

# Architecture with Internet of Things for Counting Passengers in Urban Bus

Julio Cezar Rutke, Janine Kniess  
Programa de Pós-Graduação em Computação Aplicada  
Universidade do Estado de Santa Catarina  
Email: rutke.julio@gmail.com, janine.kniess@udesc.br

**Abstract**—This paper proposes an Architecture based on Internet of Things for passenger counting in bus. The Architecture provides informations about the number of passengers per route in a public transportation system that adopts integrated billing. Experiments were performed with microcontrollers boarded on buses and with the COAP and Ble 6LowPan protocols. Cameras were installed on the buses and Computational Vision algorithms in the microcontrollers analyzed the images to detect people. The results showed that the Internet of Things protocols, COAP and Ble 6LowPan offer low overhead and response time.

## I. INTRODUÇÃO

Segundo estudos realizados [1], a maior parte da população vive na zona urbana. A estimativa é que em 2050 o número de pessoas vivendo na zona urbana seja quase o dobro da população rural. O crescimento desse êxodo, traz consigo problemas de infraestrutura. Uma possível solução é tornar as cidades mais inteligentes.

Uma cidade inteligente liga a infraestrutura física, a infraestrutura de Tecnologia da Informação (TI), a infraestrutura social e a infraestrutura de negócios para alavancar a inteligência coletiva da cidade [2].

Uma mobilidade inteligente pode auxiliar uma cidade no processo de torná-la inteligente, reduzindo engarrafamentos e diminuindo o número de acidentes de trânsito.

Nas cidades de médio a grande porte, o oferecimento de um transporte público eficiente é essencial para melhorar a qualidade da mobilidade urbana. A mobilidade urbana refere-se a facilidade com que um deslocamento pode ser feito até um determinado lugar, partindo de qualquer ponto de uma cidade [3].

No Brasil, o modelo mais utilizado de transporte coletivo é o ônibus. As rotas percorridas por uma frota de ônibus apresentam características diferentes ao longo do dia em nível de lotação. Neste contexto, informações sobre o número efetivo de pessoas que trafegam em cada rota pode proporcionar uma melhor distribuição dos ônibus em relação as rotas.

Diversas empresas utilizam o sistema de bilhetagem eletrônica para realizar a contagem de passageiros [4]. O passageiro possui um número de créditos em um cartão dotado de *chip* e memória.

No modelo de bilhetagem eletrônica, a informação sobre o número de passageiros por trajeto percorrido pode ser imprecisa. Isto ocorre, pelo fato da maioria das empresas adotarem um sistema de bilhetagem eletrônica integrado entre

terminais. Esse modelo, representado pela Figura 1 é comum no Brasil, principalmente nas cidades de médio e grande porte porque facilita o trânsito de passageiros entre terminais com um número reduzido de bilhetes.

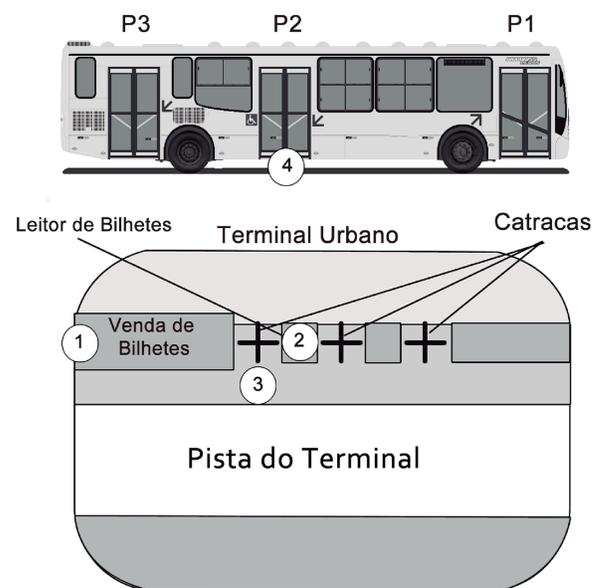


Fig. 1. Sistema de bilhetagem integrado, Fonte: próprio autor.

## A. Cenário de Aplicação

O modelo genérico de um sistema de bilhetagem eletrônica integrado, consiste dos seguintes elementos como descrito na Figura 1. Para que o usuário possa acessar um terminal urbano, ele necessita de um bilhete que pode ser adquirido fora do terminal (define-se terminais como estruturas físicas por onde os ônibus circulam, iniciando e/ou terminando uma determinada rota) (representado na Figura 1 pelo número 1). Após adquirir o bilhete, o passageiro deve registrá-lo em um leitor (número 2 na Figura 1) e acessar o terminal urbano passando por uma catraca (número 3 na Figura 1).

A partir deste ponto, o passageiro pode entrar em qualquer um dos ônibus presentes no terminal pela(s) porta(s) traseira(s) e/ou centrais (quando houver) (representadas por P2 ou P3 na Figura 1) sem a necessidade de registrar o bilhete novamente (número 4 na Figura 1). Adicionalmente, no sistema integrado, quando um ônibus chega a um novo terminal, o passageiro

pode trocar de ônibus, acessando-o também pela(s) porta(s) traseira(s) e/ou centrais (*P2* ou *P3* na Figura 1) (quando houver) sem um novo pagamento. Durante a rota entre um terminal e outro, existem paradas de embarque/desembarque. Nessas paradas os passageiros descem pelas porta traseiras e/ou centrais (*P2* e *P3* na Figura 1) e embarcam pela porta frontal (*P1* na Figura 1). Ao entrar no ônibus através de uma parada intermediária entre terminais, o passageiro deve registrar um bilhete no leitor disponível no ônibus (*P1* na Figura 1).

O gerenciamento eficiente da frota de ônibus no modelo integrado depende principalmente de uma contagem precisa do número de passageiros que trafegam nas diferentes rotas e horários. A detecção de pessoas no ônibus se apresenta como um desafio, pois os passageiros possuem tamanhos e formas diferentes e podem portar acessórios tais como, malas e mochilas. Adiciona-se a essas dificuldades o número significativo de pessoas que podem acessar o ônibus em horários de pico.

Neste contexto, este trabalho apresenta uma arquitetura para detecção e contagem de pessoas com Internet das Coisas (IoT) para o sistema de bilhetagem integrado. O transporte público urbano da cidade de Joinville, norte do estado de Santa Catarina, Brasil, é o ambiente de estudo de caso. Em Joinville, o transporte público através de ônibus é concessionado para empresas privadas e é dividido em terminais urbanos integrados.

