



## MP08 - Serveis de Xarxa

### UF4 - Serveis d'àudio i vídeo

#### EXERCICI PRÀCTIC - Servidor DLNA i RTMP

Pol Jiménez

<https://github.com/pjimenez21/ubuntu-dlna>

Imagen utilizada:

<https://hub.docker.com/r/gabekangas/owncast>

#### Objectius

Administra serveis de vídeo identificant les necessitats de distribució i adaptant-ne els formats.

- Descriu la funcionalitat del servei de vídeo.
- Instal·la i configura un servidor de vídeo.
- Configura el client per a l'accés al servidor de vídeo.
- Reconeix i utilitza formats de compressió de vídeo digital.
- Utilitza tècniques de sindicació i subscripció de vídeo.
- Elabora documentació relativa a la instal·lació i administració del servidor de vídeo.

#### Enunciat

**ATENCIÓ: CAL APROVAR TANT LA PART 1 COM LA PART 2 PER A APROVAR AQUESTA PRÀCTICA**

#### Part 1

##### EXERCICI 1.a- (10%) -

Explica amb les teves pròpies paraules, **posant exemples (si no poses exemples, no s'avaluarà)**, què és i com funciona el protocol UPnP. (Cal explicar les diferents fases del protocol -addressing, discovery, etc. -)

**R:**

**UPnP (Universal Plug and Play): Un protocolo de red que tiene como objetivo simplificar la configuración de los dispositivos de red para los usuarios finales (routers, impresoras, cámaras IP, etc)**

**Para que el dispositivo sea capaz de comunicarse y configurarse de forma sencilla tiene que pasar por estas etapas:**

**Addressing: Los dispositivos de la red recibirán una dirección IP única que lo identifica en la red con el objetivo de comunicarse con otros dispositivos. Por ejemplo al visitar una web, cuando colocamos una dirección en el Chrome y este la utiliza para encontrar y mostrar el contenido correspondiente.**



**Discovery:** Los dispositivos escanearán si encuentran otros dispositivos utilizando mensajes de descubrimiento UPnP para identificar los dispositivos disponibles en la red. Por ejemplo, cuando un pc hace un escaneo de los dispositivos con el objetivo de encontrar una impresora disponible para poder imprimir un documento.

**Description:** Como bien dice su nombre aquí los dispositivos compartirán la información sobre sus especificaciones (capacidades - funciones) "para que se conozcan entre sí" y así se encontraran en comunicación y podrán utilizarse entre si. Ejemplo: Producto en línea - Descripción: "Teléfono inteligente de última generación con cámara de alta resolución, pantalla táctil y capacidad de almacenamiento de 128 GB."

**Control:** En esta etapa los dispositivos se enviarán comandos entre sí para poder ejercer ciertas funciones (controlar funcionamiento). Esto ocurre cuando utilizamos un móvil e instalamos una aplicación para controlar la televisión a través de nuestro dispositivo.

**Eventing:** Como dice su nombre (eventos) esto son envíos de señal que puedan hacer los dispositivos a otros. Esto por ejemplo, cómo a día de hoy se compran muchas cámaras IP con la que podemos controlar/monitorear nuestra cámara. Pues también estas cámaras nos envían una notificación cuando ha detectado una actividad sospechosa para notificarnos.

#### EXERCICI 1.b - (10%) -

Explica detalladament, amb exemples (si no poses exemples, no s'avaluarà), què és DLNA i com funciona un servidor DLNA.

**R:**

DLNA (Digital Living Network Alliance) permite que con los dispositivos modernos tecnológicos, como las smart tv, ordenadores, móviles y tablets, tengan la máxima facilidad para poder compartir y reproducir contenido multimedia, como fotos, música y vídeos, en una red local. Para lograr esto está utilizando el UPnP (Universal Plug and Play), que es lo que está haciendo que podamos escanear los dispositivos de nuestro alrededor y establecer la conexión con ellos sin tener que hacer una configuración manual sofisticada. DLNA es muy útil ya que acepta muchos formatos de archivos a la hora de hacer los intercambios de nuestro material multimedia.



Un servidor DLNA puede estar en forma de dispositivo o de una aplicación que tiene la capacidad de poder almacenar y compartir contenido multimedia en una red local.

Por ejemplo un servidor DLNA puede ser en forma de ordenador con un software especializado, un dispositivo de almacenamiento conectado a la red (NAS) o incluso un router con funcionalidades de servidor.

Para que un servidor DLNA pueda ser utilizado por otros dispositivos DLNA, debe estar conectado a la misma red local y estar configurado de manera que tenga la compartición abierta de los contenidos multimedia a través de las carpetas que se compartan dentro de la red local.

Los dispositivos DLNA pueden acceder a estas carpetas compartidas y buscar los contenidos multimedia disponibles. Una vez que un dispositivo DLNA ha encontrado un servidor DLNA y ha accedido a una carpeta compartida, puede buscar, seleccionar y reproducir los contenidos multimedia disponibles.

Por ejemplo si disponemos de una configuración en nuestra red local que nos permite disfrutar de nuestra música favorita en la Smart TV sin problemas. Para que esto ocurra, tenemos que disponer de un servidor multimedia instalado en un ordenador y una Smart TV conectada a la misma red. El servidor está configurado para compartir una carpeta que contiene una variedad de archivos de música en formato MP3.

Una vez que estamos en el menú de contenidos multimedia de tu Smart TV, se puede buscar y seleccionar el servidor multimedia que has configurado en nuestro ordenador. A partir de ahí, tendremos acceso a la carpeta compartida con nuestra música. Ahora podremos explorar y reproducir directamente en nuestra Smart TV los archivos de música que tengamos disponibles.

## EXERCICI 2 - (10%) -

Explica detalladament, amb exemples (si no poses exemples, no s'avaluarà), què és RTP, RTMP, RTSP, i HLS, i quina és la diferència.

Podeu consultar aquests enllaços:

<https://www.youtube.com/watch?v=eQA1wJyYmZ8>

[https://www.streamingvideoprovider.com/live-streaming-platform/streaming\\_video\\_protocols](https://www.streamingvideoprovider.com/live-streaming-platform/streaming_video_protocols)

**R:**

Los protocolos RTP, RTMP, RTSP y HLS están especialmente destinados para la transmisión de datos de audio y vídeo a través de Internet.



### **Protocolo RTP (Real-Time Transport Protocol):**

Es ampliamente utilizado para la transmisión en tiempo real de datos de audio y vídeo a través de redes de comunicaciones. Su principal aplicación se encuentra en escenarios que requieren transmisiones en vivo, como videoconferencias o eventos deportivos en tiempo real.

### **Protocolo RTMP (Real-Time Messaging Protocol):**

Se emplea para transmitir datos de audio y video en tiempo real en Internet. Su uso se centra principalmente en la transmisión de videos en vivo y el streaming de audio. Es compatible con plataformas reconocidas, tales como YouTube y Facebook.

### **El protocolo RTSP (Real-Time Streaming Protocol):**

Se destaca por su capacidad para transmitir audio y video en tiempo real a través de redes de comunicaciones. Es empleado principalmente en aplicaciones de videovigilancia y transmisiones en vivo.

### **Por último, el protocolo HLS (HTTP Live Streaming):**

Se basa en HTTP (Protocolo de Transferencia de Hipertexto). Este protocolo se caracteriza por dividir el vídeo en segmentos más pequeños, permitiendo su descarga independiente y brindando la posibilidad al usuario de reproducir el video mientras se descarga. HLS es compatible con plataformas como Apple, Netflix y Amazon Prime.

