FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA E INOVAÇÃO TECNOLÓGICA.
INSTITUTO DE ENSINO SUPERIOR FUCAPI
COORDENAÇÃO DE GRADUAÇÃO EM
SISTEMAS DE INFORMAÇÃO

PROTÓTIPO DE SISTEMA DE BEACON BASEADO EM REDE WIFI

MARCELO BARBOSA DA ROCHA

MANAUS
2015

MARCELO BARBOSA DA ROCHA

PROTÓTIPO DE SISTEMA DE BEACON BASEADO EM REDE WIFI

Monografia apresentada ao Curso de Graduação em Sistemas de Informação do Instituto de Ensino Superior FUCAPI – CESF, como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação. Área de Concentração: Ciência da Computação

Orientador(a): Carlos Augusto de Araújo Mar, M.Sc.

MANAUS

2015

Deve ser gerada através do preenchimento do Formulário Eletrônico de Elaboração da Ficha Catalográfica, disponível no link: http://www.uece.br/biblioteca/index.php/entrega-de-trabalho.

X000x Sobrenome, Nome do 1º autor. (citado na folha de rosto)
Título principal: subtítulo./Nome completo do 1º autor, Nome completo do 2º autor, Nome completo do 3º autor; orientação [de]. – Local: ano.
Nº de folhas.: il.(se houver ilustração); 30 cm.

Inclui bibliografias: f.(nº da folha em que se encontra)
Trabalho de Conclusão de Curso (Graduação em) – Universidade Estadual do Ceará – (UECE).

1. Assunto. 2. Assunto. 3. Assunto. I. Sobrenome, Nome do 2º autor. II. Sobrenome, Nome do 3º autor. III. Sobrenome, Nome do orientador (orient.). IV. Universidade Estadual do Ceará – UECE. V. Título.

CDU

MARCELO BARBOSA DA ROCHA

PROTÓTIPO DE SISTEMA DE BEACON BASEADO EM REDE WIFI

Monografia apresentada ao Curso de Graduação em Sistemas de Informação do Instituto de Ensino Superior FUCAPI – CESF, como requisito parcial para obtenção do Título de Bacharel em Sistemas de Informação. Área de Concentração: Ciência da Computação

Aprovada em: , por:

Carlos Augusto de Araújo Mar, M.Sc.
(Orientador)

Membro da Banca Dois
Faculdade de Filosofia Dom Aureliano Matos – FAFIDAM
Universidade do Membro da Banca Dois - SIGLA

Membro da Banca Três
Centro de Ciências e Tecnologia - CCT
Universidade do Membro da Banca Três - SIGLA

Membro da Banca Quatro
Centro de Ciências e Tecnologia - CCT
Universidade do Membro da Banca Quatro - SIGLA

MANAUS

2015

À minha família, por sua capacidade de acreditar em mim e investir em mim. Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE CÓDIGOS-FONTE

LISTA DE ABREVIATURAS E SIGLAS

URNA	Universal Real-Time Navigational Assistance
IDE	Ambiente Integrado de Desenvolvimento
UUID	Universally Unique Identifier
API	Application Programming Interface

SUMÁRIO

1 INTRODUÇÃO

A internet das coisas segundo (ASHTON, 2009) refere-se a convergência tecnológica onde as coisas do mundo real, ou seja, objetos usados no nosso dia-a-dia estarão conectados entre si e a rede de mundial de computadores. Nesse novo mundo as "coisas" conectadas passarão a ser mais inteligentes, coletar informações ao seu redor e realizar tarefas automaticamente sem a interferência humana.

Em meio aos vários objetos que aos poucos estão dando forma a internet das coisas estão os beacons. Beacons são pequenos dispositivos que utilizam a tecnologia bluetooth para detectar a presença de outros dispositivos (smartphones, tablets, etc), dentro do seu raio, que também tenham bluetooth ativo e iniciar uma ação nesses mesmos dispositivos através de um aplicativo intermediário previamente instalado.

Os beacons podem agir de forma ativa, permitindo que as aplicações com que eles interagem, possam prover aos usuários informações com base no seu contexto de localização. Ou de forma passiva simplesmente registrando que um determinado dispositivo entrou em seu raio de detecção, esses registros podem ser trabalhados por outros sistemas para obter a localização aproximada do usuário em relação ao beacon.

A aplicabilidade dos beacons são diversas. Podem ser utilizados para mapeamento de fluxo de usuários em shoppings, feiras e eventos diversos; localização em ambientes fechados (indoor); ou em uma loja de varejo, enviar uma notificação com informação de ofertas de produtos para os smartphones de clientes quando eles passarem por uma determinada área da loja. Também existem outras possibilidades, como: checkins automáticos e até pagamentos através de dispositivos móveis.

1.1 PROBLEMA

Segundo (TEIXEIRA, 2014) beacons não são inteligentes, toda a inteligência fica sob responsabilidade do aplicativo instalado no dispositivo que irá interagir com o beacon. Atualmente no mercado existem dois padrões principais para comunicação entre beacons e aplicações, o iBeacon criando e mantido pela Apple, é um padrão proprietário que funciona somente com os dispositivos da própria fabricante e o Eddystone criando e mantido pela Google, é um padrão aberto (open-source) que pode ser utilizado por qualquer dispositivo. Em ambos os padrões o beacon age conforme a afirmação anterior, a total dependência de uma aplicação

pode representar um problema em alguns casos, como por exemplo no cenário em que uma loja de varejo deseja utilizar beacons para prover informações para os seus clientes com base nos dados de seus próprios perfis e também histórico das últimas compras realizadas, que estão armazenados em seus servidores de banco de dados, mas não deseja disponibilizar por razões de segurança, um serviço público que exponha os dados dos clientes diretamente para um aplicativo.

Neste último cenário seria mais seguro que os dados podessem ser tratados, caso fosse necessário, pelo próprio beacon antes de serem enviados para a aplicação instalada no dispositivo do cliente que está interagindo com ele. Além disso, o acesso do serviço sendo realizado através do beacon representa mais segurança para a rede interna da loja, pois neste cenário o beacon é um dispositivo que pode ser controlado pelo próprio administrador da rede.

1.2 OBJETIVOS

Neste tópico serão abordados os objetivos geral e específicos desta monografia.

1.2.1 Objetivo Geral

Desenvolver um protótipo de sistema de beacon baseado na tecnologia wifi para aumentar a taxa de transferência de dados e a segurança de aplicações baseadas em beacons bluetooth.

1.2.2 Objetivos Específicos

- Pesquisar e definir as tecnologias, ferramentas e suas configurações para criar o beacon wifi;
- Implementar o algoritmo do protocolo de comunicação para o beacon e os dispositivos;
- Desenvolver o aplicativo cliente para comunicação;
- Desenvolver um serviço para prover as informações para o sistema utilizando o beacon e o aplicativo cliente;
- Validar o protótipo quanto ao aumento da taxa de transferência de dados e segurança.

1.3 JUSTIFICATIVA

Seguindo o propósito da internet das coisas, em disponibilizar dispositivos cada vez mais inteligentes que possam executar tarefas automaticamente sem a intervenção humana, este

trabalho tem o intuito de desenvolver um beacon capaz não somente de detectar a presença dos dispositivos que entrarem em seu raio de ação, mas também um beacon mais inteligente que os encontrados atualmente no mercado, podendo processar dados internamente e também comunicar diretamente com serviços internos ou externos.

1.4 TRABALHOS RELACIONADOS

(CHOI; PARK; LEE, 2015) descreveram um sistema de gerenciamento de energia de escritório capaz de reduzir o consumo de energia de computadores, monitores e luzes de um escritório com a utilização de beacons baseados em bluetooth e um aplicativo móvel. Enquanto vários beacons foram colocados em lugares estratégicos do escritório de forma a cobrir todos os espaços, um aplicativo móvel determina, com a ajuda dos beacons, se o usuário entrou ou saiu do escritório e modificam o modo de economia de energia dos computadores, monitores e luzes. O sistema proposto pode reduzir o consumo de energia em um escritório sem causar desconforto aos usuários. No caso da utilização de beacons wifi, com um único dispositivo, seria possível cobrir toda uma sala do escritório.

(BOHONOS *et al.*, 2007) descreveram um sistema de software e hardware chamado URNA que permite a comunicação de informações relevantes de localização para uma pessoa cega através de um celular com bluetooth. O sistema tem como alvo principal ajudar um pedestre cego na travessia de ruas de um cruzamento, informando o nome da rua e o estado atual do semáforo. Com o uso de beacon wifi, visto que a tecnologia bluetooth tem um curto alcance, o sistema poderia ser estendido com a criação de um aplicativo para os motoristas, cuja a funcionalidade seria comunicar com o beacon e informá-los sobre a presença de pedestres cegos a medida que o mesmo se aproximasse de um semáforo.

(CHAWATHE, 2008) descreve um método de localização de dispositivos móveis em ambientes fechados usando beacons baseados em bluetooth. Nesse caso o raio de detecção limitado da tecnologia bluetooth é usado como vantagem. Mas para isso é preciso estabelecer de forma correta a posição de cada beacon, de modo que os beacons não sofram com interferência de raio de outro beacon.

1.5 MÉTODO DE INVESTIGAÇÃO

A metodologia de desenvolvimento proposta para este trabalho é composta das seguintes etapas:

A pesquisa de referencial teórico sobre beacons será baseada nas principais fontes encontradas no mercado e/ou disponibilizado na internet, como livros, artigos, tutoriais e revistas. Com intuito de fornecer ao trabalho um embasamento teórico necessário para o entendimento dos beacons, os conceitos envolvidos e suas características.

A pesquisa de tecnologias e ferramentas necessárias para definição da arquitetura e construção de um beacon baseado na tecnologia wifi utilizando o microcomputador Raspberry Pi, juntamente com seu sistema operacional raspbian baseado na distribuição linux debian. Bem como a forma de detecção dos dispositivos cliente; e também o protocolo de comunicação a ser utilizado entre o beacon e esses dispositivos.

Na etapa de desenvolvimento serão implementados: o protótipo do aplicativo cliente que fará uso das principais ferramentas encontradas para o desenvolvimento android, sendo o Android Studio como IDE e o Framework Android; e o serviço responsável por implementar o protocolo de comunicação definido na etapa anterior. Como objetivo desta etapa, o aplicativo deverá ser capaz de entender e utilizar o serviço de protocolo de comunicação com o beacon para o recebimento de suas mensagens em forma de notificação.

