

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ARGUS LUCONI ROSENHAIM

PROJETO DE DIPLOMAÇÃO

EMBEDDED STREAMING

Porto Alegre

2010

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

EMBEDDED STREAMING

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

ORIENTADOR: Marcelo Götz

Porto Alegre

2010

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

ARGUS LUCONI ROSENHAIM

EMBEDDED STREAMING

Este projeto foi julgado adequado para fazer jus aos créditos da Disciplina de “Projeto de Diplomação”, do Departamento de Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Marcelo Götz, UFRGS

Doutor pela Universität Paderborn, Alemanha

Banca Examinadora:

Prof. Marcelo Götz, UFRGS

Doutor pela Universität Paderborn, Alemanha

Prof. João Cesar Netto, UFRGS

Doutor pela Université Catholique de Louvain, Bélgica

Vinicius Vasconcellos, Gerente do Grupo RBS

MBA Engenheiro Eletricista pela UFRGS, Porto Alegre

Porto Alegre, novembro de 2010.

DEDICATÓRIA

Dedico este trabalho a Senhora Maria Tereza da Rosa Luconi, conhecida como Vó Tereza, que acompanhou muito próxima e quase na integra todo este período de graduação, bem como o dos meus irmão antes de mim.

AGRADECIMENTOS

Primeiramente a pessoa que me apresentou esta idéia e foi muito importante no sentido de motivar sua realização, a amiga Raquel Hoshino.

Aos professores que acompanharam de perto, em especial as orientações do Professor Doutor Marcelo Götz e as dicas e soluções do Professor Doutor João Cesar Netto.

Ao colega de UFRGS Thiago Santini, com seus e-mails rápidos e decisivos.

Ao desenvolvedor Thiago de Freitas Oliveira Araújo, que foi de suma importância para a solução do projeto, mostrando o poder da Comunidade de Software Livre.

A revisão de Patrícia Diniz, ágil e eficiente.

Aos familiares, não apenas pelo suporte e a atenção aos intermináveis e incomensuráveis assuntos, mas pela oportunidade de um estudo superior de qualidade.

RESUMO

Este projeto consiste no desenvolvimento de um dispositivo portátil capaz de transmitir vídeos em tempo real por meio da internet, sendo estes para o uso em *broadcast*, *webcast* ou mesmo de ponto a ponto, utilizando conexões de banda larga sem fio móveis.

Palavras-chaves: Streaming. Sistemas Embarcados. Webcast.

ABSTRACT

The objective of this project it is to develop a mobile device capable of straming live footage thru the internet, to be use in broadcast, webcast or peer to peer, using broadband wireless links.

Keywords: Streaming. Embedded System. Webcast

SUMÁRIO

1 - Introdução	11
2 - Contexto do projeto	13
2.1 - Entrevistas	13
2.1.1 - LiveU	14
2.1.2 - Rede TV!	15
2.1.3 - Universo Online	16
2.1.4 - Grupo Abril	17
2.2 - Definições	19
3 - Análise de alternativas	21
3.1 - Protocolo.....	22
3.2 - Servidor.....	23
3.3 - Cliente	26
3.4 - Dispositivo Encoder	27
4 - Métodos, Processos e Dispositivos	30
4.1 - Taxa de Transmissão	30
4.2 - VLC	34
4.2.1 - Cross-Compile.....	35
4.2.2 - Instalação dos binários	36
4.3 - Ffmpeg	37
4.3.1 - Ffmpeg2theora.....	37
4.3.2 - Oggfwd	38
4.4 - Compilação ffmpeg2theora	39
4.5 - Filesystem Debian para ARM	39
4.5.1 - Bootloader - Uboot.....	40
4.5.2 - Kernel.....	40
4.5.3 - Formatação da Flash SD	41
4.5.4 - FileSystem.....	41
4.5.5 - SSH.....	43
4.6 - Instalação ffmpeg2theora	44
4.7 - Captura	45
4.7.1 - WebCam.....	46
4.8 - Bateria	47
5 - Resultados alcançados.....	48
6 - Conclusão	49

LISTA DE ILUSTRAÇÕES

Figura 1 - LiveU LU-30.....	14
Figura 2 - Sistema de transmissão utilizando a internet como meio.....	21
Figura 3 - Placa de desenvolvimento em processadores ARM.....	29
Figura 4 - Aplicativo Bandwidth Monitor NG.....	31
Figura 5 - Distribuição de resoluções usuais.....	32
Figura 6 - Console utilizado diretamente na placa de desenvolvimento.....	43
Figura 7 - Placa de Captura de vídeo USB.....	46

LISTA DE TABELAS

Tabela 1 - Estrutura padrão de tópicos.....	12
Tabela 2 - Soluções a serem escolhidas para o sistema.....	20
Tabela 3 - Textes de qualidade x taxa de transmissão.....	33

1 INTRODUÇÃO

Desde o surgimento da internet e principalmente com a sua popularização nos anos 90, muitos estudiosos a classificaram como a futura substituta dos meios de comunicação que até então se conhecia. Mais de vinte anos se passaram e tais meios não desapareceram, e sim incorporaram entre seus serviços como forma de ferramenta a rede mundial de computadores.

Entre os motivos pelo qual a internet não assumiu o espaço das outras mídias está a falta de mobilidade que um computador impõe. Inicialmente tratavam-se apenas dos computadores pessoais de mesa (*Desktops*) com suas conexões discadas lentas. Posteriormente proliferaram-se os computadores portáteis (*Notebooks*) e nos últimos dez anos as redes sem fio (*Wi-Fi*) e móveis (3G). Embora atualmente encontrarmos computadores portáteis e conexões móveis rápidas, ainda existem barreiras com relação à geração e o consumo de informações: a leitura física do jornal impresso ainda é mais prático do que em um *notebook*; ouvir uma emissora de rádio no carro é mais cômodo do que destinar um computador para esta função.

Porém, para o acesso a informações tem surgido muitos dispositivos portáteis que possuem características de telefones celulares e computadores portáteis. Com dimensões e processamento de uma máquina estruturada e a mobilidade e conectividade de um telefone

Esses dispositivos são conhecidos como MID - *Mobile Internet Device* – Dispositivo Móvel de Internet, e tem ganhado muito mercado após os recentes lançamentos da Apple Inc,. Tais equipamentos permitem acesso inúmeros à conteúdos presentes na internet como publicações, vídeos e músicas, bem como visualização de transmissões ao vivo de áudio e vídeo.

Contudo no que se refere à produção de conteúdo ainda se fazem necessárias grandes estruturas e diversos equipamentos para garantir agilidade e qualidade que um veículo de

comunicação preza em seu material. O objetivo desta pesquisa foi de desenvolver um sistema para a transmissão de material audiovisual com qualidade e a maior mobilidade possível, com a melhor relação custo - benefício considerando como principais fatores: a qualidade mínima requerida por uma empresa comercial que gere ou utilize de transmissões ao vivo, o tipo e a qualidade dos diversos tipos de conexões com a internet que se pode dispor, e por fim o menor custo agregado à produção contínua de conteúdo, a fim de facilitar o acesso ao produto. A tabela 1 apresenta a estrutura deste relatório.

Tabela 1 - Estrutura dos tópicos

Capítulo	Conteúdo
1. Introdução	Necessidades, oportunidades ou outras motivações que levaram ao desenvolvimento do projeto
2. Contexto do Projeto	Informações necessárias para definir o escopo do projeto; Especificações do projeto, premissas e condições de contorno
3. Análise de alternativas:	Avaliação das alternativas técnicas para implementação do projeto dentro das especificações apresentadas no capítulo anterior; pesquisa bibliográfica, justificativa técnico econômica da alternativa escolhida
4. Métodos, Processos e Dispositivos	Apresentação do projeto propriamente dito. Fluxogramas, memoriais de cálculo, circuitos, desenhos, diagramas, etc.
5. Resultados Alcançados	Descrição dos testes e resultados alcançados
6. Conclusão	Conclusões sobre o projeto realizado

2 CONTEXTO DO PROJETO

Atualmente para uma emissora de televisão realizar uma transmissão ao vivo são necessárias horas de preparação dias antes do evento que se deseja capturar. Envolvendo uma grande quantidade de profissionais e equipamentos. O local de transmissão é um fator relevante, pois deve atender as condições mínimas de visibilidade com a antena da emissora, para que se possa montar um link direcional, que consiste em uma ligação feita entre antenas de rádio frequência com ângulos de abrangência muito pequenos. Porém não se almeja no momento sobrepor a qualidade de tal sistema, uma vez que é um procedimento muito bem desenvolvido e explorado, envolvendo equipamentos de muita qualidade. Para se atingir tal qualidade com um sistema que utilize a internet como meio ainda são necessários avanços na própria rede de dados que se dispõe no país.

Para auxiliar na definição dos parâmetros deste projeto foram realizadas entrevistas com profissionais técnicos e administrativos de alguns veículos de comunicação, a fim de avaliar as soluções utilizadas e quais as características principais e quais as que se deseja encontrar em tais dispositivos.

2.1 ENTREVISTAS

Durante esta etapa de definições dos requisitos foram realizadas entrevistas, previamente com os profissionais experts da área de estudo, por meio de um questionário foram questionados os aspectos técnicos e também econômicos da proposta deste trabalho.

