

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
PROGRAMA DE PÓS-GRADUAÇÃO  
DEPARTAMENTO ACADÊMICO DE ELETRÔNICA  
ESPECIALIZAÇÃO EM TECNOLOGIA E SOFTWARE LIVRE

ERIC HENRIQUE GARCIA MAZURKIEVIZ

**BABÁ ELETRÔNICA COM SOFTWARE LIVRE**

MONOGRAFIA DE ESPECIALIZAÇÃO

CURITIBA  
2017

ERIC HENRIQUE GARCIA MAZURKIEVIZ

## **BABÁ ELETRÔNICA COM SOFTWARE LIVRE**

Monografia de Especialização, apresentado ao Curso de Especialização em Tecnologia e Software Livre, do Departamento Acadêmico de Eletrônica, da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Especialista.

Orientador: Prof. Msc. Christian C. S. Mendes

CURITIBA  
2017



## TERMO DE APROVAÇÃO

### BABÁ ELETRÔNICA COM SOFTWARE LIVRE

por

**Eric Henrique Garcia Mazurkieviz**

Esta monografia foi apresentada às 19 horas do dia 11 de maio de 2017 como requisito parcial para a obtenção do título de ESPECIALISTA EM TECNOLOGIA E SOFTWARE LIVRE, do Programa de Pós-Graduação da Universidade Tecnológica Federal do Paraná. O candidato foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após a deliberação, a Banca Examinadora considerou o trabalho aprovado.

---

Msc Christian Carlos Souza Mendes  
UTFPR

---

Msc Leandro Batista de Almeida  
UTFPR

---

Msc Fabiano Kuss  
SERPRO

---

Prof. Msc Lincoln Herbert Teixeira

## **AGRADECIMENTOS**

Ao Professor Lincoln e meu Orientador Christian, pela atenção e imensa paciência ao desenvolvimento desse trabalho.

Aos meus pais, Fátima e Cezar, e minha irmã, Nataly, que me apoiaram e aconselharam a persistir.

A minha querida esposa Vanessa, que não me deixou desanimar e por varias vezes compreendeu minha ausência.

A minha amada filha Alice, que pelo amor incondicional me incentivou a fazer esse trabalho.

## RESUMO

MAZURKIEVIZ, Eric Henrique Garcia. **Babá Eletrônica com Software Livre**. 2017. 41 páginas f. Monografia (Curso de Especialização em Tecnologia e Software Livre), Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná. Curitiba, 2017.

A rotina dividida entre trabalho doméstico e cuidada de bebês e crianças em lares brasileiros é muito comum, e muitas vezes por um minuto de desatenção, o pior pode ocorrer, aumentando as estatísticas de acidentes domésticos com eles. Este trabalho apresenta a importância de monitorar as crianças quando os pais não estão presentes no mesmo lugar, fazendo uso de dispositivos eletrônicos chamados Babas-eletrônicas. O objetivo é apresentar um protótipo funcional de baixo custo, desenvolvido com ferramentas gratuitas e de código aberto, com características e desempenho semelhantes aos modelos comerciais encontrados no varejo.

**Palavras chave:** Babá Eletrônica. Raspberry Pi. Software Livre.

## **ABSTRACT**

MAZURKIEVIZ, Eric Henrique Garcia. **Baby Monitor with Free Software**. 2017. 41 pages f. Monograph (Specialization in Technology and Free Software), Academic Department of Electronic, Federal Technological University of Paraná. Curitiba, 2017.

The routine of domestic working and care of babies and children in Brazilian homes is very common, and often for one minute of inattention, the worst can occur, increasing the statistics of domestic accidents with them. This work presents the importance of monitoring children when parents are not present in the same place of them, making use of electronic devices called baby monitors. The objective is to present a functional prototype of low cost, developed with free tools and open source, and with features and performance like commercial models found in the retail.

**Keywords:** Baby Monitor. Raspberry Pi. Free Software.

## LISTA DE FIGURAS

FIGURA 1 – MODELO COMERCIAL DE BABÁ ELETRÔNICA. ....	12
FIGURA 2 - TOPOLOGIA DE UM SISTEMA BÁSICO DE MONITORAMENTO COM CÂMERAS IP. ....	15
FIGURA 3 - RASPBERRY PI MODELO B .....	18
FIGURA 4 – TOPOLOGIA DO SISTEMA PROPOSTO PARA O PROTÓTIPO .....	21
FIGURA 5 – ADAPTADOR DE SOM USB E MICROFONE .....	22
FIGURA 6 – ADAPTADOR WIRELESS USB .....	24
FIGURA 7 - RASPBERRY PI CÂMERA .....	27
FIGURA 8 – ACESSO A PÁGINA DO SERVIÇO DE STREAMING .....	29
FIGURA 9 – MENU DE CONFIGURAÇÃO DE VÍDEO .....	31
FIGURA 10 – TESTE DO TEMPO DE RESPOSTA DO VÍDEO .....	32

## LISTA DE TABELAS

TABELA 1 - CRIANÇAS VÍTIMAS DE ACIDENTES, SEGUNDO IDADE E TIPO DE ACIDENTE.....	17
TABELA 2 - CUSTO DO DESENVOLVIMENTO DO PROTÓTIPO .....	33
TABELA 3 - PREÇO DE BABÁS ELETRÔNICAS ENCONTRADOS NO VAREJO .....	33



## LISTA DE ABREVIATURAS, SIGLAS E ACRÔNIMOS

SBC	<i>Single Board Computer</i>
USB	<i>Universal Serial Bus</i>
CPU	<i>Central Processor Unit</i>
GPU	<i>Graphical Processor Unit</i>
SoC	<i>System-on-Chip</i>
HDMI	<i>High-Definition Multimedia Interface</i>
QoS	<i>Quality of Service</i>

# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>11</b>
1.1	PROBLEMA .....	12
1.2	OBJETIVOS.....	13
1.2.1	OBJETIVO GERAL .....	13
1.2.2	OBJETIVOS ESPECÍFICOS .....	13
1.3	JUSTIFICATIVA.....	14
1.4	ESTRUTURA DO TRABALHO .....	14
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>15</b>
2.1	SISTEMAS DE MONITORAMENTO E VIGILÂNCIA .....	15
2.2	ACIDENTES DOMÉSTICOS ENTRE CRIANÇAS.....	16
2.3	RASPBERRY PI.....	17
2.4	O SISTEMA OPERACIONAL LINUX.....	19
<b>3</b>	<b>DESENVOLVIMENTO DO PROTÓTIPO .....</b>	<b>21</b>
3.1	TOPOLOGIA PROPOSTA PARA O PROJETO .....	21
3.2	CONFIGURAÇÃO DO ADAPTADOR DE SOM USB COM MICROFONE .....	22
3.3	CONFIGURANDO A RASPBERRY PI COMO UM ROTEADOR .....	24
3.4	CONFIGURANDO O STREAMING DE AUDIO E VIDEO .....	27
<b>4</b>	<b>RESULTADOS E CONSIDERAÇÕES FINAIS.....</b>	<b>30</b>
4.1	RESULTADOS OBTIDOS COM O STREAMING DE ÁUDIO E VÍDEO .....	30
4.2	CUSTOS DO DESENVOLVIMENTO DO PROTÓTIPO .....	32
4.3	CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS .....	33
	<b>REFERÊNCIAS.....</b>	<b>35</b>
	<b>ANEXOS .....</b>	<b>37</b>

# 1 INTRODUÇÃO

Os primeiros meses de vida de um recém-nascido são os que mais requerem cuidados por parte dos pais.