Na etapa de validação da solução ocorrerá a comparação dos resultados obtidos a respeito da capacidade máxima do raio de detecção dos dispositivos clientes, a rapidez na transferência dos dados aos dispositivos e o nível de segurança proporcionado na utilização da tecnologia wifi no beacon em relação a tecnologia bluetooth.

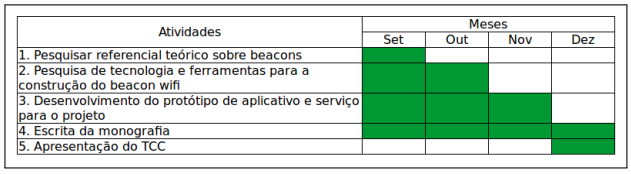
A etapa de escrita a monografia acontecerá durante todo o período de execução deste trabalho.

1.6 ESTRUTURAÇÃO DA MONOGRAFIA

XXXXXXXXXXXXXXXXXXXXXXXXX

1.7 CRONOGRAMA

Figura 1 – Cronograma do trabalho



Atividades	Meses			
	Set	Out	Nov	Dez
1. Pesquisar referencial teórico sobre beacons				
2. Pesquisa de tecnologia e ferramentas para a construção do beacon wifi				
3. Desenvolvimento do protótipo de aplicativo e serviço para o projeto				
4. Escrita da monografia				
5. Apresentação do TCC				

Fonte: Elaborado pelo autor

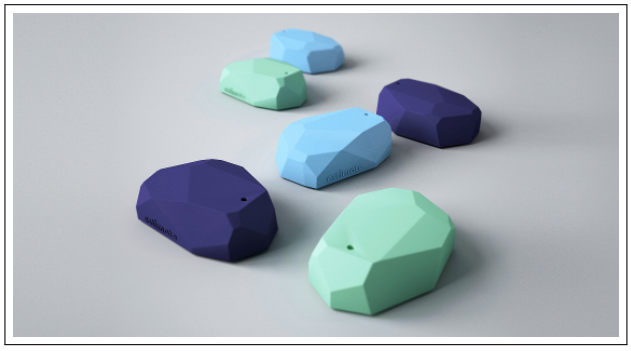
2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem por finalidade apresentar os principais conceitos necessários para o desenvolvimento do presente trabalho. Serão abordados

2.1 BEACONS

(DANOVA, 2014) conceitua que beacons são dispositivos de hardware de baixo custo que são pequenos o suficiente para serem fixados em uma parede ou um balcão. Eles utilizam bateria de forma eficiente e conexão bluetooth low energy (de baixo consumo) para enviar mensagens ou notificações para smartphones ou tablets.

Figura 2 – Beacons



Fonte: Estimote

Os beacons são considerados dispositivos integrantes da próxima geração da internet, chamada de internet das coisas, onde terão um papel fundamental na forma de comunicação entre as mais variadas instituições como: lojas de varejo, locais de eventos, supermercados, restaurantes e instituições de ensino; e as pessoas.

Entre os padrões de beacons existentes atualmente no mercado, dois se destacam por serem desenvolvidos pelas duas empresas – Apple e Google – mais importantes e inovadoras da área tecnológica. A seguir, os padrões serão explicados com detalhes.

2.1.1 iBeacon

> O iBeacon é uma nova tecnologia que estende os Serviços de Localização no iOS. Seu dispositivo iOS pode alertar apps quando você se aproxima ou sai de um local com um iBeacon. Além de monitorar um local, o app pode estimar sua proximidade

> a um iBeacon (por exemplo, uma vitrine ou caixa em uma loja). Em vez de usar latitude e longitude para definir o local, o iBeacon usa um sinal de baixa energia de Bluetooth, detectado pelos dispositivos iOS. (APPLE INC., 2015)

Esta tecnologia foi introduzida pela Apple a partir do seu sistema iOS versão 7, através dela as aplicações podem reagir aos sinal de beacons próximos ao dispositivo do usuário. Para isso o iBeacon envia pequenos pacotes de dados contendo o seu ID e a força de sinal. Como podemos ver abaixo:

Quadro 1 – Informações presentes em um pacote de sinal em iBeacons

Campo	Tamanho	Descrição
UUID	16 bytes	Número identificador de um conjunto de beacons
Major	2 bytes	Usado para identificar um subconjunto de beacons dentro do conjunto de beacons
Minor	2 bytes	Usado para identificar individualmente um beacon dentro de um subconjunto

Fonte: Elaborado pelo autor

Os valores contidos no pacote são utilizados de maneira hierárquica pelo sistema iOS para determinar o beacon que o usuário está próximo.

Quadro 2 – Hierarquização de informações em iBeacons

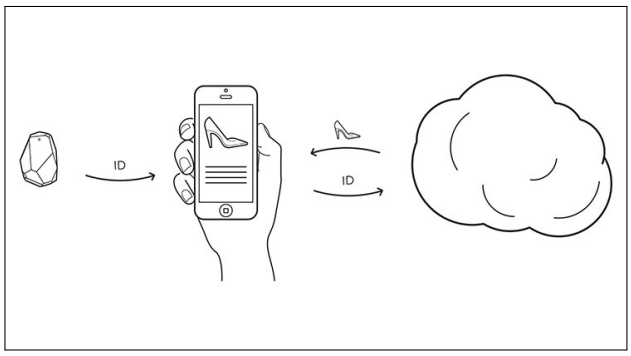
Localização		Manaus	São Paulo	Rio de Janeiro
UUID		AAAAAAAA-BBBB-CCCC-DDDD-EEEEEEEEEEEE		
Major		1	2	3
Minor	Bebidas	10	10	10
	Higiene	20	20	20
	Limpeza	30	30	30

Fonte: Elaborado pelo autor

No quadro acima, está exemplificado o caso envolvendo uma empresa com filiais em três cidades: Manaus, São Paulo e Rio de Janeiro. As três filiais compartilham o mesmo UUID que identifica de forma única a empresa. A identificação de cada filial fica sob responsabilidade do Major; no exemplo 1 para Manaus, 2 para São Paulo e 3 para Rio de Janeiro; E a identificação de cada setor dentro das filiais fica sob responsabilidade do Minor, no exemplo 10 para Bebidas, 20 para Higiene e 30 para Limpeza.

Segundo (APPLEINSIDER STAFF, 2013) iBeacons estão presentes em todas as lojas oficiais da Apple nos Estados Unidos. Através da utilização da aplicação oficial da loja e os iBeacons, os clientes podem ter acesso a uma camada extra de informações e serviços disponíveis nas lojas.

Figura 3 – Dinâmica do padrão iBeacon



Fonte: Estimote

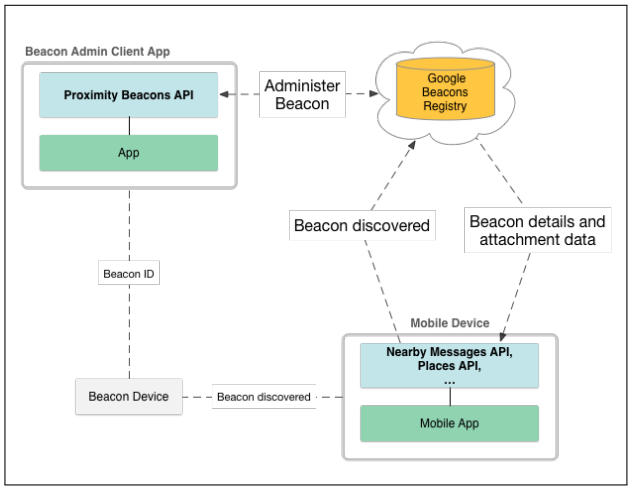
2.1.2 Eddystone

Eddystone é um formato de beacon aberto do Google e que funciona com os sistemas Android e iOS. Eddystone inclui três tipos de estrutura para transmissão de dados, adequados para diferentes cenários. (GOOGLE DEVELOPERS, 2015)

O padrão Eddystone é uma parte integrante da plataforma de beacons da Google. Com sua utilização é possível:

- Permitir que as aplicações reajam ao contexto do usuário através de anexos (attachments) beacons;
- Monitorar o status de uma frota de beacons a estrutura de telemetria do padrão Eddystone;
- Transmitir dados para utilização da Web Física.

Figura 4 – Dinâmica do padrão Eddystone



Fonte: Google Developers

A plataforma de beacons do Google é composta pelos próprios beacons, pelo padrão

Eddystone e pela API de proximidade de beacons.

Os beacons que suportam as especificações do padrão Eddystone podem transmitir dados de um único tipo de estrutura ou intercalar entre uma estrutura e outra. Por exemplo, transmitir repetidamente 50 estruturas do tipo Eddystone-UID seguidas de uma estrutura Eddystone-TLM.

O padrão Eddystone especifica atualmente três tipos de estruturas: Eddystone-UID, Eddystone-URL e Eddystone-TLM.

2.1.2.1 Eddystone-UID

Eddystone-UID contém o número identificador do beacon, que pode ser utilizado por um aplicação para disparar uma ação para o usuário. No Eddystone-UID, assim como ocorre no iBeacon, também é enviado um pequeno pacote de dados contendo um namespace e instance, detalhes no quadro abaixo:

Quadro 3 – Informações presentes em um pacote de sinal em Eddystone

Campo	Tamanho	Descrição
Namespace	10 bytes	Identificador de um conjunto de beacons
Instance	6 bytes	Usado para identificar individualmente um beacon

Fonte: Elaborado pelo autor

Para gerar um namespace a especificação do padrão Eddystone recomenda utilizar os dez primeiros bytes do hash SHA-1 gerado a partir de um nome de domínio. Exemplo de namespace gerado a partir do domínio “fucapi.br” utilizando a linguagem python:

Figura 5 – Código de exemplo para geração de namespace em python

```python
import hashlib
import binascii

# Converte nome de domínio em bytes
domain_bytes = str.encode('fucapi.br')

# Gera o hash SHA-1
hash_sha1 = hashlib.sha1(domain_bytes)

# Devolve o hash em hexadecimal
hash_hex = hash_sha1.hexdigest()

# Converte o hexadecimal em bytes pegando os 10 primeiros
bytes10 =  bytearray.fromhex(hash_hex)[0:10]

# Converte os bytes em hexadecimal
namespace = binascii.hexlify(bytearray(bytes10))

print(namespace)
```

Fonte: Elaborado pelo autor

Ao executar o código a saída é:

1755ba6780003245d85c

2.1.2.2 Eddystone-URL

Eddystone-URL é a base para um novo conceito criado pela Google, chamado de Web Física, onde o usuário não precisará de um aplicativo dedicado para interpretar e executar as ações ao utilizar beacons fixados em objetos do mundo real. O conteúdo transmitido pelos beacons segue o mesmo padrão das URLs interpretadas pelos navegadores web, assim o usuário poderá acessar o conteúdo – em forma de webapp ou website – sem precisar baixar e instalar um aplicativo dedicado.

