# Curso de Sistemas de Informação

#### SISTEMA PARA RASTREIO COLABORATIVO UTILIZANDO BEACONS

Otávio Montemezzo de Souza Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil otaviom.souza@gmail.com

Francisco Assis Moreira do Nascimento Professor Orientador Faculdades Integradas de Taquara – Faccat – Taquara – RS – Brasil assis@faccat.br

#### Resumo

Este artigo apresenta o desenvolvimento de um sistema de informação para dispositivos móveis voltado para o rastreamento de pertences, de forma colaborativa, utilizando *beacons*. O aplicativo consiste em uma rede, onde os usuários podem cadastrar os seus pertences, que estão acoplados em *beacons* e então, ter total controle sobre ele, podendo rastreá-lo de forma sonora para saber sua distância aproximada, marcá-lo como perdido para que outros usuários do sistema também possam o rastrear em qualquer lugar do mundo, além de determinar a distância máxima permitida para ele se afastar. Para que a rede colaborativa do aplicativo se torne eficaz, ele possui um ranking dos usuários que mais colaboram no sistema, e também um incentivo para compra de *beacons* com desconto.

**Palavras-chave**: aplicativo, rastreio, beacon, colaborativo.

#### SYSTEM FOR COLLABORATIVE TRACKING USING BEACONS

#### Abstract

This paper presents an information system for mobile devices oriented to collaboratively tracking belongings using beacons. The mobile application consists of a network, where users can register their belongings, which have embedded beacons, and then can have full control over it, tracking it by means of sound to know its approximate distance, mark it as lost so that other system users can also track it from anywhere in the world, and to set the allowed maximum distance at which it may move away. In order to the collaborative network of the mobile application becomes really effective, there is a ranking of the more collaborative users in the system, and also a score-based incentive that can be used to purchase beacons with discount.

**Key-words:** app, tracking, beacon, collaborative.

# 1 INTRODUÇÃO

Conforme SEBRAE (2015), a internet já estimulou as pessoas a fazerem amigos, influenciarem outras pessoas, efetuarem compras, enviarem documentos, dentre muitas outras atividades. Mas, agora ela tem pela frente um desafío ainda maior, que é o de facilitar a interação das pessoas com o mundo real, utilizando os objetos que estão ao nosso redor e ainda desconectados da rede. Esta nova fase da internet será mais interativa e dinâmica conectando tudo ao mundo virtual.

Segundo GOETTEN (2015), o universo virtual criou uma ligação muito forte entre pessoas e empresas, mas os *beacons* podem alterar radicalmente esse cenário. Os *beacons* possibilitam a ligação das pessoas com o mundo real por meio de dispositivos móveis. A grande disseminação dos dispositivos móveis influenciará esse movimento. A breve transição de plataformas, o aumento do uso de novas tecnologias e o uso cada vez maior de sensores intensifica essa trajetória. Os *beacons* viabilizarão essa revolução, que tornará a experiência dos consumidores única e diferenciada.

Pensando nos dias atuais, pode-se verificar um problema muito comum: as pessoas cada vez mais vivenciam situações de perda de algum objeto, como chaves, carteiras, bolsas e, até mesmo, animais de estimação. Grande parte destes casos acontecem pela distração devido ao grande acúmulo de tarefas ou estresse.

Uma das possíveis formas de conectar estes objetos à internet (Internet das Coisas) (BAHGA, 2014) é com acoplamento de *beacons* (GOETTEN, 2015). Os *beacons* podem se comunicar com outros dispositivos através de *bluetooth* e transmitir determinadas informações.

Um dos principais desafios do setor de tecnologia é desenvolver um software para promover o encontro destes objetos sem depender de pessoas com boa índole. Para que este rastreio seja possível, cada objeto deverá possuir um dispositivo eletrônico sinalizador chamado de *beacon*, que é um dispositivo sinalizador *bluetooth* 4.0 de baixo consumo de energia.

O aplicativo, que realizaria a leitura da sinalização do *beacon*, se basearia em uma rede colaborativa, onde cada smartphone com o aplicativo instalado funciona como uma antena, ou seja, quando um usuário sinalizar ao software que seu objeto está perdido, outros usuários que possuírem o aplicativo estarão aptos para rastreá-lo e mandarem a informação para o smartphone do dono, tudo isso de forma automática, formando assim uma grande área de abrangência.

Tendo em conta esta demanda ressaltada através de um problema muito comum na vida das pessoas, o principal objetivo deste trabalho consiste justamente no desenvolvimento de um aplicativo Android que permite o rastreio de objetos perdidos, através de *beacons*, formando uma grande área de abrangência, de forma colaborativa entre os usuários do mesmo, permitindo assim rastrear objetos em vários lugares do mundo.

Para isto se fez necessário o desenvolvimento de um aplicativo Android para rastreamento de *beacons*; a integração entre os diversos usuários através de um serviço Web, a fim de formar a rede colaborativa; e a criação de alguma forma de estímulo para que os usuários auxiliem os outros usuários, que procuram por seus *beacons*, podendo ser através de um ranking ou de uma pontuação que lhe gere benefícios, como descontos na aquisição de *beacons*;

Nas próximas Seções serão apresentados os principais conceitos e tecnologias adotados na realização deste trabalho.

### 1.1 Conceitos e tecnologias

#### 1.1.1 Beacon

*Beacon* (TOWNSEND, 2014) é um nome mais amigável para a tecnologia de sistema de proximidade em ambientes fechados. Este dispositivo também é comumente chamado de iBeacon, pois ele utiliza um protocolo criado e padronizado pela Apple (IBEACON, 2015), que tem este mesmo nome.

A Figura 1 mostra um hardware *beacon* desmontado. A primeira e a última parte da imagem são apenas a estrutura que protegem o hardware (placa verde com componentes) e a sua bateria. O dispositivo pode ser de um tamanho menor que sua bateria, que tem aproximadamente 2 cm de diâmetro.

Stomman Authority Coles 31/

Figura 1 – Ilustração detalhada de um b*eacon* desmontado

Fonte – BLAGDON (2013)

A tecnologia utiliza *BLE - Bluetooth Low Energy* (Bluetooth de baixa energia), que garante a alta durabilidade de sua bateria, para transmitir um código universal único de identificação chamado de UUID. Este identificador pode também ser combinado com os códigos *Major* e *Minor* para determinar a localização de um dispositivo.

O seu funcionamento se dá através da proximidade de outros dispositivos *bluetooth*, com isso sua identificação é transmitida para este dispositivo, que deverá possuir um software para tratamento destas informações.