Segundo as entrevistas realizadas no mês de Outubro de 2010 nas empresas Rede TV!, Universo Online e o Grupo Abril, apresentarem que a solução atual utilizada por estes veículos, é o Live U, proveniente de uma companhia israelense.

2.1.1 LiveU

A empresa *LiveU* foi a pioneira e detem praticamente todo o mercado brasileiro no que se refere à soluções de *uplink* via celular para soluções de broadcast, sendo uma das mais utilizadas no mundo. A figura 1 mostra o principal produto da empresa.



Figura 1 – LiveU LU-30

Consiste em um sistema de captura de vídeo via conexão *Firewire*, presente na maioria das câmeras profissionais de vídeo na qualidade *standard* (720 x 480) que utiliza mais de uma conexão de internet móvel 3G para multiplexar e transmitir vídeos e arquivos para um servidor a fim de ser retransmitido por uma emissora.

Nas informações obtidas com os entrevistados, o modelo LU-30 consiste em um *hardware* de um micro-computador não embarcado (x86), com a captura de 4 portas IEEE-1394 (Firewire), contendo até 7 *modems* 3G, um *display touchscreen* de 7" para *preview* e configurações, bateria interna com duração máxima de 2 horas. Ainda pode-se utilizar conexões *wi-fi* ou ethernet para realizar as transmissões.

2.1.2 Rede TV!

A RedeTV! é uma rede brasileira de televisão, com instalações em São Paulo, Rio de Janeiro, Belo Horizonte, Recife e Fortaleza. A sede da RedeTV! encontra-se no CTD - Centro de Televisão Digital, situado em Osasco, na Grande São Paulo. Iniciou suas atividades no dia 15 de novembro de 1999. É a mais jovem rede de televisão aberta, dentre as cinco maiores redes do Brasil. A programação da emissora é voltada principalmente ao entretenimento, com diversos programas direcionados a segmentos específicos: humorísticos, *talk shows*, jornalísticos, esportivos, séries, programas de entrevista e femininos.

A entrevista ocorreu com o Senhor Abraão Dantas Farina, Gerente de Tecnologia Digital, responsável pela TI da emissora. Possuem no total sete equipamentos do modelo LU-30 da LiveU, e utilizam diariamente para transmissões ao vivo de todo o país veiculando seu sinal aberto e também de TV à Cabo. Realizam a transmissão direta para um servidor local instalado na emissora, utilizando um aplicativo no ambiente *Windows* também fornecido pela *LiveU* para então conectar ao *Switch* de produção.

A Rede TV ! utiliza este sistema desde fevereiro de 2010, quando adquiriram os equipamentos. As mochilas de transmissão, como assim as qualificam, possuem seis *links* de dados 3G, sendo de três operadoras de telefonia diferentes, a fim de garantir a existência de sinal.

A emissora investe no uso de tais dispositivos, apesar de ter ciência que a cobertura de internet 3G no país não é favorável. Outro ponto negativo é que uma hora e meia de bateria não é considerado suficiente pelos profissionais, que devem sempre manter proximidade com fontes de rede elétrica. Para a operação dos dispositivos foi necessário realizar um treinamento com os profissionais de campo, uma vez que existem diversas configurações e devido a cobertura de telefonia o funcionamento é incerto.

O profissional particularmente mostrou interesse na pesquisa em estudo, e por sua posição na empresa tem condições de incentivar a aquisição de novas tecnologias, caso a pesquisa torne-se um produto de mercado.

2.1.3 Universo Online

O Universo Online (UOL) é um provedor de conteúdo e um provedor de acesso à Internet brasileira, que foi criado pela empresa Folha da Manhã, que edita o jornal Folha de S. Paulo.

No UOL o entrevistado foi Senhor Derek Sismotto Flores, responsável pelo UOL Notícias, ligado a TV UOL. A empresa utiliza o mesmo equipamento anteriormente citado, porém pelas características do seu mercado faz um uso curioso desta tecnologia. Sendo um Provedor de Internet bem como um Portal de Notícias, o *site* do Uol necessita de imediatismo, porém que o conteúdo seja sob demanda. A empresa não busca, na maioria do tempo, uma transmissão ao vivo, o que torna a princípio o uso de um sistema de transmissão incoerente.

Contudo a cidade de São Paulo, devido as suas dimensões, acarreta um aumento no tempo para geração de conteúdo pós produzido. Um exemplo dado pelo profissional, foi um caso de um acontecimento recorrente na cidade, como um prédio em chamas sendo apagado pelos bombeiros em um ponto específico da cidade. Nesta situação se fosse enviar um profissional (*câmera man*) para registrar as imagens e aguardá-lo retornar com o material gravado para edição e inserção no portal da emissora., haveriam transcorrido no mínimo duas horas do fato, considerando deslocamento e tráfego. Para solucionar tal problema, seguindo o mesmo exemplo, o profissional é enviado com a mochila (conforme citado no item 2.1.1) quando realiza a captura e transmite ao mesmo tempo, porém somente para o servidor do Portal, que em tempo real edita a matéria e dispõe o vídeo sob demanda no seu site.

Os principais pontos negativos que Senhor Flores apontou foram: a baixa autonomia, indicando a duração de bateria como no máximo uma hora e trinta minutos, e as dificuldades com o sinal de telefonia 3G. O profissional adicionou que em um evento onde há grande concentração, principalmente de profissionais da área de comunicação, com muitos telefones e computadores com conexão de internet 3G, o funcionamento do sistema *LiveU* é praticamente nulo.

2.1.4 Grupo Abril

O Grupo Abril é um conglomerado brasileiro de mídia, com sede em São Paulo. O grupo é proprietário de algumas editoras, entre elas a Editora Abril, responsável pela revista *Veja*, a Editora Ática e a Editora Scipione, estas especializadas em livros didáticos. Possui participação em outras áreas como televisão, onde é proprietária da MTV Brasil e da TVA. É o segundo maior grupo de comunicação do Brasil perdendo apenas para as Organizações Globo.

Nesta empresa foram realizadas várias entrevistas, desde cargos administrativos à técnicos. Primeiramente conversou-se com Senhor Daniel Trocoli, no cargo de Consultor de Marketing, responsável pela contratação desta tecnologia. Embora o Grupo Abril trabalhe principalmente com conteúdo impresso de revistas, possui também um Portal na internet, onde utiliza a tecnologia para transmissões ao vivo de fatos relacionados aos seus produtos. São locados dois equipamentos através de contrato anual, sendo um destinado unicamente à Revista *Veja*, e outro utilizados pelas demais Revistas.

O Grupo Abril optou por utilizar um servidor externo, uma vez que suas transmissões são unicamente para a internet, não necessitando assim da infraestrutura de dados para servir o conteúdo. Inicialmente foi utilizada uma empresa brasileira, contudo a operação foi

considerada insatisfatória, e então iniciaram a operação com um Portal localizado nos Estados Unidos chamado *Live Stream*. O *Live Stream* possui acordos com o fabricante do *LiveU* para fácil configuração de seu serviço. Por tal uso o Grupo Abril paga um espaço para a postagem de conteúdo sob demanda, mais um limite de horas mensais para transmissão ao vivo, e um adicional caso tal limite seja transposto. O profissional apontou que existem muitas restrições de tecnologias de câmeras a serem utilizadas com o sistema locado, bem como muitas vezes o atraso entre a transmissão e a recepção do sinal chega até a quarenta segundos, o que representa uma característica negativa do seu uso.

Também foi entrevistado o profissional Senhor Carlos Eduardo Jorge, jornalista que trabalha diretamente na Revista *Veja*, responsável pela configuração e operação das transmissões ao vivo realizadas tanto com o *LiveU* como por Computadores. O Senhor Carlos Eduardo Jorge, considera a qualidade do material obtido com a *LiveU* suficiente, ainda mais por se tratar de uma veiculação apenas para internet, porém denotou também que a duração da carga da bateria é insuficiente para uma produção externa e que a conexão é quase sempre incerta devido as redes de telefonia móvel. Com relação à resolução e taxas de transmissão adotadas pelo Portal, eles se assemelham aos definidos a esta pesquisa, uma vez que o foco é vídeos para internet.

Por meio do profissional Helio Miyake chegou-se a Junior João Golçalves, que foi o responsável pelo projeto de implantação da tecnologia vigente de *streaming* utilizada pelo Grupo Abril. Ele explicou como foi o processo de escolha da tecnologia, e que embora existam muitos pontos negativos do seu uso, é uma solução necessária para a empresa atualmente. Mostrou também bastante interesse no produto deste projeto, encaminhando a descrição realizada e suas características para Senhor Manoel Lemos, Diretor de TI do Grupo Abril.

Utilizando-se de correio eletrônico, posteriormente a realização das entrevistas, houve um retorno do Grupo Abril com interesse de testar e avaliar a solução final desta pesquisa, para possíveis aquisições futuras.

