



SEI-SICITE 2021

Pesquisa e Extensão para um mundo em transformação

Protótipo de Software Servidor Embarcado em Dispositivo IoT para Controle de Semáforo

Prototype of Software Embedded Server in IoT Device to Traffic Light Control

Guilherme Malaquias Novaes¹, Lucio Agostinho Rocha²

RESUMO

Atualmente no Brasil as vias urbanas das grandes cidades possuem uma quantidade elevada de tráfego de veículos. Como consequência, o grande número de veículos acarreta um maior fluxo nas vias urbanas. Este projeto tem o objetivo de desenvolver um software servidor embarcado para o controle de semáforos de trânsito com Internet das Coisas. A motivação é oferecer uma solução de baixo custo que permita estudar alternativas de controle de tráfego conforme a demanda do trânsito em vias urbanas. A metodologia deste projeto foca em controlar um sinal luminoso de um dispositivo semafórico real com requisições Hypertext Transfer Protocol (HTTP) enviadas para a rede sem fio de um Raspberry Pi. Esse dispositivo está conectado ao semáforo por meio de portas de propósito geral e relés. As requisições HTTP definem a abertura e o fechamento de uma via de trânsito para oferecer um controle dinâmico da passagem de veículos conforme a demanda do tráfego. O resultado apresentado é o desenvolvimento de um software servidor embarcado para consultar e modificar dados de semáforos em um site de gerência na nuvem. Concluímos que é possível oferecer uma solução de baixo custo para a gestão de vias urbanas com Internet das Coisas.

Palavras-chave: Internet das Coisas; Gerência de Tráfego Urbano; Sistemas Embarcados

ABSTRACT

Nowadays in Brazil the urban roads of the big cities have a high traffic of vehicles. The high number of vehicles implies in high flow in urban roads. This project has the goal of developing an embedded server software to control traffic lights with the Internet of Things. The motivation is to offer a low-cost solution to allow study alternatives to traffic control according to the traffic demand on urban roads. The methodology of this project focus on control a traffic light with Hypertext Transfer Protocol (HTTP) requests send to a wireless network of a Raspberry Pi. This device is connected to traffic lights by general-purpose ports and relays. The HTTP requests define the open and close of the traffic road to offer a dynamic control to the traffic flow according to the traffic demand. The result presented is the development of an embedded software server to query and modify traffic light data in a management cloud site.

Keywords: Internet of Things; Management of Urban Traffic; Embedded Systems.

1 INTRODUÇÃO

Atualmente no Brasil as vias urbanas das grandes cidades possuem uma quantidade elevada de tráfego de veículos nos horários de "rush". De acordo com o IBGE, no estado do Paraná trafegam mais de 8 milhões de veículos em todo estado, sendo que mais de 1 milhão de veículos trafegam só em Curitiba (IBGE, 2021) (KOWALSKI,2021). Como consequência, esse grande número de veículos acarreta um maior fluxo de veículos nas vias urbanas.

Neste projeto de iniciação tecnológica foram pesquisadas soluções de gerência de tráfego urbano com semáforos inteligentes. Uma empresa brasileira com sede em Uberlândia/MG desenvolveu um projeto de semáforo inteligente em meados de 2013, porém, por se tratar de um produto comercial, foi encontrada pouca

¹ Curso de Engenharia Elétrica, Universidade Tecnológica Federal do Paraná, Apucarana, Paraná, Brasil;
guilhermenovaes@alunos.utfpr.edu.br

² Curso de Engenharia de Computação, Universidade Tecnológica Federal do Paraná, Apucarana, Paraná, Brasil;
luciorocha@utfpr.edu.br



informação com relação ao projeto desta empresa (SEMINT Semáforos, 2013). Um projeto similar é o de um sistema de semáforos inteligentes instalado em Cuiabá em 2017 (G1 MT, 2017). O sistema utilizado na cidade utiliza uma central de controle para controlar semáforos remotamente conforme a demanda do fluxo do trânsito. Outro projeto similar relacionado à pesquisa é um trabalho de mestrado (MUGNELA, 2012) no qual foi criado um sistema que calcula a melhor forma de atuação do semáforo de acordo com o tempo de abertura e fechamento das vias. O autor informa que há necessidade de um controle que contemple o comportamento dinâmico das relações entre a malha viária e a frota ao longo dos anos e considera também as restrições tecnológicas e políticas da implementação dos controladores. Além disso, a pesquisa apresentou um controle semaforico como o tema de estudo do trabalho de mestrado.