O *BLE* também é conhecido como *bluetooth* Inteligente ou *bluetooth* 4.0 (BLUETOOTH, 2015). Ele utiliza menos da metade da energia do que as comunicações via *bluetooth* padrão, porém possui uma velocidade de transmissão um pouco menor. Ele trabalha na mesma faixa de transmissão do *bluetooth* convencional, mas utiliza um conjunto diferente de canais. Foi projetado para uso em aparelhos que possuam uma bateria de pouca capacidade, como por exemplo os *beacons*. Estima-se que uma bateria comum pode durar pelo menos doze meses com o dispositivo *beacon* sempre ligado.

O UUID (Identificação Universal Única) é um código que contém 32 dígitos hexadecimais divididos em cinco grupos através de traços, como por exemplo: 11111111-2222-3333-4444-5555555555555555. É um sistema de identificação que permite que um número único seja gerado por dispositivo, o objetivo do ID é distinguir os seus *beacons* de todos os outros *beacons* em redes fora de seu controle. Tecnicamente uma organização pode utilizar vários UUIDs, se necessário, por exemplo para identificar os locais em diferentes pontos ou para identificar diferentes unidades de negócio.

O *Major* ou código principal é um código de dois bytes, que é utilizado para distinguir e identificar um grupo de *beacons*, como por exemplo, em qual loja de uma grande rede está aquele *beacon*, fazendo com que seja possível que o dono da rede saiba em qual loja o consumidor está.

O *Minor* ou código secundário é um código de dois bytes e identifica *beacons* individuais, como por exemplo: um *beacon* que está localizado atrás de uma loja tem apenas um Minor, o que faz com que o dono da loja saiba a localização do consumidor.

O *TX Power* é utilizado para determinar a proximidade do *beacon*. Através da potência da transmissão (intensidade do sinal) é possível obter uma estimativa aproximada de distância, mas para isso o dispositivo deverá passar por uma calibração com antecedência, que geralmente ocorre de forma automática.

### 1.1.2 Plataforma Web

Para o desenvolvimento da plataforma web foi utilizado o Java (MENDES, 2009) que é uma linguagem de programação orientada a objetos desenvolvida na década de 1990 pela empresa Sun Microsystems. Um software desenvolvido neste linguagem pode ser executado em qualquer sistema operacional, isto é possível através da sua JVM (Java Virtual Machine - máquina virtual Java), pois ela é responsável pela interpretação do seu código compilado (bytecode), sendo assim responsável pela integração com o sistema operacional.

A linguagem é extremamente poderosa, pois possui suporte para desenvolvimento de aplicações desktop, web e mobile (THJT, 2015). Para o desenvolvimento deste trabalho, foi definido o uso da linguagem Java porque o desenvolvimento de aplicativos Android nativos se dá obrigatoriamente nesta linguagem. A parte servidora da aplicação também foi desenvolvida com a linguagem para haver uma padronização.

Os serviços web que foram utilizados (WEB SERVICES, 2015) são uma maneira fácil de integrar sistemas, que utiliza como base um arquivo texto estruturado, possibilitando que aplicações possam interagir com outros sistemas desenvolvidos em plataformas diferentes.

Cada aplicação pode ser desenvolvida em uma linguagem diferente, mas as mensagens, que serão trocadas entre elas, são traduzidas para um texto estruturado universal como XML (Extensible Markup Linguage) (XML, 2015) e JSON (JSON, 2015).

A linguagem JSON (JavaScript Object Notation) foi desenvolvida baseada em um subconjunto da linguagem JavaScript (JSON, 2015). JSON é um formato de texto que visa facilitar a troca de informação de forma extremamente leve através da internet, fazendo com que sistemas desenvolvidos em linguagens diferentes possam se comunicar, além de ser uma linguagem de fácil compreensão tanto para computadores como para humanos.

Por ser um formato extremamente leve, permite o processamento de uma grande quantidade de dados de forma rápida e eficiente. Por este motivo, foi utilizado para comunicação entre o servidor e o aplicativo móvel, visando minimizar os problemas decorrentes da péssima velocidade de conexões com a Internet dos dispositivos móveis e ao grande número de requisições que o servidor deverá suportar.

Para facilitar o desenvolvimento foi utilizado o Spring (SPRING, 2015), que é um framework para linguagem Java de código aberto, criado por Rod Johnson. Um dos objetivos do Spring é simplificar a programação, porque abstrai códigos que não sejam específicos do sistema, com isso o desenvolvedor se preocupa em apenas desenvolver realmente o que é

importante para o seu projeto. O Spring possui diversos módulos, tais como, aqueles para persistência de dados e segurança da aplicação.

Os diferenciais do framework são: injeção de dependências, suas rotinas que estão prontas para uso, facilidade na aprendizagem, criação de testes unitários de forma fácil (SPRING, 2015).

#### 1.2.3 Plataforma Mobile

Android é um sistema operacional baseado em Linux desenvolvido pelo Google para dispositivos móveis (GLAUBER, 2015). Atualmente está instalado em mais de um bilhão de smartphones e tablet's (ANDROID, 2015). Este sistema conta com uma gama de funcionalidades que podem estar habilitadas dependendo do hardware do smartphone em que está instalado. Entre as diversas funcionalidades estão por exemplo: GPS, *Bluetooth*, Acelerômetro, Sensor de Luminosidade, Câmera, Wireless e outras.

O sistema operacional provê acesso aos recursos de hardware aos aplicativos que programadores irão desenvolver em linguagem Java e em um padrão especificado pela versão mínima do Android escolhida para suporte (ANDROID, 2015).

### 1.3 Sistemas Correlatos

# 1.3.1 TrackR

Conforme o site do produto, o TrackR (TRACKR, 2015) é um pequeno dispositivo do tamanho de uma moeda que se prende facilmente aos seus itens valiosos. Com o aplicativo TrackR, o usuário pode localizar qualquer item perdido ou extraviado em segundos tocando um alarme, usando um indicador de distância ou pela rede colaborativa de GPS.

O sistema TrackR possui um grande ponto fraco, ele não estimula o usuário a manter o Bluetooth do seu smartphone ligado, para que mesmo quando não estiver procurando por nenhum de seus pertences, ele esteja ajudando outros usuários através da sua rede colaborativa, com isso, sua rede colaborativa não funciona com eficiência, pois o usuário desliga o *Bluetooth* para economizar bateria, utilizando o aplicativo somente quando perde algum de seus pertences. Além disso, conforme relatos dos usuários brasileiros nas redes sociais do software, eles teriam efetuado a compra do TrackR e já se passaram meses e ainda não teriam recebido o produto, portanto o software não é bem avaliado no Brasil.

A Figura 2 é uma captura do site da TrackR, onde mostra as principais funcionalidades do sistema que é fornecido, como por exemplo, indicador de distância de um objeto, tocar um sinal no dispositivo perdido, rede colaborativa de GPS, design pequeno do *beacon* entre outras.