2.2 DEFINIÇÕES

Dentro das possibilidades e utilizando as orientações obtidas no decorrer das entrevistas o produto desejado tem os seguintes requisitos:

- Qualidade – Buscou-se criar um sistema que tenha uma qualidade no produto final mínima esperada para uma solução comercial de audiovisual, considerada a resolução *standard* como referência;
- Baixo custo – Uma vez que existe um sistema profissional aprovado e largamente utilizado que apresenta esta mesma solução, um dos grandes diferenciais desta pesquisa é o baixo custo agregado ao sistema e sua utilização. Distinguindo-se da solução atual, possibilitando não apenas o uso comercial como também de treinamento e educacional. e para soluções comunitárias e sem fins lucrativos. O principal custo encontra-se na placa de desenvolvimento, que representa menos de R\$ 300,00;
- Mobilidade – Existindo equipamentos importados que realizem operações similares, teve-se como meta igualar-se neste requisito. Além de eliminar a necessidade de preparações prévias, bem como as limitações de locações, uma vez que praticamente todo território das regiões metropolitanas, onde concentram-se a grande maioria das transmissões ao vivo, possuem cobertura de banda larga móvel;
- Autonomia – Poucos sistemas de transmissão hoje trabalham utilizando baterias ou outra forma de energia que não seja da rede elétrica pública, pela

quantidade de energia exigida pela transmissão em si. O produto desenvolvido é mantido por baterias próprias, o que possibilita a mobilidade desejada, porém também possui um baixo consumo que consegue obter um maior rendimento da carga disposta com o menor peso possível, proveniente das baterias;

- Operacionalidade – A fim de reduzir a estrutura necessária para se realizar uma transmissão ao vivo de uma locação externa, não apenas em equipamentos, mas também em profissionais, buscou-se um sistema de fácil operação podendo esta ser realizada pelo próprio operador de câmera, com comando simples e resposta objetivas, possibilitando uma rápida interação com o sistema;
- Compatibilidade – Aspirou-se não criar restrições com relação ao tipo de equipamentos e redes de dados o sistema poderia operar, a fim de não ter de adequar o usuário aos requisitos mas utilizar os protocolos e padrões mais usuais presentes no mercado.

Com a definição dos requisitos para o projeto é possível a escolha das soluções que se deseja agregar ao sistema, que se encontram na tabela 2, e embasadas no Capítulo 3, Análise das Alternativas.

Tabela 2 – Soluções a serem escolhidas para o sistema

Recurso	Descrição
3.1 Protocolo	Tipo de encapsulamento utilizado para se trafegar na internet com o conteúdo
3.2 Servidor	Programa utilizado para receber e retransmitir o sinal
3.3 Cliente	Programa utilizado para a visualização do material transmitido
3.4 Encoder	Hardware que irá realizar a captura, codificação, encapsulamento, conexão com a internet e envio do material

3 ANÁLISE DE ALTERNATIVAS

O projeto consiste em montar uma estação transmissora multimídia portátil, utilizando como fonte o sinal de vídeo gerado por uma câmera filmadora e como meio a internet. O serviço pode ser utilizado para um sistema de *multicast*, que consiste em enviar o mesmo serviço a vários destinatários, utilizado na forma de *webcast*, ou seja, visualização via internet. Sendo esta a fonte principal do sinal, ou com apenas um destino, para ser mixado com outros sinais e efeitos por parte de uma emissora ou produtora de vídeo, na forma de *broadcast*.

A imagem 2 ilustra as possibilidades de funcionamento, onde se pode notar que a diferença ocorre sempre após o servidor que irá receber a transmissão inicial.

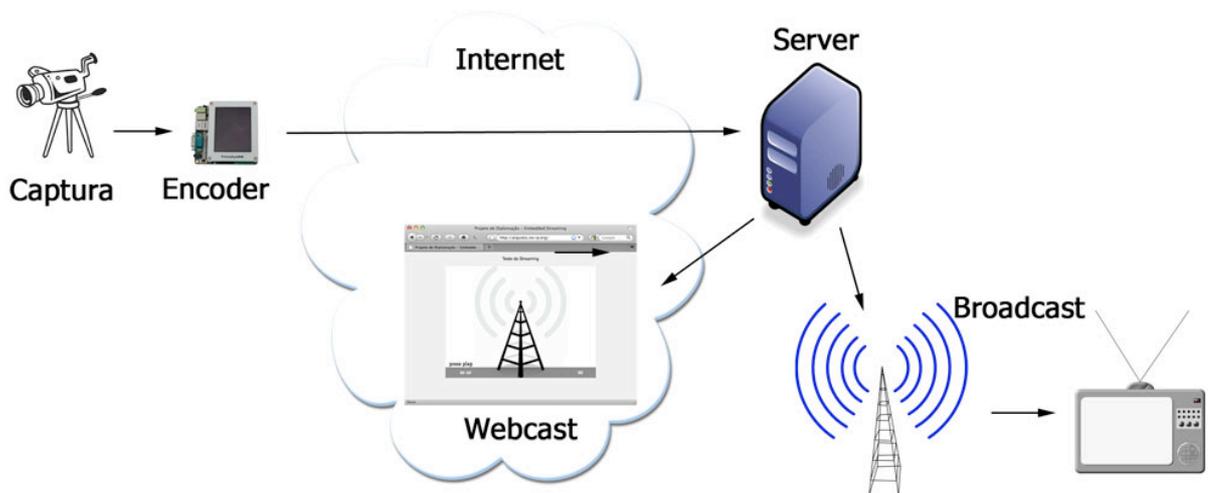


Figura 2 – Sistema de transmissão utilizando a internet como meio

A seguir são especificadas as escolhas feitas para o projeto, com base nas definições anteriormente citadas.

3.1 PROTOCOLO

A escolha do formato de encapsulamento do sinal, conhecido como protocolo, teve sua base principalmente na abrangência que o formato tem, no número de diferentes usuários poderia alcançar. Embora o formato *Flash* seja o mais utilizado na atualidade, mérito do *website Youtube* que o utiliza, possui um consumo de banda acima dos demais, além de ser um formato proprietário, bem como também são o *Real Player*, o *Quick Time* e o *Windows Media Video*. Estes outros ainda apresentam o inconveniente de serem suportados apenas em um sistema operacional específico nativamente.

Foi então escolhido o *Ogg* [1] pela sua característica mais marcante em relação aos outros formatos, que é ser composto de código aberto e livre (*Open Source*). Embora não seja nativo para a maioria dos sistemas operacionais, não se faz necessário este requisito para este protocolo em específico como será descrito no item 3.3 deste documento.

Outra grande vantagem de se utilizar software livre para o desenvolvimento é a gama de aplicativos compatíveis já desenvolvidos ou ainda em pesquisa, e com isso toda a comunidade de desenvolvedores sempre dispostos a auxiliar nas dúvidas e desafios dos que atuam no meio.

O próprio sistema operacional utilizado no dispositivo é uma versão personalizada customizada do *Linux*, sistema operacional que é o principal divulgador das Licenças GPL (*General Public Licence*) responsáveis por resguardar tais códigos e sistemas.

Dentro do encapsulamento ainda existem os *codecs* de áudio e vídeo, que são os programas responsáveis pela codificação e decodificação do sinal. O encapsulamento *Ogg* traz junto o *codec* de vídeo *Theora* [2] e o de áudio *Vorbis* [3]. Para se fazer a transmissão utilizando o encapsulamento *Ogg* é necessário que haja um software que faça a codificação do sinal desejado para o formato *Theora/Vorbis*, e para se assistir tal transmissão é necessário um software cliente que interprete tais protocolos.

Para realizar os testes iniciais, a partir de um sistema x86 (computador de uso pessoal) foi utilizado o aplicativo *VLC Media Player* [4] que é capaz de servir tanto como encoder do sinal como cliente para sua visualização. Este aplicativo é multiplataforma, ou seja, pode ser utilizado em praticamente qualquer sistema operacional comercial ou não comercial, existentes atualmente. Sendo que foi utilizado tanto nos sistema OS X da Apple Inc. como em versões de Sistema Operacional *Linux* como as distribuições Ubuntu 10.04 e Mandriva 2010.

3.2 SERVIDOR

Para se fazer realizar uma transmissão *streaming* pela internet é necessário ter uma origem ou um destino conhecido. Ou o sinal é enviado de um local conhecido, e qualquer cliente pode selecionar de onde quer receber o sinal, semelhante aos canais de televisão que estão fixos em uma rádio frequência específica, ou a máquina que deve receber permanece aguardando novas conexões para reproduzir. Neste caso é necessário identificar o seu local, que seria o endereço de IP (*IP Address* é o endereçamento numérico que toda máquina conectada a uma rede de computadores recebe para realizar a troca de dados), para então enviar.

Este requisito presente, e toda e qualquer troca de dados na internet, ainda obedecem à dois requisitos presentes no sistema de redes de computadores, que são: o Ip Válido e o Ip Fixo.

O Ip Válido significa que a máquina ou dispositivo que deseja enviar ou receber o sinal esteja ligado diretamente à internet, sendo seu endereço de Ip visível para qualquer outra máquina ou dispositivo conectados também à internet. Porém, a prática mais comum consiste em utilizar apenas um Ip Válido para cada rede interna de residências e empresas, dividido entre vários Ips locais através de roteadores, isto por economia em se tratando de contratação de serviços de dados e por proteção dos próprios usuário contra outros softwares e usuários mal intencionados. Para realizar a troca de dados uma das pontas da troca deve ter Ip Válido.

Uma alternativa é utilizar de tabelas de roteamento para se levar o sinal desejado a máquina escolhida, mas costuma adicionar fatores de erro ao serviço desejado.