Na arquitetura proposta, câmeras e equipamentos de baixo custo como, microcontroladores e sensores foram instalados no ônibus para realizar a contagem.

A detecção dos passageiros foi realizada através da análise de sequências de imagens com as técnicas: Subtração de Fundo[5], Características Haar [6] e Histogramas de Gradientes Orientados [7].

Este trabalho está organizado como segue: na Seção II são apresentados os trabalhos relacionados. Na Seção III apresenta-se a abordagem proposta. Os experimentos e os resultados são apresentados na Seção IV. As conclusões são apresentadas na Seção V.

## II. REVISÃO DE LITERATURA

Na literatura pode-se encontrar trabalhos relacionados com esta pesquisa. Entretanto, identificou-se que as abordagens não atendem os requisitos do sistema de bilhetagem integrado, bem como não oferecem uma solução baseada em tecnologias e protocolos de Internet das Coisas.

No contexto da contagem de pessoas com o uso de câmeras cita-se a proposta de Boreiko [8] que propõe um modelo estrutural para contagem de passageiros e rastreamento do transporte público. O sistema faz a contagem através de duas câmeras alocadas no teto, acima das portas do ônibus, e identifica as pessoas que passam.

Durante a rota, a cada parada, quando as portas do ônibus abrem o sistema grava as imagens através das câmeras. Após fechar as portas, um controlador Raspberry Pi, agrega as imagens gravadas juntamente com as informações de localização

geográfica coletadas através de um GPS. Os dados são transmitidos através de um modem 3G. Um servidor externo ao ônibus calcula o número de passageiros e o rastreamento do veículo pelas coordenadas. Por fim, as informações coletadas e processadas são apresentadas através de uma representação por Redes de Petri. Neste trabalho, observa-se que todo o processamento é realizado externamente ao ônibus podendo agregar um alto custo na transmissão dos vídeos pela rede 3G. Protocolos de IoT não foram analisados no escopo desta proposta.

No trabalho de Escolano [9] realizado na cidade de Manilla, nas Filipinas, câmeras foram instaladas nas portas de ônibus para coletar as imagens. O sistema baseia-se em uma estimativa de fluxo óptico para detectar e acompanhar o embarque e desembarque dos passageiros. O fluxo óptico utiliza sequências de imagens para identificar o sentido em que o passageiro passa. Para o cálculo da detecção nas imagens a função *Horn-Schunk* [10] é aplicada. Após a contagem das pessoas, os dados são processados por um controlador de lógica fuzzy que determina o horário do envio de um novo ônibus. Nesse projeto o autor utilizou o dispositivo ni myRIO e protocolos específicos para IoT não foram considerados no seu desenvolvimento e experimentos.

O artigo apresentado por Mukherjee [11], propõe um *framework* para detecção de pessoas e seu rastreamento em estações de trem. O *framework* é dividido em 3 partes: detecção das pessoas, rastreamento e validação. A etapa de detecção é realizada através de câmeras colocadas em cima das entradas da estação. Utilizando o algoritmo de *Hough* [12] para detectar formas circulares e também o algoritmo de *HOG* [7], o sistema identifica pessoas pelas formas circulares de suas cabeças. Na etapa de rastreamento, o sistema analisa os quadros posteriores de movimento através da identificação do fluxo óptico pelo método *Horn-Schunk* [10]. O sistema executa a etapa de validação a fim de evitar falsos positivos na identificação de pessoas, como outras partes do corpo (p.ex. mãos), ou até duplicações. Mukherjee não especifica como a transmissão dos dados de contagem é realizada. Além disso, seus experimentos são realizados em uma estação de trem, não sendo aplicável no sistema de bilhetagem integrado para ônibus.

No trabalho de Perng [13], é apresentado um algoritmo composto por três etapas: Subtração de fundo, Detecção de pessoas e Rastreamento e Contagem.

Com base em uma entrada de vídeo, o algoritmo aplica uma técnica de subtração de fundo para detectar movimentos na imagem e aplica alguns processamentos morfológicos, a fim de eliminar ruídos e masterizar a imagem. O algoritmo realiza um escaneamento na imagem em busca de características que possam definir uma pessoa, como cabeça e ombro. Com a pessoa identificada, o sistema faz um rastreamento na imagem com base em algoritmos gulosos. Neste trabalho, os autores não realizaram testes de transmissão dos dados gerados no ônibus e não utilizaram conceitos de IoT.

O trabalho apresentado por [14], utiliza para a detecção de cabeças das pessoas o descritor de características de HOG realizando o treinamento das imagens com um classificador SVM.

Para o rastreamento, foram utilizados algoritmos baseados nos filtros de Kalman. Diferentemente da arquitetura proposta neste artigo, os autores realizaram os experimentos em uma sala. Ainda, testes com a transmissão de dados e aspectos de IoT não foram considerados.

Os trabalhos apresentados não atendem as características do cenário de aplicação proposto neste artigo. De modo geral, apresentam mecanismos para detectar pessoas em imagens e realizar a contagem. No entanto, nenhum deles considera o processamento das imagens em microcontroladores embarcado no ônibus e a transmissão dos dados de contagem para na rede interna e externa já convertidos em formato texto. Outro aspecto não considerado nos trabalhos relacionados é a utilização de protocolos de Internet das Coisas.

Na seção seguinte, apresenta-se a arquitetura para a detecção e contagem de passageiros em ônibus que adotam o sistema de bilhetagem integrado, e os aspectos que favorecem o uso de protocolos e tecnologias de Internet das Coisas na abordagem proposta.

### III. ABORDAGEM PARA CONTAGEM DE PASSAGEIROS

O sistema realiza a contagem automática de pessoas que passam pela porta do ônibus através de técnicas de detecção de objetos e análise de imagens obtidas por câmeras de vídeo. A arquitetura do sistema de detecção de pessoas e transmissão de dados foi especificada com a finalidade: oferecer suporte para a contagem do fluxo de passageiros (entrada e saída) em cada trajeto percorrido pelo ônibus e prover informações coletadas no ônibus para a empresa de transporte coletivo.