Phone Finder Distance Indicator Item Ringer Hotter? Colder? The TrackR app displays the With the quick tap of your finger, TrackR can Can't find your phone? Press your TrackR distance between you and your items device's button to ring your phone, even if it's under the mail! Finding keys has never been on silent! We make finding your phone too letting you know how far those pesky keys are easier. easy. Crowd GPS Separation Alerts Wallet Thin Design Lost something? TrackrR's Crowd GPS Forgot your phone again? Custom 2-way TrackR bravo uses beautiful anodized network will help you find it. When a TrackR separation alert notifies you before you leave aluminum construction for the thinnest and user is within range of your lost item, you items behind! most durable tracking device. receive a GPS update.

Figura 2 – Ilustração das funcionalidades do sistema

Fonte – Captura de tela do site do produto (TRACKR, 2015)

A Figura 3 é uma captura da tela inicial do aplicativo desenvolvido pela TrackR, onde mostra as opções para adicionar um novo dispositivo a sua conta, efetuar login e aprender mais sobre o aplicativo.



Figura 3 – Tela inicial do aplicativo

Fonte – Captura de tela inicial do aplicativo (TRACKR, 2015)

### 1.2.2 Tile

Conforme o site do sistema, Tile (TILE, 2015) é uma etiqueta Bluetooth que permite o rastreamento de objetos. Com seu aplicativo é possível encontrar um item ativando um som no dispositivo, ver no mapa onde o objeto se encontra, e utilizar a sua rede comunitária para rastrear objetos.

O aplicativo Tile compartilha do mesmo ponto fraco do aplicativo TrackR, a falta de estímulo ao usuário a manter o uso do aplicativo acionado para ajudar outros usuários e também a falta de informações para importação do produto para o Brasil.

A Figura 4 é uma captura da tela inicial do aplicativo desenvolvido pela Tile, nela é possível utilizar as funcionalidades de ver todos os meus dispositivos cadastrados, cadastrar novos dispositivos, e visualizar os dispositivos no mapa.



Figura 4 – Tela inicial do aplicativo

Fonte – Captura de tela inicial do aplicativo (TILE, 2015)

A Figura 5 é uma captura do site da Tile, onde mostra as principais funcionalidades do sistema que é fornecido, como por exemplo, tocar um sinal no dispositivo perdido, ver o dispositivo no mapa, procurar o seu telefone através de um botão no seu *beacon*.

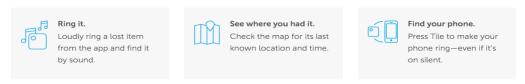


Figura 5 – Ilustração das funcionalidades do sistema

Fonte – Captura de tela do site do produto (TILE, 2015)

A Tabela 1 mostra de forma analítica as principais funcionalidades implementadas nos aplicativos correlatos e no aplicativo desenvolvido, denominado de Wanted. É possível notar que o Wanted possuí características para facilitar e fidelizar o seu uso, através de sua integração com as redes sociais e incentivos gerados aos usuários para compra de Beacons, o que torna a sua rede colaborativa muito mais eficiente do que a dos aplicativos correlatos.

Tabela 1 – Comparativo das principais funcionalidades dos sistemas

Funcionalidade	Wanted	TrackR	Tile
Rede Colaborativa (mapa) de Rastreio	Possuí	Possuí	Possuí
Rastreio Sonoro	Possuí	Possuí	Possuí
Compra de Beacons diretamente no aplicativo	Possuí	Não possuí	Não possuí
Uso do mesmo Beacon por mais de um usuário	Não possuí	Não possuí	Possuí
Incentivo aos usuários para uso contínuo do aplicativo gerando pontos e descontos na compra de Beacons	Possuí	Não possuí	Não possuí
Integração com redes sociais	Possuí	Não possuí	Não possuí

**Fonte: Autor** 

#### 2 WANTED

Na presente pesquisa foi desenvolvido o sistema Wanted, um aplicativo para o sistema operacional Android, que é voltado para o rastreio de pertences de forma colaborativa, ou seja, os usuários do sistema colaboram com outros para que encontrem os seus pertences, que foram perdidos. Além disso, o aplicativo Wanted possui outras funcionalidades para facilitar o rastreio e também uma forma de fidelização dos usuários ao uso contínuo do sistema. Para que o rastreio de um pertence seja possível, ele deverá possuir um *beacon* acoplado a ele, e assim o aplicativo será capaz de rastreá-lo via *bluetooth*.

### 2.1 Funcionalidades

O aplicativo Wanted necessita que o usuário esteja com o seu *bluetooth* sempre ativo, então sempre que o aplicativo é acionado e o *bluetooth* estiver desativado ele solicita que o usuário o ative, conforme demonstrado na Figura 6.

Figura 6 – Solicitação do aplicativo para ativar o Bluetooth



Fonte - Autor

Para iniciar o uso do aplicativo, os usuários precisam criar uma conta. Para que esta tarefa não seja burocrática, Wanted permite o cadastro de usuários no sistema de forma extremamente fácil, pois ele possui integração com as redes sociais Facebook e Google.

Na Figura 7, pode-se ver o cadastro de usuários de forma manual, solicitando-se apenas dados básicos para criação de uma conta para acesso ao aplicativo.

Wanted

Nome

E-mail

Senha

Data de nascimento

Sexo

Masculino

✓

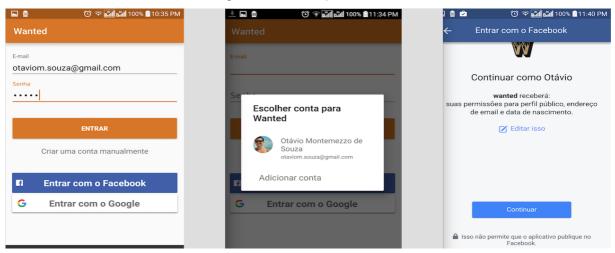
INSCREVER-SE

Figura 7 – Tela para criação de conta

Fonte-Autor

A autenticação no aplicativo exemplificada na Figura 8 se dá na mesma forma do cadastro de usuário. O utilizador pode escolher entre preencher o seu usuário e senha de forma manual ou efetuar autenticação através da integração com redes sociais através dos mesmos botões.

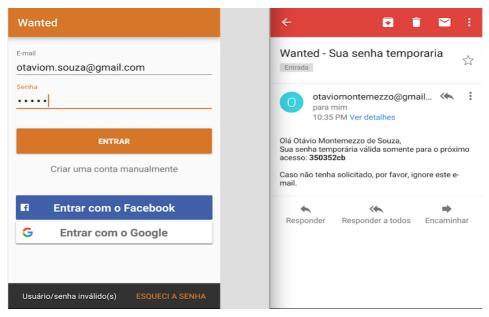
Figura 8 – Autenticação de usuário



Fonte – Autor

Caso o usuário esqueça sua senha, no momento da autenticação, o aplicativo Wanted exibe uma mensagem no rodapé com uma opção para recuperar sua senha, recebendo então as instruções através de um e-mail conforme a Figura 9.

