

Rede wireless de sensores EPOSMote III para indicação de vagas livres e ocupadas em estacionamentos

Authors

Daniel Maragno, Ricardo Vieira Fritsche e Thales Zirbel

Justificativa

Com o explosivo crescimento no número de veículos, encontrar vagas para estacionar tem se tornado uma tarefa estressante. Uma forma de melhorar essa situação é indicar ao motorista exatamente em que local do estacionamento há uma vaga livre. Fazer isso com o uso de recursos humanos é inviável.

Com o barateamento dos microcontroladores e desenvolvimento da IoT, torna-se viável imaginar um cenário em que cada vaga do estacionamento torna-se inteligente e conectada, podendo dar feedback em tempo real sobre o seu estado de ocupação.

Uma vez que cada vaga se torna um "dispositivo IoT", fica fácil imaginar outros benefícios agregados, como a possibilidade de analisar os dados gerados, por exemplo, para entender a taxa de ocupação e demanda em diferentes horários, permitindo otimização da gestão dos estacionamentos. Além disso, cria-se a oportunidade de conectar essas vagas a uma localização geográfica real, podendo oferecer feedback imediato no dispositivo móvel do motorista sobre a localização da vaga mais próxima em um mapa interativo. Há também espaço para oferecer facilidades como reservar uma vaga remotamente ou mesmo agregar recursos como reconhecimento de placas veiculares.

O dispositivo EPOSMote III, aliado ao protocolo TSTP se mostram ideais para a implementação da solução citada. Dessa forma, esse é o tema escolhido para o projeto final da disciplina de Sistemas Operacionais II, com foco em IoT.

Benefícios

Benefício para os motoristas

1. Rápida localização da vaga livre para estacionar, sem estresse.
2. Economia de combustível e desgaste com o carro circulando.
3. Economia de tempo, mais tempo para realizar suas compras.
4. Evita conflitos desagradáveis, brigas e disputas pela vaga de estacionamento.

Benefícios para a Empresa/Gestor

1. Melhora o fluxo de tráfego com os motoristas sendo guiados imediatamente para as vagas disponíveis.
2. Reduz o número de funcionários na área do estacionamento.
3. Diminui a carga de estresse do motorista, fazendo com que fique mais à vontade para depois permanecer no local e realizar suas compras.
4. Evita irritações do motorista, e conseqüente deslocamento para um estabelecimento concorrente porque não conseguiu estacionar seu veículo.
5. Permite realizar uma gestão baseada em dados reais de fluxo de uso do estacionamento.

Benefícios para o Meio Ambiente

1. Reduz emissões de CO₂, contribuindo para luta contra o efeito estufa global.

Objetivos

Esse projeto tem como objetivo mostrar a viabilidade do desenvolvimento de uma solução para permitir que as vagas em estacionamento se tornem dispositivos IoT, auxiliados por sensores ultra sônicos em

microcontroladores de baixo custo, conectados entre si em uma rede wireless, oferecendo feedback em tempo real para os motoristas sobre o estado de ocupação de cada vaga.

De forma sucinta, esses são os objetivos gerais:

1. Software básico para o dispositivo de vaga, permitindo indicar em um LED se a vaga está livre, além de enviar essa informação para o gateway em tempo real.
2. Software básico para o gateway, que irá receber as mensagens dos dispositivos de vaga e enviar para a API RESTful, atualizando o estado das vagas na aplicação web.
3. API RESTful para permitir a comunicação do gateway com a aplicação web.
4. Desenvolvimento de uma aplicação web para realizar a gestão dos estacionamentos e vagas, permitindo visualizar o estado de ocupação de cada vaga.

