

1. INTRODUÇÃO

Nesta segunda etapa do desenvolvimento o foco foi a compreensão e implementação do sistema de câmeras que serão responsáveis por monitorar os jogadores na sala. Foi utilizada a Raspberry Pi 4 cedida por Calebe para conseguir fazer o controle e transmissão das câmeras, o modelo em questão possui 8GB de RAM, se trata de uma Raspberry Pi 4 Model B Rev 1.4, com arquitetura ARM, a seguir a explicação de como foi feito.

2. SISTEMA OPERACIONAL

O sistema que estava disponível na placa não era próprio para a aplicação, então, com a permissão do professor, o cartão da placa foi devidamente formatado e nele foi instalado o sistema *Raspberry Pi OS Lite (64-bits)*, sistema esse baseado em Debian. A instalação foi feita através do software oficial da empresa e nele foram feitas algumas configurações como a ativação do SSH e criação do perfil (Usuário: pi; Senha: eroom).

3. FERRAMENTAS

Assim que conectado com o terminal da placa via SSH, foram necessárias as instalações de algumas ferramentas para o funcionamento ideal do sistema:

- ffmpeg: compressão e streaming de vídeo;
- v4l-utils: verificar dispositivos de vídeo;
- git: instalar mjpg-streamer do repositório;

4. PASSO A PASSO

Conectar a raspberry a uma alimentação de 5V, cabo de rede e Webcams que serão usadas, com a placa energizada e conectada na rede, acessamos seu terminal via SSH por algum computador, no meu caso, o acesso foi feito através de um terminal Linux Ubuntu por uma WSL. Utilizando o comando “*ssh <usuário>@<IP da placa>*” a conexão é estabelecida e o terminal pede a senha para permitir o acesso, a seguir faço a atualização dos pacotes e faço instalação as ferramentas:

- `sudo apt install ffmpeg v4l-utils git -y`
- `sudo apt install cmake libjpeg62-turbo-dev build-essential -y`

Clonar e compilar o mjpg-streamer:

- `git clone https://github.com/jacksonliam/mjpg-streamer.git`
- `cd mjpg-streamer/mjpg-streamer-experimental`
- `make clean`
- `make`
- `sudo make install`

Depois disso, teremos tudo para:

- verificar se a câmera está funcionando (`v4l2-ctl --list-devices`),
- testar a captura (`ffmpeg -f v4l2 -i /dev/video0 out.mp4`),
- transmitir via rede (`mjpg-streamer` ou até `ffmpeg` com RTSP/HTTP).

Com tudo feito, vamos iniciar a transmissão, com o comando “`v4l2-ctl --list-devices`” conseguimos ver toda a lista de dispositivos de vídeo, nessa lista os dispositivos estarão separados por categorias, para uma webcam USB o que aparece é o seguinte:

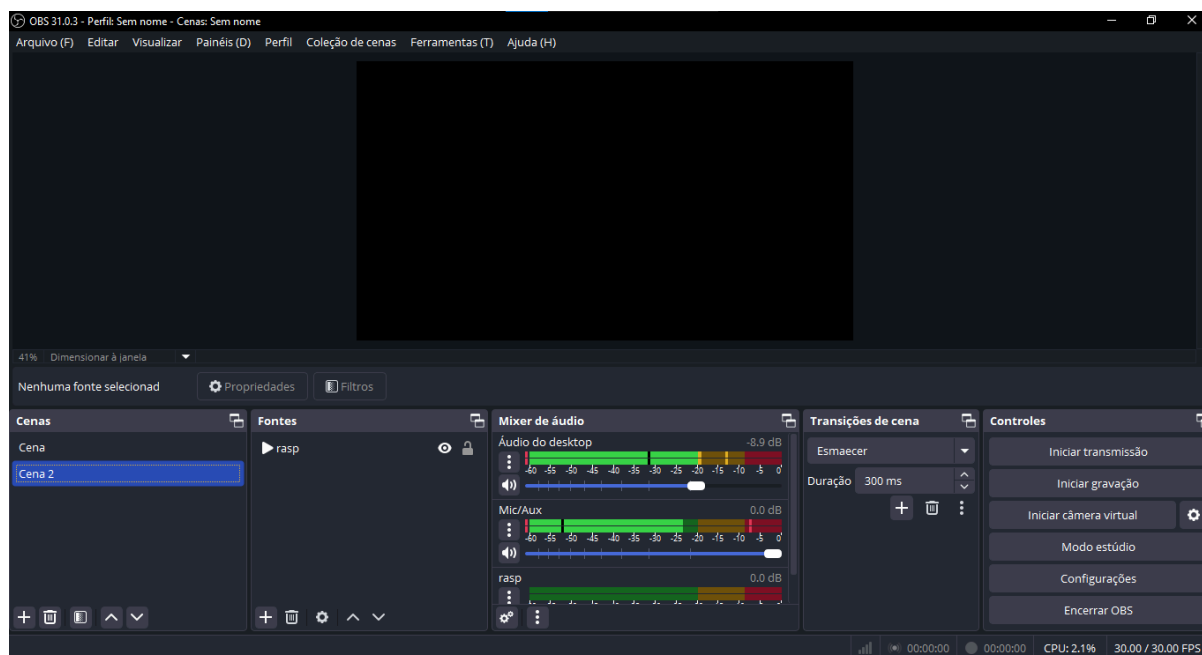
```
USB Composite Device: DV20 USB (usb-0000:01:00.0-1.3):
    /dev/video0
    /dev/video1
    /dev/media4
```