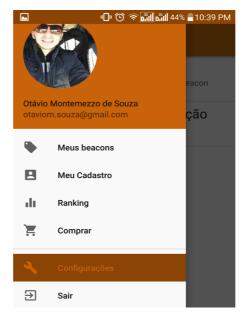
Figura 9 – Opção para recuperação de senha de acesso



Fonte - Autor

A tela principal do aplicativo Wanted, após o usuário se autenticar, exibe uma listagem de seus *beacons*, mas ele possuí um menu a sua esquerda o qual possibilita acesso as outras funcionalidades do sistema conforme a Figura 10.

Figura 10 – Menu principal do aplicativo



Fonte - Autor

No menu Meu cadastro, o usuário poderá gerenciar sua conta, podendo alterar seus dados, senha, foto (com as opções de selecionar da galeria, ou tirar uma foto através do aplicativo) e até mesmo excluir a sua conta, se assim o desejar, como exemplificado na Figura 11.

Figura 11 – Tela para gerenciar conta

Fonte – Autor

Na tela "Meus Beacons", o usuário irá ver os seus *beacons* e os que estão sendo detectados. Ao clicar sobre um deles, poderá cadastrá-lo em sua conta, caso o mesmo não seja de outro usuário do sistema, definindo assim o seu nome e sua foto (com as opções de

selecionar da galeria, ou tirar uma foto através do aplicativo) e também remover *beacons* vinculados a sua conta. Além disso, as principais funcionalidades do sistema são encontradas na tela da Figura 12.

E Beacons

Chave Carro
Distância: Fora de alcance

Carteira
Distância: Fora de alcance

Descrição
Chave Carro

Perdi meu beacon
Permite que outros usuários da rede ajudem a rastrear seu beacon
gerando um histórico de localização

Manter a uma distância de 0 metros

PHISTORICO

RASTREAR

SALVAR

EXCLUIR

Figura 12 – Tela para gerenciar beacons

Fonte – Autor

Ao clicar no botão Rastrear, o usuário poderá rastrear o seu *beacon* de forma sonora e visual, onde quanto mais perto ele estiver, mais rápido será reproduzido o som, junto com esta informação é exibida a sua distância aproximada em metros. Na Figura 13, é possível perceber que os metros são exibidos em tons de verde variando para tons de amarelo e tons de vermelho a medida que o *beacon* fica mais distante.

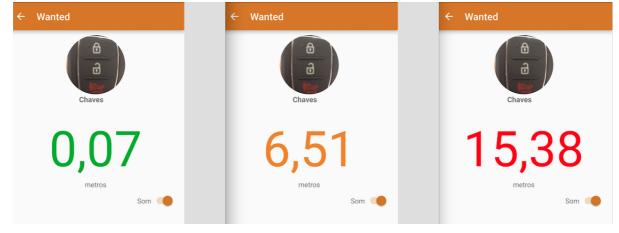


Figura 13 - Rastreamento sonoro de *beacons* 

Fonte – Autor

Caso o usuário defina um valor para o campo "Manter a uma distância de", o sistema controlará a distância aproximada que o *beacon* está e, caso ele a ultrapasse ou fique fora de alcance, o sistema irá notifica-lo, conforme mostrado na Figura 14. Esta notificação se repetirá em um intervalo, que pode ser configurado na tela de configurações do sistema. Com esta funcionalidade, os pais de uma criança poderão, por exemplo, monitorar sua distância na beira da praia.

Figura 14 – Notificação exibida quando ultrapassada distância permitida



Fonte – Autor

A principal funcionalidade do sistema pode ser ativada pelo botão "Perdi meu beacon". Com ela, o usuário permitirá que outros usuários do aplicativo fiquem aptos para rastrearem seu pertence, assim ajudando de forma colaborativa. Quando um pertence for encontrado por outro usuário do sistema, o dono deste pertence receberá uma notificação (Figura 15). Ao clicar na notificação, é exibida uma tela apontando o seu local no mapa e o horário em que foi encontrado. Através disso o usuário pode clicar para abrir este local no programa rotas GPS em seu telefone, calculando então uma rota até o local em que o pertence foi encontrado. Caso o usuário deseje acessar o histórico dessas localizações de um determinado pertence, ele poderá clicar no botão Histórico. Sempre que uma localização for mostrada no mapa, ela possuirá um círculo em sua volta, que significa o raio de precisão do GPS no momento da detecção daquele *beacon*.

Caso algum pertence perdido seja rastreado pelo seu próprio dono, ele será notificado, informando que o pertence está próximo, e ao clicar na notificação, automaticamente ele irá entrar na tela de rastreio sonoro do *beacon*.

⑧ ⓒ 후 📶 🞳 45% 💆 9:42 PM 9:50 PM **# 8** OUA. 31 DE AGO DE 2016 Pesq postos gas, cx eletron Chaves encontrado ■ TCA Informática Ltda 9:50 PM 0 31/08/2016 21:30:57 Chaves está proximo Distância: 0,01 metros Sociedade 5 de Maio • Googleorian 31/08/2016 21:30:57 informações de Wanted

Figura 15 – Passos do rastreio colaborativo de beacons

Fonte - Autor

Na tela de Configurações (Figura 16), o usuário poderá configurar o tempo de intervalo entre notificações e também o tempo em que o aplicativo deve considerar para que um *beacon* seja considerado como desaparecido após a sua última leitura.

Figura 16 – Configurações personalizadas do usuário



Fonte-Autor

### 2.2 Características

Para incentivar os usuários a manterem o seu Bluetooth ligado, assim ajudando a rede colaborativa, o aplicativo Wanted possui um menu, chamado Ranking, conforme exibido na Figura 17. Nele são mostrados os usuários que mais pontos conquistaram. O aplicativo atribui 1 ponto por minuto de *bluetooth* ligado, e com isso os usuários poderão disputar uma melhor colocação no ranking.

Figura 17 – Ranking dos melhores usuários



Fonte - Autor

Além disso, os usuários poderão utilizar os pontos gerados para compra de *beacons* com desconto. Na tela do menu Comprar (Figura 18), ele poderá ler sobre os *beacons* disponíveis para venda, informar a quantidade e o número de pontos que deseja utilizar. Com isso o aplicativo calcula automaticamente o desconto aplicado na venda. A cada 10 mil pontos, o usuário ganha 1% de desconto, limitado a um máximo de 10% por compra. Após isto, o usuário deve clicar no botão Comprar via PagSeguro e, neste momento, o sistema integra as informações da venda com o PagSeguro, incluindo: os produtos selecionados, quantidade, peso dos produtos e comprador. A partir deste momento, o PagSeguro gerencia a venda, solicitando o endereço de entrega, para calcular o valor do frete dos produtos, e também cuida da cobrança, que poderá ser feita através de boleto ou cartão de crédito. Todas as vendas são vinculadas a uma conta de vendedor e o dono desta conta pode efetuar o saque do valor total vendido pelo PagSeguro.