Um estudo aponta que, apesar de as causas da mortalidade infantil no Brasil terem mudado nas últimas décadas, os níveis de mortalidade em crianças menores de um ano permanecem elevados quando comparados com os níveis de outros países com semelhantes índices de desenvolvimento econômico (BRASIL, 2009; SILVA et al., 2012).

Uma das situações mais preocupantes é a Síndrome da Morte súbita do Lactante (SMLS), definida como a morte de crianças menores de um ano de idade, de forma inesperada, que permanece sem explicação mesmo após extensa investigação (histórica clínica, revisão completa do local do óbito, cardiopatia, maus tratos, sufocação, distúrbios metabólicos e outros) e exame de necropsia, que não conseguem apontar a causa e afirmar o motivo da morte prematura (NUNES et al., 2001; FERNANDES et al., 2012).

Como a maioria das mortes ocorre no local em que o bebê está dormindo (cama, berço ou carrinho), a síndrome recebeu o nome de “morte no berço”.

Em países desenvolvidos, é a causa mais comum de mortalidade nos lactentes, sendo entre 2 a 5 meses de vida a faixa etária de maior risco (NUNES et al., 2001; FERNANDES et al., 2012).

Para que seja possível evitar tal situação entre outras, grande parte dos pais passa adotar formas de monitorar seus filhos, durante o sono ou mesmo quando não estão tão próximos do cômodo onde a se encontram, e isso geralmente ocorre com a utilização de dispositivos eletrônicos chamados de “Babás Eletrônicas”.

Esse trabalho tem o objetivo de propor o projeto de um protótipo de “babá eletrônica”, utilizando um computador de baixo custo, em específico uma SBC (*Single Board Computer*) e softwares de código aberto e gratuitos, em conjunto com outros dispositivos comuns do nosso dia-a-dia como aparelhos de celular e computadores pessoais.

## 1.1 PROBLEMA

A escolha e aquisição de uma babá eletrônica não é algo tão fácil, já que com o avanço da tecnologia é possível encontrá-las com as mais diversas tecnologias existentes.

Um sistema básico de uma babá eletrônica é composto por um módulo transmissor e outro receptor, sendo que em alguns modelos é possível transmitir/receber áudio ou vídeo, ou mesmo ambos simultaneamente.

Os modelos mais procurados são os que transmitem o áudio e vídeo, e por sua vez são os com preços mais elevados, por também possuírem outros recursos, como a gravação do vídeo e áudio e sistema de visão noturna com infravermelho, por exemplo, fazendo com que variem de dezenas de reais a centenas reais.

Abaixo, é apresentado um modelo comercial que possui vários recursos dos acima citados.



**Figura 1 – Modelo Comercial de Babá Eletrônica.**  
**Fonte: <https://www.motorolastore.com/baby-monitors>**

A motivação de fazer esse trabalho é mostrar que é possível desenvolver um protótipo com custo bem abaixo dos modelos comerciais disponíveis, e com alguns dos mesmos recursos extras disponibilizados, utilizando apenas programas de código aberto e gratuito.

## 1.2 OBJETIVOS

Nesta seção são apresentados os objetivos gerais e específicos do trabalho, relativos ao problema anteriormente apresentado.

### 1.2.1 OBJETIVO GERAL

Desenvolver um protótipo funcional de baixo custo de uma Babá Eletrônica, utilizando softwares de código aberto e gratuitos.

### 1.2.2 OBJETIVOS ESPECÍFICOS

- Usar uma Raspberry Pi como o computador do projeto no lado do transmissor;
- Definir os outros periféricos a serem utilizados, como adaptadores de Som USB, Câmera, Microfone e um sistema de visão noturna utilizando LED infravermelho;
- Utilizar um aparelho de celular do tipo *Smartphone* com acesso a um *Web Browser* no lado do receptor;
- Configurar um Sistema Operacional a ser utilizado na Raspberry Pi, que será uma distribuição Linux chamada Raspian;
- Transformar a Raspberry Pi em um *Acess Point*, para permitir uma conexão mais simples de fazer a conexão dos dispositivos;
- Testar o desempenho da transmissão do vídeo e do áudio entre os dispositivos;

### 1.3 JUSTIFICATIVA

Visto a importância de monitorar os bebês durante o sono, o uso da babá eletrônica passa ser algo imprescindível nesse período de suas vidas.

A aquisição destes produtos nem sempre é possível ser feita por alguns pais, devido geralmente as condições financeiras dos mesmos e, portanto, o desenvolvimento de um equipamento com características equivalentes e com custo bem inferior, passa ser algo interessante.

O foco do projeto é apresentar um protótipo que tenha os mesmos recursos dos que os modelos disponíveis no mercado, e que possa ter um desempenho igual ou até mesmo superior que os mesmos citados.

### 1.4 ESTRUTURA DO TRABALHO

O trabalho terá a estrutura abaixo apresentada.

Capítulo 1 - Introdução: é apresentado o tema, o problema, os objetivos da pesquisa, a justificativa e a estrutura geral do trabalho;

Capítulo 2 – Referencial teórico: será abordado como funcionam sistemas de monitoramento em modo geral, dados sobre o índice de acidentes domésticos entre crianças e jovens, além de ser apresentado o computador de baixo custo que será utilizado no desenvolvimento do projeto em conjunto com o sistema operacional a ser instalado com ele;

Capítulo 3 – Desenvolvimento do protótipo: este capítulo irá apresentar a topologia proposta para o sistema e os passos do seu desenvolvimento;

Capítulo 4 – Resultados e considerações finais: a apresentação dos resultados obtidos com o protótipo e avaliação do seu desempenho, além de propor melhorias com o objetivo de transformar o mesmo em um produto de mercado.