O Ip Fixo é um tipo de serviço que deve ser contratado separadamente com as prestadoras, o que acarreta maiores custos. A única forma de se realizar a troca de dados é possuindo uma das pontas da comunicação conhecida, o que faz com que este serviço seja mais caro, pois as redes trabalham normalmente com endereços aleatórios. Cada vez que a modem é reiniciado o endereço se altera. Existem algumas explicações para tal fato, a primeira é a falta de endereços suficientes para todos os usuários, o que leva a alteração permanente a cada requisição, e a outra é de que sem um Ip Fixo é difícil prover um serviço aos outros, motivo pelo qual as operadoras cobram mais por tal aplicação. Uma solução para tal questão é a utilização de DNS Dinâmico (*Domain Name Service* – Serviço de Nomes de Domínios), que é um serviço hierárquico de nomeação que atualiza em um servidor externo o endereço de rede que se deseja saber toda vez que ele é alterado, porém mantendo sempre o mesmo nome publicamente, como é o caso do <http://argusr.no-ip.org> que sempre se destina a mesma máquina, mesmo que o serviço de internet seja reiniciado e o endereço alterado.

Ao se tratar de um serviço em que nenhum dos pontos são fixo ou conhecidos a solução é utilizar um ponto intermediário, conhecido como servidor. Caso a situação desejada seja enviar o mesmo sinal para vários clientes o servidor torna-se indispensável. Alguns protocolos ainda suportam que o emissário projeta o sinal para todos os clientes, mas considerando as condições que normalmente se apresentam nos *links* disponíveis e/ou móveis não é aconselhável, pois pode tornar o serviço instável pelo excesso de clientes.

Para o encapsulamento escolhido foi utilizado o servidor de *Icecast* [5], originalmente criado e largamente utilizado para serviços de *streaming* de áudio em *Web Rádios*. A partir da versão 2.2.0 o servidor passou a suportar vídeo em *Ogg* no *codec Theora*, o que tornou o mais indicado para esta aplicação.

É de fácil configuração e não demanda manutenção, sendo que novas conexões são controladas pelos encoders, sem novas configurações e necessitando apenas a senha registrada inicialmente. Para os testes realizados o servidor foi instalado e configurado apenas uma vez, passando-se meses de uso e testes sem necessárias manutenções. Embora a máquina utilizada não possua Ip Fixo por ser uma conexão do tipo residencial, esta cadastrada no serviço de *DNS Dinamico* no host www.no-ip.org. Como não possui nenhum roteador em seu caminho a rede externa recebe o Ip Válido, não necessitando de tabelas de roteamento, e sim apenas liberação na segurança da máquina, o seu *Firewall*.

Com isto é possível originar e receber o sinal de qualquer máquina ligada a rede mundial de computadores, mesmo que esteja atrás de roteadores, uma vez que existe um ponto fixo para se enviar o sinal e um ponto fixo, sendo o mesmo, para se reenviar o sinal.

Com a difusão das *Web Radios* pelo mundo criaram-se muitos servidores de *Icecast* (protocolo de software livre de transmissão multimídia) disponíveis aos usuários da internet, alguns pagos que oferecem inúmeros recursos, porém outros gratuitos, que trazem restrições como o número de conexões ou a qualidade máxima. Outros não oferecem restrições sobre a qualidade ou número de espectadores, mas como única exigência de que seja utilizado o *player* embarcado em um site específico, onde são postados anúncios responsáveis pela manutenção financeira do serviço.

3.3 CLIENTE

Uma das principais adversidades da troca de dados na internet é o fato de que ambos os lados da comunicação terem os mesmos requisitos para que a comunicação seja eficiente. Neste caso trata-se dos codecs já mencionados, que são necessários tanto para quem codifica como para quem decodifica o sinal.

Embora o protocolo e codecs escolhidos sejam de código aberto e gratuitos, não significa que todos fabricantes os tenham incluído em seus produtos. Porém cada usuário pode simplesmente instalar os pacotes distribuídos livremente na internet.

Para simplificar este problema, da presença ou não dos codecs necessários, é proposto o uso de um *media player html* presente em um site na internet. Basta o usuário acessar um *link* e o *player* já está presente, sem a necessidade de utilizar aplicativos da máquina, que podem também não estar presentes de fábrica. É claro que o usuário não estará livre de ter alguns requisitos mínimos em sua máquina, mas busca-se solucionar este problema para o maior número de usuários conjuntamente.

Inicialmente foi testado e aprovado um *player* chamado *Cortado Java Applet*[6] que é um aplicativo java para leitura de *streamings Ogg/Theora* embarcado em um *html*. O *software* teve um bom desempenho, mas, não apresentava opções de controles como *Full Screen* (tela cheia), entre outros. Para ser utilizado necessita da instalação prévia da *Java Virtual Machine*, todavia como é exigida pelos sites de *Home Banking* já se encontra presente em muitos computadores pessoais.

Foi escolhido então outro *player* embarcado chamado *Oiplayer* [7] que utiliza uma tecnologia recente chamada HTML5 que vem sendo aplicada nativamente nos navegadores em geral. Uma grande vantagem deste *player* é o recurso chamado *FallBack*, que significa literalmente “cair para”, ao *player* indicado anteriormente Cortado Java Applet. Então é utilizada uma tecnologia nova, o HTML5, e caso ela não esteja presente o usuário já é

automaticamente direcionado a solução Java. O *Oiplayer* ainda pode direcionar o usuário a uma fonte *Flash*, que é outro protocolo de **streaming** largamente utilizada, mas possui como desvantagens ser de uso comercial, logo possui custo, e de consumir mais espaço na banda da conexão utilizada. Por estes motivos este protocolo não foi escolhido frente ao *Ogg*.

Para uma aplicação profissional o *player* embarcado em html não é necessário, sendo melhor utilizar outros aplicativos de maior qualidade e que tenham melhores controles sobre o sinal apresentado. Este protocolo funciona em todo e qualquer aplicativo de visualização de vídeos que tenham os devidos codecs instalados, em qualquer sistema operacional, sem restrições. Por exemplo, para o uso em uma emissora, deve-se utilizar um aplicativo que permita a visualização em tela cheia do computador, para que quando o vídeo for misturado com as demais fontes não sejam necessários recortes e ajustes.

3.4 DISPOSITIVO ENCODER

Todo este procedimento de envio de material multimídia através da internet pode ser realizado hoje facilmente com um microcomputador pessoal (PC) que tenha as entradas necessárias para se capturar um vídeo, mesmo que seja de uma *webcam*, que se tornou praticamente nativo a todo PC fabricado atualmente.

O desafio proposto foi de realizar esta atividade em um dispositivo pequeno e portátil, que tenha as características mínimas necessárias. Atualmente existem no mercado alguns dispositivos importados que realizam este tipo de transmissão, mas são de custo elevado, fisicamente pesados e de pouca autonomia em horas de funcionamento. Consistem basicamente de um hardware de um *desktop slim* com uma tela *touch screen* (sensível ao toque) e utiliza de vários *modems* de internet móvel 3G, para obter melhor qualidade e redundância na operação.

O objetivo deste trabalho não é sobrepor à qualidade de um produto já comercializado atualmente, mas de apresentar uma alternativa com melhor relação entre qualidade e custo do serviço. Ainda sobre qualidade, tais dispositivos foram desenvolvidos para mercados como o Norte Americano e Europeu, que possuem redes de banda larga 3G com qualidade e confiabilidade, que não é o caso do Brasil. Várias empresas nacionais de grande porte já utilizam tais equipamentos, na sua maioria em condições de teste, e revelam que por condições da rede de telefonia móvel nacional o desempenho destes produtos não justifica o seu uso pela falta de confiabilidade do sinal.

Como plataforma para desenvolver o encoder do sinal de vídeo foi escolhida a placa de desenvolvimento *Friendly Arm – Mini 2440* (Figura 3) [8]. Tal escolha baseia-se na disponibilidade de informações referente a este dispositivo, não apenas do fabricante como a comunidade de desenvolvedores que o adotou como uma ótima plataforma para realizarem seus projetos.

Possui um processador da família Arm 9 e uma série de IOs (drivers de entrada e saída). Os principais motivos da escolha deste *hardware* foram a disponibilidade, uma vez que havia sido adquirida anteriormente a realização do projeto, sua versatilidade, baixo custo mesmo considerando as tarifas de importação, tamanho reduzido tendo apenas 10 cm x 10 cm de área, um display de LCD com *touch screen* de 3.5”, uma porta *usb host* permitindo a conexão de outros dispositivos de conexão e captura, além de conexões serial e ethernet, bem como uma vasta documentação e a possibilidade de se obter suporte ao desenvolvimento dentro da própria universidade, devido as linhas de pesquisa e ensino realizadas por alguns professores. Ainda existe a possibilidade do uso do modelo *Micro2440*, que possui o mesmo hardware de processamento porém sem os IOs, sendo necessária a instalação posterior, mas reduz mais ainda o custo e o espaço ocupado.



Figura 3 – Placa de desenvolvimento em processadores ARM

Porém o projeto não se restringe a este hardware, possuindo uma instalação versátil, é adaptável a praticamente qualquer outro dispositivo de aplicação embarcável com um processador de 32bits e os *drivers* de entrada e saída necessários para a captura e transmissão do sinal.

4 MÉTODOS, PROCESSOS E DISPOSITIVOS

Uma vez escolhidas as características básicas do sistema que se deseja foram iniciadas as tarefas de construção e testes. Alguns requisitos de mercado podem não se aplicar as necessidades deste projeto, ou por não se dispor dos mesmos recursos, como uma exigência de qualidade de vídeo *High Definition*, ou mesmo por poder consumir menos recursos, como bateria e taxa de transmissão, por exemplo.