#### A. Arquitetura do Sistema

A arquitetura do sistema para a detecção e contagem de pessoas, e para a transmissão dos dados coletados no ônibus é mostrada na Figura 2, incluindo os componentes para percepção do ambiente, tratamento dos dados coletados, interface de comunicação, armazenamento de dados e aplicação.

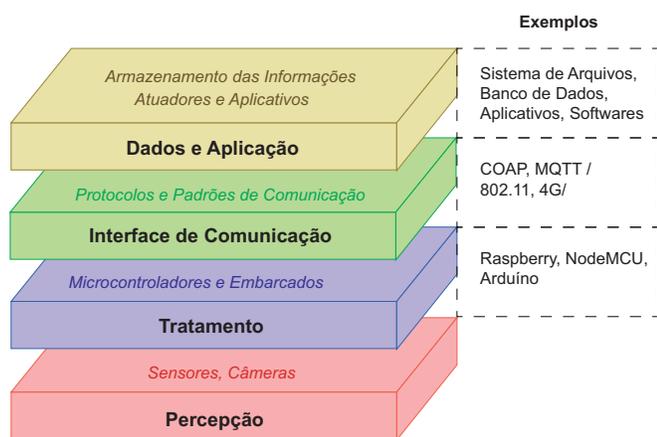


Fig. 2. Arquitetura do Sistema, Fonte: próprio autor.

A Camada de Percepção tem a função de perceber o ambiente e coletar informações através de periféricos, como sensores e câmeras. Utilizou-se câmeras para realizar a gravação

do fluxo de entrada e saída do ônibus. Após os dados serem coletados, necessita-se tratar estes dados, uma vez que a informação obtida pode ser uma pessoa ou várias, ou mesmo um objeto como, uma mala ou uma mochila transportada por um passageiro. Esse cuidado com as imagens coletadas é realizado pela Camada de Tratamento, que irá realizar o processamento de algoritmos que tem como função identificar pessoas nas imagens e contabilizar o fluxo de passagem pela porta do ônibus (entrada ou saída).

A Camada de Tratamento é composta por placas de desenvolvimento, como Arduino, Nodemcu ou Raspberry Pi. Na operacionalização do sistema, as câmeras estão interligadas a um Raspberry Pi 3B responsável pelo processamento da Camada de Tratamento. As imagens são obtidas e armazenadas pela Camada de Percepção e são processadas pela Camada de Tratamento. Após os dados serem coletados, devem ser transmitidos pela Camada Interface de Comunicação para que possam ser apresentados.

Conforme descrito na Figura 1 da Seção I-A, um ônibus pode conter duas ou três portas. Dessa forma, cada porta deve conter um módulo de contagem composto por uma câmera e um microcontrolador para coletar as imagens e tratá-las. Existe a necessidade de haver comunicação entre os módulos de contagem para que o sistema possa agregar os dados de cada porta e assim obter uma contagem efetiva do número de passageiros que entraram ou saíram pelas portas em cada uma das paradas realizadas pelo ônibus.

A comunicação entre os microcontroladores, bem como outros aspectos relacionados a comunicação do sistema com a rede externa ao ônibus para disponibilização dos dados são gerenciadas pela Camada de Interface de Comunicação (Figura 2). No sistema, os dados obtidos no ônibus são descarregados para um servidor externo, utilizando-se uma tecnologia de rede local como o WiFi 802.11 em algum ponto de parada do ônibus ou então na garagem da empresa. Outras tecnologias de comunicação podem ser adotadas na transmissão como as redes celulares ou *Low-Power Wide-Area Network* (LPWAN). Entretanto, o custo será maior em relação ao 802.11.

A Camada de Dados e Aplicações concentra a função de armazenamento das informações e apresentação dos dados aos usuários finais. A implementação desta camada pode estar em uma nuvem computacional ou em um servidor externo da empresa.

#### B. Contagem de Passageiros

O esquema de funcionamento da solução para a detecção e contagem de passageiros no ônibus foi organizado nos seguintes módulos: (I) Detecção e acionamento de porta (Camada de Percepção da Figura 2); (II) Contagem de passageiros (Camada de Tratamento); (III) Agregação, Comunicação e Aplicação (Camada de Tratamento, Camada de Interface de Comunicação e Camada de Dados e Aplicação, respectivamente). O fluxograma do algoritmo de detecção e contagem encontra-se descrito na Figura 3.

O sistema inicia com a abertura de porta do ônibus. Quando o motorista aciona alguma porta, seja para abrir ou fechar, o

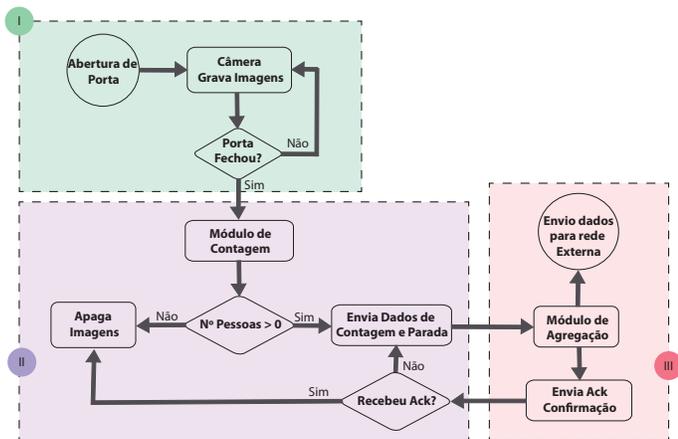


Fig. 3. Fluxograma do Algoritmo de Detecção e Contagem, Fonte: próprio autor.

sistema interno do ônibus emite um pulso elétrico por meio de um fio que está conectado à placa do módulo de Contagem. Assim o sistema identifica que ocorreu um acionamento de porta, e o módulo de Contagem de passageiros é notificado do evento. O módulo de Contagem é formado pela câmera que irá coletar as imagens, juntamente com a placa de desenvolvimento que irá processar as imagens e realizar a contagem, no caso desse projeto, um Raspberry Pi 3B.

Ao receber o sinal do módulo de Detecção e acionamento, o módulo de Contagem de passageiros através do microcontrolador recebe as imagens capturadas, após a abertura da porta. Esse processo se mantém até a porta ser fechada. Neste momento, o módulo de Detecção e de acionamento da porta notifica o módulo de Contagem do fechamento. Logo após, o módulo de Contagem processa e analisa as imagens para efetuar a contagem de passageiros.