## 2 FUNDAMENTAÇÃO TEÓRICA

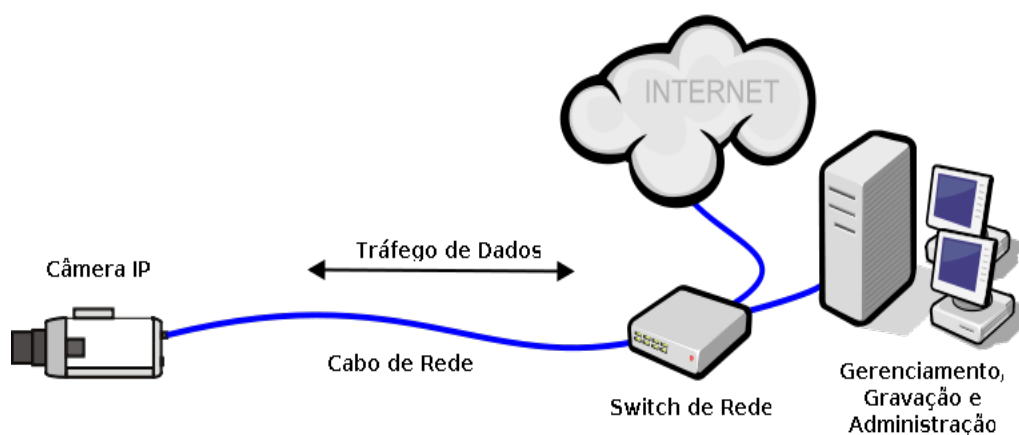
Este capítulo irá abordar sobre como é um sistema básico de monitoramento, além de haver uma explanação sobre dados de acidentes domésticos envolvendo crianças, e sobre os dispositivos eletrônicos e softwares a serem utilizados no desenvolvimento do protótipo, para que tenhamos um referencial teórico sobre o assunto e possa assim ser justificado o desenvolvimento deste trabalho.

### 2.1 SISTEMAS DE MONITORAMENTO E VIGILÂNCIA

Segundo o dicionário Prebiram, o significado da palavra vigilância é o “Ato ou efeito de vigiar”, “ter zelo” e “atenção por algo”, sendo exatamente isso o objetivo dos sistemas de monitoramento.

Um dos tipos de sistemas mais comuns encontrados no mercado são os chamados de CFTV, que significa Circuito Fechado de Televisão, que é uma versão traduzida da sigla em inglês mais conhecida como CCTV (Closed-Circuit Television).

Esse tipo de sistema pode ser montado em vários tipos de configurações, com diversos tipos de tecnologia, abaixo é apresentado um exemplo de topologia básica desse sistema, que inclusive é similar ao ser desenvolvido nesse trabalho.



**Figura 2 - Topologia de um sistema básico de monitoramento com câmeras IP.**

Fonte: <http://www.guiadocftv.com.br>.

Como visto na figura 2, o protocolo de comunicação comumente usado pelas câmeras, pela central e outros dispositivos ligados ao sistema são basicamente protocolos comuns de rede TCP/IP, que por vezes podem ser Ethernet 802.3 (redes cabeadas) ou 802.11 (redes sem fio).

Este trabalho não irá contemplar detalhes desses tipos de rede, apenas nos capítulos posteriores será um pouco mais explanado para ser feita a configuração do protótipo a ser desenvolvido.

## 2.2 ACIDENTES DOMÉSTICOS ENTRE CRIANÇAS

O acidente doméstico tem se revelado como uma das principais causas dos atendimentos, internações, incapacidades e óbitos em crianças, em vários países e tem contribuído, de forma considerável, para manter elevada a taxa de morbimortalidade infantil. (CORDEROABADET et. al., 1989; SCHVARTSMAN, 1987; SANTOS, 1988; JONES, 1993)

Um acidente por definição é considerado um acontecimento imprevisto ou inesperado, mas não significa que não é algo que possa ser evitado, pois conforme aponta estatísticas do Ministério da Saúde, acidentes ou lesões não intencionais são as principais causas de morte de crianças de 1 a 14 anos no Brasil.

Ao todo, cerca de cinco mil crianças morrem e 110 mil são hospitalizadas anualmente.

Segundo um estudo o publicado na Revista Brasileira de Pesquisa e Saúde, o ambiente domiciliar é apontado como o principal local de ocorrência dos acidentes, em que predominam as quedas, aspirações de corpo estranho, queimaduras, afogamentos em banheiras e intoxicações.

Os menores de um ano são as principais vítimas, em decorrência da sua fase de desenvolvimento, marcada pela curiosidade e pela dependência do adulto.

Na tabela 1, constam os resultados de uma pesquisa, quanto ao tipo de acidente e a incidência por faixa etária.



VARIÁVEL	IDADE CATEGORIZADA				Total
	De 3 a menor de 8 meses	De 8 meses a menor de 4 anos	De 4 a menor de 8 anos	De 8 anos a 12 anos	
Tipos de Acidentes					
Queda	4	25	9	28	66
Choque elétrico	1	2	1	2	6
Corte	-	1	-	4	5
Queimadura	-	2	-	4	6
Colisão	-	-	2	2	4
Escorregão	1	5	1	2	9
Obstrução de vias aéreas	1	1	-	3	5
Intoxicação	-	1	-	2	3
<b>Total</b>	<b>7</b>	<b>37</b>	<b>13</b>	<b>47</b>	<b>104</b>

**Tabela 1 - Crianças vítimas de acidentes, segundo idade e tipo de acidente.**

Fonte: [http://www.saocamilo-sp.br/pdf/mundo\\_saude/155558/A03.pdf](http://www.saocamilo-sp.br/pdf/mundo_saude/155558/A03.pdf).

Como aponta o estudo, as crianças são muito vulneráveis aos acidentes e que o ambiente doméstico pode trazer diversos riscos.

Portanto, o constante monitoramento sobre elas é primordial na prevenção dos acidentes, mas em muitos momentos de nossa rotina do dia-a-dia não podemos deixar de fazer nossas tarefas, é onde entra o auxílio de dispositivos, como as Babás eletrônicas.