Premissas

1. Disponibilidade de pessoas com conhecimento técnico do EPOS, EPOSMote III e protocolo TSTP para responder a eventuais dúvidas que os membros do grupo possam ter.
2. Disponibilidade de no mínimo 4 dispositivos EPOSMote III para criação do protótipo de um estacionamento, demonstrando o funcionamento da solução.
3. Disponibilidade de sensor ultrassônico com viabilidade de conexão no EPOSMote III
4. Disponibilidade de LEDs verdes e vermelhos para conectar ao EPOSMote III

Entregas

Serão realizadas 4 entregas até a data limite de envio do trabalho final, seguindo o modelo de um MPV (Produto viável mínimo), onde os recursos mais fundamentais para realizar o objetivo do projeto serão entregues primeiro, de forma que havendo tempo hábil, serão realizados refinamentos e melhorias opcionais.

As entregas são detalhadas abaixo e serão realizadas em ordem.

1. **Instalação e ferramental tecnológico e pré-código (15/10/2017)**

1. Visão geral esquemática da visão geral do funcionamento do projeto
2. Ser capaz de compilar aplicações para o EPOSMote III capaz de:
 1. Acender e apagar leds conectados
 2. Fazer a leitura de valores do sensor ultra-som (fazer leitura pela porta serial)
 3. Fazer a leitura de valores do sensor ultra-som e enviar pelo protocolo TSTP (fazer a leitura através de outro ###EPOSMote III que irá servir como sink e enviar o valor lido pela porta serial)
3. Avaliar a viabilidade do EPOSMote sink enviar mensagens para a API via conexão WIFI ou GSM
 1. Definir estratégia de localização dos sensores
 2. Sensor faz broadcast do seu identificador único com a leitura do sensor, cabe ao gateway fazer a leitura e associar em um mapa
 3. Programar manualmente a localização de cada sensor
 4. Sensor faz broadcast perguntando sua localização, recebe a informação do sink e grava em seu sistema de arquivos
 5. Sensor descobre a sua localização conversando com o sink

2. **Software básico para o dispositivo de vaga (22/10/2017)**

1. Detectar se a vaga está ou não ocupada com um sensor ultrassônico
 1. Enviar informação de ocupação ao gateway
 2. Acender um LED verde ou vermelho conforme a ocupação da vaga
2. Sincronizar com um gateway

3. **Software básico para o gateway (29/10/2017)**

1. Receber mensagens TSTP dos dispositivos e encaminhar para a API e enviar a resposta para o

dispositivo, caso seja necessário

4. API RESTful (05/11/2017)

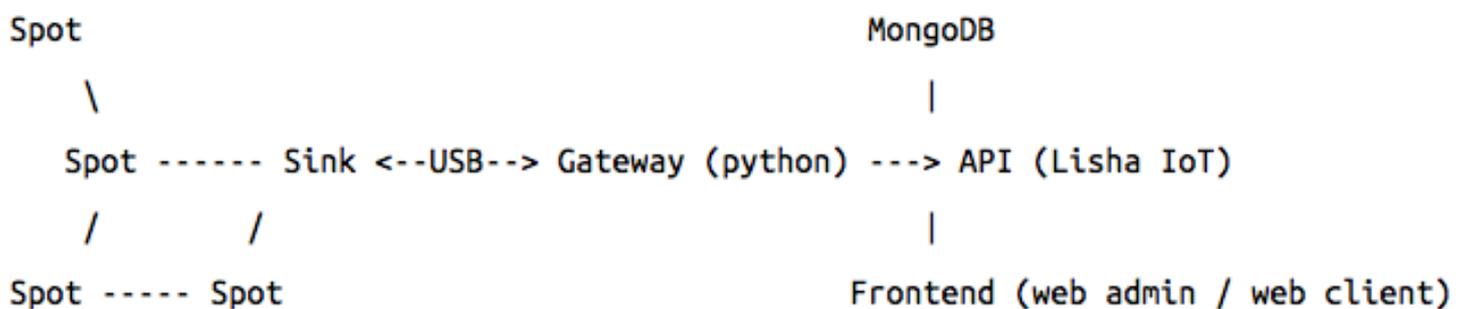
1. Permitir um dispositivo de hardware marcar a sua vaga como ocupada ou desocupada