2.1.2.3 Eddystone-TLM

Eddystone-TLM especifica uma estrutura apropriada para transmissão de dados sobre os próprios beacons, permitindo que os mesmos possam ser gerenciados. Essa estrutura pode ser enviada juntamente com Eddystone-UID ou Eddystone-URL. A aplicação utilizada pelo usuário deve ser preparada para retransmitir os dados enviados pelo Eddystone-TLM para um serviço externo responsável por prover dados de gerenciamento.

A estrutura pode conter os seguintes dados sobre os beacons:

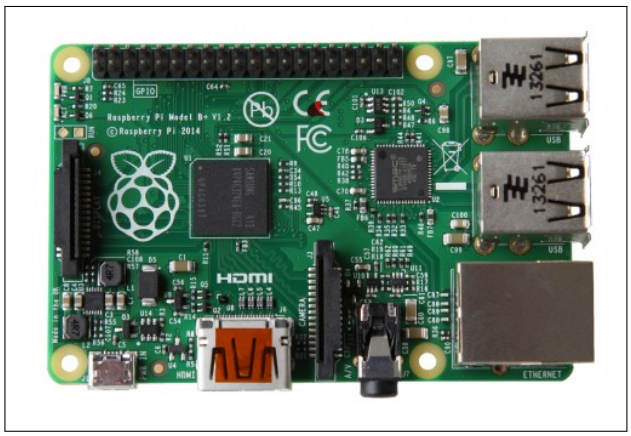
- Tensão da bateria, que pode ser usado para estimar o nível de bateria restante em um beacon;
- Temperatura do beacon;
- Número de pacotes enviados desde a última vez que o beacon foi ligado ou reiniciado;
- Tempo de atividade desde a última vez que o beacon foi ligado ou reiniciado.

2.2 RASPBERRY PI

De acordo com a (RASPBERRY PI FOUNDATION, 2015) Raspberry Pi é um microcomputador de dimensões mínimas, semelhante ao tamanho de um cartão de crédito, que pode ser conectado a um monitor e usa teclado e mouse padrão.

Apesar de ser um microcomputador, é possível realizar a maioria das atividades da mesma forma que são realizadas em um computador pessoal convencional. Como navegar na

Figura 6 – Raspberry Pi



Fonte: Raspberry Pi Foundation

internet, ouvir arquivos de audio e visualizar videos, criar documentos de textos, planilhas e apresentações. Raspberry Pi é desenvolvido pela Raspberry Pi Foundation, instituição que tem como principal objetivo, promover o estudo da ciência da computação e assuntos relacionados.

Atualmente, em sua mais nova versão – Raspberry Pi 2, é possível utilizar tanto sistema Linux, quanto Windows. Oficialmente é fornecida uma distribuição baseada no linux Debian chamada Raspbian, no caso do Windows a própria Microsoft disponibiliza uma versão adaptada do Windows 10, chamada Windows 10 IoT (Internet of Things).

Além das dimensões mínimas, suas principais características estão relacionadas ao processador, armazenamento e porta GPIO (General Purpose Input/Output). O processador é baseado na arquitetura ARM, a mesma presente nos processadores de smartphones e tablets, o que proporciona baixo consumo de energia e também baixo aquecimento ao ponto de não necessitar de um dissipador de calor. O armazenamento de dados, inclusive do sistema operacional, é feito através do uso de um cartão SD, essa forma é muito vantajosa para realização de backup do sistema e também trocar ou alternar rapidamente o sistema utilizado no microcomputador. A porta GPIO é composta por um conjunto de pinos programáveis que são responsáveis pela comunicação de entrada e saída de sinais digitais, através dela é possível fazer comunicação com equipamentos externos ou periféricos. Como por exemplo LEDs, motores, sensores, entre outros.

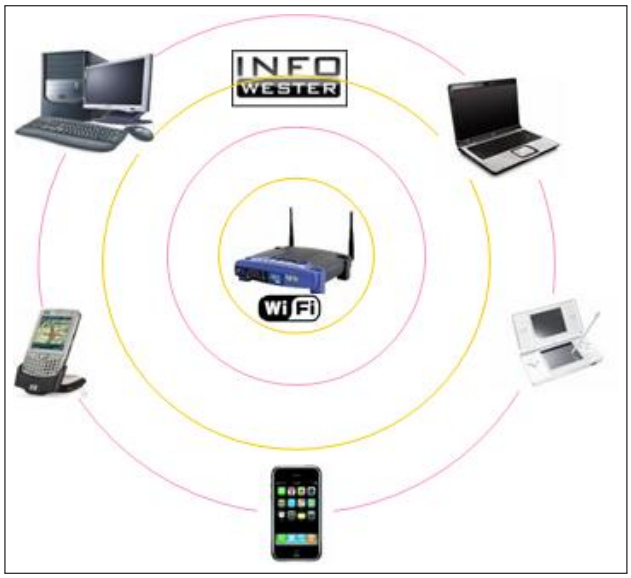
2.3 WIFI

Segundo (ALECRIM, 2013) WiFi é um conjunto de especificações para redes locais sem fio (WLAN - Wireless Local Area Network) baseada no padrão IEEE 802.11. O termo WiFi

é um acrônimo de Wireless Fidelity, a Wifi Alliance é a instituição responsável por manter as especificações e certificar os produtos que utilizam esta tecnologia.

A flexibilidade desta tecnologia é um fator de forte aderência para o mercado, pois através dela é possível conectar, formando uma rede, os mais variados tipos de dispositivos, como: smartphones, tablets, computadores, impressoras, TVs, Consoles de Jogos, Câmeras IP, entre outros. O uso mais comum da tecnologia WiFi é através do uso de roteadores sem fio para acesso à internet nos mais variados tipos de lugares, como: hotéis, restaurantes, aeroportos, bares, shoppings, universidades, praças públicas, entre outros.

Figura 7 – Rede WiFi



Fonte: Infowester

Como mais de uma rede WiFi podem existir no mesmo lugar ou em lugares próximos, é preciso que as mesmas sejam identificadas com um SSID (Service Set Identifier) que consiste em um conjunto de até 32 caracteres. Popularmente é conhecido como “nome” da rede WiFi.

Ao longo do tempo a Wifi Alliance tem mantido o padrão 802.11 em constante evolução, a cada nova versão é atribuída uma letra para efeito de representação. As diferenças básicas entre um padrão e outro estão relacionadas a três características principais: alcance de sinal, frequência de operação e velocidade de transmissão de dados.

Quadro 4 – Versões do padrão 802.11

Versão	Alcance	Frequência	Velocidade
a	35m	5 GHz	54 Mbps
b	35m	2.4 GHz	11 Mbps
g	38m	2.4 GHz	54 Mbps
n	70m	2.4/5 GHz	74 Mbps

Fonte: Elaborado pelo autor

2.4 MQTT

> MQTT é um protocolo de mensagens leve baseado em publicação/assinatura. Um ativador crítico de Internet of Things (IoT), o MQTT também pode ser usado para sistemas de mensagens corporativos confiáveis para dispositivos móveis, permitindo comunicações seguras, confiáveis, para a próxima geração de aplicativos móveis resilientes. (MAYNARD, 2015)

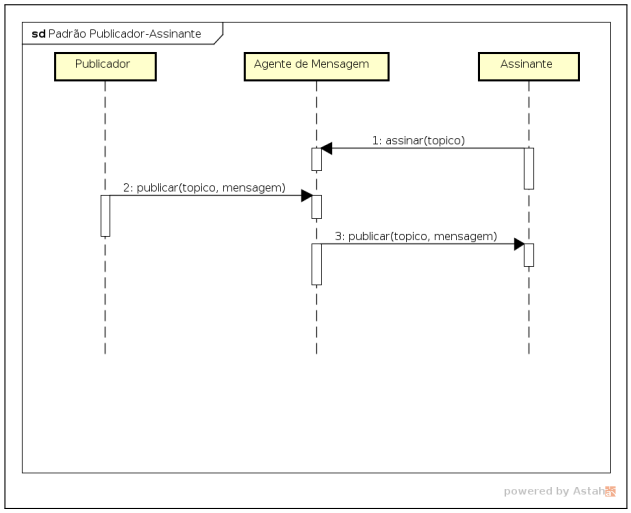
De acordo com (MQTT.ORG, 2015) MQTT é um protocolo para conexão máquina a máquina ou para internet das coisas. Ideal para aplicações móveis devido ao seu tamanho pequeno, baixo consumo de energia, pacotes de dados minimizados e distribuição eficiente de informações para um ou mais receptores. Ele foi projetado para transportar mensagens com base no padrão publicador/assinante, ou seja, para o envio de uma mensagem existe o papel de um publicador e para o recebimento de uma mensagem existe um ou mais assinantes. Este padrão promove o desacoplamento entre as partes, pois publicador e assinante não conhecem a existência um do outro. Para isso é necessário um terceiro componente conhecido por ambos, chamado agente de mensagens, responsável por receber as mensagens referentes a um tópico do publicador e enviá-las aos assinantes que assinam o mesmo tópico de mensagem. O padrão publicador/assinante é uma alternativa ao tradicional padrão cliente-servidor, onde o cliente conhece e se comunica diretamente com o servidor.

Suas principais características são:

- Especificação aberta e de fácil adoção;
- Provê conectividade para dispositivos inteligentes através de mensagens;
- Oferece opções de conectividade otimizados para sensores e dispositivos remotos;
- Voltado para dispositivos com pouca memória e baixo poder de processamento;
- Ideal para redes limitadas com pouca largura de banda, alta latência e conexão instável;
- Promove escalabilidade na implantação e gerenciamento de soluções.