### 2.3 RASPBERRY PI

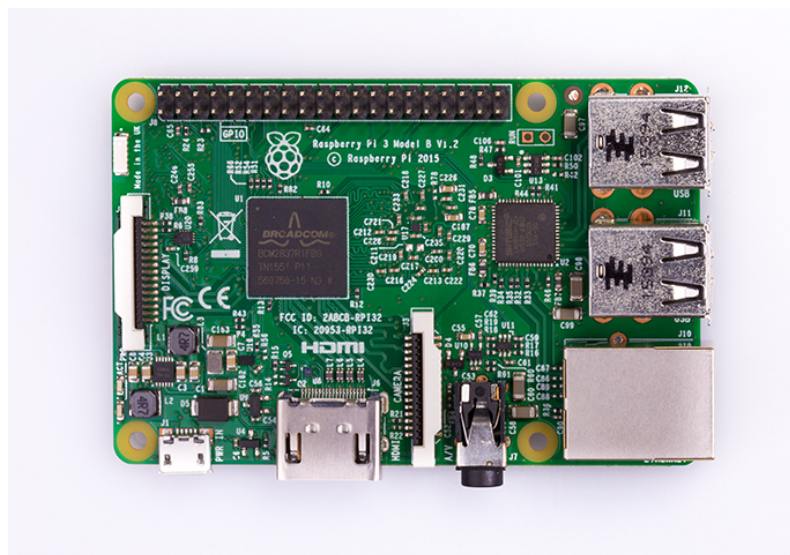
A Raspberry Pi é um computador de baixo custo, do tamanho de um cartão de crédito, desenvolvido por Eben Upton e sua equipe, que vieram então a criar Raspberry Pi Foundation em 2009 no Reino Unido.

Esse computador é denominado como uma SBC (*Single Board Computer*), que é composto de um processador Multimídia da empresa Broadcom, que é do tipo SoC (*System-on-Chip* – em português, sistema em um chip) , ou seja, a grande parte do sistema, como a CPU (Unidade Central de Processamento), a GPU (Unidade de Processamento Gráfico), o hardware de áudio e de comunicações são integrados em um só componente, estando apenas a memória RAM localizada externamente, além de outros componentes discretos necessários para o sistema.

O que a torna tão interessante, é que para utilizá-la é necessário apenas um cartão de memória do tipo *SD Card*, pois a mesma não possui nenhum tipo de disco rígido em sua composição, uma fonte de alimentação de 5V com uma corrente de pelo menos 700mA, (isso para os modelos que o processador é de apenas um núcleo) sendo o ideal a partir de 2A (que no caso atenderia as versões de quatro núcleos), além de um monitor ou televisão, teclado e mouse para fazer as configurações iniciais.

Até o presente momento desse trabalho, a Raspberry Pi Foundation já lançou cerca de 12 modelos de Raspberry Pi, sendo as mudanças basicamente no seu tamanho físico, capacidade do processador (versões com 1 ou 4 núcleos) e quantidade de memória RAM (256MB, 512Mb e 1Gb), sendo que seu preço pode variar de US\$5 a US\$35.

Para o desenvolvimento desse trabalho, foi escolhido a Raspberry Pi 3 Modelo B, igual o da figura 3.



**Figura 3 - Raspberry Pi Modelo B**  
Fonte: Raspberry Pi Foundation (2017)

É o modelo que possui mais capacidade de processamento e quantidade de memória RAM, sendo as suas especificações as seguintes:

- Processador Quad-Core 64-bit de 1,2Ghz;
- 1GB de Memória RAM;
- Conector Ethernet;
- Adaptador Wifi 802.11n integrado;
- Bluetooth 4.1 BLE integrado;
- Conector de vídeo HDMI;
- 4 portas USB 2.0;
- Interface para câmera (CSI);
- Interface para display (DSI);
- Slot para cartão micro SD;
- Conector de áudio e vídeo;
- GPIO de 40 pinos;

## 2.4 O SISTEMA OPERACIONAL LINUX

O Linux é um dos avanços tecnológicos mais importantes do século XXI, além de seu impacto sobre o crescimento da internet e do seu lugar como uma tecnologia capacitadora para uma série de dispositivos baseados em computador, o desenvolvimento do Linux tem sido um exemplo de como projetos colaborativos podem ultrapassar o que as pessoas e empresas individuais podem fazer sozinhas (NEGUS, 2012).

Segundo NEGUS, o Linux é um sistema operacional que é construído por uma comunidade de desenvolvedores de software ao redor do mundo e liderado por ser criador Linus Torvalds.

Várias organizações empacotam esse Kernel do Linux e as oferece ao público como uma “distribuição”, ou também chamado *distro*. Algumas das mais comuns distribuições são a Red Hat, SuSE, Debian e variantes desses mesmos projetos, apenas para citar alguns.

A Raspberry Pi, que é placa a ser utiliza nesse projeto, suporta diversos sistemas operacionais baseados em Linux, e isso pode ser encontrado na página da Raspberry Pi Foundation na seção de Downloads.

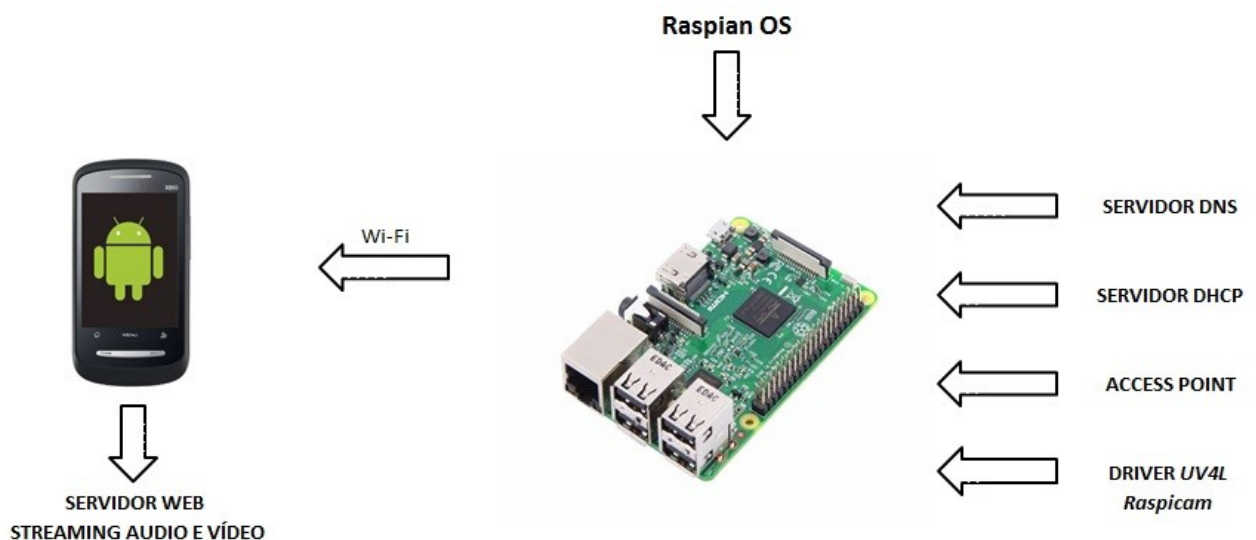
Nesse trabalho será utilizado um sistema operacional chamado de *Raspian*, que é uma distribuição específica para Raspberry Pi, que é baseada na distribuição Debian, um dos projetos de distribuição Linux de maior sucesso.