A detecção das pessoas é efetuada através da análise das imagens com técnicas de Visão Computacional. No escopo deste trabalho foram utilizados os algoritmos de HOG, Harr e Subtração de Fundo.

A contagem é realizada quando uma pessoa é identificada em uma determinada área da imagem. Os limites dessa área são representados por duas linhas horizontais de uma ponta até a outra. A altura das linhas (superior e inferior) servem como base para contagem de uma pessoa. Em seguida, o sistema percorre cada frame do vídeo, e aplica-se ao frame as técnicas propostas. Finalmente, uma suavização nas bordas é utilizada nos frames. A suavização permite realçar as bordas dos objetos que permite uma identificação mais apurada. Com base nisso um vetor de contornos é gerado. Neste vetor, um contorno refere-se a uma pessoa identificada. O algoritmo percorre o vetor de contornos e verifica se cada contorno está dentro da área demarcada pelos limites inferior e superior. Se a condição for atendida a pessoa é contabilizada.

Para fins de contagem, no microcontrolador são armazenados os seguintes dados: número de entradas, número de saídas, identificador do módulo da porta (cada porta do ônibus tem um módulo de Contagem) e identificador de parada. Cada

variável tem 3 (três) bytes, totalizando apenas 12 (doze) bytes no microcontrolador. O identificador de parada é um número inteiro que é incrementado a cada vez que o módulo de Detecção e de acionamento da porta é ativado. O identificador de parada é zerado quando o ônibus termina o trajeto total e retorna para a garagem (local para abastecimento e limpeza) da empresa. O sistema utiliza dados de localização obtidos de um GPS agregado ao módulo de contagem para identificar o final do percurso. O sistema obtém os identificadores de latitude e longitude do GPS e compara com as informações de localização da garagem pré programadas. Ao identificar a proximidade dos pontos, assume-se que o ônibus adentrou à garagem.

Conforme mencionado, um ônibus contém mais de uma porta, portanto necessita de mais de um módulo de contagem. O módulo Agregador coleta e agrega as informações de cada porta. Nos experimentos apresentados na Seção IV utilizou-se um NodeMCU ESP32 e os resultados mostraram que o microcontrolador foi eficiente na transmissão dos dados para a rede externa.

Tanto no módulo de Contagem, quanto no de Agregação deve haver um módulo de Comunicação embarcado. Este módulo é responsável por transmitir as informações dos módulos de Contagem para o Agregador, assim como enviar dados do Agregador para uma rede externa ao ônibus.

Ao receber o sinal de fechamento de portas, o módulo de Contagem inicia a análise das imagens e realiza a contagem de passageiros. Caso não haja a identificação de pessoas o sistema exclui as imagens obtidas e não enviará dados ao módulo Agregador. Por outro lado, se houver a contabilização, o módulo de Comunicação nas portas divulga via rede sem fio para o módulo de Agregação o número de passageiros que entraram ou saíram do ônibus entre os eventos de abertura e fechamento de porta.

O módulo de Agregação ao receber os dados, processa e mantém atualizado o número de passageiros no ônibus a cada instante de tempo. O módulo de Agregação envia uma mensagem de reconhecimento (ACK\_Contagem) ao módulo de Comunicação informando a conclusão do processo. Em seguida, as imagens armazenadas no módulo de Contagem são eliminadas.

O tamanho dos vídeos gravados entre a abertura e o fechamento da porta tem em média 10 MB podendo alcançar um valor expressivo ao final da rota (ver Seção IV). Em virtude dessa característica decorrente da utilização de câmeras, aliada aos problemas provocados pela baixa capacidade de armazenamento e processamento dos microcontroladores tem-se um grande desafio: manter o mínimo de dados nos microcontroladores ou na rede sem prejudicar a contagem de passageiros.

Com o objetivo de reduzir o congestionamento na rede interna e externa do ônibus, definiu-se que o módulo de Contagem envia ao Agregador os dados da contagem já processados e em modo texto. O Algoritmo 1, apresenta as estratégias do módulo de Agregação para gerenciar e processar os dados recebidos do módulo de Comunicação (situado

no mesmo microcontrolador que o módulo de Contagem) e transmite os dados para a rede externa.

### Algoritmo 1: Algoritmo de Agregação

```

1 início
2   Módulo de Agregação recebe os dados dos módulos de
   Contagem
3   stringContagem = contabilizacao();
4   idParada = stringContagem.substring(0, 3);
5   idPorta = stringContagem.substring(3, 3);
6   numeroEntradas = stringContagem.substring(6, 3);
7   numeroSaidas = stringContagem.substring(9, 3);
8   paradaExiste = false;
9   Percorre por todas paradas armazenadas anteriormente
10  para cada u ∈ lista.DeParadas faça
11    se idParada == u.idParada então
12      paradaExiste = true;
13      u.numeroEntradas += numeroEntradas;
14      u.numeroSaidas += numeroSaidas;
15      totalNoMomento +=
        numeroEntradas-numeroSaidas;
16    fim
17  fim
18  se paradaExiste == false então
19    listaDeParadas.add(idParada, numeroEntradas,
        numeroSaidas);
20  fim
21  enviaConfirmacao("ACK_Contagem");
22 fim

```

O módulo de Agregação mantém uma lista de paradas como variável global, nessa lista estão as informações de cada parada do ônibus. A informação de parada segue o modelo: "001001015010". No modelo estão representados 4 (quatro) dados: Identificação da Porta, Identificação da Parada, Número de Entradas e Número de Saídas. Na sequência, 3 (três) dígitos representam uma variável. Por exemplo, os três primeiros dígitos "001" correspondem ao identificador da porta. Quando o ônibus inicia a rota essa lista está vazia, sendo incrementada a cada parada.

O módulo de Contagem da porta (linha 4) envia para o módulo de Agregação uma *String* com os dados da contagem de cada parada.

A *String* é tratada (ver Figura 4) com o propósito de separar os dados individualmente (linhas 5 até 8). Através da separação dos dados da string, o módulo de Agregação realiza a soma dos dados de entrada e saída das diversas portas.