Aqui se descreve o trabalho prático realizado, bem como as rotinas de testes e os caminhos tomados durante a execução do projeto.

4.1 – TAXA DE TRANSMISSÃO

Uma vez escolhidos o encapsulamento e o protocolo a ser utilizado, foi necessário uma sequência de testes para se definir a resolução e a qualidade das codificações de áudio e vídeo, que tivessem a melhor relação entre qualidade final e a taxa de transferência.

Como o meio utilizado é a internet, e em se tratando do uso de *links* de internet móvel como o 3G (rede celular), ou até mesmo de conexões de banda larga como ADSL (linha telefônica) ou *Cable* (cabo de TV por assinatura), não se pode garantir uma taxa de transferência de envio (upload) maior que 10% da máxima contratada, segundo as regras vigentes atualmente para as operadoras. Considerando que o link mais usual na atualidade seja o de 1Mbps, e a máxima garantia fornecida pelos provedores, ainda dividido entre download e upload, estima-se em 50kbps a taxa garantida máxima de envio de dados.

Como limite superior da escolha de resolução e compressão foi utilizado a taxa de 50kbps de transmissão, e como limite inferior foi considerado a qualidade do vídeo recebido. Os testes foram realizados utilizando o software VLC na plataforma Mac OS X 10.6/ *MacBook Pro 15"* com uma conexão via cabo ethernet na configuração GigaBit (1000Mbps) com o servidor *Icecast* na máquina Linux/Mandriva 2010. Foi utilizada esta montagem para

garantir que a conexão não seria o limitante em se tratando de velocidade e qualidade dos testes.

Para a visualização da taxa de transferência foi utilizado o *software Bandwidth Monitor NG* [9] que é executado diretamente no console do sistema que se deseja observar, neste caso o servidor *Linux*, configurado para mostrar a taxa média dos últimos trinta valores apresentados. A figura 4 ilustra o *software* em operação.

```

bwm-ng v0.6 (probing every 0.500s), press 'h' for help
input: /proc/net/dev type: avg (30s)
/
=====
      iface          Rx          Tx          Total
=====
      lo:            0.14 KB/s    0.14 KB/s    0.27 KB/s
      eth1:          0.00 KB/s    0.00 KB/s    0.00 KB/s
      wlan0:         6.55 KB/s    49.68 KB/s   56.24 KB/s
      eth0:         50.13 KB/s    3.33 KB/s    53.46 KB/s
=====
      total:        56.82 KB/s   53.15 KB/s  109.97 KB/s
=====

```

Figura 4 – Aplicativo Bandwidth Monitor NG

Dentre as possibilidades de resolução para a transmissão do vídeo, o formato mais usual na internet é 320x240 *pixel*, porém com o desenvolvimento de novas codificações os tamanhos vem aumentando, até mesmo a qualidade conhecida como HD 1080, que consiste em uma resolução de 1920x1080 *pixel*. A figura 5 apresenta um esquema das resoluções mais usuais para vídeo em geral [10].

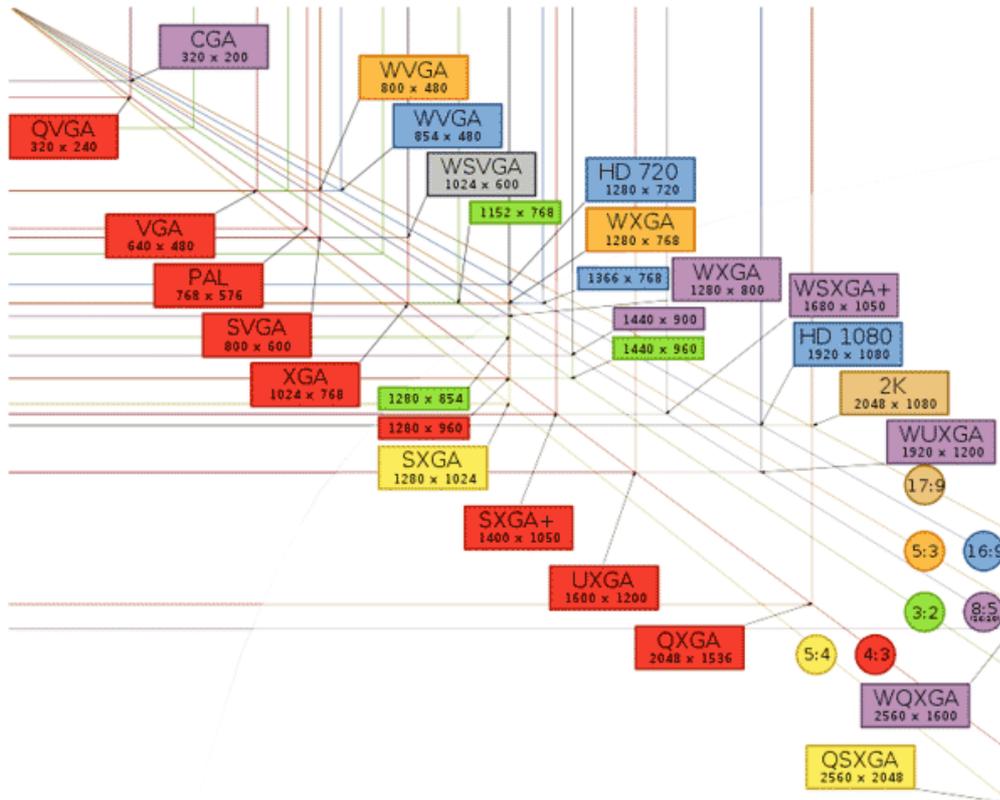


Figura 5 – Distribuição de resoluções usuais

Foram então testadas para as resoluções desde abaixo de QVGA até SVGA para a relação 4:3 (*Standard*) e WVGA e HD 720 e ainda a metade da última citada para a relação 16:9 (*Widescreen*). Para cada uma das resoluções foram variados independentemente a taxa de compressão de áudio e de vídeo. Os valores e as observações sobre todos os testes encontram-se na tabela 3.

Tabela 3 – Textes de qualidade x taxa de transmissão

	Nome	Resolução		Rel	Theo	Vorb	\overline{kbps}	Comentário			
1		160	120	4:3	56	32	10	Ruim, sem áudio			
2					64	64	17	Pixelado			
3					128	64	23	Visível porém com baixa qualidade			
4					256	64	26				
5					512	64	29				
6								1024	64		Sem alteração no áudio
7								64	128	25	
8	QVGA	320	240	4:3	64	64	17	vídeo ruim, vorbis error			
9					128	64	25	ok			
10					256	64	40	vídeo bom, truncando			
11					512	64	47	vídeo ótimo, início ruim			
12	VGA	640	480	4:3	64	64	13	instável			
13					128	64	19	ok			
14					256	64	31	ok			
15					512	64	53	ótimo			
16	PAL	768	576	4:3	64	64	6	não inicia			
17					128	64	13	tranca			
18					256	64	20	tranca			
19					512	64	30	tranca			
20	SVGA	800	600	4:3	64	64	6	tranca			
21					128	64	13	tranca			
22					256	64	20	tranca			
23					512	64	37	tranca			
24	WVGA	854	480	16:9	64	64	6	tranca			
25					128	64	13	tranca			
26					256	64	22	bom - tranca			
27					512	64	32	ótimo - tranca			
28	HD 720	1280	720	16:9	64	64	1.6	não inicia			
29					128	64	6.4	Tranca			
30					256	64	10	não inicia			
31					512	64					
32	HD / 2	640	360	16:9	64	64	8	tranca			
33					128	64	22	ok			
34					256	64	37	ok			
35					512	64	64	ótimo			

Nos primeiros testes, realizados na resolução de 160x120, exatamente a metade da mais usual, variando-se primeiramente a compressão de áudio, observou-se que para 32kbps não ocorria a transmissão do áudio. Fixou-se então a taxa mínima de áudio para 64kbps de compressão. Na mesma rotina de experimentos, ainda para a resolução de 160x120, aumentou-se a compressão do áudio para 128kbps, porém não se observou diferença na

qualidade do som. Após este teste fixou-se 64kbps como a qualidade ideal para o áudio, motivo pelo qual não foi variado nos outros experimentos.

Também neste primeiro *set* de experimentos definiu-se como limite inferior de compressão de vídeo para os testes a taxa de 64kbps, e como superior a taxa de 512kbps, sendo a primeira como mínima necessária para que o encoder de vídeo funcionasse, segundo o console do próprio aplicativo VLC, e o segundo limite em função da definição de máxima transferência de dados em 50kbps, sendo que compressões acima do valor definido apresentavam taxas de transferência acima da desejada.

Desta rotina de experimentos observou-se então que a melhor configuração para um vídeo *Standart* seria a resolução VGA (640x480) e para um vídeo *Widescreen* a metade resolução HD 720, ambas com a compressão de vídeo em 128kbps e de áudio em 64kbps. Dependendo da garantia da conexão de internet presente para a transmissão ainda é possível aumentar a qualidade do vídeo para 256kbps de compressão.

4.2 VLC

Como primeira solução para o sistema foi escolhido o uso do software VLC para - realizar a compressão e envio do sinal de vídeo e áudio desejados. Por se tratar de um *Software* Livre foi possível se obter os códigos fontes [4] para então apropriar o programa ao dispositivo alvo, neste caso a placa *Friendly Arm* (alvo), em um procedimento conhecido como compilação cruzada (*cross compile*).

