  "type": "startPreview",
  "payload": {
    "source": { ... }
  }
}
```

Detener vista previa:

```
json
{
  "type": "stopPreview",
  "payload": {
    "previewID": "preview_123"
  }
}
```

Mensajes Servidor → Cliente

Programación actualizada:

```
json
{
  "type": "scheduleUpdated",
  "payload": {
    "events": [...]
  }
}
```

Evento actual cambió:

```
json
{
  "type": "currentEventChanged",
  "payload": {
    "eventIndex": 0,
    "event": { ... }
  }
}
```

Estado de conexión OBS:

```
json
{
  "type": "obsConnectionStatus",
  "payload": {
    "connected": true
  }
}
```

Lista de escenas actualizada:

```
json
{
  "type": "sceneListUpdated",
  "payload": {
    "scenes": ["Scene 1", "Scene 2", ...]
  }
}
```

Vista previa iniciada:

```
json
{
  "type": "previewStarted",
  "payload": {
    "previewID": "preview_123",
    "hlsURL": "/hls/preview_123/playlist.m3u8"
  }
}
```

Vista previa detenida:

```
json
{
  "type": "previewStopped",
  "payload": {
    "reason": "Preview automatically stopped after 30 seconds"
  }
}
```

Error:

```
json
{
  "type": "error",
  "payload": {
```

```
        "message": "Error description"
    }
}
```

11.3 Herramientas de Línea de Comandos

hls-generator

Herramienta independiente para generar streams de vista previa HLS.

Uso:

```
bash
./hls-generator [options]
```

Opciones:

Flag	Descripción	Ejemplo
--source-type	Tipo de fuente	media, browser, ffmpeg
--source-name	Nombre para fuente en OBS	PreviewSource
--source-uri	URI/ruta a fuente	/path/file.mp4, https://...
--output-dir	Directorio de salida HLS	/tmp/hls/preview_123
--width	Ancho de video (px)	1920
--height	Alto de video (px)	1080
--duration	Duración máxima (segundos)	30

Ejemplo:

```
bash
./hls-generator \
--source-type media \
--source-name TestVideo \
--source-uri /media/test.mp4 \
--output-dir /tmp/hls/test \
--width 1920 \
--height 1080 \
--duration 30
```

Salida:

```
/tmp/hls/test/
├── playlist.m3u8
```

```
|── segment000.ts  
|── segment001.ts  
└── ...
```

11.4 Referencia de Configuración

Vea Sección 6 para documentación completa de configuración.

Referencia rápida de todos los campos de config.json:

```
json
{
  "obsWebSocket": {
    "host": "localhost",           // Hostname/IP de OBS
    "port": 4455,                  // Puerto WebSocket
    "password": "password"        // Contraseña WebSocket
  },
  "webServer": {
    "host": "0.0.0.0",             // Dirección de enlace
    "port": 8080,                  // Puerto HTTP
    "hlsPath": "hls"               // Directorio HLS (nombre de campo CORRECTO)
  },
  "schedule": {
    "jsonPath": "schedule.json",   // Ruta a archivo de programación
    "scheduleSceneAux": "scheduleSceneAux" // Nombre de escena auxiliar
  },
  "paths": {
    "hlsGenerator": "./hls-generator" // Ruta a binario hls-generator
  },
  "logging": {
    "level": "info",                // debug, info, warn, error
    "format": "text"                // text o json
  }
}
```

11.5 Glosario

Auxiliary Scene (scheduleSceneAux) Una escena oculta de OBS usada por Scene Scheduler para hacer staging de fuentes antes de que se necesiten. Las fuentes se precargan aquí, luego se mueven a la escena objetivo durante la transición.

CEF (Chromium Embedded Framework) El motor de navegador embebido usado por OBS para renderizar fuentes de navegador. Requerido para funcionalidad de vista previa de browser_source.

Connection ID Identificador único asignado a cada conexión de cliente WebSocket. Usado para rastrear sesiones de vista previa y asegurar limpieza apropiada. Formato: conn_<timestamp>_<random>.

Duration La duración de tiempo que las fuentes de un evento permanecen activas antes de limpieza. Especificado en formato HH:MM:SS.

EventBus Sistema interno pub/sub que sincroniza estado a través de componentes de Scene Scheduler (motor de programación, conexión OBS, clientes WebSocket).

FFMPEG Source Tipo de fuente de OBS para entradas de streaming de red (RTSP, RTMP, SRT, RTP, HLS, etc.).

HLS (HTTP Live Streaming) Protocolo de streaming usado por sistema de vista previa. Los medios se segmentan en pequeños chunks (archivos .ts) con un manifiesto de playlist (.m3u8).

Hot-Reload Recarga automática de schedule.json cuando se detectan cambios en archivo, sin reiniciar Scene Scheduler.

Idempotent Una operación que puede llamarse múltiples veces de forma segura sin cambiar el resultado más allá de la llamada inicial. Las operaciones de limpieza de Scene Scheduler son idempotentes.

Media Source Tipo de fuente de OBS para archivos de video/audio locales (MP4, MKV, MOV, etc.).

Preview Prueba en tiempo real de una fuente antes de agregarla a un evento. Genera stream HLS de 30 segundos reproducible en navegador.

Preview ID Identificador único para sesión de vista previa. Usado en URLs HLS y para rastreo/limpieza.

Scene Una colección de fuentes en OBS. Cada evento cambia a una escena específica.

Staging La ventana de 30 segundos antes de que se dispare un evento, durante la cual las fuentes se precargan en la escena auxiliar.

Source Cualquier elemento de contenido en OBS: archivos de medios, páginas de navegador, streams de red, imágenes, etc.

VLC Source Tipo de fuente de OBS usando bibliotecas VLC para reproducción de medios. Soporta playlists y codecs exóticos.

WebSocket Protocolo de comunicación full-duplex usado por OBS (para control) y Scene Scheduler (para sincronización de frontend).

12. Tarjeta de Referencia Rápida

Comandos Esenciales

Linux:

```
bash
# Iniciar Scene Scheduler
./scenescheduler

# Iniciar con variables de entorno
OBS_WS_PASSWORD="secret" ./scenescheduler

# Validar config
jq . config.json

# Validar programación
jq . schedule.json

# Hacer backup de programación
cp schedule.json schedule.backup.json

# Reiniciar programación
echo "[]" > schedule.json

# Verificar si está ejecutándose
ps aux | grep scenescheduler

# Detener
pkill scenescheduler

# Ver logs (si redirigidos)
tail -f scenescheduler.log
```

Windows:

```
cmd
REM Iniciar Scene Scheduler
scenescheduler.exe

REM Iniciar con variables de entorno
set OBS_WS_PASSWORD=secret
scenescheduler.exe
```

```

REM Validar config (requiere jq.exe)
jq . config.json

REM Hacer backup de programación
copy schedule.json schedule.backup.json

REM Reiniciar programación
echo [] > schedule.json

REM Verificar si está ejecutándose
tasklist | findstr scenescheduler

REM Detener (Ctrl+C o Task Manager)
taskkill /IM scenescheduler.exe /F

REM Ver logs (abrir en Notepad)
notepad scenescheduler.log

```

Archivos Clave

Archivo	Propósito	Ubicación
scenescheduler	Ejecutable principal	Directorio de aplicación
config.json	Configuración	Igual que ejecutable
schedule.json	Programación de eventos	Especificado en config
hls-generator	Generador de vista previa	Especificado en config
hls/	Archivos HLS de vista previa	Especificado en config

URLs de Interfaz Web

Acceso Local (misma máquina):

```

Vista Monitor: http://localhost:8080/
Vista Editor: http://localhost:8080/editor.html
WebSocket: ws://localhost:8080/ws

```

Acceso de Red (desde otros dispositivos):