Uma variável de controle (linha 8) tem a função de identificar se a parada recebida já existe ou precisa ser criada. Suponha a seguinte situação: quando um ônibus efetua uma parada, cada porta realiza a contagem individualmente e ao final gera uma *String* com os 4 (quatro) dados acima mencionados. Cada porta enviará seus dados para agregação, dessa forma, o registro da parada em específico na lista de paradas deve ser criado apenas uma vez. Portanto, quando uma *String*

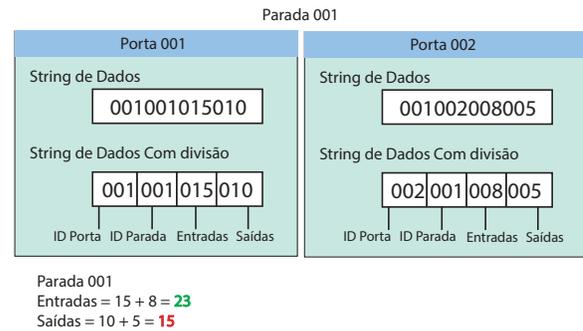


Fig. 4. Agregação da String de Contagem das Portas, Fonte: próprio autor.

com a informação de uma porta for enviada ao Agregador, o módulo deve verificar se esta parada já foi registrada por outra porta. Caso sim, agrega os números de entrada e saída. Caso contrário, deve registrar a nova parada na lista.

Justifica-se o mecanismo de controle do identificador de parada pelo fato que, as portas podem fechar em tempos diferentes ou a rede pode entregar os dados ao módulo Agregador em tempos não concomitantes. O algoritmo percorre a lista de paradas em busca de algum registro para adicionar a contagem (linhas 10 até 17). Caso nenhum identificador de parada seja encontrado na lista (linha 18), o identificador recebido é adicionado à lista (linha 19). O Agregador envia uma mensagem de confirmação para o módulo de Contagem (linha 21), avisando que os dados foram recebidos e a contagem adicionada.

## IV. EXPERIMENTOS E RESULTADOS

Nesta seção descreve-se os experimentos e os resultados com a solução proposta. Em suma, nesta seção apresentam-se: (i) descrição dos cenários de testes; (ii) resultados com a contagem de passageiros e com o mecanismo de comunicação dos elementos de hardware instalados no ônibus; (iii) análise dos resultados obtidos com os experimentos.

### A. Ambiente de Testes

No cenário alvo deste trabalho (descrito na Seção I-A), um dos grandes desafios é a dificuldade para realizar a contagem de pessoas em face do fluxo significativo de passageiros que podem entrar ou sair do ônibus ao mesmo tempo. Neste contexto, identificou-se que a Visão Computacional é uma ferramenta adequada para a detecção e contagem de passageiros neste trabalho. Portanto, câmeras foram instaladas no ônibus para a captura de imagens do fluxo de entrada e saída das portas.

Experimentos foram realizados no ambiente real (ônibus). Na implementação utilizou-se os seguintes componentes de hardware: Raspberry Pi versão 3, que possui um processador BCM2837 de 4 núcleos de 64 bits ARM Cortex-A53 a 1.2GHz, memória RAM de 1GB e um módulo de comunicação sem fio no padrão 802.11 b/g/n [15]; placa NodeMCU [16]; e uma câmera monocular da marca Logitech modelo HD

Webcam C270 que possui uma resolução de 1280 x 720 pixels [17].

O desenvolvimento do algoritmo para a contagem de passageiros foi realizado na linguagem de programação Python versão 2.7 com auxílio da biblioteca de visão computacional Opencv na versão 3.3.0. Como ambiente de desenvolvimento para escrever o código de operação do NodeMCU, utilizouse o IDE do Arduino versão 1.8.1 devido a maior oferta de bibliotecas e à compatibilidade com a linguagem C/C++.

A transmissão de mensagens entre os módulos de Contagem e Agregação é realizada através da tecnologia *Bluetooth Low Energy* (disponível no node ESP32 e no RASPBERRY PI 3). O protocolo de transporte é o *UDP* e o *BLE 6LoWPAN* foi usado para fornecer suporte IP. As mensagens são enviadas do módulo Agregador para um servidor externo via o protocolo de camada de Aplicação *Constrained Application Protocol* (CoAP).

O desempenho do algoritmo para a detecção e contagem de passageiros pode sofrer variações em função das características do ambiente. Alguns fatores podem contribuir para essas variações como, o elevado número de pessoas que podem entrar ou sair do ônibus ao mesmo tempo, variações na iluminação do ônibus decorrentes da abertura e fechamento das portas, condições climáticas ou hora do dia, desconexões do sistema de comunicação causados por ruídos do motor do ônibus, interferências de outras redes e oclusão de pessoas.

Considerando os fatores supracitados, que influenciam no desempenho da solução proposta, identificou-se as métricas descritas a seguir como as que melhor se adéquam à avaliação de desempenho desta solução.

- Taxa de Acerto: número real de pessoas que entraram/saíram do ônibus, comparado com a contagem efetuada pelo sistema de contagem.
- Sobrecarga: número de pacotes de contagem enviados pelo sistema/número total de mensagens geradas na rede.
- Perda de Pacotes: número de pacotes enviados pela origem (microcontrolador no ônibus)/ número de pacotes recebidos pelo destino (servidor).

Os testes foram realizados nos ônibus da Empresa Transtusa na cidade de Joinville-SC. Nos experimentos a seguir avaliou-se a transmissão dos dados para uma rede externa ao ônibus e o potencial para a identificação e contagem de pessoas das seguintes técnicas da literatura: Subtração de Fundo [5], Características Haar [6] e Histogramas de Gradientes Orientados [7].

### B. Análise da Transmissão dos Dados Coletados

Um requisito essencial para a viabilidade do sistema é a transmissão dos dados coletados através de uma rede de comunicação para uma nuvem computacional ou para um servidor na empresa.

Através de entrevistas junto a empresa <sup>1</sup> de transporte coletivo, identificou-se que os ônibus permanecem parados nas seguintes situações: nos terminais urbanos e na garagem

da empresa. Nos terminais urbanos, o número de ônibus é significativo e o tempo de parada varia entre 5 minutos a 10 minutos. Na garagem, o tempo de parada dos ônibus pode variar de 3 minutos a 20 minutos. Concernente ao posicionamento dos ônibus na garagem, os mesmos são enfileirados por ordem de chegada, sendo que muitos ônibus ficam fora do raio de alcance do sinal da antena da rede sem fio.