Como queremos a imagem, o que precisamos é desse `/dev/video*`, nesse caso, o print foi feito com apenas uma câmera conectada, mas ainda assim aparece mais de um `/dev/video`, as possíveis causas são as seguintes:

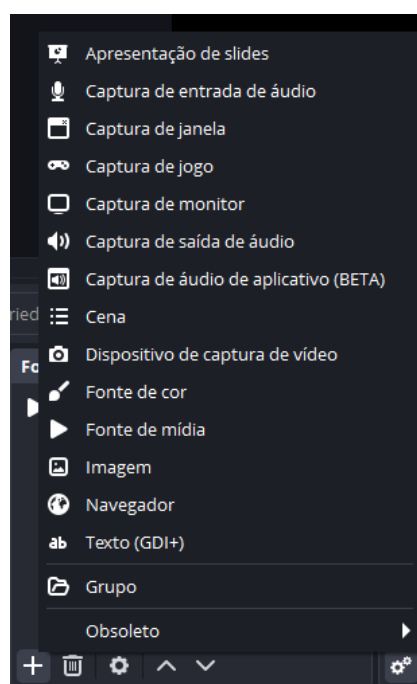
- Algumas câmeras USB criam um `/dev/video0` para o fluxo de vídeo principal e outro `/dev/video1` para metadados ou stream em resolução diferente.
- Dispositivos com microfone embutido podem gerar nós extras.
- Em webcams mais modernas, um nó pode ser só para captura estática (fotos) e outro para streaming de vídeo.

Basta testar ambos no momento da transmissão para descobrir o ideal para ser usado, para isso, precisamos agora iniciar a transmissão com o `mjpg-streamer`, usamos o comando “`cd ~/mjpg-streamer/mjpg-streamer-experimental`” para entrarmos no diretório e executamos “`./mjpg_streamer -i ./input_uvc.so -d /dev/video0 -r 1920x1080 -f 30 -o ./output_http.so -w ./www -p 8080`”. Com esse comando definimos qual o dispositivo de vídeo será transmitido (`/dev/video0`), a resolução (1920x1080), a taxa de atualização dos quadros (30fps) e a porta usada (8080).

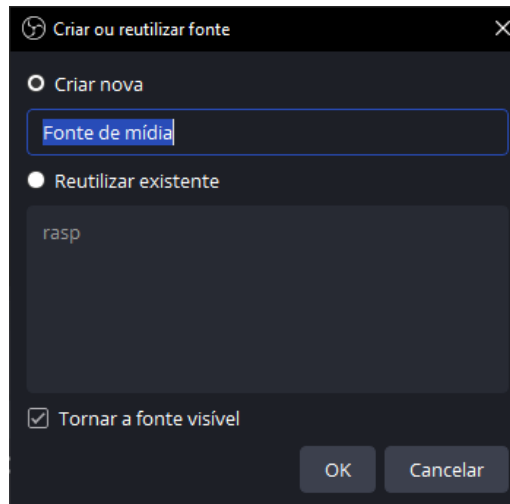
Feita a transmissão, precisamos acessar ela agora, podemos vê-la pelo navegador utilizando o comando “`http://<IP-da-Raspberry>:8080/?action=stream`”, outra opção viável é o uso do aplicativo OBS, uma plataforma open source para gravação e transmissão de vídeo, por ele podemos organizar a tela de forma que possamos ver as múltiplas câmeras. Tela inicial OBS:



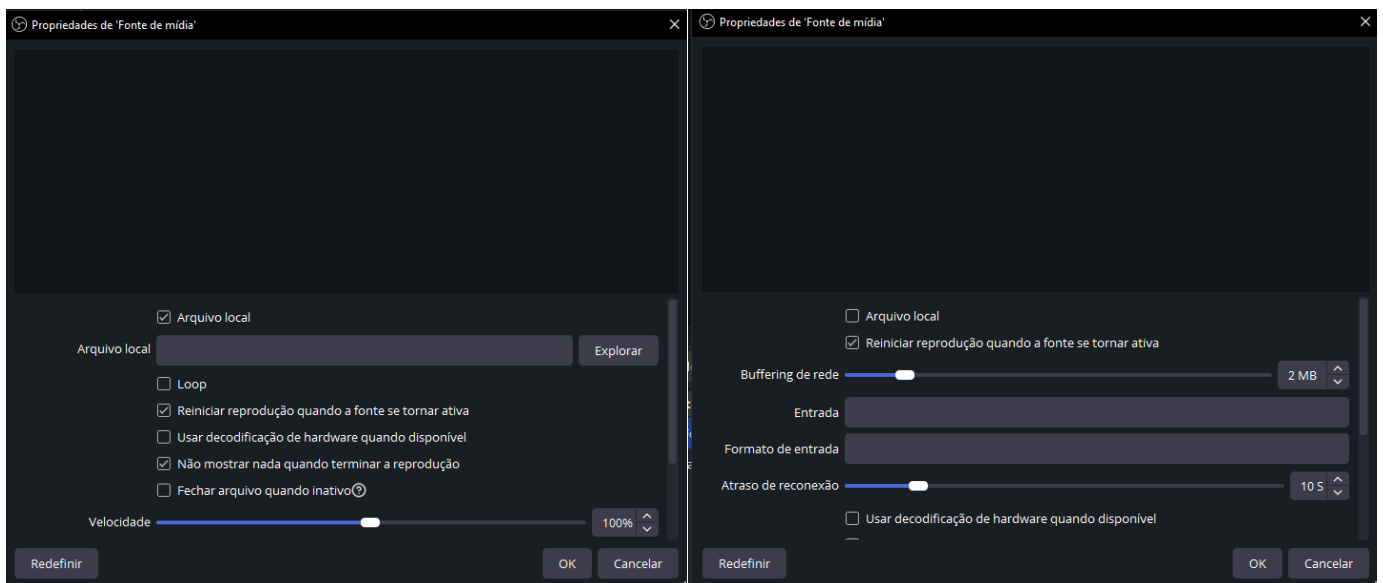
Com ele aberto, precisamos apenas criar uma Cena, nomear ela, e com ela selecionada, adicionamos uma fonte, ao clicar para adicionar uma nova fonte irá aparecer uma série de opções:



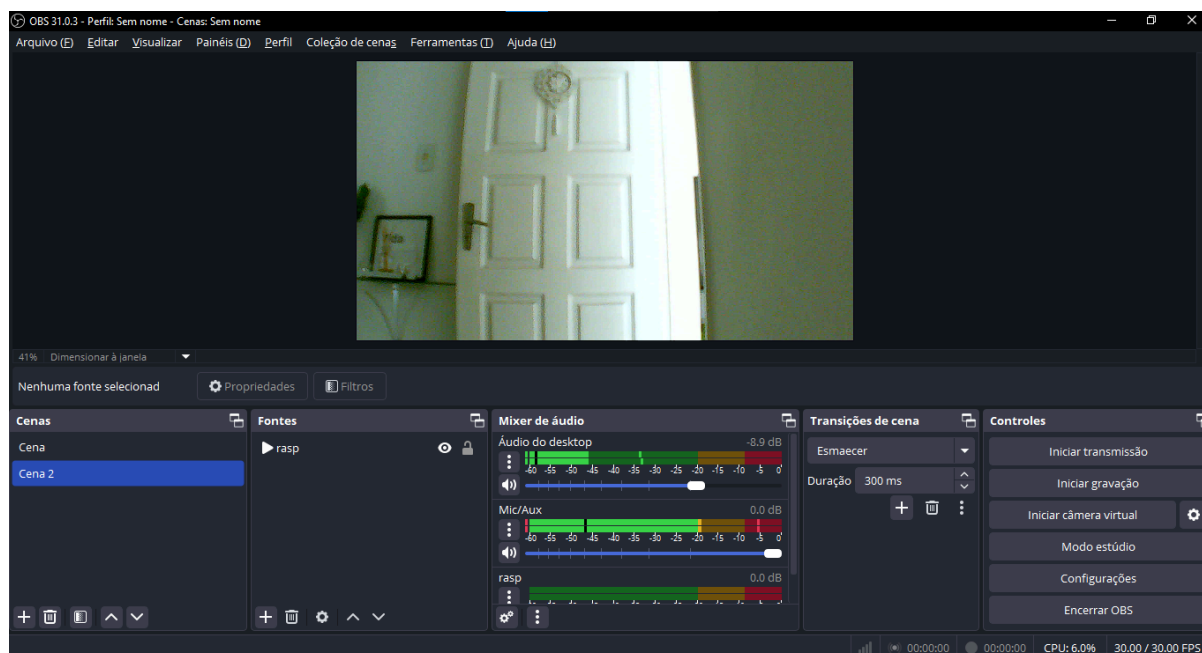
A correta nesse caso seria a de “Fonte de mídia”



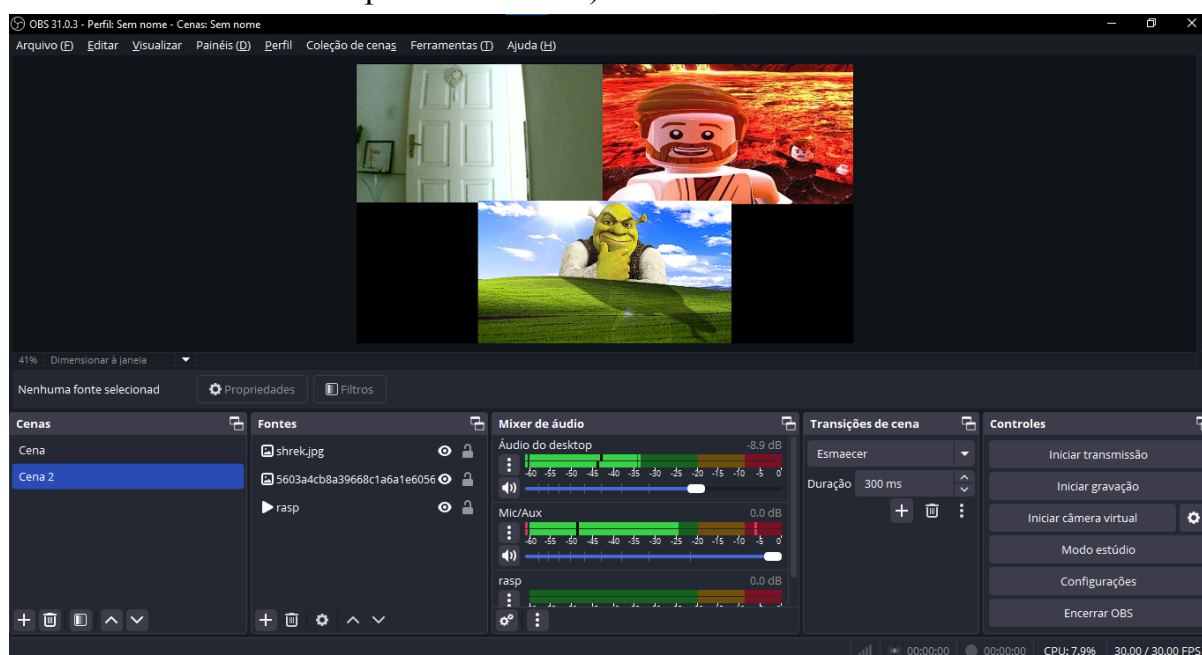
Basta escolher um nome para a fonte e confirmar, na tela a seguir desmarcamos a opção de arquivo local e novas opções irão surgir