Comprar E-mail já cadastrado R\$ 107,91 Pulseira Bea Identificação O e-mail **otaviomontemezzo@gmail.com** já está cadastrado no PagSeguro. 0 x 23.9 x 2.6 mm. Bateria: CR2016, Distancia: Ate 70 metros sem Informe a senha: R\$ 119,90 VALOR TOTAL DOS ITENS 1135335446 PONTOS À UTILIZAR: DESCONTO APLICADO: VALOR FINAL (SEM FRETE): R\$ 107,91 Obs: O frete será calculado pelo PagSeguro OH Usar outro e-mail COMPRAR VIA PAGSEGURO naaseauro

Figura 18 - Compra de Beacons com desconto

Fonte – Autor

### **3 DESENVOLVIMENTO**

O modelo de processo de software adotado para o desenvolvimento deste projeto foi o linear (ou em cascata) incremental. Neste tipo de processo de software, as atividades estão dispostas de forma que, ao final de uma atividade se dará o início de outra, o processo é completamente sequencial, só podendo trocar de operação no momento que a fase anterior estiver finalizada. Este é um processo de software bastante adequado para este projeto, que possuí apenas um integrante. Assim, o desenvolvimento foi dividido em cinco etapas: análise, projeto, codificação, testes, e implantação. A parte incremental do processo de software se deve ao fato de se tratar de um projeto de pesquisa, em que eventualmente podem surgir novos requisitos ao longo da realização do projeto, levando à necessidade de um novo ciclo sequencial para tratá-los.

### 3.1 Análise

A fase de análise deste projeto foi iniciada fazendo-se o levantamento de requisitos necessários para o desenvolvimento do sistema, pesquisando-se e comparando-se alguns sistemas correlatos e também trocando-se ideias com diversas pessoas à respeito do sistema. Com isso, foi possível definir todas as funcionalidades que deveriam ser implementadas. A Tabela 2 mostra todos os requisitos funcionais identificados e a Tabela 3 todos os requisitos não funcionais do sistema.

Tabela 2 – Requisitos Funcionais do Sistema

Requisito Funcional	Descrição do Requisito
RF001 – Criar conta	Permitir o cadastro de um novo usuário solicitando apenas dados básicos e
manualmente ou através de redes	criação de um login e senha para acesso à aplicação. Deve também permitir
sociais	que o crie o seu usuário através das redes sociais (Facebook e Google)
RF002 – Autenticação de usuário RF003 – Manter usuário	Permitir que os usuários efetuem o seu login no aplicativo com seu usuário
	e senha ou com sua rede social Permitir que o usuário altere os seus dados cadastrais, assim como altere
	sua senha e exclua a sua conta do sistema.
RF004 – Manter beacon	Permitir cadastrar, alterar e excluir beacons
	Permitir que o usuário rastreie um beacon de forma sonora, onde quanto
RF005 – Rastrear beacon de	mais perto estiver o beacon mais rápido será reproduzido o som, junto com
forma sonora	esta informação deve ser exibida a sua distância aproximada em metros.
RF006 – Configurar alarme de	Permitir que o usuário configure qual a distância máxima que um
distância máxima do beacon	determinado objeto pode ficar longe do seu smartphone.

RF007 - Rastreador colaborativo dos beacons entre os usuários do sistema

RF008 - Ranking de usuários que mais ajudam outros usuários
RF009 - Compra de beacons com desconto
RF010 - Configuração de preferências dos usuários

Permitir que um determinado usuário marque o seu beacon como perdido, através disso outros usuários do aplicativo estarão aptos para também rastrearem este, assim ajudando de forma colaborativa. Quando um beacon for encontrado por outro usuário do sistema, o dono deste beacon receberá uma notificação apontando o seu local no mapa e o horário em que foi encontrado.

Possuir um ranking, onde irá mostrar através de pontos os usuários que mais tempo ficam com seu Bluetooth ligado, ajudando assim os outros usuários do sistema a rastrearem os seus beacons perdidos.

Possuir venda de beacons, que permita que o usuário adquira um Beacon com descontos utilizando a sua pontuação.

Possuir que o usuário ajuste suas preferências de tempo para novas notificações e de tempo para considerar um beacon desaparecido.

Fonte: Autor

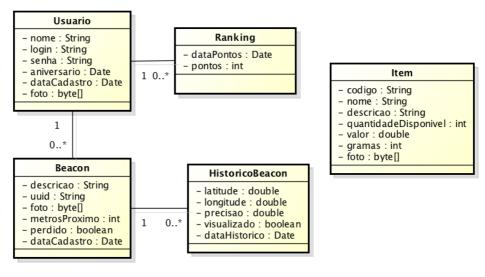
Tabela 3 – Requisitos Não Funcionais do Sistema

Requisito não Funcional	Descrição
Alta disponibilidade	O sistema deve master-se sempre disponível para acesso.
Segurança	O sistema deve manter a integridade e consistência dos dados.
Alto desempenho	O sistema deve garantir rápidos tempos de resposta para as operações executadas.

Fonte: Autor

Foi então desenvolvido um diagrama de domínio para identificação das classes, atributos e suas associações. Ver na Figura 19, o diagrama de domínio, que mostra as cinco classes utilizadas no sistema.

Figura 19 – Diagrama de domínio



Fonte – Autor

Para identificação dos atores do sistema e as operações que eles poderão realizar, foi desenvolvido um diagrama de casos de uso. Existe somente um ator no sistema e este ator possuí acesso a todas as operações, como mostrado na Figura 20.

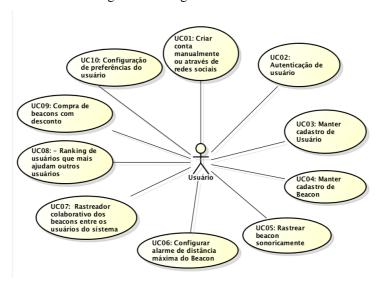


Figura 20 – Diagrama de casos de uso

Fonte - Autor

# 3.2 Projeto

Com as informações obtidas na fase de análise, foi possível partir para a fase de projeto dos diagramas da aplicação, incluindo as sequências de operações mais vitais do sistema, como por exemplo, o rastreio colaborativo entre dois smartphones, mostrado na Figura 21, e a estrutura de implantação, mostrada na Figura 22.