### 3 DESENVOLVIMENTO DO PROTÓTIPO

Neste capítulo serão apresentados os procedimentos para o desenvolvimento do protótipo da babá eletrônica.

#### 3.1 TOPOLOGIA PROPOSTA PARA O PROJETO

Para o desenvolvimento desse protótipo, foi proposta uma topologia conforme a figura 4.



**Figura 4 – Topologia do Sistema proposto para o Protótipo**

**Fonte: Autor**

Como visto na figura acima, a Raspberry Pi irá funcionar como um roteador, para facilitar o acesso de outros dispositivos a ela, onde estará rodando um servidor WEB, com uma página disponível com a transmissão do vídeo e o áudio em tempo real.

Os hardwares a serem utilizados para montar essa configuração são:

- Raspberry Pi 3 Modelo B;

- Fonte de alimentação 5V / 2,4 A;
- Cartão Micro SD Card de 16GB Classe 10;
- Raspberry Pi Câmera V1.3 de 5Mp;
- Adaptador de Som USB, para fone e microfone;
- Microfone P2 Flexível;

Nos tópicos seguintes serão apresentados os passos da configuração dos periféricos utilizados, e quanto à parte de instalação do sistema operacional, será apresentado no apêndice desse trabalho.

### 3.2 CONFIGURAÇÃO DO ADAPTADOR DE SOM USB COM MICROFONE

A Raspberry Pi em nenhuma de suas versões disponibiliza uma entrada de áudio, apenas uma saída em um conector do tipo Jack de 3,5mm, e nesse projeto é necessário a entrada, para que possa ser feito o uso de um microfone, para captar os sons externos, que no caso será da criança a estar monitorada, e enviado para os dispositivos de recepção.

A solução encontrada foi utilizar um adaptador de som USB com um microfone externo, como apresentado da figura 5.



**Figura 5 - Adaptador de Som USB e Microfone**  
Fonte: Autor

O Sistema Operacional Raspian, reconhece os dispositivos de áudio integrados a CPU como o padrão a ser utilizado, portanto, é necessário que seja configurado o adaptador como o novo dispositivo padrão, para isso, deve-se primeiramente verificar se o adaptador de áudio USB foi reconhecido pelo sistema, usando o comando `dmesg`, e verificando as informações de sua saída, deverão ser apresentadas as seguintes mensagens:

```
[347.041411] input: C-Media Electronics Inc.USB PnP Sound
Deviceas/devices/platform/soc/3f980000.usb/usb1/11/11.2/11.2:1.3/000
3:0D8C:013C.0003/input/input2.
```

```
[347.106063] hid-generic 0003:0D8C:013C.0003: input,hidraw0: USB HID
v1.00 Device [C-Media Electronics Inc. USB PnP Sound Device] on usb-
3f980000.usb-1.2/input3.
```

O próximo passo é modificar o arquivo de configuração de som, que está localizado em `/usr/share/alsa/alsa.conf`, com o seguinte comando:

```
$ nano /usr/share/alsa/alsa.conf
```

Ao abrir o arquivo o mesmo será apresentado com a seguinte configuração:

```
defaults.ctl.card 0
defaults.pcm.card 0
```

O valor deve ser mudado de `0` para `1` para que o adaptador de som USB passe a ser o dispositivo padrão de som.

### 3.3 CONFIGURANDO A RASPBERRY PI COMO UM ROTEADOR

Para facilitar o acesso de outros dispositivos à Raspberry Pi, foi utilizado um adaptador USB *wireless*, com a capacidade de funcionar como AP, o mesmo é apresentado na figura 6.



**Figura 6 - Adaptador wireless USB**  
**Fonte: Autor**

Para fazer o adaptador wireless funcionar da forma necessária, foi instalado os softwares *hostapd* e *dnsmasq*, ambos os softwares são livres e gratuitos, e estão disponíveis nos repositórios do Raspian, sendo o primeiro para configura-lo como um *Acess Point* e o segundo como um servidor DNS e DHCP, dando uma funcionalidade de roteador a Raspberry Pi.

Os softwares foram instalados com o seguinte comando:

```
$ sudo apt-get install hostapd dnsmasq
```



Depois de instalados os softwares, foi preciso configurar a interface a ser utilizada, que no caso é a `wlan0`, atribuindo um endereço IP fixo para ela, editando o arquivo `/etc/network/interfaces` da seguinte forma:

```
iface wlan0 inet static
address 192.168.50.1
netmask 255.255.255.0
network 192.168.50.0
Broadcast 192.168.50.255
```

Em seguida, é reiniciada a interface com os comandos `ifdown wlan0` e `ifup wlan0`, para que a configuração acima passasse a vigorar.

Para o adaptador operar como AP, foi criado um arquivo de configuração em `/etc/hostapd/hostapd.conf`, com as seguintes informações:

```
interface=wlan0           #Interface a ser utilizada como AP
driver=nl80211            #Driver utilizado pelo adaptador
ssid=Pi3-Baba             #Nome da rede
hw_mode=g                 #Habilitado a banda de 2,4Ghz
channel=6                  #Canal utilizado
macaddr_acl=0             #Habilitar o padrão 802.11n
auth_algs=3               #Habilita o QoS sobre o Wifi
ignore_broadcast_ssid=0   #Requer que cliente saiba nome da rede
Wpa=2                      #Usar criptografia WPA2
wpa_passphrase=raspi1234  #Senha da rede
```

```

wpa_key_mgmt=WPA-PSK      #Usar chave pré-compartilhada
wpa_pairwise=CCMP
rsn_pairwise=CCMP
ieee80211n=1              #Suporte a 802.11n

```

Para que o serviço do *hostapd* passe a iniciar automaticamente a partir da próxima reinicialização do sistema, é preciso editar o arquivo */etc/default/hostapd*, na linha onde consta *DAEMON\_CONF=""*, indicando o endereço do arquivo de configuração, passando a ser *DAEMON\_CONF="/etc/hostapd/hostapd.conf"*.

Para finalizar, é preciso que o *dnsmasq* passe a operar como o servidor DHCP e DNS da interface, para isso será editado o arquivo de configuração padrão, que se encontra em */etc/dnsmasq.conf*.

É possível fazer diversas formas de configurações, mas para o objetivo desse trabalho, foi necessário alterar apenas algumas linhas de configuração, que foram as seguintes:

```

interface=wlan0          #Interface a ser utilizada
address=192.168.50.1     #Endereço atribuído ao servidor
dhcp-range=192.168.50.5,192.168.50.150    #Range de IPs
domain=baba.local       #Nome do domínio

```

A partir desse ponto, a Raspberry Pi já está acessível a conexões externas para outros dispositivos, faltando apenas a configuração do serviço de streaming de vídeo e áudio, que será abordado no próximo tópico.