Após a compilação havia duas opções de se instalar o programa no dispositivo alvo: carregar a imagem dos sistemas de arquivos presente na placa e adicionar os binários do aplicativo; ou utilizando a placa com seu sistema rodando arrastar os binários por uma conexão e depositá-los no local específico. A segunda opção foi adotada, uma vez que o dispositivo alvo já se apresentava com um servidor de troca de arquivos (ftp) funcional, e para

tornar possível a primeira opção seria necessário recopilar o *kernel* do sistema *host* para permitir a leitura do tipo de imagem do sistema de arquivos utilizada no sistema alvo, que se trata de um procedimento complicado e demorado em função de se tratar de uma distribuição x86.

4.2.1 *Cross-Compile*

Tal função é necessária quando o alvo não possui espaço suficiente, ou quando seu processamento levaria muitas horas a mais do que realizando em um computador de plataforma x86 (host). Para esta tarefa são necessários o compilador, aqui utilizando o *arm-linux-gcc*, que é o compilador da linguagem C para os dispositivos da plataforma ARM de processadores, os códigos fonte do programa que se deseja compilar, e uma série de bibliotecas referentes a compilação e as funcionalidades desejadas ao programa final.

O compilador ARM utilizado foi o *arm-linux-gcc-4.3.2* instalado em um *host Ubuntu* 10.04.1, os fontes do VLC são da versão 1.1.4, e das bibliotecas utilizadas ressalta-se a de *sout* responsável pela possibilidade de transmissão do *vlc*, *vorbis*, *theora* e *ogg* responsáveis pelos codecs e container respectivamente.

Uma vez formatado o ambiente para a compilação, esta foi realizada inicialmente apenas acrescentando os pacotes que se deseja utilizar, sem excluir nenhum outro, deixando a customização para uma próxima etapa. Dente as dificuldades de se gerar os binários para esta solução esteve em como cria-los sem que se misturassem ao sistema de arquivos do sistema *host*, tendo ocorrido durante as tentativas até mesmo a inutilização da instalação do *host*.

Foi possível a utilização ágil de mais de um sistema operacional por meio de um aplicativo que gera máquinas virtuais, todas com sistemas de arquivos independentes permitindo situações diferentes de ambiente, configurações de software e instalações de

hardware. O sistema padrão da máquina utilizada é o *Mac OS X*, e o aplicativo que cria as máquinas virtuais utilizado foi o *Parallels Desktop 6*.

4.2.2 Instalação dos Binários

A placa alvo possui vários tipos diferentes de conexões, como a serial RS232 utilizada para comandos, console e *debug*, bom como a ethernet e *usb*, ambas utilizadas para troca de arquivos. Para se fazer a instalação de *kernel e filesystem* (sistema de arquivos) é mais comum se utilizar a *usb*, porém quando o sistema operacional encontra-se ativo os protocolos de rede ip através da porta *ethernet* permitem a troca via *ftp*, mais ágil.

Em uma máquina virtual foi instalado uma distribuição *Windows XP*, necessária para utilizar o Hiperterminal responsável por comandos de serial RS 232 e também aplicativo proveniente do fabricante do sistema alvo para a instalação de *kernel e filesystem*. Através do Hiperterminal foi aberta uma janela de console onde se pode definir uma senha para o usuário *root* no sistema operacional ativo na placa alvo, necessário para a troca de arquivos.

Por meio de uma rede ip foi possível utilizar o protocolo de transferência de dados conhecido como ftp, que vem originalmente instalado na distribuição *linux Qtopia* da placa de desenvolvimento utilizada. Configurando endereços de ip pertencentes a mesma sub-máscara de rede foi utilizado o cliente de *ftp Fireftp* do navegador *web Mozilla Firefox* (no host sob o *Mac OS X*) para se conectar ao sistema alvo e ter acesso a toda sua árvore de diretórios.

Embora todos binários referentes ao programa VLC encontram-se no diretório */usr* as pastas subsequentes são pré-existent no sistema, contendo os binários de aplicativos já instalados. Este foi a principal dificuldade da geração dos binários, pois no sistema host existem muitos aplicativos nativamente no sistema, e a simples compilação do VLC, sem indicar um local diferenciado (*--prefix*), torna maçante a busca dos arquivos necessários.

As primeiras tentativas de gerar os binários em locais diferenciados resultaram em uma instalação com erros nos *links* do programa quando instalado no sistema alvo, o que tornou inválido o seu funcionamento.

Após várias tentativas de compilar o VLC para a plataforma ARM, com diferentes versões de binários, bem como diferentes versões do *toolchain* (*arm-linux-gcc*), e ainda algumas postagens no Fórum de desenvolvedores da Vídeo Lan [<http://forum.videolan.org/viewtopic.php?f=13&t=82971> com o usuário ArgusR] foram encontrados problemas intransponíveis até o presente momento, o que levou a busca de outra solução (aplicativo) para a captura, compactação e transmissão do sinal de áudio e vídeo.

4.3 FFMPEG

Na procura por uma nova solução para o *streaming* de vídeos no formato *theora* foi encontrada um aplicativo desenvolvido pela empresa Holoscópio de Minas Gerais – Brasil chamado *Landell*, antes conhecido como SLTV, destinado a edição e transmissão de vídeos para eventos. Lendo sua documentação descobriu-se que utiliza de alguns programas baseados no *ffmpeg* para fazer a conversão para *Theora*, e posteriormente o encapsulamento em *Ogg*.

Ffmpeg é um solução multiplataforma em *Open Source* para gravar, converter e transmitir áudio e vídeo. Para a conversão em *Theora* é utilizado o *Ffmpeg2theora*, e para a transmissão em *Ogg* o aplicativo *Oggfwd*.

4.3.1 *Ffmpeg2theora*

Consiste em um conversor para criar arquivos audiovisuais no formato *Ogg Theora*. Possui apenas uma interface por linha de comando, com opções para capturar sinais de entrada no dispositivo ou mesmo arquivos de áudio e de vídeo.

Foram realizados testes na plataforma *Ubuntu* (x86) mostrando um ótimo desempenho, muitas opções de controles como resolução, taxas de compressão de áudio e de vídeo bem como qualidade desejada de áudio e de vídeo. Apresenta ainda a opção de salvar o arquivo captura e convertido no disco local e também a inserção de metadata, que seria a identificação de origem, autor e descrição do conteúdo visualizáveis nos aplicativos clientes que irão receber o sinal.

Como exemplo foi executado o seguinte comando:

```
$ ffmpeg2theora /dev/video0 -f video4linux2 -x 320 -y 240 --inputfps 10 --videobitrate 30 --audiobitrate 30 --channels 1 -o -
```

Onde *ffmpeg2theora* executa o programa; */dev/video0* referência ao dispositivo de captura desejado, neste caso uma placa de captura de vídeo usb EasyCap; *-f video4linux2* especifica o formato de entrada do vídeo; *-x 320 -y 240* a resolução de saída; *--videobitrate 30 --audiobitrate 30* as taxas de compressão de áudio e vídeo; *--channels 1* uma saída mono.

A configuração é bem intuitiva, e uma vez conectada no servidor é onde são feitas as alterações para se obter o melhor desempenho de conexão e qualidade.

4.3.2 *Oggfwd*

É um aplicativo muito simples que transporta um vídeo em *Ogg* e o conecta a um servidor *Icecast*. O comando necessário para seu funcionamento é:

```
$ ffmpeg2theora ... | oggfwd argusr.no-ip.org 8000 senha /tcc
```

Onde *ffmpeg2theora ...* é o comando utilizado para captura e comprimir o sinal desejado; *| oggfwd* gera o pipe (container) e chama o aplicativo para a conexão; *argusr.no-ip.org 8000* indicam o servidor e a porta para a conexão; *senha /tcc* representa a senha para o

servidor e *mount point* (local de montagem desta transmissão, sendo que o mesmo servidor pode receber inúmeras transmissões)

4.4 COMPILAÇÃO FFMPEG2THEORA

Bem como para o VLC existe a opção de compilar cruzadamente o *ffmpeg2theora* e o *oggfwd* para se adicionar ao sistema de arquivos presente na placa (Qtopia), porém não foram encontradas referências de como se compilar o *ffmpeg2theora* para uma plataforma diferente da *linux* x86 ou compilar para plataformas windows.

Foi então encontrada no servidor de pacotes *Debian*, que consistem em uma distribuição *linux*, uma sessão com estes aplicativos desejados já compilados para a plataforma ARM, no formato de pacotes *.deb*. Inicialmente foram diretamente adicionados ao *filesystem* *Qtopia*, e depois de solucionados problemas de *links* e bibliotecas necessárias deparou-se com o problema de que estes pacotes *debian* necessitem de bibliotecas C de versões diferentes das presentes no *Qtopia*.

Uma vez que não foram encontradas referências de como recompilar tais programas para os processadores ARM, porém foram encontrados configurados no repositório *debian*, a solução procurada neste momento foi de um *filesystem* que possuísse as bibliotecas C nas versões necessárias.

4.5 FILESYSTEM DEBIAN PARA ARM

Sendo as bibliotecas do *Qtopia* incompatíveis com os pacotes *debian* encontrados, a solução foi utilizar um *filesystem* *debian*. Encontrou-se na internet um tutorial de como fazer um *filesystem* *debian* para a placa Friendly Arm Mini2440, juntamente de uma imagem pronta de um *filesystem*, porém o tamanho excede a memória flash disponíveis na placa de desenvolvimento.