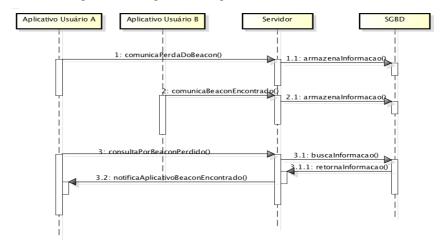
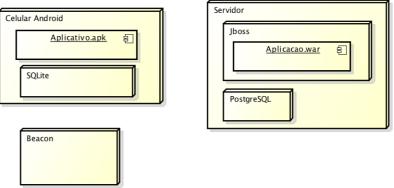


Figura 21 – Diagrama de sequencia de rastreio colaborativo

Fonte – Autor

Conforme a Figura 21 demonstra, o aplicativo deverá realizar a consulta no servidor para verificar se os seus objetos perdidos já foram localizados, ou seja, o servidor não fica encarregado de enviar está informação ao aplicativo.

Figura 22 – Diagrama de implantação



Fonte – Autor

A Figura 22 mostra a estrutura de implantação do sistema, no telefone há uma instância do aplicativo e de um banco de dados em arquivo SQLite, o Beacon é completamente independente do aplicativo, podendo ser utilizado para outros aplicativos que utilizem Beacons. Na parte servidora, o servidor de aplicação Jboss é o responsável por rodar a aplicação web do sistema. Esta aplicação é dependente de uma instância do banco de dados PostgreSQL.

### 3.3 Codificação

Com as fases de análise e projeto concluídas, foi possível seguir para a fase de codificação do sistema. O sistema foi codificado em duas etapas, a etapa do aplicativo mobile para Android, e a etapa do serviço web para receber as informações do aplicativo.

O desenvolvimento deste software foi feito com a utilização de duas IDE (*Integrated Development Enviroment*, ou Ambiente Integrado para desenvolvimento), para o desenvolvimento do aplicativo mobile foi utilizada a IDE Android Studio (STUDIO, 2015) e para desenvolvimento do servidor, a IDE NetBeans (GONÇALVES, 2006).

A manutenção do software foi feita através do uso de ferramentas para controle de versão, utilizando ramos separados para as versões de desenvolvimento e de produção. A aplicação servidora e o aplicativo móvel possuem controles de versão distintos com o uso da ferramenta Git (SILVERMAN, 2013).

Para o funcionamento do aplicativo é exigido Android com versão 4.0 ou superior, devido a limitação do sistema operacional na comunicação com Beacons com versões Android inferiores das citadas.

A principal parte da codificação deste sistema acontece na leitura dos Beacons, através do *bluetooth dos* aparelhos. Para facilitar realizar esta leitura, foi utilizada a biblioteca de código aberto AltBeacon (ALTBEACON, 2015) que provê diversas classes e comandos, que são explicados na Figura 23.

Figura 23 – Código fonte para leitura dos beacons

```
//instância o gerenciador dos beacons
BeaconManager mBeaconManager = BeaconManager.getInstanceForApplication(getApplicationContext());
//define para que rode também em segundo plano
mBeaconManager.setBackgroundMode(true);
//determina o tempo em milissegundos de cada leitura
mBeaconManager.setBackgroundScanPeriod(1100);
//determina o intervalo de tempo milissegundos entre cada leitura
mBeaconManager.setBackgroundBetweenScanPeriod(10 * 1000);
//determina o layout do UUID que será lido, neste caso utilizando o padrão Apple - IBEACON
mBeaconManager.getBeaconParsers().add(new BeaconParser().
        setBeaconLayout(BeaconLayout.IBEACON.layout()));
//determina o método que será acionado após a leitura dos beacons
mBeaconManager.setRangeNotifier(new RangeNotifier() {
    public void didRangeBeaconsInRegion(Collection<Beacon> beacons, Region region) {
        //método recebe a lista dos Beacons lidos naquele momento
        System.out.println("Beacons lidos: " + beacons.size());
});
//aciona o gerenciador para começar a varredura de beacons
mBeaconManager.startRangingBeaconsInRegion(new Region(UNIQUE_RANGING_ID, null, null));
```

Fonte - Autor

Conforme a Figura 23, o gerenciador dos beacons foi setado para atuar em modo de segundo plano, através do método "setBackgroundMode", o que permite a leitura dos beacons mesmo que o usuário não esteja na tela do aplicativo. Através dos métodos "setBackgroundScanPeriod" e "setBackgroundBetweeScanPeriod" foi possível determinar a característica de leitura de beacons do aplicativo. O primeiro método determina o tempo de cada leitura e o segundo método o intervalo entre estas leituras. O tempo configurado neste caso foi de 1100 milissegundos para o tempo de leitura e de 10 segundos para o intervalo entre as leituras, esta configuração influencia diretamente no consumo de bateria do aplicativo. O método "didRangeBeaconsInRegion" é acionado pela biblioteca toda vez em que um ou mais beacons são lidos, recebendo então como parâmetro uma coleção de Beacons.

# 3.4 Testes

A aplicação desenvolvida passou por diversos testes funcionais. O principal teste realizado a nível de funcionalidade envolveu o rastreio colaborativo, onde foram utilizados três *smartphones* distintos, um deles possuía um beacon vinculado a sua conta, e estava com o seu *beacon* marcado como perdido, e o seu *bluetooth* desligado. Com este cenário, no momento em que algum dos outros smartphones passassem por este *beacon* perdido deveriam comunicar ao servidor suas informaçõesApós isso, o usuário dono do *beacon* deveria receber uma notificação, alertando sobre o rastreio do mesmo. Com este teste, foi possível identificar algumas falhas, que foram prontamente corrigidas As falhas encontradas com este teste foram:

- a) O proprietário do *beacon* recebia notificações repetidas, pois a todo momento os celulares que o estavam rastreando enviavam a mesma informação, por continuarem ao lado do objeto perdido; tendo sido então implementada uma validação para correção deste problema.
- b) Em alguns momentos, a informação do *beacon* perdido, que estava sendo rastreado, não era enviada ao servidor; tendo sido identificado o motivo como sendo devido a um erro que ocorria ao se acessar as coordenadas GPS do *smartphone*; tendo sido então possível corrigir o método que obtinha as informações das coordenadas GPS.

Outro teste bastante importante realizado foi o de consumo de bateria, pois a aplicação utiliza *bluetooth* e se comunica via internet várias vezes com o servidor para checar informações. Para a realização deste teste, foram deixados três *smartphones* com o aplicativo rodando durante todo um ciclo de vida da sua bateria. Foi possível então verificar pelos gráficos de consumo de bateria do sistema operacional, que o aplicativo estava consumindo mais de 30% da bateria com o uso do *bluetooth*. Com esta informação foi possível verificar que a leitura dos *beacons* estava consumindo muita bateria e, para correção do problema, foram aplicadas melhorias no código, que lia as informações dos *beacons*, pois a biblioteca utilizada oferecia configurações que permitiram diminuir o consumo de bateria. Após o ajuste, foi efetuado novamente o teste e verificado que o consumo foi reduzido para, em média, 4%.