Abaixo está uma simples representação da interação entre os atores envolvidos no padrão publicador-assinante.

Figura 8 – Padrão Publicador-Assinante



Fonte: Elaborado pelo Autor

2.5 ACTIVEMQ

Segundo (POSTA, 2011) ActiveMQ é um serviço de mensageria de código aberto que pode servir como espinha dorsal para uma arquitetura de aplicações distribuídas baseadas em mensagens.

Mensageria não representa somente uma forma de comunicação entre aplicações, mas também uma forma de integração entre elas. ActiveMQ implementa a Java Message Service (JMS) API, parte integrante da especificação J2EE, para permitir a criação, envio, recebimento e leitura de mensagens. Oferecendo uma solução de código aberto para comunicação assíncrona e de baixo acoplamento, provendo a integração das mais diferentes plataformas através da orientação a mensagens.

ActiveMQ provê as seguintes características:

- Conformidade com JMS – Como já citado anteriormente, ActiveMQ é uma implementação da JMS API. O seu uso promove importantes benefícios, incluindo envio assíncrono de mensagem, modos de entrega persistente e não persistente, e durabilidade de mensagem para assinantes;
- Conectividade – ActiveMQ possibilita conexão com uma ampla variedade de protocolos, incluindo HTTP/S, SSL, MQTT, STOMP, TCP, UDP e muitos outros;

- Persistência e Segurança – ActiveMQ permite a configuração de suas opções para persistência e segurança. Por exemplo, a persistência de dados pode realizada via KahaDB, mas também pode ser modificada para realizar via JDBC. A autenticação pode utilizar o modo padrão via arquivo de propriedade ou o padrão JAAS;
- Desenvolvimento de aplicações usando Java – O modo mais comum de desenvolver aplicações de envio e recebimento de mensagem utilizando ActiveMQ é utilizando Java;
- Integração com servidores de aplicação – É possível integrar ActiveMQ com vários servidores de aplicação, como: Apache Tomcat, Tomee, Geronimo e JBoss;
- APIs Cliente – Apesar de ActiveMQ implementar uma especificação da plataforma Java, através do uso de APIs cliente é possível desenvolver aplicações que utilizem ActiveMQ utilizando várias outras linguagens, como: C, C++, C#, Perl, PHP, Python, Ruby e outras;
- Agrupamento de servidores – É possível fazer que vários servidores rodando ActiveMQ possam trabalhar em conjunto formando um agrupamento de servidores. Essa opção geralmente é utilizada por motivos de escalabilidade;
- Administração simplificada – Apesar de ActiveMQ ser destinado a desenvolvedores, um profissional de TI sem conhecimentos de programação pode facilmente utilizá-lo. Seu monitoramento por ser feito não somente através de seu console web, mas também utilizando JMX ou até mesmo via linhas de comando.

2.6 SERVLET

Segundo a (ORACLE AND/OR ITS AFFILIATES, 2010) servlet é uma classe da linguagem de programação Java que é utilizada para ampliar os recursos de servidores que hospedam aplicações acessadas por meio do modelo de programação solicitação-resposta.

Embora servlet possa responder a qualquer tipo de solicitação, é muito comum a sua utilização no desenvolvimento de aplicações hospedadas em servidores web. Nesse caso, servlet define classes específicas para trabalhar com o protocolo HTTP.

A classe HttpServlet define os métodos doGet e doPost para tratar as requisições para este protocolo. Sendo doGet recomendado para requisições que não alterem o estado da aplicação, como por exemplo uma consulta e o método doPost recomendado para enviar dados a serem processados, como ocorre em formulários HTML. Exemplo abaixo:

Código-fonte 1 – Exemplo do método doGet

```
1  import java.io.IOException;
2  import java.io.PrintWriter;
3  import javax.servlet.ServletException;
4  import javax.servlet.http.HttpServlet;
5  import javax.servlet.http.HttpServletRequest;
6  import javax.servlet.http.HttpServletResponse;
7
8  public class HelloWorld extends HttpServlet {
9    public void doGet(HttpServletRequest request,
       HttpServletResponse response) throws ServletException,
        IOException {
10     PrintWriter out = response.getWriter();
11     out.println("<html>\n" +
12       "<head><title>Hello World</title></head>\n" +
13       "<body>\n" +
14       "<h1>Hello World</h1>\n" +
15       "</body></html>");
16   }
17 }
```

As classes HttpServletRequest e HttpServletResponse são utilizadas para acessar as informações de requisição e resposta respectivamente. Como no trecho de código abaixo:

Código-fonte 2 – Exemplo de uso HttpServletRequest

```
1  protected void doPost(HttpServletRequest request,
     HttpServletResponse response) throws ServletException,
     IOException {
2    String nome = request.getParameter("nome");
3    String sobrenome = request.getParameter("sobrenome");
       response.setContentType("text/html");
4    PrintWriter out = response.getWriter(); out.println("Bem
```

```
    Vindo<h3>"+nome+" " +sobrenome+"</h3>");
  out.close();
}
```

O código fonte abaixo é utilizado para acessar o dispositivo de saída para a resposta:

Código-fonte 3 – Acessando dispositivo de saída

```
PrintWriter out = response.getWriter();
```

2.7 KISMET

Segundo (KERSHAW, 2011) é um analisador de rede, sniffer, e um sistema de detecção de intrusão. Kismet trabalha em conjunto com placas de rede que suportam o modo de monitoramento de rede, também chamado de modo promíscuo, que suportem os padrões 802.11a, 802.11b, 802.11g e 802.11n. Kismet é diferente das outras ferramentas de detecção porque trabalha em modo passivo, ou seja, ao invés de transmitir dados, somente escuta as transmissões. Quando o modo de monitoramento é ativado, a placa de rede fica indisponível para outras finalidades.

Suas principais características são:

- Trabalha em modo passivo coletando pacotes e detectando redes, inclusive redes com SSID oculto;
- Pode descobrir o canal de rede mais congestionado;
- Permite integração com GPS;
- O arquivo de captura de pacotes é compatível com outras ferramentas de análise de rede como Wireshark e Aircrack-ng;

Kismet pode ser divido em trê partes: Drone, Servidor e Cliente. Kismet Drone pode ser utilizado para escutar pacotes de rede remotamente e enviá-los para um Kismet Servidor analisar os pacotes. Kismet Servidor pode trabalhar em conjunto com Kismet Drone ou sozinho, sua função é analisar os pacotes de rede. Kismet Cliente se comunica com Kismet Servidor para exibir as informações de redes e pacotes coletados.

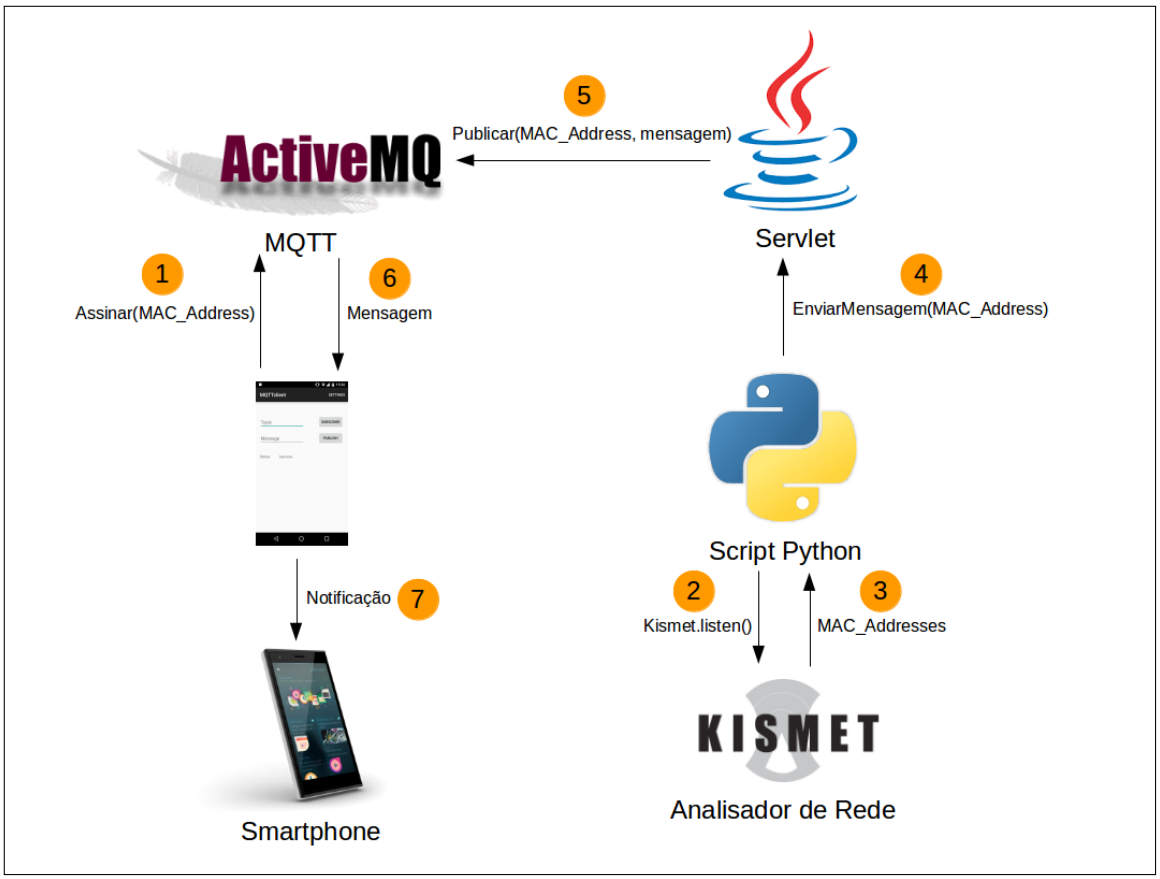
3 SOLUÇÃO

Este capítulo tem por finalidade apresentar a integração entre as tecnologias e ferramentas que fazem parte da solução proposta por este trabalho.