Para facilitar o desenvolvimento, sem ter o espaço como limitante, optou-se então por utilizar uma memória externa, também do tipo flash, na forma de um cartão SD (*Secure Digital*). Outra grande vantagem durante o desenvolvimento é poder conectar tal cartão diretamente no sistema *host*, sem a necessidade de um protocolo de rede para fazer o carregamento dos arquivos.

4.5.1 Bootloader - Uboot

Para poder utilizar uma memória externa a placa mini2440 se faz necessário a alteração do *bootloader* presente no sistema, uma vez que o originalmente instalado *Supervivi bootloader* não possuem este recurso.

Seguindo as orientações de um tutorial do site Bill's Mini2440 Forum [11] foram baixados os fontes do *Uboot*, compilados para a plataforma ARM. Porém diferente das orientações do tutorial, e seguinte outra solução, o *bootloader* foi carregado diretamente na memória Flash da placa, e não RAM como era descrito. Assim que instalado o novo *bootloader* tornou-se possível utilizar uma memória externa a *Friendly Arm* para conter o sistema de arquivos.

4.5.2 Kernel

Como a memória interna na placa estaria sendo utilizada apenas para o *bootloader*, e apontando para o *filesystem* presente no cartão SD, optou-se por recompilar o kernel e colocá-lo juntamente o *filesystem* dentro do cartão.

Seguindo as orientações encontradas no site Mini2440 – LinuxMCE [12] foi obtido o kernel e compilado com o mesmo ambiente anteriormente utilizado para configurar o *bootloader*. Durante a etapa de *make menuconfig* foi mantida a configuração original, uma vez que o sistema desejado utiliza apenas os componentes normais do hardware.

4.5.3 Formatação da Flash SD

Depois de considerar diversas fontes de informação com relação à formatação de um cartão SD para ser usada com a *Friendly Arm*, optou-se por realizar uma solução mista. Embora os passos seguidos para criação do sistema de arquivos orientassem apenas a criação de uma partição, foi criada uma segunda do tipo swap, com o tamanho aproximado de 500MB, para auxiliar a memória RAM do sistema limitada em 64MB.

Swap, denominado desta forma por representar um espaço de troca, oferece características de memória virtual ao sistema, permitindo que, quando excedida a memória física disponível, alguns trechos de dados que não estejam sendo utilizados sejam armazenados temporariamente neste espaço, até que sejam requisitados novamente e então retornem a memória.

Como o cartão SD utilizado possui 4GB de espaço, e a partição swap ocupa 500MB, restam ao sistema de arquivos 3.5GB, mais do que o necessário para um sistema embarcado. Como haverá disponibilidade de disco, pode-se implementar a backup (copia local) de todo e qualquer material que venha a ser capturado para se transmitir.

Como dito anteriormente, a utilização de uma memória externa facilita a criação do sistema embarcado, pois não exige a utilização de protocolos de comunicação de dados para o carregamento dos arquivos, sendo sempre possível a simples remoção do cartão de memórias do sistema embarcado e a conexão direta ao sistema *host*.

4.5.4 File System

Para a construção do sistema de arquivos foi seguido o tutorial *How to build a root file system for EM2440/MINI 2440 Board – Linux in embedded systems* (Como construir um sistema de arquivos para a placa EM2440/MINI2440 – *Linux* em sistemas embarcados), encontrado no site *ModBus.PL* [13], com algumas adaptações, descritas a seguir.

Para os fontes utilizou-se o sistema mínimo *debootstrap*, presente em um repositório em *Debian*. No momento da compilação o domínio www.emdebian.org encontrava-se indisponível, então foi utilizado o ftp.uk.debian.org/emdebian. O *debootstrap* consiste em um sistema de arquivos pronto, com uma configuração mínima *Debian*, onde se escolhe a arquitetura desejada no momento do *download*, através de comandos específicos do programa. Seguindo as orientações, a arquitetura utilizada foi *ARM EABI* (armel) que consiste no esforço atual de portar a distribuição *linux Debian* para a plataforma ARM, sendo a sucessora do suporte a arquitetura ARM (arm apenas) para *Debian*.

O particionamento do cartão SD foi diferenciado em relação ao orientado no site, como descrito anteriormente. Então o novo sistema de arquivos, anteriormente comprimido em um arquivo TAR, é exportado para dentro da partição destinada no cartão de memória. Foram acrescentados comandos *no fstab, hostname, console e apt/sources.list* do novo sistema de arquivos, para que possa funcionar quando inserido o cartão no sistema alvo. Também foi copiado neste momento o *kernel* anteriormente compilado, na forma de um arquivo *uImage*, para a pasta */boot* do sistema novo.

Já no sistema alvo, após alguns comandos no *uboot bootloader*, chegou-se a um console de comandos, onde novas operações destinadas a configuração do sistema de arquivos foram executadas, incluindo a adição de *PATHs* (caminho para os executáveis do sistema) e um automático e demorado segundo estágio de instalação do *debootstrap*.

Uma vez finalizada a instalação, e executados alguns comandos para a criação de console via porta serial e configuração de ip automático de rede, a placa foi reiniciada e o *bootloader* reconfigurado para desativar o *script* de inicialização destinado a configurações.

4.5.5 SSH

Quando o sistema foi iniciado o console via serial, que estava sendo utilizado até o presente momento, deixou de funcionar, aparentemente pendurado em um comando. Porém observou-se que no display na própria placa de desenvolvimento apareceu a opção de *login* via linha de comando.

Após alguns testes para solucionar o problema do console via serial, optou-se por tentar utilizar o display da *Friendly Arm*, uma vez que com uma tela de 3 polegadas apresentava caracteres muito pequenos, como mostrado na figura 6, para a instalação de um servidor *ssh* e então tornar a executar os comandos e operações no sistema host. Utilizando um teclado USB conectado a placa foi possível executar os comandos necessários.

```

ut it is not going to be installed
Depends: openssh-blacklist but it is no
ing to be installed
E: Unmet dependencies. Try 'apt-get -f install' with no p
cesses (or specify a solution).
min:2440:~# apt-get -f install
Reading package lists... Done
Building dependency tree... Done
Correcting dependencies... Done
The following packages were automatically installed and a
no longer required:
  libcurl3-gnutls libbz2-0.0-0 libidn11 mysql-common
  libcurl3-gnutls libexpat1 libkrb53 libldap2 libffi6
  libkrb53 libidn2-2-4-2 libexpat1 libffi6 libidn2-2-4-2
  libkrb53 libidn2-2-4-2 libkrb53 libidn2-2-4-2
  libkrb53 libidn2-2-4-2 libkrb53 libidn2-2-4-2
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  binutils ca-certificates app-4.3 doc-4.3 libsound2
  libcurl3-gnutls libidn2-2-4-2 libexpat1 libffi6
  libexpat1 libffi6 libidn2-2-4-2 libkrb53 libidn2-2-4-2
  libkrb53 libidn2-2-4-2 libkrb53 libidn2-2-4-2
  libkrb53 libidn2-2-4-2 libkrb53 libidn2-2-4-2
The following packages will be REMOVED:
  openssh
The following NEW packages will be installed:
  binutils ca-certificates app-4.3 doc-4.3 libsound2
  libcurl3-gnutls libidn2-2-4-2 libexpat1 libffi6
  libexpat1 libffi6 libidn2-2-4-2 libkrb53 libidn2-2-4-2
  libkrb53 libidn2-2-4-2 libkrb53 libidn2-2-4-2
  libkrb53 libidn2-2-4-2 libkrb53 libidn2-2-4-2
0 upgraded, 22 newly installed, 1 to remove and 25 not up
dat.
Need to get 11.4MB of archives.
After this operation, 25.4MB of additional disk space will
be used.
Do you want to continue [Y/n]? y
Get:1 http://mirrors.us.debian.org/mirror/main libexpat1 2.4.2-1
[107kB]
Get:2 http://mirrors.us.debian.org/mirror/main libffi6 3.2.1-1
[107kB]
Get:3 http://mirrors.us.debian.org/mirror/main app-4.3 4.3.2-1
[107kB]
Get:4 http://mirrors.us.debian.org/mirror/main binutils 2.18.1
[107kB]

```

Figura 6 – Console utilizado diretamente na placa de desenvolvimento

O *login* com o usuário *root* foi permitido na placa, porém a instalação dos pacotes desejados não era possível, uma vez que o *apt-get* retornava erros. Como havia conexão, pois

o *ping* para qualquer ip existente retornava válido, foi detectado um problema de DNS. Assim que corrigido, acrescentando o ip 8.8.8.8 (Google Public DNS) no arquivo */etc/resolv.conf*, foi possível utilizar os pacotes de instalação *apt-get*.

O pacote *sshd* foi então instalado na placa de desenvolvimento, o que tornou possível a conexão remota de terminal via rede ip. Como o ambiente de desenvolvimento é conhecido, sabe-se qual ip é esperado para que a placa assuma, já que está configurada com *dhcp* (ip automático). Embora seja uma configuração que possa complicar futuros ajustes, já que o ip fornecido por um link 3G é quase sempre diferente, este terminal é utilizado apenas para configurações avançadas, e não se faz necessário no dia a dia de uso de um produto final.