### 3.4 CONFIGURANDO O STREAMING DE AUDIO E VIDEO

Para a transmissão de áudio e vídeo, foi utilizado um driver chamado *Raspicam driver*, que foi especificamente desenvolvido para câmera oficial da Raspberry Pi Foundation, como a da figura 7.



**Figura 7 - Raspberry Pi Câmera**  
Fonte: [www.amazon.com](http://www.amazon.com)

Esse driver faz parte de um projeto chamado UV4L, que é mantido pelo desenvolvedor Luca Risolia, que já é envolvido há muito tempo com o desenvolvimento de drivers para streaming de dispositivos de vídeo e áudio em geral.

É um projeto em desenvolvimento, mas já possui diversos drivers totalmente funcionais, e todo o software e documentação é disponibilizada através de seu site <http://linux-projects.org>.

Para instalar os módulos e o driver do projeto, é necessário primeiramente adicionar o repositório do projeto, que é feito através do seguinte comando:

```
$ curl http://www.linux-projects.org/listing/uv4l_repo/lrkey.asc |  
sudo apt-key add -
```

Em seguida, é necessário atualizar o arquivo `/etc/apt/sources.list`, adicionando a seguinte linha:

```
deb http://www.linux-projects.org/listing/uv4l_repo/raspbian/ jessie main
```

O Sistema deve ser atualizado, e na sequência já é instalado o driver, através desses comandos:

```
$ sudo apt-get update  
$ sudo apt-get install uv4l uv4l-raspicam uv4l-raspicam-extras
```

Já com o driver instalado, já é possível configura-lo, editando o arquivo de configuração `/etc/uv4l/uv4l-raspicam.conf`.

Depois de configurado o driver, já é possível acessar o Webservice do streaming, utilizando endereço local definido anteriormente, e através da porta 8080, acessando a opção *Two-way Audio/Video*, como mostra a figura 8.

Os detalhes da configuração do driver serão apresentados no tópico seguinte, nos resultados obtidos.



**Figura 8 - Acesso a página do serviço de Streaming**  
Fonte: Autor

## 4 RESULTADOS E CONSIDERAÇÕES FINAIS

Neste capítulo, será apresentado os resultados obtidos com o protótipo, além das possíveis melhorias a serem feitas e a conclusão sobre o trabalho.

### 4.1 RESULTADOS OBTIDOS COM O STREAMING DE ÁUDIO E VÍDEO

De forma geral, o resultado obtido foi bem satisfatório, sendo que foi necessário testar algumas opções de configurações disponíveis pelo driver.

Em relação à transmissão do áudio, não houve necessidade de qualquer alteração ao que já estava configurado por padrão, tendo o tempo de resposta totalmente aceitável.

Quanto ao vídeo, foram feitos ajustes no arquivo de configuração, que se encontra em `/etc/uv4l/uv4l-raspicam.conf`, de forma a conseguir a melhor performance em qualidade de vídeo e no tempo de resposta possível, e as configurações ficaram as seguintes:

- Resolução: 1024 x 768
- Formato de vídeo: h264
- Frame rate: 30

Alguns outros ajustes como brilho, contraste e rotação do vídeo, estão disponíveis através de um menu de configurações, como a da figura 11.

## Camera Control Panel

**Resolution & Format**

width  height  format

NOTE: if the camera is already in use for streaming by another application/client, applying the resolution & format will not have any effect (until all the streaming sessions have been closed).

---

**Control settings**

brightness

contrast

saturation

red balance

blue balance

sharpness

rotate

shutter speed

zoom factor

iso sensitivity

jpeg quality

frame rate

horizontal mirror  enable  disable

vertical mirror  enable  disable

text overlay  enable  disable

object/face detection  enable  disable

stills denoise  enable  disable

video denoise  enable  disable

image stabilization  enable  disable

awb mode

exposure mode

exposure metering

drc strength

**Figura 9 - Menu de configurações de vídeo**  
**Fonte: Autor**

O tempo de resposta do vídeo foi medido através de um cronômetro rodando em um computador, e ao lado a mesma imagem sendo transmitida para um celular, como apontado na figura 10.

O tempo médio de reposta com a configuração feita acima, foi de aproximadamente 300 a 400ms utilizando esse procedimento.



**Figura 10 - Teste do tempo de resposta do vídeo**  
**Fonte: Autor**

#### 4.2 CUSTOS DO DESENVOLVIMENTO DO PROTÓTIPO

O principal objetivo do trabalho era conseguir desenvolver o protótipo com o um custo inferior aos modelos vendidos comercialmente no varejo.

O custo do para fazer o protótipo é apresentado na tabela 2, abaixo:



Item	Quantidade	Preço
Raspberry Pi 3 Modelo B	1	R\$ 189,00
Raspberry Pi Câmera V1.3	1	R\$ 85,00
Adaptador de Som USB	1	R\$ 8,90
Microfone P2 Flexível	1	R\$ 9,90
Fonte 5V/2,1A com cabo Micro USB	1	R\$ 19,90
Case em acrílico	1	R\$ 18,00
	TOTAL	R\$ 330,70

**Tabela 2 - Custo do desenvolvimento do protótipo**  
**Fonte: Autor**

Foram feitas pesquisas em vários sites de comércio eletrônico, de alguns modelos de babás eletrônicas com especificações próximas as desenvolvidas no projeto, e os seus valores podem ser comparados a tabela 3.

Marca	Modelo	Preço
Q-See	B200-1	R\$ 299,99
Motorola	MBP421	R\$ 422,90
Tectoy	VBT-2400	R\$ 499,00
Multilaser	BB136	R\$ 609,00

**Tabela 3 - Preço de babás eletrônicas encontrados no varejo**  
**Fonte: Autor**

#### 4.3 CONCLUSÃO E SUGESTÕES PARA TRABALHOS FUTUROS

Foi possível fazer o acesso da baba eletrônica através de celulares e laptops com bastante facilidade, sendo a única limitação encontrada, a apenas funcionar no navegador web Google Chrome, pois o mesmo tem suporte à tecnologia no driver utilizado, porém, o desenvolvedor afirma que é possível utilizar outros navegadores, mas os mesmos requerem outros softwares complementares, que por vezes são de terceiros e não oficiais do repositório do navegador, para funcionar adequadamente.

O projeto se mostrou viável e vantajoso do aspecto financeiro, considerando que a maioria dos dispositivos equivalentes possuem preços ligeiramente iguais ou superiores do que o gasto no protótipo.

O projeto foi apenas apresentado como um protótipo, mas com os resultados obtidos é possível ver que com adição de mais alguns recursos, facilmente poderia se transformar em produto de mercado.