A transmissão dos dados por meio de uma rede local como, por exemplo o 802.11 tem baixo custo de implantação e a infraestrutura de rede sem fio da empresa pode ser aproveitada sem grandes modificações. Todavia, neste cenário, o módulo Agregador atua como um centralizador dos dados, que mantém os dados até que o ônibus alcance um ponto de parada, no qual os dados podem ser enviados para um servidor externo.

Com o objetivo de analisar o desempenho da tecnologia 802.11 na transmissão dos dados coletados no ônibus (vídeo e texto), realizou-se um conjunto de experimentos, no qual os dados são transmitidos do ônibus para um servidor configurado para este fim. A transmissão foi realizada via o protocolo CoAP para um servidor, e os dados foram coletados por uma página na linguagem PHP.

A informação no formato texto consiste em dados oriundos do processamento dos vídeos realizado pelo equipamento Raspberry Pi alocado no ônibus. No módulo de Contagem, os vídeos são processados pelo Raspberry Pi e o resultado é a informação de contagem por parada no formato texto. Cada linha de texto contém 50 bytes representando o número de embarques, desembarques, o código do ônibus, um identificador da parada, data e horário. Efetuou-se 10 (dez) transmissões com 10 (dez) linhas cada uma para o servidor. Os vídeos foram transmitidos para o servidor sem nenhum tratamento.

No primeiro experimento, utilizou-se uma rede sem fio 802.11. Um roteador D'Link 524 possibilitou a conexão do Raspberry Pi no ônibus com um servidor configurado em um notebook HP Pavillon G4. Durante a transmissão, o Wireshark [19] foi utilizado para a captura pacotes que trafegaram na rede.

Por meio dos pacotes capturados com o Wireshark, mediuse o atraso médio entre o Raspberry e o servidor na transmissão do texto. Como resultado, obteve-se um atraso médio de 29,4 milissegundos para transmitir uma linha de dados (50 bytes).

No segundo experimento, foram transmitidos 10 (dez) vídeos de 6 MB e 5 (cinco) vídeos de 19 MB do Raspberry para o servidor via na rede sem fio 802.11. A transmissão dos vídeos menores resultou em um atraso médio de 23,2 segundos, enquanto os maiores levaram em média 65,0 segundos.

O gráfico da Figura 5, representa a sobrecarga na rede decorrente da transmissão dos dados no formato texto e dos vídeos na rede 802.11. Os resultados mostraram que a transmissão de vídeos aumenta significativamente a sobrecarga na rede, por exemplo, na transmissão do texto, a sobrecarga ficou abaixo de 1%, e na transmissão dos vídeos alcançou 5%.

O tempo gasto para transmitir os vídeos e a sobrecarga (ver Figura 5) confirma a necessidade de processar os vídeos previamente no microcontrolador antes de transmiti-los via

<sup>1</sup>Transporte e Turismo Santo Antonio [18]

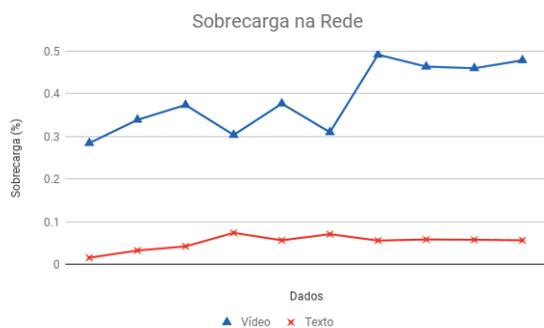


Fig. 5. Análise da Sobrecarga na Transmissão de Dados.

rede. Neste sentido, faz-se necessário o uso de uma técnica de detecção de pessoas nas imagens que não sobrecarregue o microcontrolador e ao mesmo tempo forneça precisão quanto a contagem.

Através da análise nos pacotes capturados pelo Wireshark, constata-se que, a perda de pacotes na transmissão do texto ficou abaixo 1%, enquanto a perda de pacotes na transmissão dos vídeos chegou a 2,3%.



Fig. 6. Análise da Perda de Pacotes na Transmissão de Dados.

Levando-se em consideração que cada vídeo leva em torno de 23,0 segundos, e que cada ônibus tem 3 (três) portas, pode ser necessário mais de um minuto para que os vídeos sejam descarregados via rede sem fio. Visto que, cada linha do ônibus pode ter entre 10 (dez) a 20 (vinte) paradas, chega-se a um número aproximado de 10 (dez) a 20 (vinte) minutos para descarga. Se os vídeos não forem processados antecipadamente no microcontrolador, faz-se necessário o uso vários pontos de transmissão pela cidade, ou então um tempo maior de parada do ônibus na garagem.

Outras tecnologias de transmissão, por exemplo o 3G ou 4G pode ser adotada para a transmissão de dados na arquitetura proposta, porém, deve-se adicionar o custo das operadoras das redes celulares.

### C. Análise dos Algoritmos de Contagem

Para gravação dos vídeos, utilizou-se uma câmera webcam de baixo custo com capacidade de captura de imagens HD (alta definição - 1280 x 720 pixels). Os vídeos foram recebidos e

processados pelo Raspberry Pi. As imagens foram gravadas no ônibus 1118 que percorreu a rota de número 1360 entre os terminais Norte e Pirabeiraba na cidade de Joinville-SC. A câmera foi posicionada no centro do teto do ônibus (entre uma lateral e outra do ônibus) a 45° em relação ao teto e voltada para a porta.

Para a realização dos experimentos foram gravados dois vídeos, um durante uma rota diurna e outro em uma rota noturna. Ressalta-se que as rotas são as mesmas, a diferença está no período do dia. Os 2 (dois) vídeos possuem a duração de 60 minutos e foram executados com uma taxa de 20 quadros por segundo. A gravação diurna iniciou às 17:30 com a seguinte condição climática: céu aberto e dia ensolarado. A gravação noturna iniciou às 21:35 com o céu nublado.

Após a gravação dos vídeos, as imagens relacionadas aos embarques e aos desembarques foram analisadas. A gravação entre a abertura e o fechamento de portas resultou em 145 segundos, especificamente, 10 MB (diurno) e 20 MB (noturno). Desta gravação, 23 segundos são relacionados aos embarques na rota diurna, 29 segundos aos embarques durante a noite, 40 segundos para os desembarques durante o dia e 53 segundos para os desembarques a noite. Em número de passageiros tem-se, 15 (quinze) e 9 (nove) embarques, 21 (vinte e um) e 33 (trinta e três) desembarques, todos realizados no terminal.