4.6 INSTALAÇÃO *FFMPEG2THEORA*

Agora tendo um *filesystem Debian* com as bibliotecas atualizadas, e ainda por possuir recursos de instalação de pacotes com o *apt-get*, foi possível instalar os aplicativos *ffmpeg2theora* e *oggfwd* com todas suas dependências.

No momento da instalação o único repositório indicado ainda era o *emdebian*, o que não permitiu que os pacotes fossem encontrados com *apt-get install*. Logo eles foram baixados individualmente com *wget* e instalados via *dpkg*. Em etapas seguintes foi acrescentado o repositório de pacotes *debian*, e quando requisitada a atualização do sistema via *apt-get* então todos os pacotes instalados manualmente foram reinstalados em suas versões atualizadas.

Foram realizados alguns testes com o *ffmpeg2theroa* e o *oggfwd*, utilizando arquivos salvos na memória flash diretamente para fazer a transmissão. Não se conseguiu manter a transmissão, acreditando-se que ou por falta de *codecs* necessários pelo arquivo específico de vídeo ou por configuração indevida do *ffmpeg2theora*. O comando de testes foi o seguinte:

```
ffmpeg2theora teste.mp4 -x 320 -y 240 --inputfps 10 --videobitrate 128 --audiobitrate  
64 --channels 1 -o - | oggfwfwd argusr.no-ip.org 8000 senha /tcc
```

Muito embora a transmissão não tenha se concretizado, foi possível observar no servidor a criação da transmissão (mount point /tcc), indicando que uma conexão havia sido iniciada, originada do ip externo da conexão de onde a *Friendly Arm* estava, porém em instantes o *mount point* torna a desaparecer.

4.7 CAPTURA

Para a captura do sinal de vídeo composto foi selecionada uma placa usb com o seu funcionamento já comprovado sob o sistema operacional *linux*, sendo a instalação do *driver* disponibilizada na internet. Outro motivo pela sua escolha, além de ter conexão usb, presente na placa de desenvolvimento como entrada, e possibilitar o uso de vídeo composto, possível de se obter de qualquer câmera filmadora ou até mesmo fotográfica digital, é o baixo custo, sendo adquirida através do *website* MercadoLivre.com por R\$29,00.

O dispositivo chama-se EasyCap DC60, possui uma entrada de vídeo composto, uma entrada de *S-video* e uma entrada de *audio estereo*. A figura 7 mostra a placa usb de captura em questão.

O *script* para a instalação da placa de captura podem ser obtidos no *SourceForge.net* [14], e juntamente ao pacote o arquivos *README* explica que este *driver* destina-se unicamente ao *chipset* (controlador) com ID 05e1:0408. Para se observar o ID da placa é necessário conectá-la a um computador rodando *linux* e executar o comando *lsusb*, onde pode-se observar todos os dispositivos usb conectados.



Figura 7 – Placa de Captura de vídeo USB

Então descobriu-se, que existem mais de um controlador para o mesmo modelo de dispositivo, e casualmente havia se adquirido o errado. Após a troca o *hardware* foi testado no ambiente *Ubuntu* 10.04 e funcionou perfeitamente.

Para a instalação no dispositivo alvo, o script exige como dependência a presença dos *headers* do *kernel* da distribuição. Tentou-se copiar os *linux-headers* do sistema host, pois foram utilizados para a compilação do *kernel*, porém sua copia para o sistema alvo não foram suficientes para a compilação exigida pelo *script* em questão.

Outra opção seria a instalação do pacote *build-essential*, necessário para a compilação do *kernel*, porém também não foi possível instala-lo devido a problemas nas dependências de outros pacotes. Isto acontece devido a problemas com o repositório indicado.

4.7.1 WebCam

Para realizar testes, uma vez que a placa de captura não estava operante, foram testados alguns modelos de *webcam*, por serem dispositivos de captura de vídeo, alguns também apresentam áudio, e utilizaram conexão USB.

Esta tentativa também não obteve resultado positivo pois a instalação de hardwares deste gênero necessitam do pacote *build-essential* para completar.

4.8 BATERIA

Um dos principais pontos deste projeto consiste na autonomia de uso do sistema final. Para o projeto como descrito até o momento, o consumo de corrente da placa de desenvolvimento em *standy by* (sistema em espera) gira em torno de 1100mA. Porém como não foi criado um ambiente gráfico para a aplicação o display, um dos componentes que mais consome energia, pode ser desconectado, o que faz com que a corrente média fique em torno de 600mA.

Utilizando uma bateria ácida de chumbo de 6V / 4AH ter-se-ia uma autonomia de no mínimo 5 horas de uso, considerando a tensão mínima para o funcionamento da placa.

Para uma situação com ambiente gráfico do sistema, destinado a configurações e controles, mais a conexão sem fio, seja *Wi-Fi* ou 3G, e a captura de vídeo e áudio, muito provável que este consumo suba consideravelmente. Estimando o consumo máximo de dois Amperes de corrente para esta situação, e uma bateria de 6V / 7AH, teria-se no mínimo 3 horas de operação, sem acrescentar muito as dimensões e peso do dispositivo final.

5 RESULTADOS ALCANÇADOS

Dentro da proposta realizada no início da pesquisa o projeto final não conseguiu completar integralmente os objetivos. Contudo, o estado atual encontra-se muito próximo da solução desejada, necessitando pequenos ajustes e instalação de pacotes específicos para seu funcionamento pleno.

Em se tratando de características físicas, conseguiu-se montar um dispositivo pequeno, leve e com uma autonomia superior a presente no mercado atualmente, requisitos presentes nas definições de projeto.

A construção de um sistema de arquivos dedicado mostrou-se a melhor solução para os problemas de compatibilidade das bibliotecas requeridas pelos programas. A utilização de uma memória externa de maior capacidade possibilitou operações como a compilação de *drivers*, que necessitam de grandes bases de arquivos com os *headers* do *linux* por exemplo.

Foi possível transmitir utilizando unicamente o sistema desenvolvido, mesmo que por alguns instantes, o que significa que a instalação do sistema de arquivos e dos programas necessários foram corretamente executados. Estima-se que o funcionamento parcial da transmissão se dá devido a configuração equivocada dos parâmetros de transmissão, até o presente momento não solucionado.

6 CONCLUSÃO

Com o objetivo de desenvolver uma solução para transmissão de vídeo para telejornalismo, este projeto foi iniciado sem o conhecimento real das necessidades técnicas e funcionais desta aplicação.

A oportunidade de conhecer de perto o uso de sistemas semelhantes por diversos veículos de comunicação de grande porte na mídia brasileira foi de suma importância para a escolha dos requisitos do projeto, a fim de atender de melhor forma as necessidades dos profissionais deste ramo.

A intenção de se utilizar um *hardware* dedicado para a realização do processo de compressão, encapsulamento e transmissão de um sinal de vídeo mostrou-se possível e suficiente, mostrando ser desnecessário a utilização de uma plataforma x86 para tal fim.

A criação de um sistema computacional de forma integral, interagindo diretamente com o hardware, com diferentes formas de armazenamento, *boot* de inicialização, *kernel* e sistema de arquivos mostrou-se muito atrativa, e boa parte disto se deve a capacidade rápida de executar testes e obter soluções, bem como a vasta documentação, opções diferenciadas de soluções e o apoio irrestrito de toda a comunidade de desenvolvimento de Software Livre

O trabalho proporcionou diversas oportunidades e experiências tais como tratar diretamente com desenvolvedores de diferentes partes do mundo, com diversos fusos horários e sotaques para solucionar problemas mútuos, ou mesmo alheios, sem qualquer necessidade de remuneração ou demora para se obter respostas.

Este projeto deve continuar mesmo após sua apresentação e a finalização da graduação em Engenharia Elétrica, pois além de mostrar-se muito atrativo ao desenvolvimento, despertou interesse de profissionais e empresas que necessitam de soluções voltadas a transmissão de vídeo móvel.

Muitas das dificuldades encontradas no desenvolvimento se deram pela diferença entre a linha de estudos do curso e o projeto escolhido. Porém o suporte obtido com os professores e também alunos do curso de Engenharia e Ciências da Computação forma de suma importância para a solução obtida.

REFERÊNCIAS

1. Xiph.org:Ogg <<http://www.xiph.org/ogg/>>
2. Theora, video for everyone <<http://www.theora.org/>>
3. Vorbis.com <<http://www.vorbis.com/>>
4. VideoLAN – VLC <<http://www.videolan.org/>>
5. Icecast.org <<http://www.icecast.org/>>
6. Cortado (Software) <http://en.wikipedia.org/wiki/Cortado_%28software%29>
7. OIPlayer jQuery Plugin–OpenImages <<http://www.openbeelden.nl/oiplayer>>
8. Mini2440 | S2C2440 ARM9 <<http://www.friendlyarm.net/products/mini2440>>
9. Volker Gropp <<http://www.gropp.org/?id=projects&sub=bwm-ng>>
10. VectorVideo Std<http://en.wikipedia.org/wiki/File:Vector_Video_Standards2.svg>
11. Bill’s Mini2440 Forum <<http://billforums.station51.net/viewtopic.php?f=1&t=2>>
12. Mini2440 – LinuxMCE <<http://wiki.linuxmce.org/index.php/Mini2440#Kernel>>
13. HowTo FS EM2440 <<http://www.modbus.pl/S3C2440FS.html>>
14. EasyCAP DC60 Driver <<http://sourceforge.net/projects/easycapdc60/>>