### 3.5 Implantação

A implantação do sistema se dá em duas etapas: a etapa do aplicativo e a do servidor *web*. A primeira é extremamente simples, pois é o que todos os usuários do sistema deverão fazer, bastando para isto instalar o aplicativo no smartphone, como qualquer outro aplicativo.

O próprio aplicativo se encarrega das configurações iniciais, como configuração do seu banco de dados local, por exemplo.

Para geração do arquivo de aplicativo instalável para os usuários finais, deverá ser ajustado o endereço de conexão com o servidor web instalado. Este endereço é definido no atributo estático chamado "URL" da classe "ConexaoWS", que fica no pacote "br.com.wanted.util".

Para a implantação do servidor web, a máquina precisa do banco de dados PostgreSQL 9.0 ou superior instalado e também do servidor de aplicação Jboss 7 (JBOSS, 2015). Antes da compilação do sistema, deverá ser ajustado o arquivo "jdbc.properties", que fica localizado na pasta "src/main/webapp/WEB-INF". Nele deverá ser preenchido a informação sobre a conexão com o banco de dados PostgreSQL, incluindo usuário, senha, endereço e base de dados. Após esta configuração, o sistema deve ser compilado e instalado no servidor de aplicação Jboss 7, dentro da pasta "standalone/deployments".

### **4 EXPERIMENTOS**

A capacidade essencial do sistema desenvolvido é de rastrear colaborativamente pertences perdidos. Caso um usuário tenha perdido seu animal de estimação, sua carteira na rua, ou tenha tido sua bicicleta roubada, por exemplo, ele informa ao aplicativo que o seu pertence está perdido, então o aplicativo envia esta informação para o serviço *web* do sistema. No momento em que outro usuário com o aplicativo instalado passar com seu *smartphone* próximo a um pertence, o aplicativo irá avisar automaticamente o serviço *web* passando as coordenadas GPS do local, mesmo que o aplicativo do usuário esteja fechado, pois ele continua executando em segundo plano. Então, neste momento, o serviço *web* irá checar se aquele objeto está perdido, caso o objeto esteja mesmo sendo procurado, o serviço irá notificar o telefone do dono do objeto, e ele irá poder ver em um mapa, qual o local aproximado em que seu objeto se encontra. A Figura 24 ilustra em alto nível este processo.

Smartphone 1

1. Avisa ao servidor que perdeu o objeto A

2. Armazena informações sobre o objeto perdido

5. Armazena informações sobre o objeto encontrado

6. Envia informações para o dono do objeto encontrado

4. Envia dados do Objeto encontrado para o servidor

Figura 24 – Comunicação entre os dispositivos beacons, smartphones e o servidor

Fonte – Autor

Na Figura 25 podemos observar o aplicativo em uso, rastreando de forma sonora e visual um pertence perdido, nela é possível perceber que o sistema aponta uma distância aproximada de 0,01 metros do pertence que está ao fundo, como ele está próximo a cor da fonte é verde, variando para tons de amarelo e vermelho conforme a distância aumenta. Junto com a variação da cor, o sistema também emite um sinal sonoro que possuí intervalos cada vez menores quanto mais próximo do objeto. Este sinal sonoro pode ser interrompido a qualquer momento, basta o usuário desativá-lo na chave "som" que é exibida na imagem.

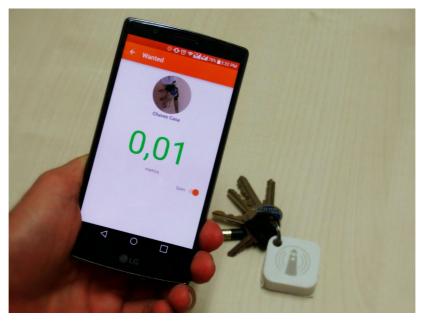


Figura 25 – Comunicação entre os dispositivos beacons, smartphones e o servidor

Fonte - Autor

### **5 TRABALHO FUTURO**

Devido ao escopo do projeto e do tempo disponível para o desenvolvimento, algumas funcionalidades secundárias não foram implementadas, e podem ser desenvolvidas em um futuro para a melhoria do sistema em questão, como por exemplo:

### a) Compartilhamento de beacons:

O aplicativo desenvolvido permite que um *beacon* seja atrelado somente a um usuário do sistema, porém poderia possuir uma funcionalidade onde fosse permitido que dois ou mais usuários utilizassem o mesmo *beacon*. Como, por exemplo, um animal de estimação seria cadastrado em todas as contas de uma família. Para que isso fosse possível, o primeiro usuário que cadastrou o *beacon* em sua conta, definiria uma chave de acesso, então outros usuários que soubessem esta chave poderiam cadastrá-lo em sua conta.

### b) Rastreamento colaborativo de beacons de forma pública:

A atual forma da rede colaborativa do aplicativo somente notifica ao dono do *beacon* quando este for encontrado, ou seja, o usuário do aplicativo que passou próximo a um *beacon* que estava perdido e acabou ajudando outro usuário não fica sabendo que este evento efetivamente ocorreu, pois de dá de forma automática através da interação entre seu dispositivo móvel e o servidor web do sistema Wanted. Desta forma não se estabelecesse uma relação de confiança entre os usuários. Como uma segunda funcionalidade, o rastreio de *beacon* poderia ter o modo privado (atual) e o modo público, o qual permitiria que as duas partes fossem notificadas no momento em que um *beacon* fosse encontrado, proporcionando assim que o usuário que rastreou o *beacon* perdido de alguém possa pegar este objeto para devolver ao seu dono, podendo conversar com o mesmo através de um chat. No momento, em que um usuário encontra um *beacon* de outro usuário e faz a devolução, ele poderia ser recompensado com um determinado número de pontos, que hoje são utilizados no ranking e na compra de *beacons*.

# c) Configuração de horários para alarme de beacons:

O aplicativo pode possuir uma funcionalidade para que se registre horários em que determinado *beacon* deve ficar próximo. Por exemplo, um usuário deseja que sua carteira sempre fique perto quando estiver no horário de trabalho. Hoje no aplicativo o usuário não pode determinar o horário destas notificações, apenas determinar uma distância máxima que o pertence pode se distanciar.

### 6 CONCLUSÃO

Este trabalho de pesquisa e desenvolvimento envolveu a criação de um aplicativo para dispositivos móveis, chamado Wanted, que deverá auxiliar as pessoas em seu dia a dia, trazendo a elas mais comodidade para que evitem que percam pertences importantes e os rastreie de forma colaborativa, caso isto aconteça. Os *beacons*, que são utilizados para o rastreio dos pertences neste aplicativo, estão ficando cada vez mais populares, porém sua comercialização ainda é um pouco limitada em alguns países.