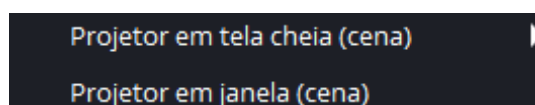
Aqui é necessário apenas alterar o que está em entrada e por o mesmo link usado para acessar a câmera no navegador, “*http://<IP-da-Raspberry>:8080/?action=stream*”, clica em “ok” e a imagem já estará sendo mostrada



Com as câmeras adicionadas, podemos reorganizar a maneira que elas aparecem na tela para facilitar o monitoramento (imagens meramente ilustrativa pois não tenho outras câmeras para demonstrar)



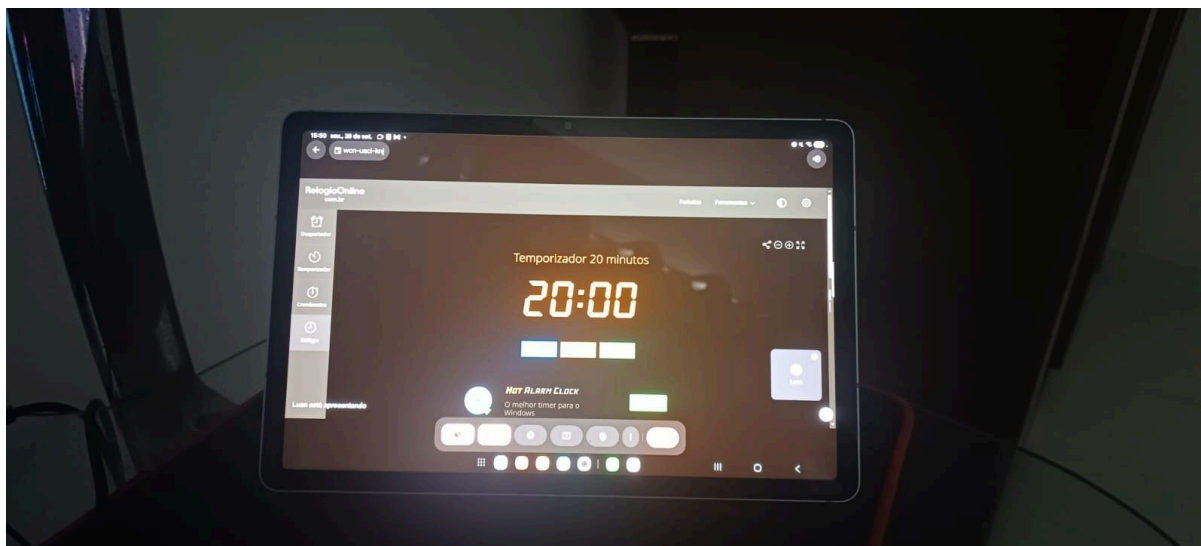
Com tudo devidamente configurado, podemos clicar com o botão direito na cena que estamos usando, no meu caso a “Cena 2” e selecionar alguma das opções a seguir:



Como elas mostram, a segunda cria uma janela com a cena, então o monitor pode redimensionar ela à vontade, e na primeira, podemos escolher um monitor e nele aparecerá a cena em tela cheia.

5. CAPTURA DE ÁUDIO E VISUALIZAÇÃO DO TEMPO

Como foi acordado com Calebe, essas duas partes poderiam ser feitas utilizando um tablet em transmissão em alguma plataforma de videoconferência (meet, zoom, discord) por onde tanto os monitores quanto os jogadores poderão se comunicar ligando o microfone quando necessário, e para a visualização do tempo, seria transmitido na chamada um cronômetro de algum site (Ex: <https://relogioonline.com.br/cronometro/>), com isso, os monitores podem controlar o tempo no início, durante e final do desafio. Exemplo abaixo:



6. CONSIDERAÇÕES FINAIS

Todas as partes desenvolvidas por mim parecem funcionar devidamente dentro dos testes que foram feitos, um potencial problema para a transmissão das imagens pode ser a qualidade da rede que a placa estará, quando ela esteve conectada via cabo LAN no roteador, não demonstrou travamentos nem atraso na transmissão, porém, quando usado o Wifi, foi percebido um atraso de aproximadamente 2,5 segundos e pequenos travamentos, pode ser relevante a disponibilidade de um cabo de rede para garantir maior estabilidade.