3.1 VISÃO GERAL DA SOLUÇÃO

A solução proposta neste trabalho contempla a utilização de beacon wifi para que os usuários de dispositivos inteligentes, como smartphones, possam ser receber informações em forma de notificação. A figura abaixo demonstra o fluxo e os componentes envolvidos na solução:

Figura 9 – Visão Geral



Fonte: Elaborado pelo autor

O beacon wifi é construído a partir do microcomputador Raspberry Pi contendo o sistema operacional Raspbian instalado juntamente com Kismet, python, TomEE, ActiveMQ e mais duas antenas wifi de conexão usb. Uma antena para detecção dos dispositivos e outra para conexão dos mesmos dispositivos via wifi com o beacon. A conexão via wifi com o beacon

poderá ser utilizada tanto para comunicação da aplicação com o agente de mensagens, quanto para compartilhamento de sinal de internet usando um cabo conectado a porta ethernet do Raspberry Pi.

Para que ocorra a detecção dos dispositivos, nesse caso o smartphone, é necessário que os mesmos estejam com o wifi ligado. A detecção fica sob responsabilidade do Kismet, que através da antena wifi em modo de monitoramento capturam os dispositivos com wifi ligado dentro do seu raio de alcance.

Uma vez que o dispositivo é detectado, para que as notificações sejam enviadas aos dispositivos, seguindo o protocolo MQTT, é necessária a assinatura de algum tópico de mensagem, o que na solução proposta acontece de forma implícita através da aplicação, que assina um tópico representando o MAC address do dispositivo. Desta forma, é possível identificá-lo de forma única, assim as mensagens poderão conter informações com conteúdo baseado no perfil do usuário do dispositivo. Mas nada impede que o usuário possa também assinar outros tópicos de seu interesse, assim receberá informações categorizadas. Como por exemplo, em uma loja de varejo o usuário receber notificações contendo informação de novos produtos, ofertas ou promoções.

O script feito com a linguagem de programação python é responsável por coletar os dados de detecção gerados pelo Kismet através do método kismet.listen(), a medida que os dados são coletados o MAC address é filtrado para em seguida ser enviado pelo script para o servlet que contém o serviço da solução.

O serviço, que fica hospedado no servidor de aplicações TomEE, tem como função principal passar as informações a serem enviadas pelo agente de mensagens. Ao receber o MAC address de um dispositivo detectado, o serviço poderá acessar outras fontes de dados, como um banco de dados ou webservice para extrair dados com base na identificação do usuário através do MAC address e formatar mensagens personalizadas para o usuário. Citando novamente o exemplo do uso de beacon em uma loja de varejo, o usuário poderá receber uma notificação contendo recomendações de produtos com base nos últimos produtos comprados por ele.

O agente de mensagens utilizado pela solução proposta é o ActiveMQ, configurado para trabalhar internamente usando o protocolo MQTT. É ele de fato quem conhece e entrega as mensagens aos dispositivos. Todas as interações tanto de assinatura, quanto de publicação serão realizadas através do ActiveMQ. Também é possível que outras aplicações de uso administrativo sejam integradas para realizar o gerenciamento dos tópicos de mensagens, assinantes ou até

envio direto de mensagem sem intermédio de um serviço.

O aplicativo instalado no smartphone do usuário é o responsável por gerar a notificação assim que uma mensagem enviada pelo ActiveMQ é entregue. Mas para que isso ocorra, na aplicação existe um serviço implementado que é iniciado no momento que o dispositivo é ligado, este serviço é gerenciado pelo próprio sistema operacional do smartphone, ficando ativo enquanto o dispositivo permanecer ligado, o que garante que a notificação seja disparada mesmo que o usuário não esteja com a aplicação aberta.

3.2 INSTALANDO RASPBIAN

O sistema operacional Raspbian é instalado em um cartão SD comum, sendo recomendado a utilização de cartões com no mínimo 8GB de espaço. O arquivo (zip) de imagem oficial pode ser obtido através do endereço:

<https://www.raspberrypi.org/downloads/raspbian/>

Após baixar o arquivo 2015-09-24-raspbian-jessie.zip (versão atual), o mesmo deve ser descompactado. Para isso o comando abaixo deve ser executado:

```
1 $ unzip Downloads/2015-09-24-raspbian-jessie.zip
```

O arquivo resultante será 2015-09-24-raspbian-jessie.img. Para gravar a imagem no cartão SD, o comando abaixo deve ser executado:

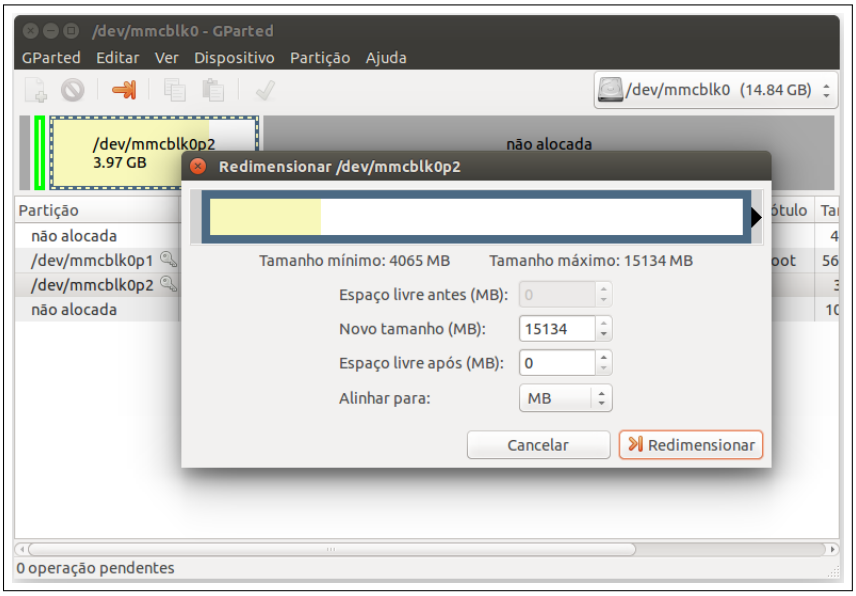
```
1 $ dd if=/caminho/ate/imagem of=/dev/sdx
```

Onde /caminho/ate/imagem representa literalmente o caminho até onde está a imagem que foi descompactada no comando anterior e /dev/sdx é a particão que representa o cartão SD no sistema operacional.

Por fim, é necessário expandir a partição reservada aos arquivos do sistema Raspbian (ext4) para ocupar todo o espaço restante disponível no cartão SD. Para isso é recomendada a utilização do programa GParted, selecionando a opção de redimensionamento e em seguida

alterar o valor do campo “Novo tamanho (MB)” para o máximo valor disponível em megabytes. Conforme imagem abaixo:

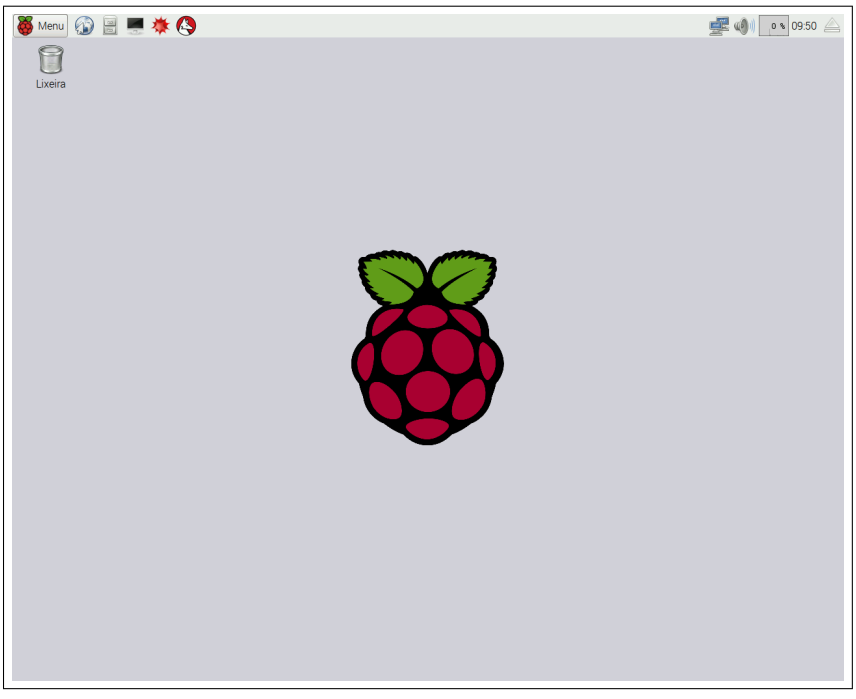
Figura 10 – Expandindo partição com GParted



Fonte: Elaborado pelo autor

Após expandir a partição no cartão SD, o sistema estará pronto para ser iniciado.

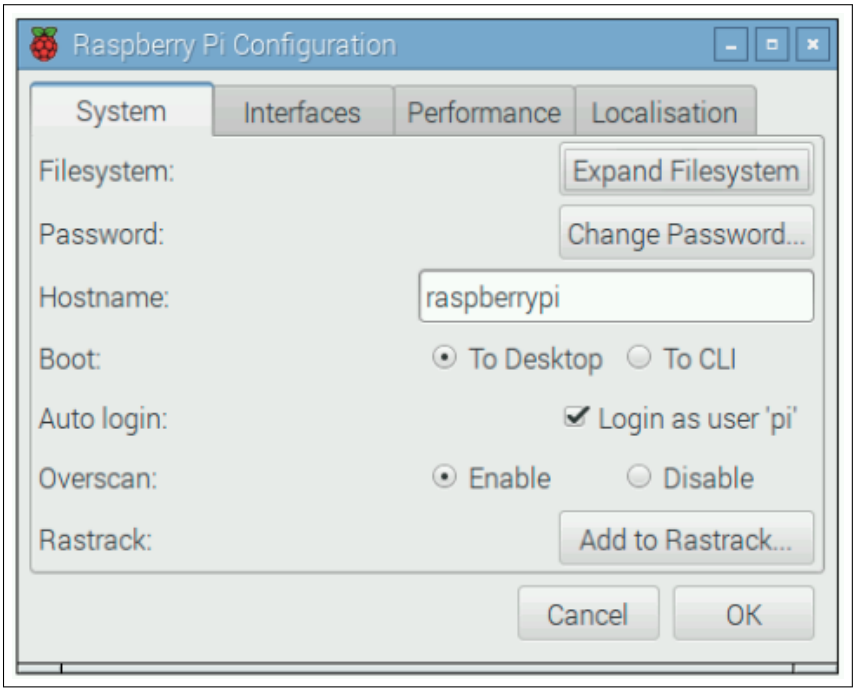
Figura 11 – Área de trabalho do Raspbian



Fonte: Elaborado pelo autor

Para completar a expansão, a opção “Expand Filesystem” no programa de configuração do sistema deve ser utilizada. O programa de configuração está disponível em: Menu»Preferências»Raspberry Pi Configuration. Conforme figura abaixo:

Figura 12 – Programa de configuração do Raspbian



Fonte: Elaborado pelo autor

3.3 CRIANDO ACCESS POINT

Esta seção apresenta todos as ferramentas e configurações necessárias para fazer com que o Raspberry Pi torne-se um ponto de acesso wifi.