Las diferencias entre estos protocolos son las siguientes:

RTP, RTMP y RTSP son protocolos de transmisión en tiempo real, mientras que HLS es un protocolo de transmisión en tiempo diferido.

RTMP es un protocolo propietario desarrollado por Adobe, mientras que RTP, RTSP y HLS son protocolos abiertos.

RTMP y HLS se basan en el protocolo HTTP, mientras que RTP y RTSP no lo hacen. HLS utiliza la técnica de descarga progresiva para ofrecer una mejor calidad de video, mientras que RTP, RTMP y RTSP transmiten el video de forma continua.



### EXERCICI 3- (20%) -

**3.1 - (10%)** Els dos servidors multimèdia per a xarxes locals que estan més implantats en l'actualitat són KODI i PLEX. Fes una taula comparativa entre els dos.

**R: Ambos KODI y PLEX ofrecen soporte multiplataforma, interfaz de usuario, reproducción de medios y gestión de bibliotecas de medios. PLEX cuenta con sincronización en tiempo real, servidor de transmisión en vivo, soporte de aplicaciones móviles, integración de servicios de transmisión en línea, acceso remoto a través de Internet, transcodificación de medios, control parental, compartir bibliotecas con otros usuarios, actualizaciones frecuentes, capacidad de streaming en múltiples dispositivos y soporte de subtítulos y audio multilingüe.**

**Por otro lado, KODI no ofrece sincronización en tiempo real ni servidor de transmisión en vivo. Además, PLEX no permite la personalización de la interfaz.**

Característiques	KODI	PLEX
Soporte multiplataforma	Sí	Sí
Interfaz de usuario	Sí	Sí
Reproducción de medios	Sí	Sí
Gestión de bibliotecas	Sí	Sí
de medios		
Sincronización en	No	Sí
tiempo real		
Servidor de transmisión	No	Sí
en vivo		
Soporte de aplicaciones	Sí	Sí
móviles		
Integración de servicios	Sí	Sí
de transmisión en línea		
Acceso remoto a través	Sí	Sí
de Internet		
Transcodificación de	Sí	Sí
medios		
Control parental	Sí	Sí
Compartir bibliotecas	Sí	Sí
con otros usuarios		
Actualizaciones	Sí	Sí
frecuentes		
Capacidad de streaming	Sí	Sí
en múltiples dispositivos		
Personalización de la	Sí	No
interfaz		
Soporte de subtítulos	Sí	Sí
y audio multilingüe		



**3.2 - (10%)** Explica detalladament com es configura el reproductor multimèdia VLC per tal que actui com a client DLNA. Podeu llegir el següent article:

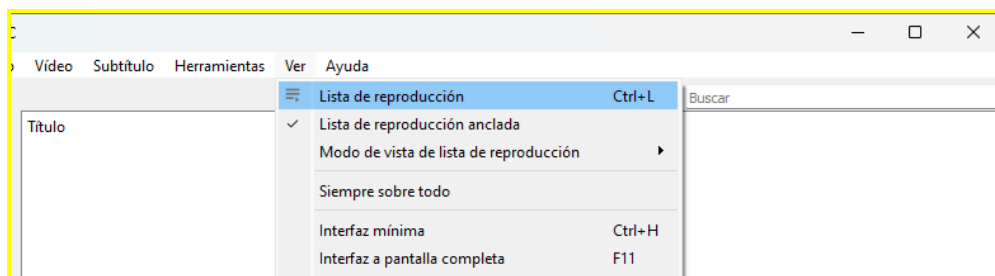
<https://www.vlchelp.com/access-media-upnp-dlna/>

**R:**

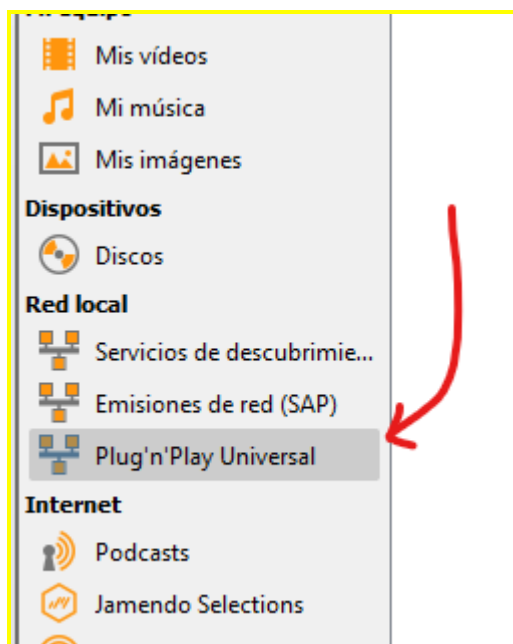
**VLC Media Player tiene la característica que nos va a permitir acceder a dispositivos y archivos multimedia en nuestra red local a través de UPnP/DLNA . Para hacerlo haremos los siguientes pasos:**

**1- Abrimos VLC en nuestro ordenador.**

**2- Nos dirigiremos a la parte superior de la ventana en "Ver" y seleccionaremos "Lista de reproducción" en el menú desplegable.**



**3- En la columna izquierda de la lista de reproducción, buscaremos la opción "Universal Plug'n'Play" (UPnP).**

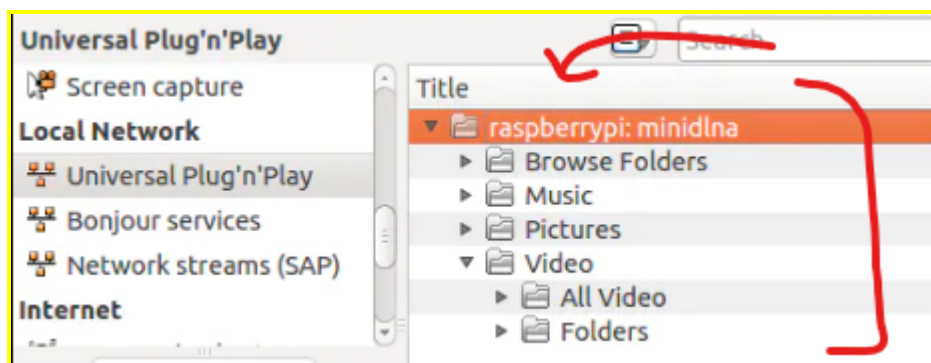




4- Podremos observar la lista de dispositivos y servidores DLNA disponibles en nuestra red local. Entonces simplemente tendremos que seleccionar el dispositivo o servidor al que queremos acceder.

(Dependiendo del dispositivo, nos podría llegar a solicitar credenciales de acceso para determinado dispositivo o servidor DLNA.)

Una vez que tengamos el dispositivo o servidor seleccionado ya tendremos la libertad de poder navegar por los archivos multimedia disponibles y reproducirlos en VLC.



#### EXERCICI 4 - (10%):

Explica què és IGMP snooping, en què consisteix, i com afecta en la transmissió de continguts multimèdia.

**R:**

Imaginemos que tenemos una red con varios dispositivos que desean recibir un flujo de datos multicast enviado por el servidor. Sin IGMP Snooping, todos los switches en la red transmitirán el flujo de datos a todos los dispositivos, incluso aquellos que no están interesados en el flujo multicast. Esto puede generar congestión en la red y hacer que los dispositivos no reciban correctamente los paquetes multicast.

Con IGMP Snooping, el switch escucha los mensajes IGMP enviados por los dispositivos que desean recibir el flujo de datos multicast. A través de estos mensajes, el switch sabe qué dispositivos están interesados en el flujo de datos multicast y puede enviar los paquetes solo a esos dispositivos, reduciendo la congestión de la red y mejorando la eficiencia en la transmisión de los contenidos multimedia.