Como trabalhos futuros, é possível sugerir que as seguintes funcionalidades possam ser adicionadas ao protótipo:

- Um sistema de visão noturna, utilizando LED's do tipo IR;
- Configurar o acesso a redes externas, para que os dispositivos conectados não estejam limitados somente a ela, tendo por exemplo acesso à internet, sendo isso possível configurando um NAT, e por segurança implantar um tipo de Firewall;
- Desenvolver um sistema de rotação da câmera, que em alguns modelos comerciais possuem, utilizando alguns servo-motores ou motores-de-passo, tudo comandado via o GPIO da placa.

## REFERÊNCIAS

BRASIL. Ministério da Saúde. Secretaria de Vigilância em Saúde. Secretaria de Atenção à Saúde. **Manual de vigilância do óbito infantil e fetal e do Comitê de Prevenção do Óbito Infantil e Fetal**. 2. ed. Brasília: Ministério da Saúde, 2009. p. 96 (Série A. Normas e Manuais Técnicos).

Dnsmasq – Disponível em: <http://www.thekelleys.org.uk/dnsmasq/doc.html>  
<Acessado em: 24 de fevereiro de 2017>

FERNANDES, A. et al. Síndrome da morte súbita do lactente: o que sabem os pais. **Acta Pediátrica Portuguesa**, v. 43, n. 2, p. 59-62, mar./abr., 2012.

Hostapd – Disponível em:  
<https://wireless.wiki.kernel.org/en/users/documentation/hostapd>  
<Acessado em: 18 de março de 2017>

NEGUS, Christopher . Linux - A Bíblia: O mais abrangente definitivo guia sobre Linux 8ª Ed., 2014 – Rio de Janeiro, Editora Alta Books, p. 5 – 26.

NUNES, M. L. et al. Síndrome da morte súbita do lactente: aspectos clínicos de uma doença subdiagnosticada. **Jornal de Pediatria**, Rio de Janeiro, v. 77, n. 1, p. 29-34, fev., 2001.

Priberam. 2017. Definição da palavra vigilância. [Online] . Disponível em:  
<https://www.priberam.pt/dlpo/vigil%C3%A2ncia> <Acessado em: 12 de Abril de 2017>

Raspberry pi Foundation – Disponível em: <https://www.raspberrypi.org/>  
<Acessado em: 17 de fevereiro de 2017>

SCHARSTSMAN, S. (1997) Acidentes na infância. in: CARVALHO, O Manual de pediatria. Rio de Janeiro: Guanabara Koogan.

SILVA, V. L. S da. et al. Mortalidade infantil na cidade de Pelotas, estado do Rio Grande do Sul, Brasil, no período 2005-2008: uso da investigação de óbitos na análise das causas evitáveis. **Epidemiologia e Serviços de Saúde**, Brasília, v. 21, n. 2, p. 265-274, 2012.

Sociedade Brasileira de Pediatria. **Aspiração de corpo estranho**. São Paulo: Sociedade Brasileira de Pediatria; 2010.

UNIVERSIDADE TECNOLÓGICA Federal do Paraná. **Comissão de Normalização de Trabalhos Acadêmicos Normas para elaboração de trabalhos acadêmicos/** Universidade Tecnológica Federal do Paraná, Comissão de Normalização de Trabalhos Acadêmicos. - Curitiba: UTFPR, 2016.

UV4L – User Space Video4Linux. Disponível em: <https://www.linux-projects.org/uv4l/>  
<Acessado em: 23 de janeiro de 2017>

## ANEXOS

### ANEXO A – Foto do protótipo



## ANEXO B – Arquivo de configuração do driver UV4L

```

# This file contains the default values of SOME of all the available
# options for the UV4L core module, the raspicam driver, and the
# streaming server front-end.
#
# This file is referenced by the 'uv4l_raspicam' init script.
# Alternatively, it can be passed as argument to
# the '--driver-config-file' driver option and/or to the
# '--config-file' option of uv4l.
#
# Please refer to the 'uv4l', 'uv4l-raspicam' and 'uv4l-server' manual
# pages for all the valid options and values or for more informations
# about their meaning.

# NOTE: for multi argument options you must specify one arg per line,
# e.g --min-object-size 80 120 becomes:
# min-object-size = 80
# min-object-size = 120

#####
# uv4l core options
#####

driver = raspicam
#video_nr = 0
auto-video_nr = yes
#verbosity = 6
#syslog-host = localhost
#syslog-port = 514
#frame-timeout = 5000
    frame-buffers = 4
#drop-bad-frames = yes
#relaxed-ownership = yes

#####
# raspicam driver options
#####

encoding = h264
#width = 1024
#height = 768
framerate = 30
#custom-sensor-config = 2

### dual camera options:
#stereoscopic-mode = side_by_side
#camera-number = 1
#decimate = yes
#swap-eyes = yes

```

```
### still and/or video options:
#quality = 85
#stills-denoise = yes
  video-denoise = no
#raw = yes

### h264 options:
  profile = high
  level = 3.2
  bitrate = 1700000
#intra-refresh-mode = dummy
  intra-period = 15
  inline-headers = yes
#quantisation-parameter #arg

### video overlay options:
npreview = yes
fullscreen = yes
#osd-layer = 2
#opacity = 255
### preview window <x, y, w, h>:
preview = 480
preview = 240
preview = 320
preview = 240

### post-processing options:
# text-overlay = yes
# text-filename = /usr/share/uv4l/raspicam/text.json
# object-detection = yes
# object-detection-mode = accurate_tracking
# min-object-size = 80
# min-object-size = 80
# main-classifier = /usr/share/uv4l/raspicam/lbpcascade_frontalface.xml
# secondary-classifier =/usr/share/uv4l/raspicam/lbpcascade_frontalface.xml

### image settings options:
# sharpness = 0
# contrast = 0
# brightness = 50
# saturation = 0
# iso = 400
# vstab = yes
# ev = 0
# exposure = auto
# awb = auto
# imgfx = none
# metering = average
# rotation = 0
# hflip = no
# vflip = no
# shutter-speed = 0
# drc = off
# red-gain = 100
# blue-gain = 100
# text-annotation = HelloWorld!
# text-annotation-background = yes
### ROI <x, y, w, h> normalized to [0, 1]
# roi = 0
# roi = 0
# roi = 1
```

```

# roi = 1
### ISP blocks
# black-level-compensation = yes
# lens-shading = yes
# automatic-defective-pixel-correlation = yes
# white-balance-gain = yes
# crosstalk = yes
# gamma = yes
# sharpening = yes

### advanced options:
# statistics = yes
# output-buffers = 3

### serial Number & License Key:
# serial-number = #arg
# license-key = #arg

#####
# streaming server options
#####

### path to a separate config file that will be parsed by the streaming
server
### module directly when it's loaded,
### in which you are allowed to specify all the streaming server options
### listed below in the short form "option=value" instead of the longer
### "--server-option = --option=value" form that you must use
### in this configuration file.
#server-config-file = #path

server-option = --port=8080
#server-option = --bind-host-address=localhost
#server-option = --md5-passwords=no
#server-option = --user-password=raspil234
#server-option = --admin-password=raspil234
### To enable 'config' user authentication
#server-option = --config-password=raspil234

### HTTPS options:
server-option = --use-ssl=yes
server-option = --ssl-private-key-file=/etc/ssl/private/selfsign.key
server-option = --ssl-certificate-file=/etc/ssl/private/selfsign.crt

### WebRTC options:

server-option = --enable-webrtc=yes
# server-option = --enable-webrtc-datachannels=yes
# server-option = --webrtc-datachannel-label=uv4l
# server-option = --webrtc-datachannel-socket=/tmp/uv4l.socket
server-option = --enable-webrtc-video=yes
server-option = --enable-webrtc-audio=yes
server-option = --webrtc-receive-video=yes
server-option = --webrtc-receive-datachannels=yes
# server-option = --webrtc-received-datachannel-socket=/tmp/uv4l.socket
server-option = --webrtc-receive-audio=yes
server-option = --webrtc-received-audio-volume=4
server-option = --webrtc-prerenderer-smoothing=yes
server-option = --webrtc-recdevice-index=0
# server-option = --webrtc-vad=yes