O controle de semáforos geralmente é feito de forma manual com temporização fixa, isto é o semáforo é pré-programado para abrir e fechar vias e seus períodos em cada fase porém também existe outro ajuste cujo o qual o semáforo tem uma pré-programação para alguns horários do dia onde ele funciona de outra maneira nos horários de maior fluxo de trânsito. Ter soluções que permitam monitorar e modificar o tempo de passagem de veículos de acordo com a demanda é útil para otimizar o tráfego nas vias mais congestionadas. O monitoramento tem o potencial de facilitar o fluxo do trânsito durante as diferentes horas do dia.

Neste artigo queremos responder a seguinte pergunta: é viável implementar um servidor, capaz de receber requisições HTTP para controlar um semáforo de forma variável, permitindo uma adaptação ao fluxo do tráfego? Para responder a essa pergunta realizamos um estudo de caso com a implementação de um software servidor embarcado em um dispositivo voltado para aplicações de IoT. O objetivo é oferecer uma solução de baixo custo que permita estudar alternativas de controle de tráfego conforme a demanda do trânsito em vias urbanas. A novidade do trabalho é acrescentar a comunicação para permitir o controle remoto em tempo real. Este projeto permite controlar o sinal luminoso de um dispositivo semaforico real com requisições HTTP (Hypertext Transfer Protocol) enviadas para a rede sem fio de um Raspberry Pi (RASPBERRY PI FOUNDATION, 2021). Esse dispositivo está conectado ao semáforo por meio de portas de propósito geral e relés. As requisições HTTP definem a abertura e o fechamento de uma via de trânsito para oferecer um controle dinâmico da passagem de veículos conforme a demanda do tráfego. O software servidor é capaz de consultar e modificar dados de semáforos em um site de gerência na nuvem.

O restante deste trabalho é apresentado como segue. A Seção Metodologia descreve a metodologia utilizada para a criação do protótipo de servidor Web embarcado; a Seção de Resultados mostra os resultados obtidos na execução do protótipo; finalmente, a Seção Conclusão faz as considerações finais.

2 METODOLOGIA

Neste projeto foi realizado um estudo preliminar com a metodologia Canvas de gestão de projetos para aferir os potenciais benefícios do projeto para os usuários. A seguir, são apresentados os principais itens considerados no desenvolvimento do modelo de negócios deste projeto.

a) Usuários do Sistema: o usuário desse sistema será um administrador de tráfego urbano. Os beneficiários indiretos desse sistema são quaisquer motoristas de trânsito urbano, como por exemplo, motoristas particulares de empresas de transporte, transporte urbano (por exemplo, companhias de ônibus), agentes públicos de trânsito (ambulância, bombeiros, viaturas, etc.), e até mesmo pedestres.

b) Proposta/Oferta de valores: esse sistema busca reduzir o tempo de espera dos motoristas nas vias urbanas.

c) Como o sistema será ofertado: o sistema será ofertado em uma plataforma de nuvem acessível por administradores de tráfego urbano.

d) Como o sistema poderá ser comercializado: o sistema poderá ser ofertado como parte de um pacote de software acessível em uma plataforma de nuvem. O usuário poderá manter um contrato de prestação de



serviços onde o fornecedor se compromete a manter o sistema sempre ativo e operacional. No momento, estamos na fase de protótipo, sendo a comercialização uma proposta futura.

e) Custos operacionais: para manter o sistema operacional, além dos recursos humanos para a manutenção do aparato de software e hardware, os principais custos previstos são a hospedagem do sistema em um host servidor próprio. Até o presente momento, todo o desenvolvimento foi baseado no uso de plataformas de software gratuitas, porém limitadas quanto à quantidade de requisições de acesso e ao armazenamento de dados. Para a implantação do sistema será necessário um servidor próprio de alto desempenho com acesso à Internet de alta velocidade.