**En resumen, IGMP Snooping permite a los switches controlar el flujo de datos multicast en la red y enviar los paquetes solo a los dispositivos que lo necesitan, mejorando así la calidad de la transmisión de contenidos multimedia.**

Part 2

**Pots fer el 5A o el 5B (el 5B té menys puntuació).**

### **EXERCICI 5 A - (50%) -**

En el següent tutorial, que hi ha al següent enllaç, s'explica com instal·lar un servidor DLNA, RTMP, o HLS a un Ubuntu:

<https://chapuboot.blogspot.com/2018/09/servidor-dlna-ubuntu-1804.html>

Extrapola aquesta informació per a crear una imatge de Docker (fent el teu propi Dockerfile), basada en Ubuntu 22.04, en la que implantaràs un servidor DLNA.

Has d'usar el reproductor VLC que tens instal·lat a Windows perquè es connecti al servidor DLNA que has implantat i reproduïxi un vídeo en format MP4 que hauràs guardat dins d'un volum de docker (que hauràs creat).

Per fer funcionar el servei tens dues opcions:

1- Crea un fitxer shellscript que creï una imatge a partir del Dockerfile i, posteriorment, creï un contenidor a partir d'aquesta imatge, arrencant així el servei, obrint els ports apropiats, creant els volums necessaris, etc, perquè funcioni sense problema i el VLC s'hi pugui connectar.

2- L'alternativa al shellscript és fer un docker-compose.yaml

Has de posar captures de pantalla de tot el procés, i també que demostrin que funciona.

**Si no hi ha demostració de que funcioni, s'avaluarà l'exercici amb un 0.**

Finalment, has de:

- Pujar la teva imatge docker (creada a partir del teu Dockerfile) al teu dockerhub, i tindrà el següent nom:  
`elteunodusuari/ubuntu-dlna`





- Crea un repo a github, i inclou-hi els teus fitxers: Dockerfile, shellscript, docker-compose.yaml. No oblidis que has d'incloure un README.md. El nom del repo de github serà també:  
`elteunodusuari/ubuntu-dlna`

### EXERCICI 5 B - (20%) - **HE ESCOGIDO ESTA OPCIÓN**

Has d'aconseguir, a l'igual que a l'exercici 5A, que el VLC media player es connecti a un servidor de vídeo, però ara no crearàs tu el Dockerfile, sinó que podràs reutilitzar qualsevol imatge de docker que vingui configurada amb un servidor DLNA, RTMP, o HLS:

Teniu informació sobre imatges docker de servidors de RTMP al final de l'arxiu README.md del repositori de proves de dockers: <https://github.com/vnaranj1/vnaranj1xtec>

Per fer funcionar el servei tens dues opcions:

- 1- Crea un fitxer shellscript que creï un contenidor a partir d'alguna d'aquestes dues imatges, arrencant així el servei, obrint els ports apropiats, creant els volums necessaris, etc, perquè funcioni sense problema i el VLC s'hi pugui connectar.
- 2- L'alternativa al shellscript és fer un docker-compose.yaml

Has de posar captures de pantalla de tot el procés, i també que demostrin que funciona.

**Si no hi ha demostració de que funcioni, s'avaluarà l'exercici amb un 0.**

Finalment, has de:

- Crea un repo a github, i inclou-hi els teus fitxers: shellscript i / o docker-compose.yaml. No oblidis que has d'incloure un README.md. El nom del repo de github serà:  
`elteunodusuari/ubuntu-dlna`

### **Lliurament:**

Un document pdf amb totes les respostes i les passes fetes per a cada exercici, amb les corresponents captures de pantalla d'ajuda a les explicacions.

**També cal lliurar, aquí en el pdf, els enllaços als repositoris de dockerhub i de github.**

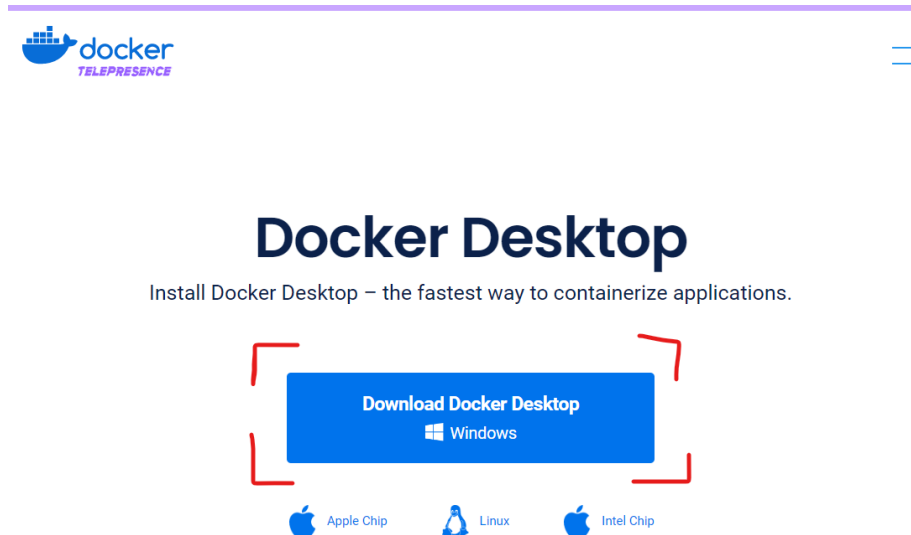
Patró que ha de seguir el nom del document:

M8-UF4-EP-NomCognom.odt

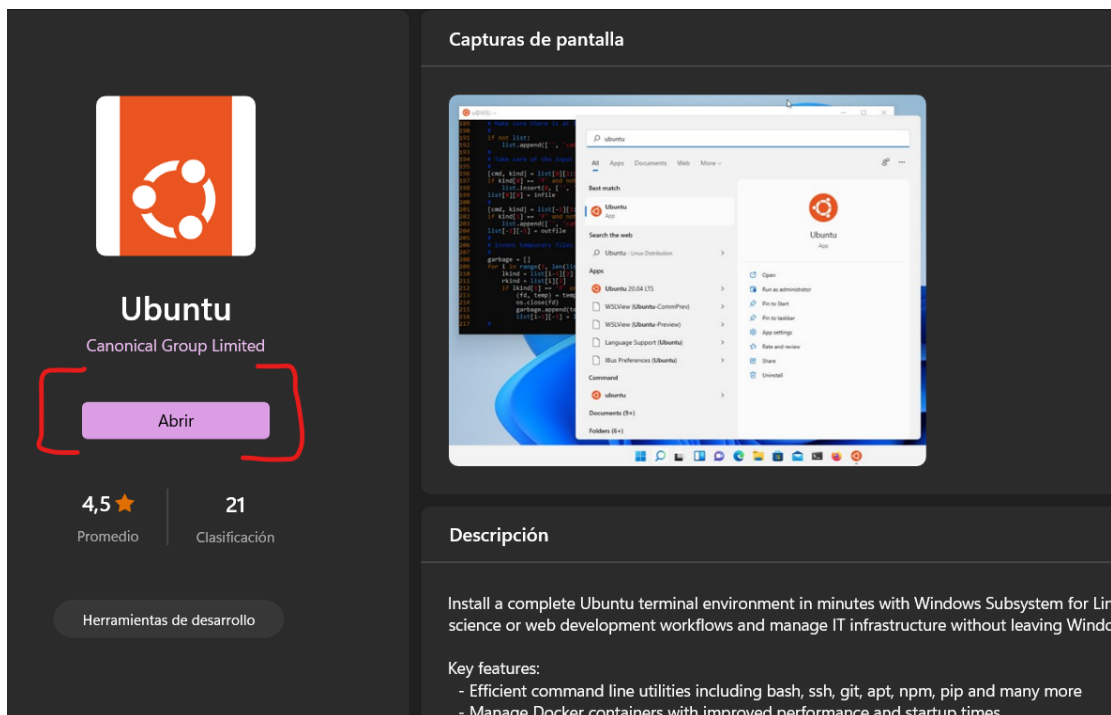
**Atenció: si el nom del fitxer no compleix aquest patró  
NO ES CORREGIRÀ LA PRÀCTICA.**