A metodologia utilizada auxiliou muito na criação do sistema, pois houve surgimento de novos requisitos ao longo do projeto, e o processo incremental auxiliou nesta organização, deixando assim o desenvolvimento mais claro e objetivo.

A versão final do aplicativo está pronta para ser implantada, pois o mesmo está completamente funcional para o objetivo proposto, que é o rastreamento de pertences de forma colaborativa entre os usuários do sistema. Porém, o aplicativo pode ainda receber diversas novas funcionalidades, que iriam revolucionar ainda mais o uso desta tecnologia e atrair ainda mais o interesse dos usuários. O aplicativo desenvolvido funciona apenas em dispositivos móveis com o sistema operacional Android, assim, outra possível melhoria seria o desenvolvimento do mesmo aplicativo para outros sistemas operacionais móveis existentes.

O aplicativo Wanted foi divulgado para vários usuários de tecnologias móveis e, por todos eles, foi muito bem aceito. A opinião geral sobre o aplicativo é que ele revoluciona o uso da tecnologia, auxiliando na vida das pessoas, até mesmo, sem que elas mesmo percebam isto.

### REFERÊNCIAS

GOETTEN, Vicente. **Beacons: uma revolução na experiência do consumidor.** Disponível em: <a href="http://blog.totvs.com/beacons-uma-revolucao-na-experiencia-do-consumidor/">http://blog.totvs.com/beacons-uma-revolucao-na-experiencia-do-consumidor/</a>. Acesso em: 24. set de 2015.

BAHGA, Arshdeep. Internet of Things (A Hands-on-Approach). Editora VPT. Atlanta, 2014.

TOWNSEND, Kevin. Getting Started with Bluetooth Low Energy: Tools and Techniques for Low-Power-Networking. O'Reilly Media. Newton, 2014.

TEXEIRA, Fabrício. **Tudo o que você precisa saber para começar a brincar com iBeacons.** Disponível em: <a href="http://arquiteturadeinformacao.com/ux-em-espacos-fisicos/tudo-o-que-voce-precisa-saber-para-comecar-a-brincar-com-ibeacons/">http://arquiteturadeinformacao.com/ux-em-espacos-fisicos/tudo-o-que-voce-precisa-saber-para-comecar-a-brincar-com-ibeacons/</a>>. Acesso em: 24. set de 2015.

THJT. **The History of Java Technology.** Disponível em: <a href="http://www.oracle.com/technetwork/java/javase/overview/">http://www.oracle.com/technetwork/java/javase/overview/</a>. Acesso em: 24 set. de 2015.

JSON, JavaScript Object Notation. Disponível em: <a href="http://www.json.org/">http://www.json.org/</a>>. Acesso em: 24 set. de 2015.

ANDROID, **Sistema Operacional Android.** Disponível em: <a href="https://www.android.com/intl/pt-BR\_br/history/">https://www.android.com/intl/pt-BR\_br/history/</a>. Acesso em: 24 set. de 2015.

SILVERMAN, Richard. Git – Guia Prático. Novatec. São Paulo, 2013.

GONÇALVES, Edson. Dominando Netbeans. Ciência Moderna. Rio de Janeiro, 2006.

GLAUBER, Nelson. Dominando o Android - Do Básico ao Avançado. Novatec. São Paulo, 2015.

IBEACON, iBeacon Guide. Disponível em: <a href="http://www.ibeacon.com/">http://www.ibeacon.com/</a>. Acesso em: 24. set de 2015.

BLUETOOTH, **Bluetooth Smart.** Disponível em: <a href="http://www.bluetooth.com/Pages/Bluetooth-Smart.aspx">http://www.bluetooth.com/Pages/Bluetooth-Smart.aspx</a>. Acesso em: 24. set. de 2015.

SPRING, **Introdução ao Spring Framework.** Disponível em: <a href="http://www.devmedia.com.br/">http://www.devmedia.com.br/</a> introducao-ao-spring-framework/>. Acesso em: 24. set. de 2015.

WEB SERVICES, **Introdução a Web Services.** Disponível em: <a href="http://www.devme dia.com.br/artigo-net-magazine-54-introducao-a-web-services/10726">http://www.devme dia.com.br/artigo-net-magazine-54-introducao-a-web-services/10726</a>. Acesso em: 24. set. de 2015.

SEBRAE, **Inovação e Tecnologia: internet das coisas**. Disponível em: <a href="http://www.sebrae.com.br/sites/Portal">http://www.sebrae.com.br/sites/Portal</a> Sebrae/artigos/Inova%C3%A7%C3%A3o-e-Tecnologia:-internet-das-coisas>. Acesso em: 24. set. de 2015.

KONTAKT, **iBeacon Parameters: UUID, Major and Minor.** Disponível em: <a href="https://support.kontakt.io/hc/engb/articles/201620741-iBeacon-Parameters-UUID-Major-and-Minor">https://support.kontakt.io/hc/engb/articles/201620741-iBeacon-Parameters-UUID-Major-and-Minor</a>>. Acesso em: 24. set. de 2015.

BLAGDON, Jeff. **iBeacons could solve indoor mapping.** 2013. Disponível em: <a href="http://www.theverge.com/2013/9/11/4718082/ios-7-ibeacons-could-solve-indoor-mapping-make-shopping-better">http://www.theverge.com/2013/9/11/4718082/ios-7-ibeacons-could-solve-indoor-mapping-make-shopping-better</a>. Acesso em: 24. set. de 2015.

STUDIO, Android. **The official Android IDE.** Disponível em: < https://developer.android.com/sdk/index.html>. Acesso em: 09. set. de 2016.

MENDES, Douglas Rocha. Programação Java com ênfase em orientação a objetos. Novatec. São Paulo, 2009.

XML. Extensible Markup Language. Disponível em: <a href="http://www.w3.org/XML/">http://www.w3.org/XML/</a>. Acesso em: 09 set. de 2016.

JBOSS. **Jboss Aplication Server 7.** Disponível em: <a href="https://docs.jboss.org/author/display/AS7/Documentation">https://docs.jboss.org/author/display/AS7/Documentation</a> Acesso em: 10 set. de 2016.

TRACKR. The TrackR Bravo. Disponível em: <a href="https://www.thetrackr.com/">https://www.thetrackr.com/</a>. Acesso em: 11 nov. de 2015.

TILE. **The Tile App.** Disponível em: <a href="https://www.thetileapp.com/">https://www.thetileapp.com/</a>>. Acesso em: 11 nov. de 2015.

ATLBEACON. **The Open and Intereoperable Proximity Beacon Specification.** Disponível em: <a href="http://altbeacon.org/">http://altbeacon.org/</a> . Acesso em 03 de out. de 2016.