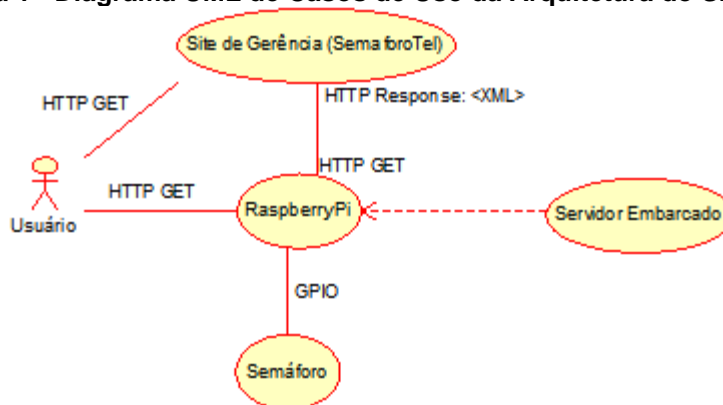
f) Parcerias: este projeto conta com a parceria do IDEPPLAN (Instituto de Desenvolvimento, Pesquisa e Planejamento de Apucarana) que tem interesse na pesquisa desse projeto.

A partir dessas informações do modelo de negócios foi possível vislumbrar um protótipo de baixo custo que serve como instrumento de validação da proposta.

2.1 Arquitetura do Sistema

A proposta deste projeto é utilizar dispositivos IoT (BRANDÃO, 2020) para interagir com equipamentos reais de trânsito de vias urbanas. Embora haja trabalhos similares na literatura (MUGNELA, 2012, SEMINT, 2013) a proposta do estudo de gestão de vias urbanas com dispositivos IoT ainda é recente. Neste projeto, essa interação é prevista para ser realizada com um dispositivo semafórico real de trânsito urbano. A criação desse protótipo permitirá avaliar algoritmos de temporização dinâmicos sem interferir nas vias. A Figura 1 mostra o diagrama UML de casos de uso da arquitetura do sistema. Nesta figura, o software servidor está embarcado no Raspberry Pi. SemaforoTel é o site que recebe requisições HTTP do software servidor embarcado. Este software se comunica com o semáforo físico através das portas de propósito geral (General Purpose Input / Output – GPIO) do Raspberry Pi.

Figura 1 - Diagrama UML de Casos de Uso da Arquitetura do Sistema.



Fonte: Autoria própria (2021).

Essa arquitetura foi implementada de duas maneiras:

1) Com um Raspberry Pi físico: O sistema utilizou um Raspberry Pi versão 3 Model B para acessar o site do servidor na nuvem. Primeiramente foi utilizado um cartão de memória de 16 GB no qual foi instalado o sistema Raspbian. Após a instalação do Raspbian no cartão de memória foi o sistema operacional no mesmo. Após a instalação do sistema operacional, o ambiente foi configurado para incluir o software servidor embarcado (CARNINO, 2021).

2) Com emulação do Raspberry Pi: A emulação foi utilizada por razões de desempenho, como alternativa para desenvolver o software servidor embarcado e para avaliar a comunicação do Raspberry Pi com o servidor da nuvem.

2.2 Software Servidor Embarcado

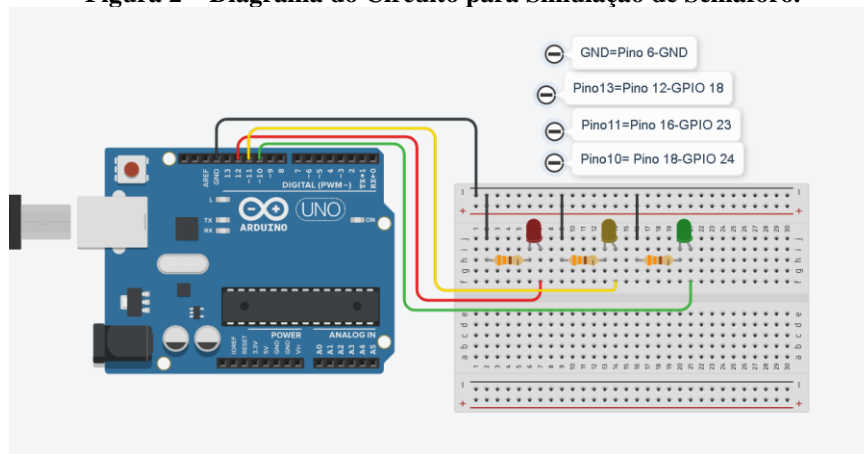


Por razões de desempenho, o software servidor embarcado foi desenvolvido no ambiente emulado com QEMU. O software servidor interage com o site da nuvem e com o dispositivo semafórico através das portas GPIO. Os seguintes softwares foram utilizados: servidor Web.py com base de dados Postgres.