O comando abaixo atualiza a lista de pacotes do gerenciador apt-get:

```
1 $ sudo apt-get update
```

Após a lista de pacotes ser atualizada, as ferramentas hostapd e isc-dhcp-server poderão ser instaladas executando o comando abaixo. São elas que, de fato, transformam o Raspberry Pi em um ponto de acesso wifi.

```
1 $ sudo apt-get install hostapd isc-dhcp-server
```

O hostapd é a ferramenta que implementa as funções de um ponto de acesso no linux e o isc-dhcp-server é o serviço que irá prover os endereços IPs aos dispositivos que conectarem no ponto de acesso.

3.3.1 Configurando o ISC DHCP Server

O primeiro arquivo a ser configurado é o dhcpd.conf através do seguinte comando:

```
$  sudo nano /etc/dhcp/dhcpd.conf
```

No comando acima está sendo utilizado o editor nano, mas outro editor pode ser utilizado.

Após o arquivo ser aberto no editor, as seguintes linhas devem ser localizadas:

```
option domain-name "example.org";
option domain-name-servers ns1.example.org, ns2.example.org
   ;
```

Após localizar, o carácter # deve ser colocado no início das duas linhas. Como abaixo:

```
#option domain-name "example.org";
#option domain-name-servers ns1.example.org, ns2.example.
   org;
```

O próximo passo é localizar as seguintes linhas:

```
# If this DHCP server is the official DHCP server for the
   local
# network, the authoritative directive should be
   uncommented.
#authoritative;
```

Em seguida, o carácter # deve ser removido da última linha. Ficando como abaixo:

```
# If this DHCP server is the official DHCP server for the
   local
# network, the authoritative directive should be
   uncommented.
authoritative;
```

Feitos os passos anteriores, as seguintes linhas devem ser acrescentadas no final do arquivo dhcpd.conf:

```
subnet 10.2.0.0 netmask 255.255.255.0 {
    range 10.2.0.10 10.2.0.50;
    option broadcast-address 10.2.0.255;
    option routers 10.2.0.9;
    default-lease-time 600;
    max-lease-time 7200;
    option domain-name "local";
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

As principais configurações definidas nas linhas acima é a sub-rede, máscara de rede e a faixa de IPs a serem distribuídas pelo serviço DHCP. Para salvar todas as configurações realizadas no arquivo dhcpd.conf, deve ser utilizada a combinação de teclas Ctrl+O e para sair do editor Ctrl+X.

O segundo arquivo a ser configurado é o isc-dhcp-server através do seguinte comando:

```
$   sudo nano /etc/default/isc-dhcp-server
```

Após o arquivo ser aberto no editor, a seguintes linhas devem ser localizadas:

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES=""
```

Após localizar, a última linha deve ser alterada acrescentando a interface wlan0 entre as aspas. Como no exemplo abaixo:

```
# On what interfaces should the DHCP server (dhcpd) serve DHCP requests?
#       Separate multiple interfaces with spaces, e.g. "eth0 eth1".
INTERFACES="wlan0"
```

Para salvar as configurações realizadas no arquivo isc-dhcp-server, deve ser utilizada novamente a combinação de teclas Ctrl+O e para sair do editor Ctrl+X.

3.3.2 Configurando o IP estático para interface wlan0

O IP estático deve ser configurado no arquivo interfaces através do comando:

```
$   sudo nano /etc/network/interfaces
```

Após o arquivo ser aberto no editor, deve ser verificado se já existe qualquer tipo de configuração relacionada a interface wlan0. Caso exista alguma, deve ser removida ou comentada com o carácter #. Devendo ser mantida somente a linha allow-hotplug wlan0.

Em seguida, devem ser adicionadas as seguintes linhas:

```
iface wlan0 inet static
    address 10.2.0.9
    netmask 255.255.255.0
```

As configurações devem ser salvas utilizando a combinação de teclas Ctrl+O e para sair do editor Ctrl+X.

Por fim, o IP estático deve ser associado a interface wlan0 através do comando:

```
$   sudo ifconfig wlan0 10.2.0.9
```

3.3.3 Configurando o hostapd

No hostapd são configurados os detalhes do ponto de acesso , como SSID e a senha do ponto de acesso. Para isso, o primeiro passo é criar o arquivo de configuração hostapd.conf na pasta /etc/hostapd. O comando abaixo, utilizando o editor nano, criará o arquivo caso ele não exista:

```
$   sudo nano /etc/hostapd/hostapd.conf
```

Após isso, já no editor, as linhas seguintes devem ser acrescentadas:

```
interface=wlan0
driver=rtl871xdrv
ssid=RPi_AP
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=Raspberry123
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Nas linhas de configuração acima, as informações referentes ao ssid e wpa_passphrase podem ser modificadas conforme necessidade. E o driver deve ser informado de acordo com o modelo do adaptador wifi utilizado.

As configurações devem ser salvas através das combinações de teclas Ctrl+O e Ctrl+X para sair do editor.

O segundo arquivo a ser configurado é o hostapd na pasta /etc/default com o comando abaixo:

```
$   sudo nano /etc/default/hostapd
```

Após o arquivo ser aberto no editor, a seguinte linha deve ser localizada:

```
#DAEMON_CONF=""
```

A linha localizada deve ser modificada retirando o carácter # e atribuindo entre as aspas o caminho até o arquivo hostapd configurado anteriormente. Conforme abaixo:

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Por último, as configurações realizadas devem ser salvas com Ctrl+O e Ctrl+X para sair.

3.3.4 Configurando o NAT

NAT é um protocolo que faz a conversão dos endereços IP de uma rede para endereços IP de outra rede. Mas para o contexto deste trabalho, o NAT será utilizado para converter os endereços IP dos dispositivos conectados para um único endereço IP para acesso à internet.

O primeiro arquivo a ser modificado é o sysctl.conf, o mesmo pode ser acessado com o comando abaixo:

```
$   sudo nano /etc/sysctl.conf
```

Após aberto no editor, a seguinte linha deve ser acrescentada no final do arquivo:

```
net.ipv4.ip_forward=1
```

A modificação deve ser salva com os comandos Ctrl+O e Ctrl+X.

Para ativar a nova configuração, o seguinte comando deve ser executado:

```
$   sudo sh -c "echo 1 > /proc/sys/net/ipv4/ip_forward"
```

O próximo passo é configurar a conversão de endereços entre as interfaces eth0 e wlan0 com os comandos abaixo:

```
$   sudo iptables -t nat -A POSTROUTING -o eth0 -j
   MASQUERADE
$   sudo iptables -A FORWARD -i eth0 -o wlan0 -m state --
   state RELATED,ESTABLISHED -j ACCEPT
$   sudo iptables -A FORWARD -i wlan0 -o eth0 -j ACCEPT
```

Para fazer com que os comandos anteriores não precisem ser executados novamente quando o Raspberry Pi for iniciado. O próximo comando deve ser executado:

```
$   sudo sh -c "iptables-save > /etc/iptables.ipv4.nat"
```

O último arquivo a ser configurado é o interfaces com o seguinte comando:

```
$   sudo nano /etc/network/interfaces
```

Após aberto no editor, a seguinte linha de ser acrescentada no final do arquivo:

```
up iptables-restore < /etc/iptables.ipv4.nat
```

A configuração deve ser salva com Ctrl+O e Ctrl+X.

3.3.5 Finalizando a Configuração

Nesta seção serão realizadas as últimas configurações do ponto de acesso wifi.

Primeiramente, é necessário iniciar os serviços do hostapd e isc-dhcp-server com os comandos:

```
$   sudo service hostapd start
$   sudo service isc-dhcp-server start
```

E por último, para iniciar os serviços sempre que o sistema Raspbian iniciar. Os comandos abaixo devem ser executados:

```
$   sudo update-rc.d hostapd enable
$   sudo update-rc.d isc-dhcp-server enable
```

3.4 INSTALANDO O KISMET

O Kismet tem um papel fundamental na solução proposta, pois ele é o responsável por detectar os dispositivos dentro da área de alcance da antena wifi.

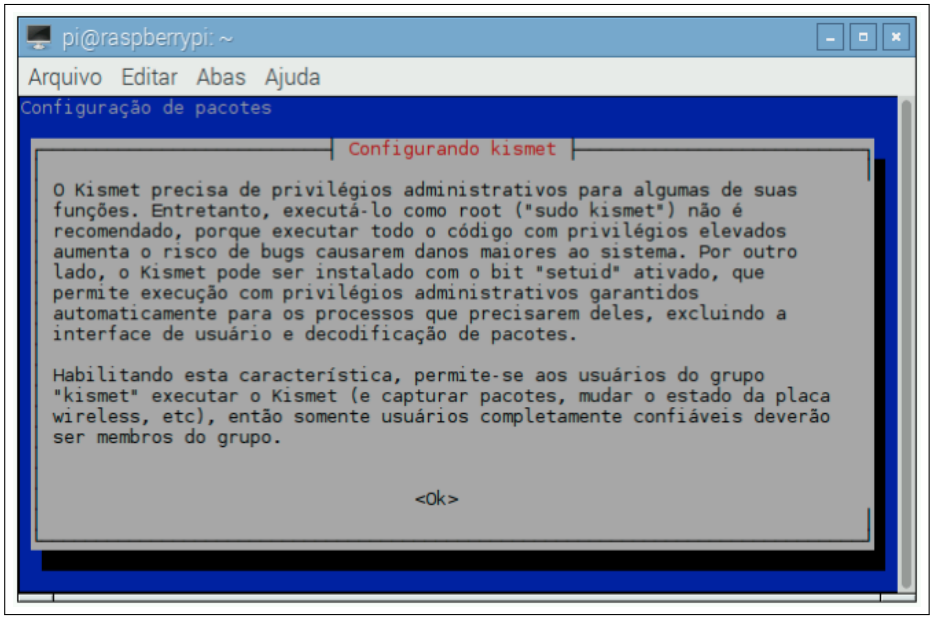
Para baixar e iniciar a instalação do Kismet execute o comando:

```
$   sudo apt-get install kismet
```

Durante a instalação, será informado que o Kismet necessita de privilégios administrativos para realizar algumas de suas funções, conforme Figura 13. No entanto, executá-lo diretamente como root pode ser algo prejudicial a segurança do sistema. Então é recomendada a instalação com o bit “setuid” ativado, assim é permitido a execução de suas funções com os privilégios administrativos garantidos para os processos que tenha essa necessidade.

Ativar o bit “setuid” significa permitir que somente usuários adicionados no grupo “kismet” possam executar funções como: captura de pacotes de rede e colocar a placa wifi em modo de monitoramento.

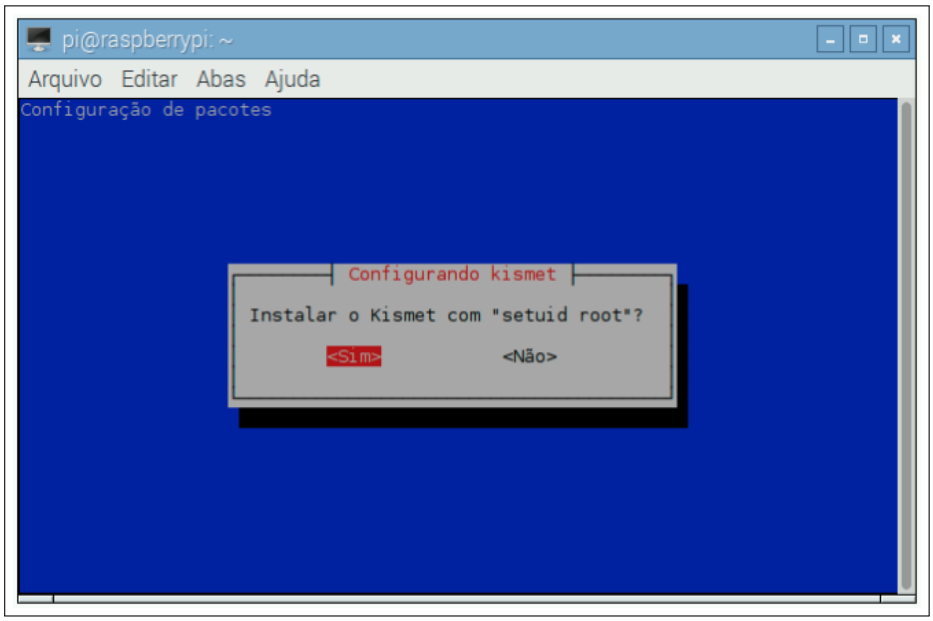
Figura 13 – Configuração Kismet - Tela 1



Fonte: Elaborado pelo autor

Após selecionar a opção <Ok> na tela acima, será questionada a instalação do Kismet com o bit “setuid” ativado. Para isso deve ser selecionada a opção <Sim> na tela abaixo:

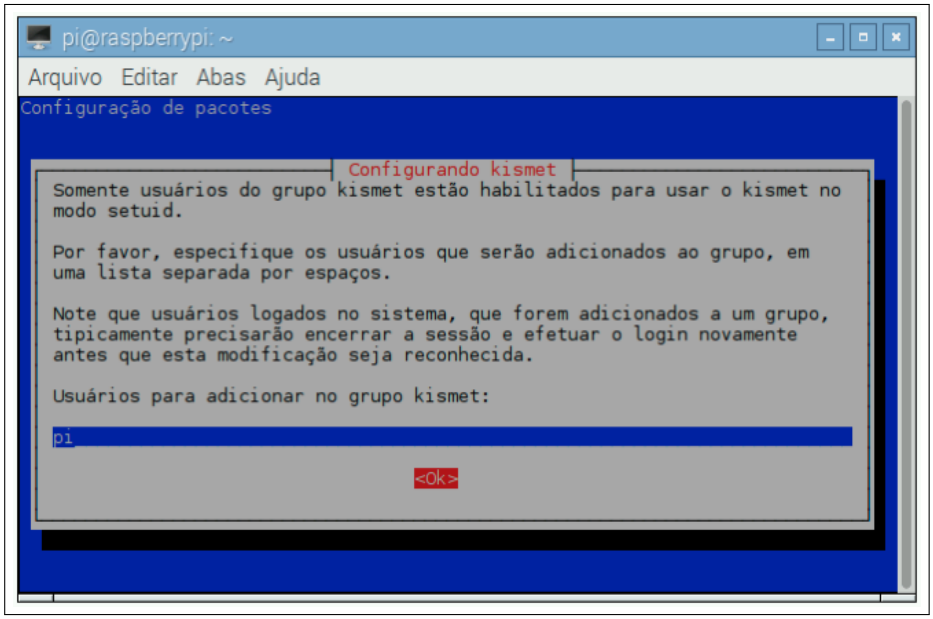
Figura 14 – Configuração Kismet - Tela 2



Fonte: Elaborado pelo autor

Em seguida, os usuários do sistema que irão executar o Kismet devem ser adicionado no grupo “kismet”. Na próxima tela, o usuário “pi” deve ser adicionado no campo texto e confirmado através da opção <Ok>.

Figura 15 – Configuração Kismet - Tela 3



Fonte: Elaborado pelo autor

Após o término da instalação, o arquivo kismet.conf deverá ser modificado. Para isso o comando abaixo deve ser executado:

```
$   sudo nano /etc/kismet/kismet.conf
```

Após o arquivo ser aberto no editor, a seguinte linha deve ser localizada:

```
gps=true
```

E modificada para:

```
gps=false
```

Em seguida será necessário informar ao Kismet a sua fonte para captura de dados. No caso desta proposta, é a interface wlan1. Para isso deve ser localizada a linha:

```
# ncsource=wlan0
```

E o carácter “#” deve ser retirado e wlan0 alterado para wlan1:

```
ncsource=wlan1
```

Após as modificações feitas acima, Ctrl+O para salvar e Ctrl+X para sair do editor.

3.5 SCRIPT DE DETECÇÃO

O script de detecção é o componente desta solução que se comunica diretamente com o Kismet para coletar os dados de pacotes capturados pelo analisador. O script é feito na linguagem python e utiliza a biblioteca KismetClient para comunicação com o Kismet Server para obter os dados de captura.

3.5.1 Instalando o KismetClient

A linguagem python já vem instalada por padrão no sistema Raspbian, não necessitando de nenhum comando adicional para seu funcionamento. Então será preciso instalar somente o KismetClient no python, usando o seguinte comando para clonar a biblioteca:

```
$   git clone https://github.com/PaulMcMillan/kismetclient.
   git
```

Após clonar a biblioteca, a pasta principal da biblioteca deve ser acessada com o comando:

```
$   cd kismetclient/
```

Por último, execute o comando para instalar:

```
$   sudo python setup.py install
```

3.5.2 Código do Script

O código do script python pode ser criado e editado em qualquer editor de texto, devendo ser salvo sempre com a extensão ".py".

No início do script é preciso importar a biblioteca KismetClient com o código abaixo:

```
from kismetclient import Client as KismetClient
```

Logo após, o seguinte código realiza a conexão com o Kismet Server:

```
address = ('127.0.0.1', 2501)
k = KismetClient(address)
```

No código abaixo o protocolo TIME é desativado, pois não é ele que contém as informações dos dispositivos alvos de captura. Em seguida o protocolo CLIENT é ativado especificando os dados relevantes da captura.

```
k.cmd('REMOVE','TIME')
k.cmd('ENABLE','CLIENT','bssid,mac,signal_dbm,firsttime,
   lasttime,datapackets')
```

O próximo passo é criar e registrar um manipulador customizado de protocolo. É no manipulador que os dados de captura serão recebidos e tratados, como no código abaixo:

```
def handle_ssid(client, bssid, mac, signal_dbm, firsttime,
   lasttime, datapackets):
  print 'bssid spotted: {} with mac {} and {} and {} and {}
      and {}'.format(bssid, mac, signal_dbm, time.strftime(
     "%D %H:%M:%S", time.localtime(int(firsttime))), time.
     strftime("%D %H:%M:%S", time.localtime(int(lasttime)))
     , datapackets)
  global flag
```

```
4    if mac == "F8:A9:D0:30:04:63" and flag == False:
5      flag = True
6      start_new_thread(sendMessage,(mac,))
7
8  k.register_handler('CLIENT', handle_ssid)
```

Código da função sendMessage utilizada no manipulador:

```
1  def sendMessage(topic):
2    macAddress = str(topic).replace(":","")
3    data = urllib.urlencode({"topic":macAddress, "message":"
       teste"})
4    u = urllib.urlopen("http://10.2.0.9:8080/BeaconServlet/
       SendMessage?%s" % data)
```

E por último, chamar a função listen():

```
1  while True:
2    k.listen()
```

O método listen() recebe os dados do Kismet Server e encaminha-os para os manipuladores registrados. No código anterior, encaminha para handle_ssid.

3.6 JAVA SERVLET

O objetivo do servlet na solução proposta é prover o serviço que recebe os endereços MAC enviados pelo script python, definir a mensagem a ser enviada e encaminhá-la ao agente de mensagens para publicação.

Java Servlets necessitam de um servidor de aplicação para serem acessíveis e executados. O servidor de aplicações que será utilizado neste trabalho é Apache TomEE.

3.6.1 Instalando o Apache TomEE

O arquivo para instalação do Apache TomEE está disponível em:

```
http://tomee.apache.org/downloads.html
```

A versão do TomEE utilizada neste trabalho é a plume. Por se tratar da versão mais completa em termos de recursos.

Após baixar o arquivo, execute o seguinte comando para descompatar e ao mesmo tempo mover a pasta descompactada para os arquivos do sistema na pasta opt:

```
$   sudo tar -zxvf apache-tomee-1.7.2-plume.tar.gz && mv
   apache-tomee-plume-1.7.2 /opt/
```

Em seguida deve ser criado e configurado um shell script para inicialização do TomEE durante a inicialização do sistema Raspbian.

Execute o comando abaixo para criar um arquivo chamado tomee-plume na pasta init.d do sistema:

```
$   sudo nano /etc/init.d/tomee-plume
```

No editor, o código abaixo deve ser redigido:

Código-fonte 4 – bash version

```
#!/bin/sh
# Tomee-Plume Init-Script

case $1 in

start)
sh /opt/apache-tomee-plume-1.7.2/bin/startup.sh
;;
```

```
9
10 stop)
11 sh /opt/apache-tomee-plume-1.7.2/bin/shutdown.sh
12 ;;
13
14 restart)
15 sh /opt/apache-tomee-plume-1.7.2/bin/shutdown.sh
16 sh /opt/apache-tomee-plume-1.7.2/bin/startup.sh
17 ;;
18
19 esac
20
21 exit 0
```

Após o código ser salvo, é preciso executar o seguinte comando para atribuir permissão ao arquivo:

```
1 $   sudo chmod 755 /etc/init.d/tomee-plume
```

Logo após atribuir a permissão, o comando abaixo precisa ser executado para que o gerenciador de scripts “update-rc.d” possa reconhecer o script tomee-plume.