5. Aplicação web de administração dos estacionamentos (26/11/2017)

1. Interface para criar usuários e instituições
2. Interface para instituição
 1. Criar estacionamentos e setores
 2. Em um mapa criar agrupamentos (clicando e definindo quantas vagas há no agrupamento feito)
 3. Associar um dispositivo de hardware a uma vaga (quando o dispositivo é conectado a rede pela primeira vez e não está associado a outra vaga)
3. Visualizar em tempo real o mapa de uso das vagas

Entrega 1: Instalação e ferramental tecnológico e pré-código (15/10/2017)

Visão geral esquemática da visão geral do funcionamento do projeto:



Spot -> envia seu machine ID e status de ocupado/livre

Sink -> traduz mensagens TSTP dos spots e dá um print

Gateway -> ouve o sink pela USB e repassa mensagens para API com a chave única do estacionamento

API -> recebe mensagens do gateway e responde requisições do frontend

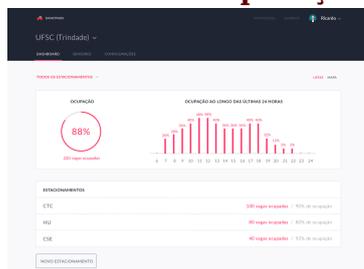
Frontend (admin) -> gerência estacionamentos, associa machine id do spot a uma vaga registrada (se possível no mapa)

Frontend (client) -> visualiza (se possível no mapa) os estacionamentos disponíveis e vagas livres por setor/agrupamento

Decisões Tomadas:

1. Utilizaremos a API do lisha.iot para enviar as mensagens da sink, através de um gateway
2. Sensores terão sua posição definidas em boot
3. Gateway sabe onde o sensor está no mapa através de um identificador único
4. A posição de cada spot será posta manualmente

Rascunhos da aplicação:



UFSC (Trindade) - 90% - 90 vagas disponíveis

OCUPAÇÃO: 90%

OCUPAÇÃO AO LONGO DAS ÚLTIMAS 24 HORAS

DETALHES:

PLANO: 90 vagas disponíveis / 100 de ocupação

ETS: 90 vagas disponíveis / 100 de ocupação

NOVA FILA

UFSC (Trindade) - 90% - 90 vagas disponíveis

OCUPAÇÃO: 90%

OCUPAÇÃO AO LONGO DAS ÚLTIMAS 24 HORAS

DETALHES:

Fila 1: 90 vagas disponíveis / 100 de ocupação

Fila 2: 90 vagas disponíveis / 100 de ocupação

Fila 3: 90 vagas disponíveis / 100 de ocupação

NOVA FILA

UFSC (Trindade) - 90% - 90 vagas disponíveis

OCUPAÇÃO: 90%

OCUPAÇÃO AO LONGO DAS ÚLTIMAS 24 HORAS

VAGAS:

1	FB9215K28190V	OCUPADA
2	0781706179780	EM VAGA
3	1092317029962	OCUPADA
4	Vaga não associada a um veículo	OCUPAÇÃO
5	Y0M05461303	OCUPADA

CRAR FILERA DE VAGA

Escreva o nome da fileira

Quantas vagas há na fileira?

Marque no mapa a localização da fileira



CRAR FILERA DE VAGA

UFSC (Trindade) - 90% - 90 vagas disponíveis

OCUPAÇÃO: 90%

OCUPAÇÃO AO LONGO DAS ÚLTIMAS 24 HORAS

DETALHES:

Fila 1: 90 vagas disponíveis / 100 de ocupação

Fila 2: 90 vagas disponíveis / 100 de ocupação

Fila 3: 90 vagas disponíveis / 100 de ocupação

NOVA FILA

UFSC (Trindade) - 90% - 90 vagas disponíveis

OCUPAÇÃO: 90%

OCUPAÇÃO AO LONGO DAS ÚLTIMAS 24 HORAS

VAGAS:

1	FB9215K28190V	OCUPADA
2	0781706179780	EM VAGA
3	1092317029962	OCUPADA
4	Vaga não associada a um veículo	OCUPAÇÃO
5	Y0M05461303	OCUPADA