Ressalta-se que durante o percurso da linha (entre os terminais) pessoas embarcaram pela porta dianteira (porta P1 da Figura 1 do ônibus), e desembarcaram pelas portas P2 ou P3 1.

No primeiro teste, a técnica de Subtração de Fundo foi aplicada sobre as imagens coletadas. Essa técnica tem por objetivo extrair o fundo das imagens comparando os pixels de frames sequenciais para a identificação de movimento. Através dos resultados obtidos, sumarizados no gráfico da Figura 7, pode-se concluir que a técnica de Subtração de Fundo, no teste diurno, mostrou uma taxa de acerto de aproximadamente 74% para o embarque e 62% para o desembarque.

No teste diurno, embarcaram 15 (quinze) pessoas no ônibus no terminal de saída, não obstante, a técnica de Subtração de Fundo detectou corretamente apenas 11 (onze) pessoas. A mesma técnica apresentou um falso positivo de 14 (quatorze) pessoas, ou seja, identificou que as mesmas desembarcaram quando não ocorreu nenhum desembarque.

A técnica de Subtração de Fundo também foi analisada no desembarque dos passageiros no período diurno ao termino da rota no terminal.

No experimento com o desembarque, desceram 21 (vinte e uma) pessoas, porém o sistema contabilizou apenas 13 (treze), e apresentou um falso positivo de 16 (dezesseis) pessoas. Com base na análise das imagens, percebeu-se que os falsos positivos ocorreram porque algumas pessoas ao subir ou descer do ônibus acabam voltando-se para atrás. Nesse caso, a técnica contabilizou o inverso do sentido da pessoa, ou seja, foi contabilizada uma saída, quando de fato tem-se uma entrada e vice-versa.

A segunda técnica avaliada foi a proposta por Viola & Jones [6], que visa primordialmente o reconhecimento de faces. Em

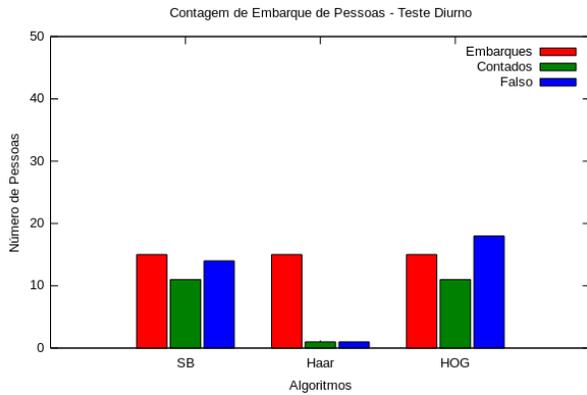


Fig. 7. Análise das Técnicas de Detecção - Embarque Diurno.

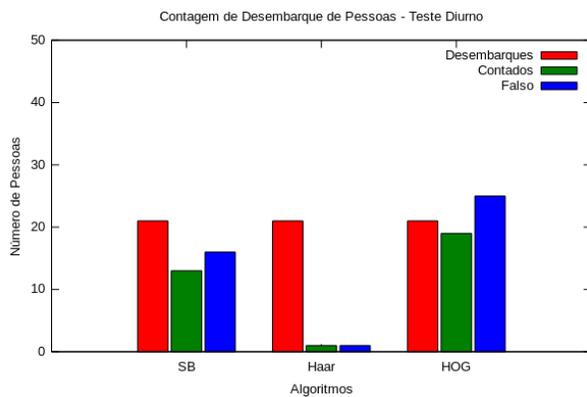


Fig. 8. Análise das Técnicas de Detecção - Desembarque Diurno.

relação a essa técnica, percebeu-se que, quando uma pessoa desembarca do ônibus, normalmente sua face está voltada para fora do veículo conforme pode ser visto na Figura 9. Consequentemente, a câmera capturou apenas a parte traseira da cabeça. Outro fator que afetou o desempenho do algoritmo é a oclusão da cena. Nos testes, constatou-se que devido a oclusão, em muitos casos, parte do rosto do indivíduo ficou escondida por algum adereço como um chapéu, ou por outra pessoa situada a frente conforme demonstrado na Figura 10.

Conforme descrito nos experimentos representados pelas Figuras 7 e 8, a técnica de [6] foi ineficiente na contagem dos passageiros em ambos os cenários, diurno e noturno, resultando em valores próximo a 0 (zero).

O algoritmo de HOG (*Histogram Oriented of Gradients*) [7] foi a terceira técnica de detecção aplicada. No algoritmo de HOG um descritor calcula o histograma da orientação dos gradientes na imagem. Neste modelo, a forma e a aparência de um objeto, na maioria das vezes, podem ser descritas pela direção das bordas ou a intensidade dos gradientes.

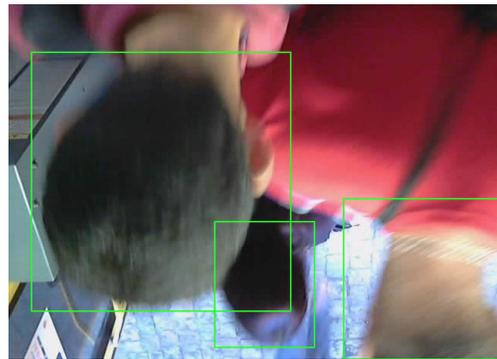


Fig. 9. Imagem do Desembarque - Câmera a 45°.



Fig. 10. Imagem do Desembarque (Oclusão) - Câmera a 45°.

Nos experimentos de contagem, o algoritmo obteve uma taxa de reconhecimento de 80%, tanto no embarque quanto no desembarque. No desembarque, 21 (vinte e uma) pessoas desceram, sendo que o algoritmo não reconheceu apenas 2 (duas) pessoas. No embarque, o algoritmo reconheceu 11 (onze) pessoas num total de 15 (quinze). No entanto, o algoritmo gerou um número superior de falsos positivos em relação aos algoritmos, Subtração de Fundo e Haar.

Especificamente, com o algoritmo de HOG observou-se que a presença de oclusão de partes do corpo como pés, mãos ou rosto provocou o não reconhecimento das pessoas. Essa conclusão foi obtida analisando as imagens de modo individual, buscando comparar com o número de acertos da contagem e identificar a razão dos falsos positivos.