```

Vista Monitor: http://<server-ip>:8080/
Vista Editor: http://<server-ip>:8080/editor.html
WebSocket: ws://<server-ip>:8080/ws

Ejemplo: http://192.168.1.100:8080/

```

Encontrando IP del servidor: - Linux: ip addr show | grep inet - Windows: ipconfig

Nota: config.json debe tener "host": "0.0.0.0" para acceso de red

Formato de Hora de Evento

Formato: HH:MM:SS (24 horas)

Ejemplos:

00:00:00 Medianoc
09:30:00 9:30 AM
14:45:30 2:45:30 PM
23:59:59 11:59:59 PM

Tipos de Fuente Comunes

```
json
// Archivo de medios (Linux)
{
  "type": "media_source",
  "name": "Video",
  "file": "/home/user/videos/file.mp4",
  "loop": false
}

// Archivo de medios (Windows)
{
  "type": "media_source",
  "name": "Video",
  "file": "C:/Videos/file.mp4",
  "loop": false
}

// Overlay de navegador
{
  "type": "browser_source",
  "name": "Overlay",
  "url": "https://example.com/page.html",
  "width": 1920,
  "height": 1080
}

// Stream de red
{
  "type": "ffmpeg_source",
  "name": "Camera",
```

```
        "input": "rtsp://camera.local/stream"
    }
```

Listado de Verificación de Troubleshooting

Scene Scheduler no inicia: - Verifique sintaxis de config.json: jq . config.json - Verifique que OBS esté ejecutándose - Verifique WebSocket de OBS habilitado (Tools → WebSocket Server Settings) - Verifique que contraseña coincide

Los eventos no se disparan: - Verifique hora del sistema: date - Verifique formato de hora de evento (HH:MM:SS) - Verifique programación cargada (interfaz web) - Revise logs para errores

Las fuentes no aparecen: - Verifique que nombre scheduleSceneAux esté configurado correctamente en config.json - Verifique que rutas de archivos sean absolutas - Pruebe acceso a archivos: cat /path/file.mp4 > /dev/null - Verifique logs para "failed to create source"

Timeout de vista previa: - Verifique que hls-generator existe: ls -la ./hls-generator - Hágalo ejecutable: chmod +x hls-generator - Verifique directorio HLS escribible: touch hls/test; rm hls/test - Pruebe fuente (archivo existe, URL alcanzable)

Interfaz web desconectada: - Backend ejecutándose: ps aux | grep scenescheduler - Puerto accesible: curl http://localhost:8080 - Verifique consola del navegador para errores (F12)

Consejos de Rendimiento

- Use codec de video H.264 (compatibilidad universal)
- Habilite decodificación/codificación por hardware
- Almacene medios en SSD (no HDD)
- Limite fuentes de navegador (uso alto de CPU)
- No programe eventos <30s aparte
- Use archivos locales (no montajes de red)
- Monitoree CPU/RAM durante ventana de staging
- Optimice JavaScript de fuentes de navegador

Listado de Verificación de Seguridad

- Establezca contraseña fuerte de OBS WebSocket
- Use variable de entorno para contraseña (no config.json)

- Enlace servidor web a localhost si no necesita acceso de red
- Configure firewall (permite solo puertos necesarios)
- Establezca permisos restrictivos de archivos:

```
bash
chmod 600 config.json schedule.json
chmod 700 hls/
```

- Backups regulares de schedule.json
- Mantenga Scene Scheduler y OBS actualizados

Obteniendo Ayuda

Documentación: - Manual completo: Manual Español v0.4.md (este documento) - Especificaciones técnicas disponibles en la carpeta docs

Antes de reportar bugs: 1. Verifique FAQ (Sección 10.1) 2. Revise sección relevante de troubleshooting (Sección 10) 3. Recopile logs y configuración 4. Cree caso de reproducción mínimo 5. Incluya versión de Scene Scheduler, versión de OBS, OS

Fin de Manual Español v0.4.md

Scene Scheduler v0.4 - 28 de Octubre, 2025