2.3 Representação do Circuito

Foi utilizada a plataforma Tinkercad para elaborar um diagrama esquemático de comunicação do sistema com um semáforo. Esse diagrama foi utilizado para projetar as portas GPIO no software servidor embarcado e compreender melhor funcionamento do sistema proposto. Nesta simulação, um dispositivo Arduino interage com leds para simular a interação com um semáforo de trânsito. A Figura 2 mostra o diagrama do circuito.

Figura 2 – Diagrama do Circuito para Simulação de Semáforo.



Fonte: Autoria própria (2021).

2.4 Integração dos Sistemas

Finalmente, o software embarcado desenvolvido no ambiente emulado com QEMU foi importado dentro do Raspberry Pi físico. As ligações apresentadas no diagrama esquemático também foram realizadas no Raspberry Pi para interagir com leds. No Raspberry Pi, no código do software servidor deve ser informado o número da GPIO, ao invés do número da porta. A fonte de alimentação e tensão recomendadas são, respectivamente, 5 Volts e 2,5 Ampères.

3 RESULTADOS

O Raspberry Pi executa continuamente o servidor HTTP e recebe comandos para mudar o led do semáforo localmente. Além disso, o servidor envia para o site o status do led no semáforo. O principal resultado obtido até o momento foi a pesquisa, o estudo e o desenvolvimento do software servidor embarcado em IoT para comunicação com dispositivos semafóricos. O software servidor no Raspberry Pi recebe requisições com o protocolo HTTP e retorna respostas padronizadas em formato XML. O software servidor foi escrito em Python e utiliza Postgres como base de dados para o registro de logs de execução.

Duas etapas de interação foram implementadas neste projeto:

1) (Local) Servidor para Semáforo (Interação direta com o semáforo de trânsito): nesta etapa, as requisições são enviadas do software servidor no Raspberry Pi para o semáforo, através das portas de propósito geral (General Purpose Input / Output - GPIO).



2) (Remoto) Servidor para SemaforoTel (Interação com o site SemaforoTel). O site SemaforoTel (SEMAFOROTEL, 2021) gera rotas urbanas entre duas coordenadas. Nesta interação, o software servidor embarcado no Raspberry Pi envia requisições HTTP para o site SemaforoTel na nuvem. O site então armazena a requisição, a processa, e devolve uma resposta em formato XML para o software servidor. São utilizadas mensagens XML para padronizar a formatação das mensagens recebidas do site. Todo o processo de desenvolvimento foi registrado no site do projeto (SILVA e NOVAES, 2021). O site utiliza mensagens em formato XML para responder ao Raspberry Pi.

Tabela 1 - Operações HTTP disponíveis no software servidor embarcado.

Operação	Sintaxe	Descrição
add	http://<HOST:PORTA>/add?id=<ID>	Adicionar um semáforo. Ex.: http://localhost:1234/add?id=2
delete	http://<HOST:PORTA>/delete?id=<ID>	Remover um semáforo. Ex.: http://localhost:1234/delete?id=2
local	http://<HOST:PORTA>/local?id=<ID>&led=<ID_LED>	Ativa um led do semáforo. Ex.:http://localhost:1234/local?id=2&led=1 Desativa um led do semáforo. Ex.: http://localhost:1234/local?id=2&led=0
nuvem	https://<HOST:PORTA>/nuvem?id=<ID>&led=<ID_LED>	Retorna as informações de um semáforo específico do site da nuvem. Ex.: https://host:1234/nuvem?id=2&led=1

Fonte: Autoria Própria (2021).