```
1 $   sudo update-rc.d tomee-plume defaults
```

E por último, executar o seguinte comando para iniciar o Apache TomEE:

```
1 $   /etc/init.d/tomee-plume start
```

3.6.2 Configurando o TomEE

Para que o Apache TomEE possa comunicar com o agente de mensagens é preciso configurar o arquivo tomee.xml localizado na pasta conf.

Para isso, o seguinte comando deve ser executado:

```
$   sudo nano /opt/apache-tomee-plume-1.7.2/conf/tomee.xml
```

A tag abaixo deve ser localizada:

```
<tomee>
  ...
</tomee>
```

E as seguintes tags “Resource” devem ser incluídas:

```
<tomee>
  <Resource id="MyJmsResourceAdapter" type="
     ActiveMQResourceAdapter">
    BrokerXmlConfig =
    ServerUrl       =  tcp://someHostName:61616
  </Resource>

  <Resource id="MyJmsConnectionFactory" type="javax.jms.
     ConnectionFactory">
    ResourceAdapter = MyJmsResourceAdapter
  </Resource>
</tomee>
```

3.6.3 Código do Servlet

O código do servlet pode variar muito dependendo da forma como os dados que compõe a mensagem serão obtidos e formatados para o usuário. Seja através de uma conexão a algum banco de dados ou consumindo webservices. Mas independente disso, o código responsável pelo envio das mensagens para o agente de mensagens sempre segue o mesmo padrão. Como será descrito abaixo:

Primeiro devemos criar uma classe estendendo a classe HttpServlet. Conforme abaixo:

```
@WebServlet("/SendMessage")
public class SendMessage extends HttpServlet {
   ...
}
```

E na mesma classe, declarar a conexão:

```
@WebServlet("/SendMessage")
public class SendMessage extends HttpServlet {

   @Resource
   private ConnectionFactory connectionFactory;
   ...
}
```

No método doGet ou doPost – o uso de um ou outro vai depender da necessidade – deve conter o código responsável pelo envio das mensagens.

O primeiro passo é armazenar os parâmetros enviados pela requisição de envio feita pelo script python. Como a seguir:

```
@Override
protected void doGet(HttpServletRequest req,
   HttpServletResponse resp) throws ServletException,
   IOException {
   String paramTopic = req.getParameter("topic");
   String paramMessage = req.getParameter("message");
   ...
}
```

Em seguida a conexão deve ser criada e iniciada:

```
try {
  Connection connection = connectionFactory.
     createConnection();

  connection.start();
  ...
} catch (JMSException e) {
  e.printStackTrace();
}
```

Logo após, uma sessão deve ser criada:

```
try {
  ...
  // Create a Session
  Session session = connection.createSession(false, Session
     .AUTO_ACKNOWLEDGE);
  ...
} catch (JMSException e) {
  e.printStackTrace();
}
```

Após criar a sessão, tópico e mensagem são criados:

```
try {
  ...
  Topic topic = session.createTopic(paramTopic);

  // Create a MessageProducer from the Session to the Topic
      or Queue
```

```
6    MessageProducer producer = session.createProducer(topic);
7    producer.setDeliveryMode(DeliveryMode.NON_PERSISTENT);
8
9    // Create a message
10   TextMessage message = session.createTextMessage(
        paramMessage);
11   ...
12 } catch (JMSException e) {
13   e.printStackTrace();
14 }
```

E por último, a mensagem é enviada:

```
1  try {
2    ...
3    // Tell the producer to send the message
4    producer.send(message);
5  } catch (JMSException e) {
6    e.printStackTrace();
7  }
```

3.7 AGENTE DE MENSAGENS

O agente de mensagens utilizado neste trabalho é o ActiveMQ, além de ser de código aberto, o mesmo já tem suporte interno ao protocolo MQTT.

3.7.1 Instalação do ActiveMQ

Para baixar o arquivo compactado de instalação do ActiveMQ, o seguinte endereço deve ser acessado:

```
1  http://activemq.apache.org/activemq-5121-release.html
```

Após baixar o arquivo, o seguinte comando deve ser executado para descompactar e mover o arquivo para pasta opt:

```
$   sudo tar -zxvf apache-activemq-5.12.1-bin.tar.gz && mv
   apache-activemq-5.12.1 /opt/
```

Em seguida, deve ser criado um link do arquivo shell script do ActiveMQ para pasta init.d do sistema:

```
$   sudo ln -sf /opt/apache-activemq-5.12.1/bin/activemq /
   etc/init.d/activemq
```

Para registrar o arquivo no gerenciador de scripts do sistema, o comando abaixo deve ser executado:

```
$   sudo update-rc.d activemq defaults
```

Para iniciar o ActiveMQ, o seguinte comando deve ser executado:

```
$   /etc/init.d/activemq start
```

3.7.2 Configuração do ActiveMQ

Para fazer com que o ActiveMQ funcione com base no protocolo MQTT é preciso alterar o arquivo de configuração.

O arquivo activemq.xml deve ser aberto com o seguinte comando:

```
$   sudo nano /opt/apache-activemq-5.12.1/conf/activemq.xml
```

No editor, a seguinte tag deve ser localizada:

```
<transportConnectors>
  ...
</transportConnectors>
```

Uma vez localizada, a tag abaixo deve ser incluída:

```
<transportConnectors>
  ...
  <transportConnector name="mqtt" uri="mqtt://0.0.0.0:1883"
     />
  ...
</transportConnectors>
```

3.8 APLICATIVO ANDROID

REFERÊNCIAS

ALECRIM, E. **O que é Wi-Fi (IEEE 802.11)?** 2013. Disponível em: <http://www.infowester.com/wifi.php>. Acesso em: 16/10/2015.

APPLE INC. **iOS: para entender o iBeacon**. [S.l.], 2015. Disponível em: <https://support.apple.com/pt-br/HT202880>. Acesso em: 11/10/2015.

APPLEINSIDER STAFF. **First look: Using iBeacon location awareness at an Apple Store**. 2013. Disponível em: <http://appleinsider.com/articles/13/12/06/first-look-using-ibeacon-location-awareness-at-an-apple-store>. Acesso em: 11/10/2015.

ASHTON, K. **That 'Internet of Things' Thing**. 2009. Disponível em: <http://www.rfidjournal.com/articles/view?4986>. Acesso em: 10/10/2015.

BOHONOS, S. *et al.* **Universal Real-Time Navigational Assistance (URNA): An Urban Bluetooth Beacon for the Blind**. 2007. Disponível em: <https://users.soe.ucsc.edu/~manduchi/Papers/health02f-bohonos.pdf>. Acesso em: 01/09/2015.

CHAWATHE, S. S. **Beacon Placement for Indoor Localization using Bluetooth**. 2008. Disponível em: <http://aturing.umcs.maine.edu/~sudarshan.chawathe/pubs/bpil.pdf>. Acesso em: 02/09/2015.

CHOI, M.; PARK, W.-K.; LEE, I. **Smart Office Energy Management System Using Bluetooth Low Energy Based Beacons and a Mobile App**. 2015. Disponível em: <https://www.deepdyve.com/lp/institute-of-electrical-and-electronics-engineers/smart-office-energy-management-system-using-bluetooth-low-energy-based-vOsfRcKGHG?articleList=%2Fsearch%3Fquery%3Dble%2Bbeacon>. Acesso em: 12/08/2015.

DANOVA, T. **BEACONS: What They Are, How They Work, And Why Apple's iBeacon Technology Is Ahead Of The Pack**. 2014. Disponível em: <http://www.businessinsider.com/beacons-and-ibeacons-create-a-new-market-2013-12>. Acesso em: 10/10/2015.

GOOGLE DEVELOPERS. **Beacons: Platform Overview**. [S.l.], 2015. Disponível em: <https://developers.google.com/beacons/overview>. Acesso em: 12/10/2015.

KERSHAW, M. **Kismet Readme**. 2011. Disponível em: <https://www.kismetwireless.net/documentation.shtml#readme>. Acesso em: 23/10/2015.

MAYNARD, N. **MQTT e IBM MessageSight: Comunicações seguras e confiáveis para a próxima geração de aplicativos móveis resilientes**. 2015. Disponível em: <http://www.ibm.com/developerworks/br/websphere/techjournal/1501_maynard/1501_maynard.html>. Acesso em: 18/10/2015.

MQTT.ORG. **MQTT**. 2015. Disponível em: <http://mqtt.org/>. Acesso em: 18/10/2015.

ORACLE AND/OR ITS AFFILIATES. **What Is a Servlet?** [S.l.], 2010. Disponível em: <http://docs.oracle.com/javaee/5/tutorial/doc/bnafe.html>.

POSTA, C. **What is ActiveMQ?** 2011. Disponível em: <http://blog.christianposta.com/activemq/what-is-activemq/>. Acesso em: 19/10/2015.

RASPBERRY PI FOUNDATION. **What is a Raspberry Pi?** 2015. Disponível em: <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>. Acesso em: 15/10/2015.

TEIXEIRA, F. **Tudo o que você precisa saber para começar a brincar com iBeacons**. 2014. Disponível em: <http://arquiteturadeinformacao.com/ux-em-espacos-fisicos/tudo-o-que-voce-precisa-saber-para-comecar-a-brincar-com-ibeacons/>. Acesso em: 09/10/2015.