**R:** Aquí mostraré todo el procedimiento que he seguido para lograr nuestro objetivo de la práctica:

**Paso 1: Instalación de Docker Desktop:**



**Paso 2: Docker Desktop solicitará que tengamos instalado Ubuntu en nuestro Windows para funcionar correctamente. Desde aquí podremos hacer los pulls necesarios:**





```
poljimenez@DESKTOP-BI5Q0U:~$
```

Paso 3: Aquí es donde ejecutaremos el script que he elaborado para que haga todas las funciones que necesitamos:

```
poljimenez@DESKTOP-BI5Q0U:~$ cat dockerfile.sh
#!/bin/bash

# Comprobar si la imagen Owncast existe localmente
if [[ "$(docker images -q owncast:latest 2> /dev/null)" == "" ]]; then
    echo "La imagen Owncast no se encontró localmente. Descargando..."
    docker pull owncast/owncast:latest
fi

# Detener y eliminar cualquier contenedor Owncast existente
docker stop owncast-container 2> /dev/null
docker rm owncast-container 2> /dev/null

# Crear volúmenes para almacenar los datos persistentes del contenedor
docker volume create owncast-data

# Iniciar el contenedor Owncast
docker run -d -p 8080:8080 -p 1935:1935 \
    -v owncast-data:/app/data \
    -v $(pwd)/owncast-videos:/app/videos \
    --name owncast-container \
    owncast/owncast:latest
poljimenez@DESKTOP-BI5Q0U:~$
```



Este código realiza las siguientes acciones:

1- Verifica si la imagen de Docker "owncast:latest" existe localmente. Si no se encuentra, se descarga la imagen desde el repositorio de Docker Hub "owncast/owncast:latest".

2- Detiene y elimina cualquier contenedor Docker existente llamado "owncast-container" para evitar conflictos o duplicados.

3- Crea un volumen de Docker llamado "owncast-data" para almacenar los datos persistentes del contenedor.

4- Inicia un nuevo contenedor Docker utilizando la imagen "owncast/owncast:latest" y realiza las siguientes configuraciones:

5- Asigna los puertos de host 8080 y 1935 al contenedor para la comunicación de la aplicación Owncast.

6- Mapea el volumen "owncast-data" del host al directorio "/app/data" dentro del contenedor para almacenar los datos persistentes.

7- Mapea el directorio "owncast-videos" actual del host al directorio "/app/videos" dentro del contenedor para almacenar los videos generados por Owncast.

8- Asigna el nombre "owncast-container" al contenedor para facilitar su identificación y gestión.

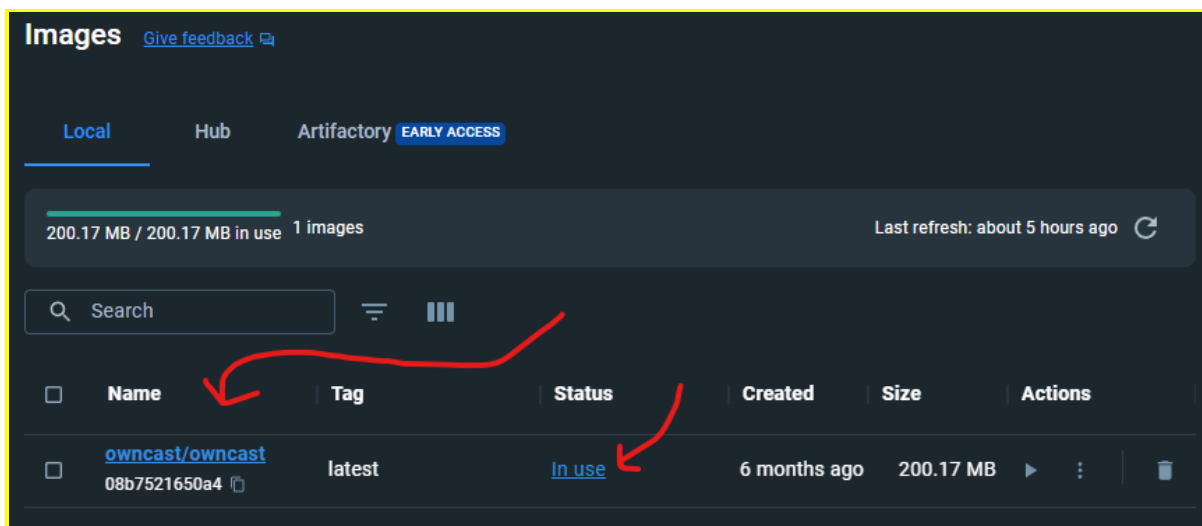
En resumen, el código comprueba y descarga la imagen de Owncast si no existe, detiene y elimina cualquier contenedor existente, crea volúmenes para almacenar datos persistentes y finalmente inicia un nuevo contenedor con la imagen de Owncast, configurando los puertos y los volúmenes necesarios.



Cuando ejecutemos el código podremos ver que todo ha ido correctamente y revisaremos que todo se haya montado correctamente:

```
poljimenez@DESKTOP-BI5Q0U:~$ ./dockerfile.sh
La imagen Owncast no se encontró localmente. Descargando...
latest: Pulling from owncast/owncast
c158987b0551: Pull complete
d9f960589d63: Pull complete
a9c9464a098a: Pull complete
f152c1b1128d: Pull complete
64e989d48323: Pull complete
Digest: sha256:bc872b31f63a901c637bc6cba396960b33658ca0e1db924ef90cf9e135a31e4b
Status: Downloaded newer image for owncast/owncast:latest
docker.io/owncast/owncast:latest
owncast-data
bb0506b399bf40378649a6da2c2817245b0b233e2bd1f8f6aadf09f9a1cedf3d
poljimenez@DESKTOP-BI5Q0U:~$
```

En el Docker Desktop comprobaremos si la imagen ha sido correctamente descargada y que el volumen se haya creado correctamente:





**Volumes** [Give feedback](#)

Create

Search

<input type="checkbox"/>	Name	Status	Created	Size	Actions
<input type="checkbox"/>	<a href="#">owncast-data</a>	in use	3 minutes ago	2.5 MB	

Paso 6: Comprobaremos si el contenedor está en funcionamiento:

**Containers** [Give feedback](#)

Search

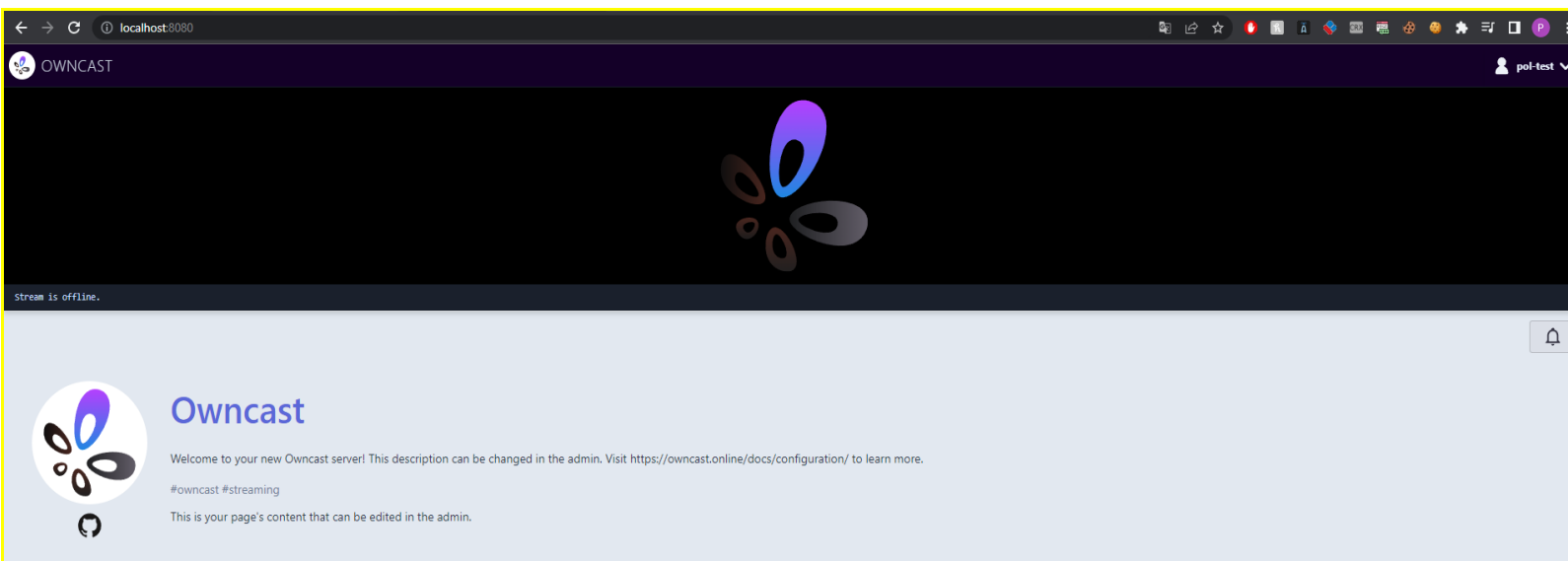
☐ Only show running containers

<input type="checkbox"/>	Name	Image	Status	Port(s)	Last started	Actions
<input type="checkbox"/>	<a href="#">owncast-container</a> bb0506b399bf	<a href="#">owncast/owncast:latest</a>	Running	<a href="#">1935:1935</a> <a href="#">8080:8080</a> <a href="#">Show less</a>	4 minutes ago	

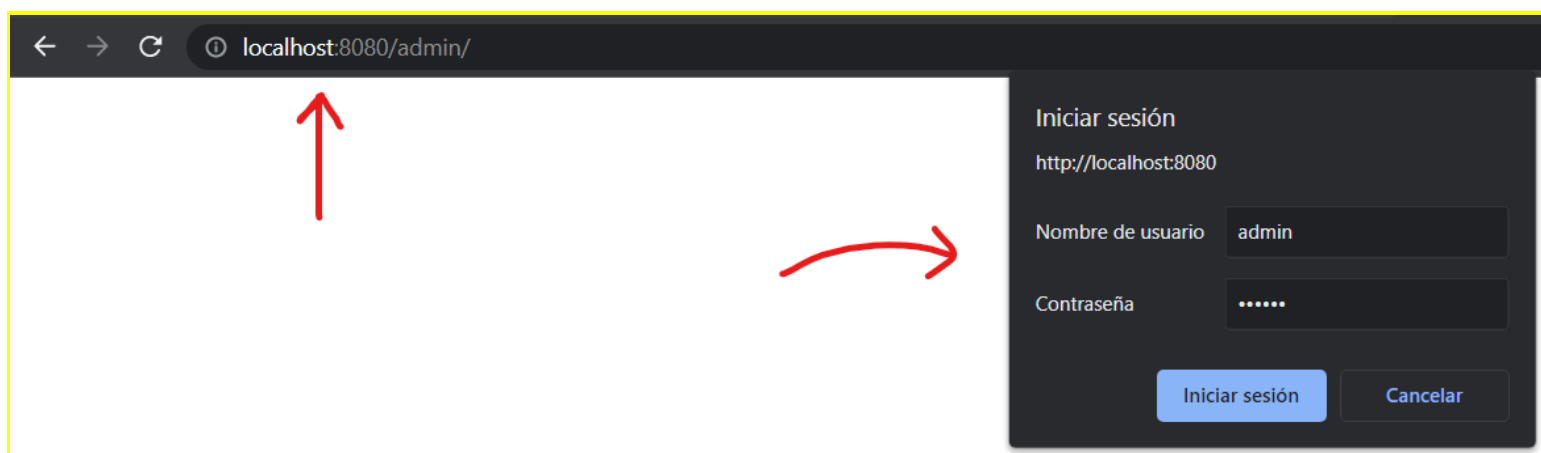
```
poljimenez@DESKTOP-BI5Q0UU:~$ docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS                               NAMES
bb0506b399bf   owncast/owncast:latest   "/app/owncast"         6 minutes ago   Up 6 minutes   0.0.0.0:1935->1935/tcp, 0.0.0.0:8080->8080/tcp   owncast-container
poljimenez@DESKTOP-BI5Q0UU:~$
```



Paso 7: Accederemos a Owncast desde nuestro navegador (<http://localhost:8080>)



Paso 8: Para acceder a la configuración de Owncast introduciremos la siguiente URL y las credenciales correspondientes por defecto:

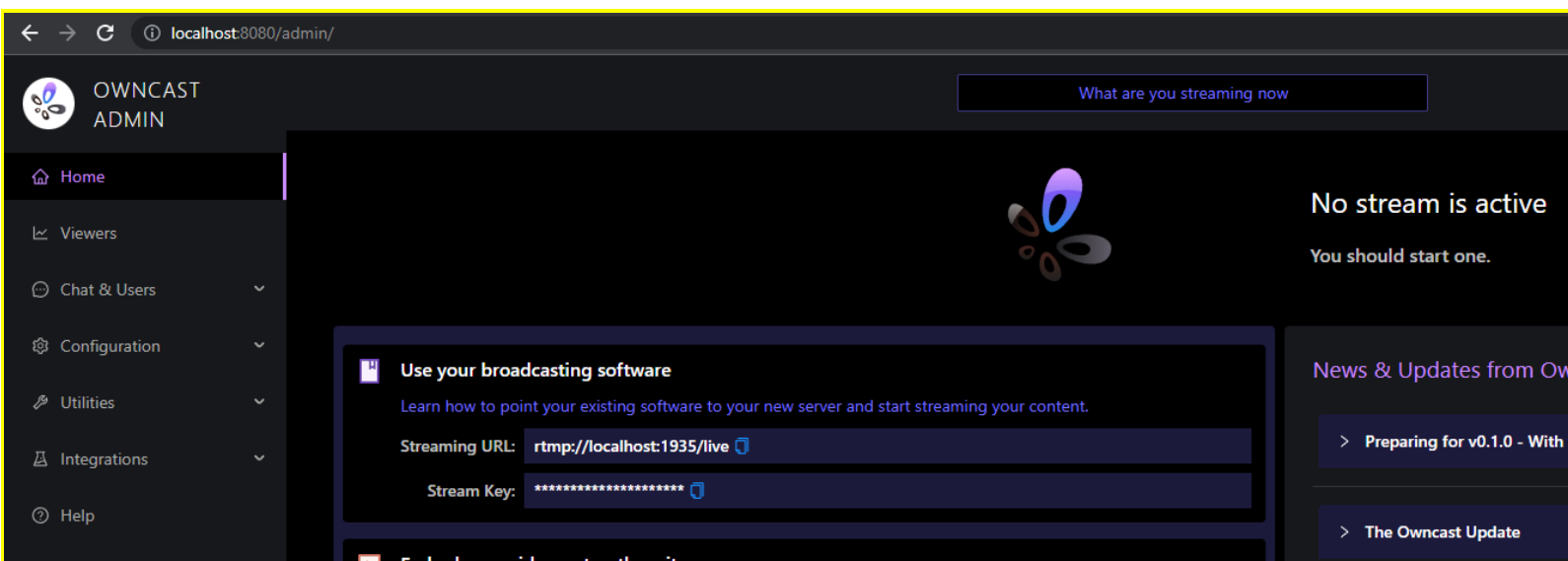


user: admin

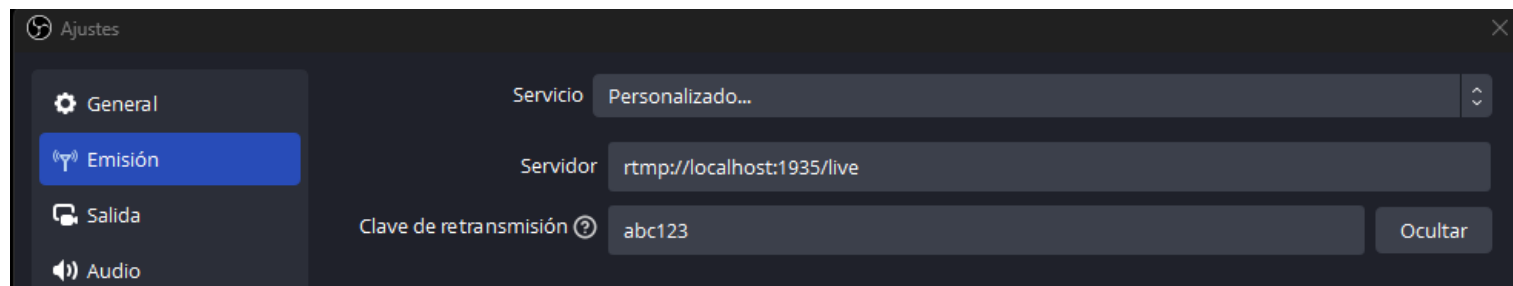
password: abc123



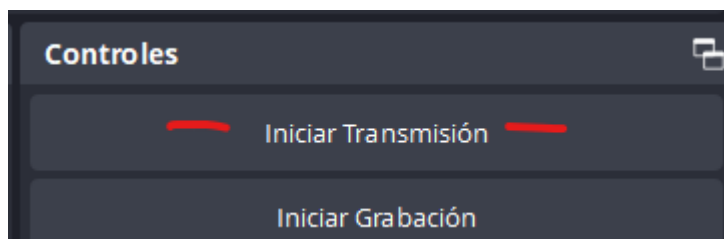
Paso 9: Y se nos mostrará la siguiente interfaz:



Paso 10: Utilizaremos el software OBS Studio para iniciar la transmisión. Iremos a los ajustes y en la sección de emisión colocaremos el Streaming URL en la sección de Servidor y la correspondiente clave. Aplicaremos los cambios.



Y iniciaremos la transmisión:

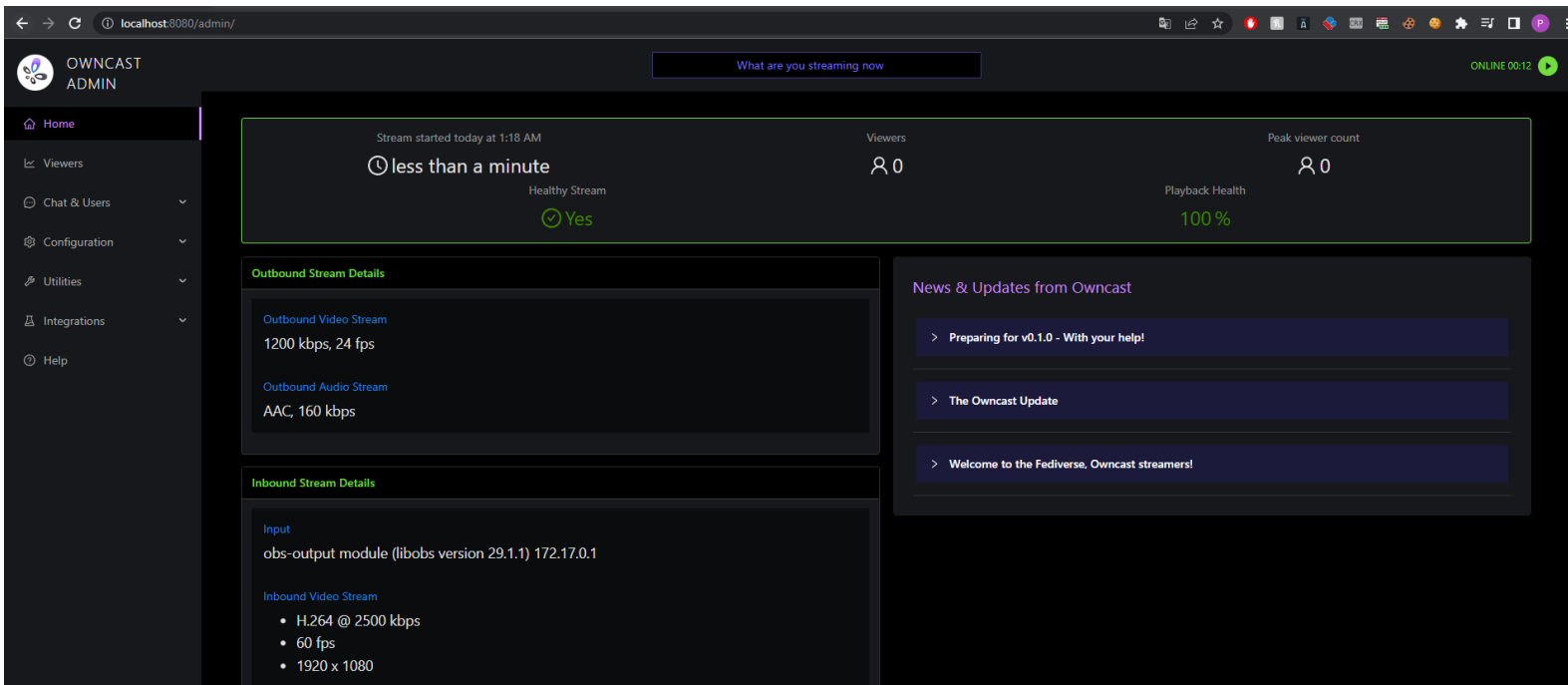




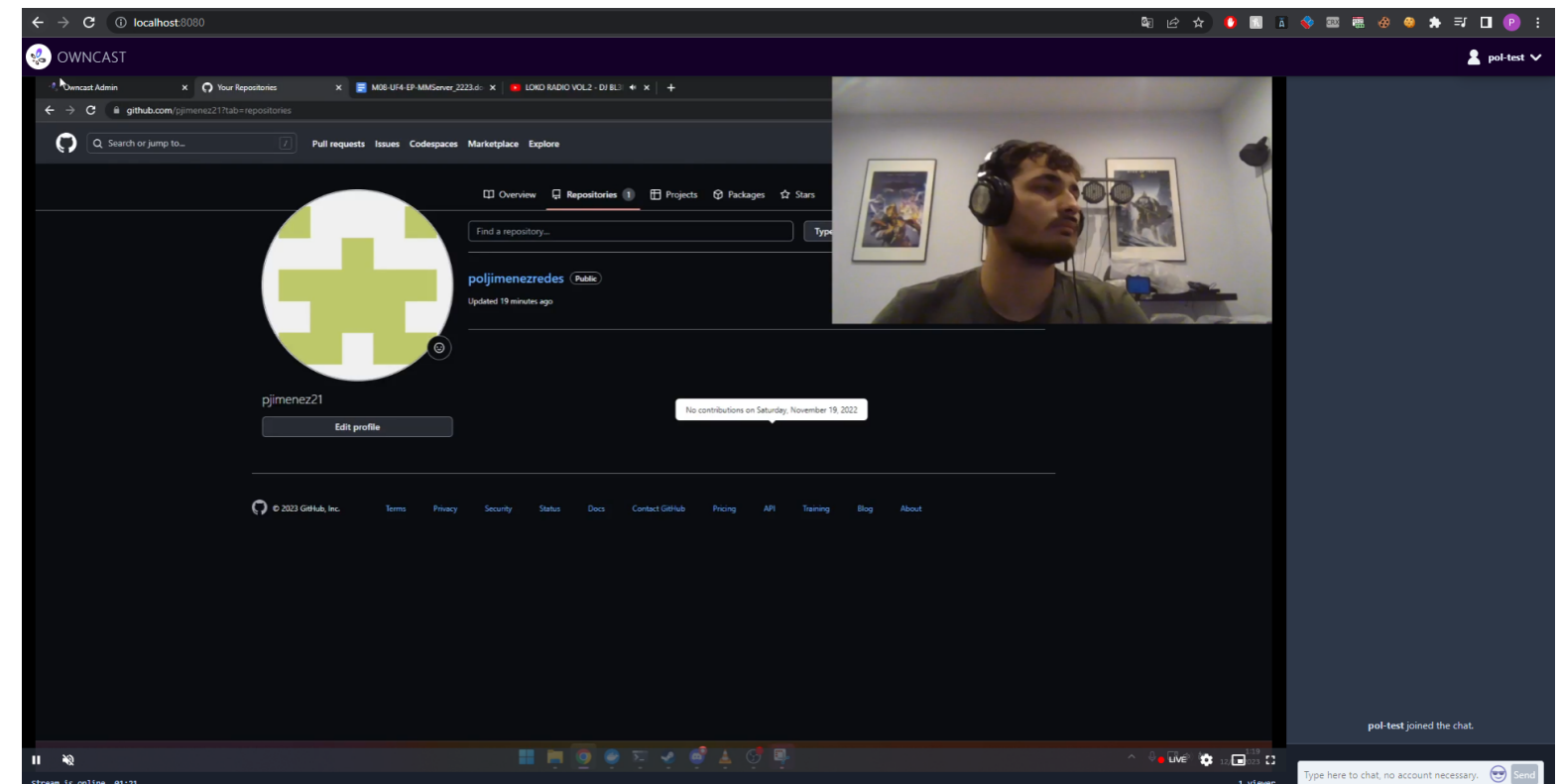


Paso 11: Podremos ver el funcionamiento:

(En el panel de admin) (<http://localhost:8080/admin/>)

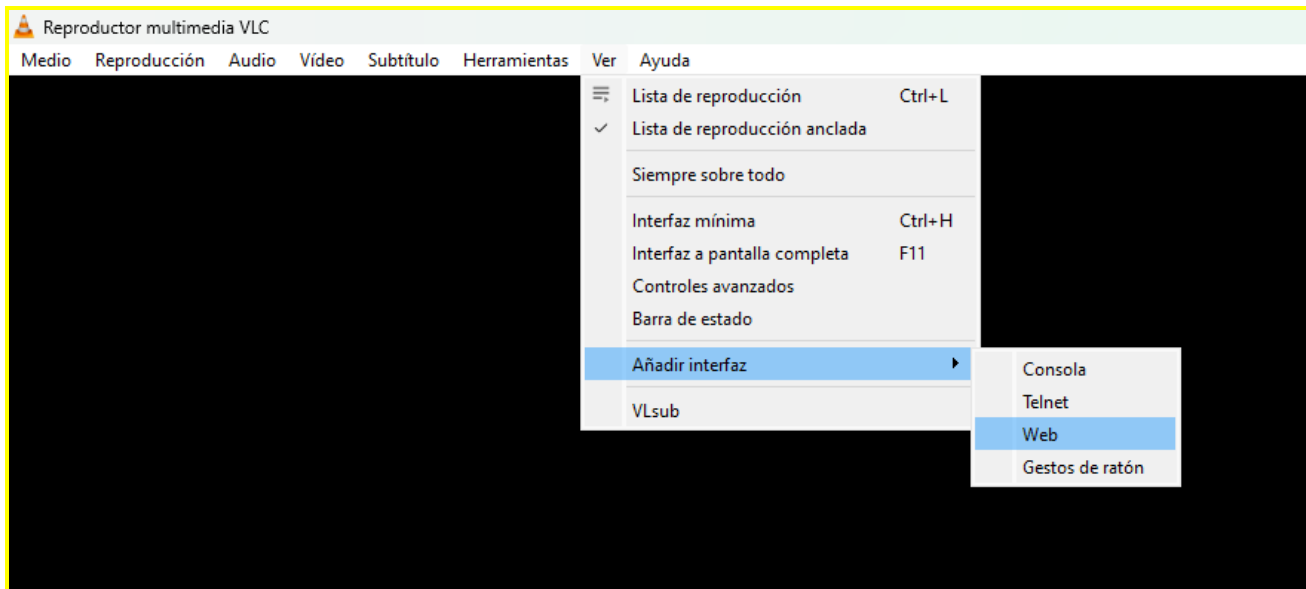


(En el panel de directo Owncast) (<http://localhost:8080/>)