```



```

server-option = --webrtc-echo-cancellation=yes
server-option = --webrtc-preferred-vcodec=0
server-option = --webrtc-enable-hw-codec=yes
server-option = --webrtc-hw-vcodec-minbitrate=256
server-option = --webrtc-hw-vcodec-maxbitrate=4000
server-option = --webrtc-hw-vcodec-startbitrate=1200
# server-option = --webrtc-max-playout-delay=34
# server-option = --webrtc-cpu-overuse-detection=yes
server-option = --webrtc-combined-audiovideo-bwe=yes
server-option = --webrtc-stun-urls=stun:stun.l.google.com:19302
# server-option = --webrtc-ice-servers=[{"urls": "stun:stun1.example.net"},
{"urls": "turn:turn.example.org", "username": "user", "credential":
"myPassword"}]
server-option = --webrtc-stun-server=yes
# server-option = --webrtc-tcp-candidate-policy=1
# server-option = --webrtc-rtcp-mux-policy=0
# server-option = --webrtc-enable-dscp=no
# server-option = --webrtc-ignore-loopback=yes
### video rendering window positions and sizes on the display.
### for each window, default values can be optionally overridden, but if
you
### do this you must specify one line for each of the four x, y, width,
height
### window properties (in that order).
### If fullscreen is set the image is stretched to the maximum available
display
### resolution from the specified size.
### window 1
# server-option = --webrtc-renderer-window=0
# server-option = --webrtc-renderer-window=0
# server-option = --webrtc-renderer-window=1024
# server-option = --webrtc-renderer-window=768
# server-option = --webrtc-renderer-fullscreen=no
# server-option = --webrtc-renderer-rotation=180
# server-option = --webrtc-renderer-opacity=255
### window 2
# server-option = --webrtc-renderer2-window=480
# server-option = --webrtc-renderer2-window=0
# server-option = --webrtc-renderer2-window=320
# server-option = --webrtc-renderer2-window=240
### window 3
# server-option = --webrtc-renderer3-window=0
# server-option = --webrtc-renderer3-window=352
# server-option = --webrtc-renderer3-window=176
# server-option = --webrtc-renderer3-window=128

### XMPP options:
# server-option = --xmpp-server=lambada.jitsi.net
# server-option = --xmpp-port=5222
# server-option = --xmpp-muc-domain=meet.jit.si
# server-option = --xmpp-room=room
# server-option = --xmpp-room-password=room_password
# server-option = --xmpp-username=me
# server-option = --xmpp-password=myspassword
# server-option = --xmpp-reconnect=yes
# server-option = --xmpp-bosh-enable
# server-option = --xmpp-bosh-tls
# server-option = --xmpp-bosh-server
# server-option = --xmpp-bosh-port
# server-option = --xmpp-bosh-hostname
# server-option = --xmpp-bosh-path

```

```

# server-option = --xmpp-bridge-host=localhost
# server-option = --xmpp-bridge-port=7999

### Janus WebRTC Gateway options:
# server-option = --janus-gateway-url=http://janus.conf.meetecho.com:8088
# server-option = --janus-gateway-root=/janus
# server-option = --janus-room=1234
# server-option = --janus-room-pin=#pin
# server-option = --janus-username=test
# server-option = --janus-token=#token
# server-option = --janus-proxy-host=#host
# server-option = --janus-proxy-port=80
# server-option = --janus-proxy-username=#user
# server-option = --janus-proxy-password=#password
# server-option = --janus-proxy-bypass=#regex
# server-option = --janus-force-hw-vcodec=no
# server-option = --janus-video-format=#code
server-option = --janus-publish=yes
server-option = --janus-subscribe=no
server-option = --janus-reconnect=yes

### Fine-tuning options:
# server-option = --connection-timeout=15
# server-option = --enable-keepalive=yes
# server-option = --max-keepalive-requests=0
# server-option = --keepalive-timeout=7
# server-option = --max-queued-connections=8
# server-option = --max-streams=5
# server-option = --max-threads=5
# server-option = --thread-idle-time=10
# server-option = --chunked-transfer-encoding=yes

### Advanced options:
# server-option = --frame-timeout=5000
# server-option = --frame-buffers=auto

### These options are specific to the HTTP/HTTPS Server
### serving custom Web pages only:
server-option = --enable-www-server=yes
server-option = --www-root-path=/usr/share/uv4l/www/
server-option = --www-index-file=index.html
server-option = --www-port=443
# server-option = --www-bind-host-address=#host
# server-option = --www-password=#password
server-option = --www-use-ssl=yes
server-option = --www-ssl-private-key file=/etc/ssl/private/selfsign.key
server-option = --www-ssl-certificate-file=/etc/ssl/private/selfsign.crt
# server-option = --www-connection-timeout=15
# server-option = --www-enable-keepalive=no
# server-option = --www-max-keepalive-requests=0
# server-option = --www-keepalive-timeout=7
# server-option = --www-max-queued-connections=8
# server-option = --www-max-threads=4
# server-option = --www-thread-idle-time=10
# server-option = --www-chunked-transfer-encoding=no

### Other options:
# server-option = --editable-config-file=#path
server-option = --enable-control-panel=yes
# server-option = --enable-rest-api=yes

```