4 CONCLUSÃO

Este projeto de iniciação tecnológica pesquisa desenvolveu um protótipo de software servidor embarcado para interação com um semáforo de trânsito. A motivação foi oferecer uma solução de baixo custo para controle de tráfego em vias urbanas. Este projeto permite controlar um sinal luminoso de um dispositivo semafórico real com requisições HTTP enviadas para a rede sem fio de um Raspberry Pi. Esse dispositivo está conectado ao semáforo por meio de portas de propósito geral e relés. As requisições HTTP definem a abertura e o fechamento de uma via de trânsito para oferecer um controle dinâmico da passagem de veículos conforme a demanda do tráfego. O software servidor também é capaz de consultar dados de semáforos em um site de gerência na nuvem. Como trabalhos futuros, até o presente momento, o protótipo interage apenas com leds, mas é esperado que o sistema agregue relés para interagir com um semáforo real de trânsito. Dependendo completamente de comandos HTTP para ligar e desligar o semáforo, ao invés de enviar uma configuração a ser instalada, poderá representar um risco quanto à perda de conexão (interferência de sinal na rede, por exemplo). Conexões criptografadas com HTTPS e requisições com tokens de permissão reduzida poderão evitar que os semáforos fiquem inoperantes e garantir uma segurança mínima quanto à interferência na rede. Até o momento, as mudanças das luzes do semáforo são determinadas pelas requisições HTTP. Os resultados preliminares apontam a necessidade de alternativas mais robustas do controle de temporização e também da qualidade da conexão de rede com o semáforo.

AGRADECIMENTOS



Agradeço ao Prof. Dr. Lucio Agostinho Rocha pela orientação e ensinamentos durante o desenvolvimento do projeto, a UTFPR pelas ferramentas para desenvolvimento do projeto, e auxílio com bolsa PIBITI do CNPq concedida no final da vigência do projeto de 03/2021 a 07/2021.

REFERÊNCIAS

BRANDÃO, Bruna. “O que é IoT – Como melhorar rotinas empresariais, industriais e pessoais com a internet das coisas?” Disponível em: <<https://maplink.global/blog/o-que-e-iot/>>. Acesso em ago. de 2020.

CARNINO, C. “Control the Raspberry Pi 2/3 GPIO pins with Swift 3.0 on Ubuntu 16.04”. Disponível em: <<https://medium.com/@ccarnino/control-the-raspberry-pi-2-3-gpio-pins-with-swift-3-0-on-ubuntu-16-04-c66ada06efe>>. Acesso em jul. de 2021.

G1 MT. Semáforos inteligentes devem começar a ser instalados em Cuiabá na segunda-feira. 2017. Disponível em: <<https://g1.globo.com/mato-grosso/noticia/semaforos-inteligentes-devem-comecar-a-ser-instalados-em-cuiaba-na-segunda-feira.ghtml>>. Acesso em mar. de 2021.

IBGE. Frota de Veículos. Disponível em: <<https://cidades.ibge.gov.br/brasil/pr/pesquisa/22/28120?indicador=28120&localidade1=0&localidade2=410690>>. Acesso em jul. de 2021.

KOWALSKI, R. L. “Frota de veículos volta a crescer em Curitiba e atinge número recorde”. Bem Paraná. Disponível em: <<https://www.bemparana.com.br/noticia/frota-de-veiculos-volta-a-crescer-em-curitiba-e-atinge-numero-recorde#.YL-6RfIKhPZ>>. Acesso em jul. de 2021.

MUGNELA, Bruno Sarno. GENPOLIS: Prototipagem e aplicação de um simulador de trânsito voltado para otimização de sinalização semafórica por meio de algoritmos genéticos. 2012. Dissertação (Mestrado em Sistemas Eletrônicos) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2012. doi:10.11606/D.3.2012.tde-21062013-121249. Acesso em jul. 2021.

RASPBERRY PI FOUNDATION. “Especificações do Raspberry Pi 3B 2015”. Disponível em: <<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>>. Acesso em mai. 2021.

SEMINT Semáforos. Como Funciona nosso Semáforo Inteligente? Disponível em <https://www.youtube.com/watch?v=_rs_j1qnkmQ>, 2013. Acesso em dez. de 2020.

SILVA, G. N. SemaforoTel. Disponível em: <<https://semafortel.herokuapp.com/>>. Acesso em mar. de 2021.

SILVA, Gustavo Nogueira e NOVAES, Guilherme Malaquias. Semáforo Inteligente. Disponível em: <<https://sites.google.com/view/semafortel/página-inicial>>. Acesso em mar. de 2021.