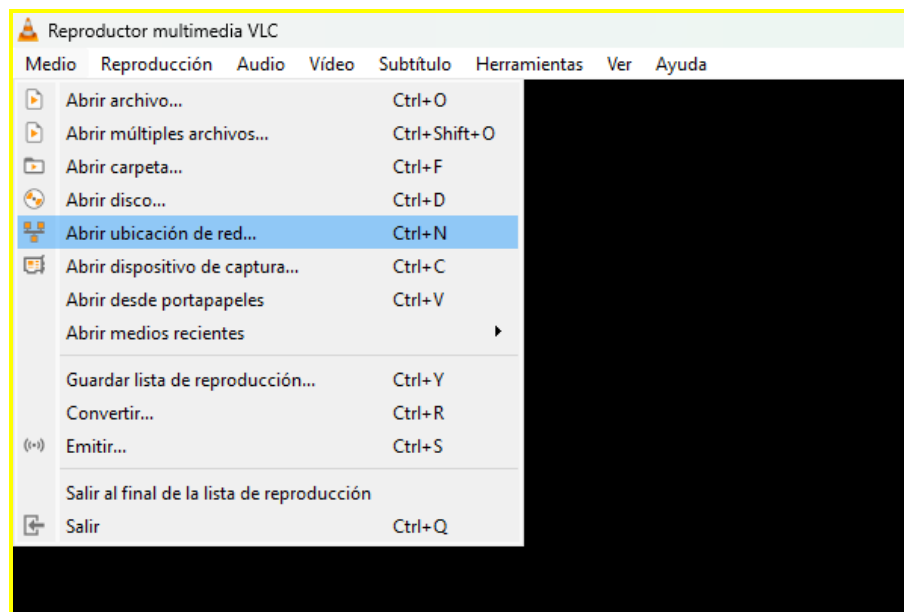


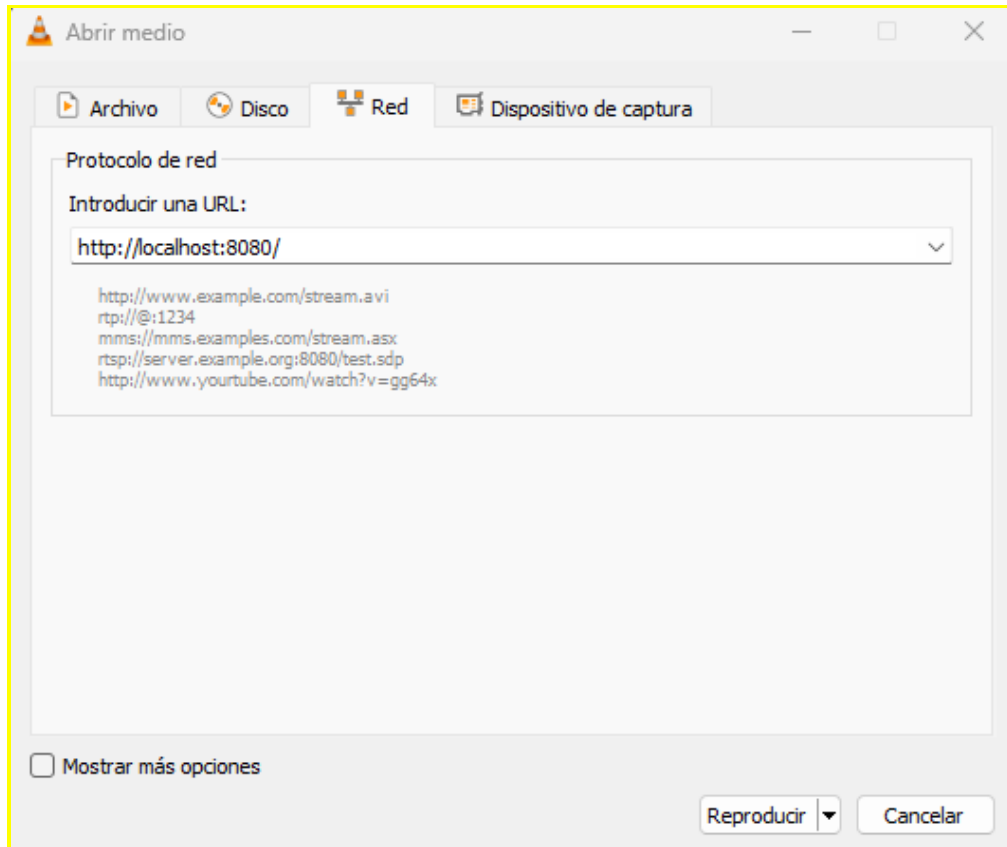


En VLC: Añadiremos la interfaz Web para posteriormente acceder al streaming.

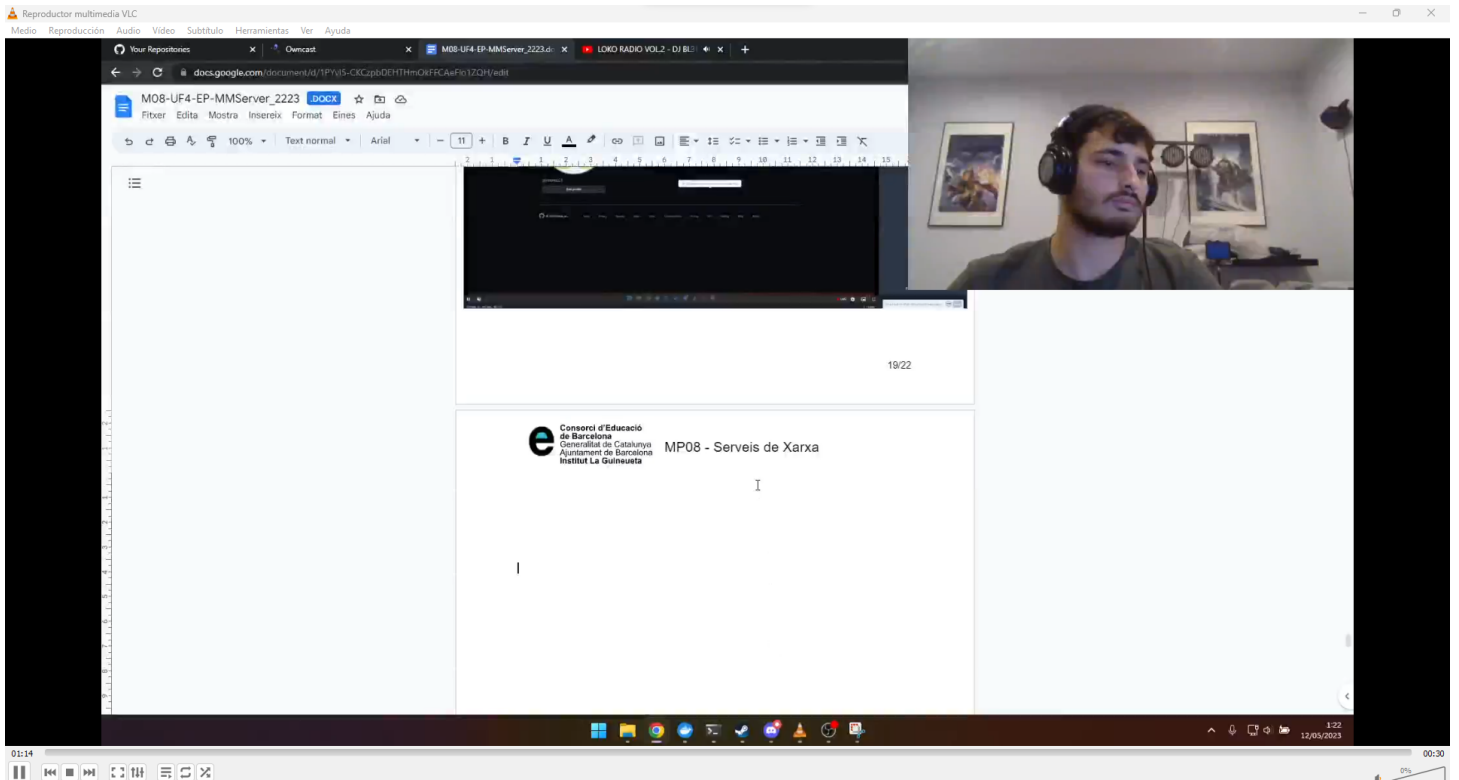


Haremos click en “Medio” en la barra de menú y seleccionaremos “Abrir ubicación de red”:





Y podremos ver el streaming:





**Data límit de lliurament:**

divendres 12 de Juny de 2023 a les 23:55 h. **Atenció: no es pot ajornar !!**

**Correcció:**

Que la resposta sigui correcta: 40%

Que la resposta sigui completa: 40%

Que el document lliurat estigui ben presentat: 20%

**IMPORTANT:**

- **Si no hi ha demostració de que funcioni, s'avaluarà l'exercici amb un 0.**
- **Si no s'inclouen els enllaços (al document a lliurar) dels repositoris de dockerhub i github quan calgui, l'exercici s'avaluarà amb un 0.**
- **Els exercicis de teoria, s'avaluaran amb un 0 si no es posen exemples, o si les explicacions no són vostres (copiades d'Internet).**

**ATENCIÓ: CAL APROVAR TANT LA PART 1 COM LA PART 2 PER A APROVAR AQUESTA PRÀCTICA**