## V. CONCLUSÕES

Neste artigo apresentou-se uma arquitetura com Internet das Coisas para contagem de passageiros em ônibus urbano. A arquitetura visa a contagem através da detecção de pessoas em imagens. Um módulo de agregação foi desenvolvido para concentrar as informações de todas as portas do ônibus e para realizar a transmissão para a rede externa, evitando sobrecarregar os módulos de Contagens acoplados nas portas do ônibus.

Realizou-se experimentos para avaliar taxas de transmissão de dados do ônibus para uma rede externa. Os resultados comprovaram que o uso de protocolos de IoT são adequados

para este tipo de aplicação, pois são mais leves e geram um número reduzido de mensagens. Ainda avaliou-se a precisão na contagem dos passageiros utilizando-se algoritmos de Visão Computacional.

No que tange a transmissão dos dados no formato texto utilizando-se o 802.11, os resultados se mostraram eficientes com baixa sobrecarga e perda de pacotes. Porém, a tecnologia apresenta limitações em algumas situações, por exemplo, se o ônibus parar na garagem fora do alcance da rede sem fio, ou então, se diversos ônibus estiverem parados ao mesmo tempo descarregando dados. Uma alternativa seria a utilização de uma tecnologia 3G/4G a fim de transmitir os dados em tempo real. Porém, acarretaria na necessidade de um módulo de transmissão 3G/4G em cada veículo e um plano de dados de internet para transmissão.

Em se tratando da contagem, os resultados mostraram que algoritmo de HOG apesar de apresentar um desempenho superior aos algoritmos de Harr e Subtração de Fundo apresentou um número maior de falsos positivos. Constatou-se que os algoritmos avaliados necessitam de um treinamento extensivo para gerar resultados melhores, e também é necessário a aplicação de técnicas de rastreamento e validação nas imagens.

#### REFERENCES

- [1] U. Nations, "Urban and rural areas 2014," Outubro 2014, [https://esa.un.org/unpd/wup/wallcharts/WUP\\_2014%20Urban-Rural%20Areas%20Wallchart.pdf](https://esa.un.org/unpd/wup/wallcharts/WUP_2014%20Urban-Rural%20Areas%20Wallchart.pdf), Acessado em: agosto de 2017.
- [2] C. Harrison, B. Eckman, R. Hamilton, P. Hartswick, J. Kalagnanam, J. Paraszczak, and P. Williams, "Foundations for smarter cities," *IBM Journal of Research and Development*, vol. 54, no. 4, pp. 1–16, 2010.
- [3] S. J. C. de Freitas, F. M. Placides, and D. Araújo, "Os desafios da mobilidade urbana," *Percurso Acadêmico*, pp. 239–241, 2015.
- [4] N. Bernini, L. Bombini, M. Buzzoni, P. Cerri, and P. Grisleri, "An embedded system for counting passengers in public transportation vehicles," in *Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on*. Senigallia, Italy: IEEE, 2014, pp. 1–6.
- [5] P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," *Video-based surveillance systems*, vol. 1, pp. 135–144, 2002.
- [6] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1. Kauai, HI, USA, USA: IEEE, 2001, pp. I–I.
- [7] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. San Diego, CA, USA, USA: IEEE, 2005, pp. 886–893.
- [8] O. Boreiko and V. Teslyuk, "Structural model of passenger counting and public transport tracking system of smart city," in *Perspective Technologies and Methods in MEMS Design (MEMSTECH), 2016 XII International Conference on*. Lviv, Ukraine: IEEE, 2016, pp. 124–126.
- [9] C. O. Escolano, R. K. C. Billones, E. Sybingco, A. D. Fillone, and E. P. Dadios, "Passenger demand forecast using optical flow passenger counting system for bus dispatch scheduling," in *Region 10 Conference (TENCON), 2016 IEEE*. Singapore, Singapore: IEEE, 2016, pp. 1875–1878.
- [10] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.
- [11] S. Mukherjee, B. Saha, I. Jamal, R. Leclerc, and N. Ray, "Anovel framework for automatic passenger counting," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. Brussels, Belgium: IEEE, 2011, pp. 2969–2972.
- [12] H. Yuen, J. Princen, J. Illingworth, and J. Kittler, "Comparative study of hough transform methods for circle finding," *Image and vision computing*, vol. 8, no. 1, pp. 71–77, 1990.
- [13] J. W. Perng, T. Y. Wang, Y. W. Hsu, and B. F. Wu, "The design and implementation of a vision-based people counting system in buses," in *2016 International Conference on System Science and Engineering (ICSSE)*, July 2016, pp. 1–3.
- [14] S. D. Pore and B. Momin, "Bidirectional people counting system in video surveillance," in *Recent Trends in Electronics, Information & Communication Technology (RTEICT), IEEE International Conference on*. IEEE, 2016, pp. 724–727.
- [15] raspberrypi.org, "Raspberrypi 3 model b," Dezembro 2017, <https://www.raspberrypi.org/products/raspberrypi-3-model-b/>, Acessado em: Dezembro de 2017. [Online]. Available: <https://www.raspberrypi.org/products/raspberrypi-3-model-b/>
- [16] NodeMCU, "Integrated development environment (ide)," Dezembro 2017, [http://www.nodemcu.com/index\\_en.html](http://www.nodemcu.com/index_en.html), Acessado em: Dezembro de 2017. [Online]. Available: [http://www.nodemcu.com/index\\_en.html](http://www.nodemcu.com/index_en.html)
- [17] logitech.com, "Logitech," Dezembro 2017, [www.logitech.com/pt-br/product/hd-webcam-c270?crd=34#specification-tabular](http://www.logitech.com/pt-br/product/hd-webcam-c270?crd=34#specification-tabular), Acessado em: Dezembro de 2017. [Online]. Available: <https://www.logitech.com/pt-br/product/hd-webcam-c270?crd=34#specification-tabular>
- [18] transtusa.com.br, "Transtusa," Dezembro 2017, <http://www.transtusa.com.br/>, Acessado em: Dezembro de 2017. [Online]. Available: <http://www.transtusa.com.br/>
- [19] wireshark.org, "Wireshark," Dezembro 2017, <https://www.wireshark.org/>, Acessado em: Dezembro de 2017. [Online]. Available: <https://www.wireshark.org/>