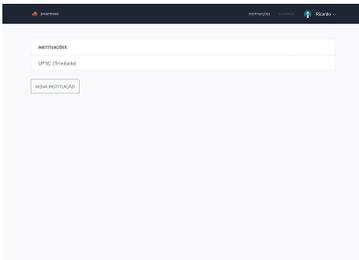
UFSC (Trindade) - 90% - 90 vagas disponíveis

OCUPAÇÃO: 90%

OCUPAÇÃO AO LONGO DAS ÚLTIMAS 24 HORAS

VAGAS:

1	FB9215K28190V	OCUPADA
2	0781706179780	EM VAGA
3	1092317029962	OCUPADA
4	Vaga não associada a um veículo	OCUPAÇÃO
5	Y0M05461303	OCUPADA



Entrega 2: Software básico para o dispositivo de vaga (22/10/2017)

Os softwares básicos foram implementados e testados. Conseguimos verificar se uma vaga está ou não ocupada com o sensor ultrassom, conseguimos enviar a informação de ocupação via gateway, conseguimos acender um led conforme a ocupação da vaga e conseguimos sincronizar com o gateway. Foi necessário criar um smartdata e um transducer para realizar o transporte entre o spot e o sink.

Entrega 3: Software básico para o gateway (29/10/2017)

O gateway está recebendo mensagens do sink, que está se comunicando com os sensores via TSTP. As mensagens são lidas e enviadas para o Lisha IoT

Entrega 4: API RESTful (05/11/2017)

Realizamos a funcionalidade do dispositivo informar se a vaga está livre ou não. Isto é realizado através da coleta das informações dos dispositivos pelo sink, que envia através do gateway para a API IoT. A aplicação então lê a disponibilidade de cada dispositivo nesta API e indica na aplicação se a vaga está ou não ocupada.

Vídeo do projeto completo funcionando com mapa (01/11/2017)

Realizamos um video demonstrando o funcionamento completo, junto da aplicação:

[Link do vídeo no Youtube:](#)

Entrega 5: Aplicação web de administração dos estacionamentos (26/11/2017)

Criamos uma aplicação para associar os spots a um mapa, permitindo visualizar a disponibilidade ou não de vagas em um determinado local. O código da aplicação está no seguinte repositório: [Repositório com código da Aplicação](#)

Código:

Acender e apagar leds conectados:

Utilizamos a aplicação já feita e disponibilizada no SVN, led_blink.

Sink realizado:

Sink que repassará os dados para a API.

Spot com sensor:

Um sensor que lê os dados do ultrassom.

Fazer a leitura de valores do sensor ultra-som:

Realizamos um transducer e um leitor. Quando lido, o transducer indica se há um erro na leitura, se a

vaga está ocupada ou se ela está livre, de acordo com "TAKEN_VALUE". [Link com informações do sensor.](#)

Gateway:

Foi realizado um gateway com um certificado para se comunicar com a API do lisha.

Fazer a leitura de valores do sensor ultra-som (fazer leitura pela porta serial):

Repositório com código da Aplicação

Pequeno teste para mostrar que se está lendo corretamente. [Video das leituras.](#)

Repositórios

Repositório com os códigos do EPOS

Repositório da Aplicação

Referências

Uso do Epos:

<https://epos.lisha.ufsc.br/EPOS+2+User+Guide>

<http://epos.lisha.ufsc.br/IoT+with+EPOS>

Informações do TSTP:

http://www.lisha.ufsc.br/pub/Resner_EUC_2016.pdf

http://www.lisha.ufsc.br/pub/Resner_SBESC_2015.pdf

http://www.lisha.ufsc.br/pub/Reghelin_MSWIM_2006.pdf

Características do sensor ultrassônico:

<https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf>

<https://www.filipeflop.com/produto/sensor-de-distancia-ultrassonico-hc-sr04/>

Acoplamento do Sensor Ultrassônico:

<https://www.filipeflop.com/blog/sensor-ultrassonico-hc-sr04-ao-arduino/>