



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

GABRIEL ECKEL PELEGRINA

**ARQUITETURA APOIADA POR BEACONS E BLUETOOTH LOW
ENERGY NO CONTEXTO DE SMART AGRICULTURE**

Assis/SP
2017



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

GABRIEL ECKEL PELEGRINA

**ARQUITETURA APOIADA POR BEACONS E BLUETOOTH LOW
ENERGY NO CONTEXTO DE SMART AGRICULTURE**

Orientador: Prof. MSc. Guilherme de Cleva Farto

Nota do orientador:	Nota do avaliador:
---------------------	--------------------

Assis/SP
2017

DEDICATÓRIA

Dedico este trabalho à minha família, à minha namorada, amigos e todas as pessoas que me apoiaram durante essa jornada e que acreditaram que seria possível alcançar esse objetivo.

AGRADECIMENTOS

Ao meu orientador Prof. MSc. Guilherme de Cleve Farto, pela paciência, apoio prestado nos difíceis momentos e pelas orientações, não somente durante este trabalho, mas também por toda a minha jornada acadêmica.

Aos professores que, durante essa jornada acadêmica, contribuíram, em qualquer aspecto, para o meu crescimento pessoal e profissional.

Aos meus familiares, Raquel Eckel Pelegrina, Marcos Antonio Pelegrina, Grazielle Eckel e Guilherme Eckel Pelegrina, que mesmo nos momentos mais difíceis, continuaram ao meu lado, apoiando-me, conduzindo-me e dando forças durante a minha vida.

À minha namorada, Maria Eduarda Santos Avanzi de Oliveira, por ser a principal razão para a conquista deste objetivo, por todo amor, compreensão e pelos maravilhosos momentos proporcionados.

E a todos que colaboraram de maneira direta ou indireta para a conclusão dessa etapa.

RESUMO

O estado atual da tecnologia da informação e comunicação está permitindo que cada vez mais o mundo esteja conectado. Um novo conceito surgiu, baseado na Internet para possibilitar a conexão de objetos físicos ao mundo virtual. Tal conceito ganhou o nome de *Internet of Things*, IoT, e sua aplicação permite diversas atividades e operações que no passado não era imaginado. Neste trabalho espera-se utilizar a aplicação de IoT juntamente com os conceitos de *Bluetooth Low Energy*, BLE, e *Received Signal Strength Indication*, RSSI, aplicados através dos *beacons* para a detecção de dispositivos inteligentes na prática do conceito de *Smart Agriculture*. As referidas tecnologias possibilitam que a comunicação seja realizada com aplicações na plataforma *Google Android* que será responsável por apresentar dados referentes ao perfil do dispositivo detectado ou à localização do mesmo na fazenda. Desta forma foi desenvolvida uma aplicação capaz de detectar *beacons*, validar a sua localização e distância através do indicador RSSI, bem como exibir informações referentes à região a qual o *beacon* pertence, possibilitando a análise e tomada de decisões precisas pelo agricultor. Como complemento, uma aplicação *Web* foi implementada, com intuito de suprir as informações necessárias para que a aplicação mobile valide os *beacons* e exiba de maneira correta o resultado para o usuário.

Palavras-chave: *Smart Agriculture, Google Android, Internet Of Things, Beacons, Bluetooth Low Energy, Received Signal Strength Indication.*

ABSTRACT

The current state of information and communication technology is allowing more and more the world to be connected. A new concept emerged, based on the Internet to enable the connection of physical objects to the virtual world. Such concept has gained the name of Internet of Things, IoT, and its application allows diverse activities and operations that in the past were not imagined. In this work is expected to use the IoT application together with the concepts of Bluetooth Low Energy, BLE, and Received Signal Strength Indication, RSSI, applied through the beacons for the detection of intelligent devices in the practice of the concept of Smart Agriculture. These technologies enable the communication to be performed with applications on the Google Android platform that will be responsible for presenting data regarding the profile of the detected device or its location on the farm. In this way, was developed an application capable of detecting beacons, validating their location and distance through the RSSI, as well displaying information regarding the region to which the beacon belongs, enabling the analysis and accurate decision making by the farmer. As a complement, a Web application was implemented in order to supply the necessary information so that the mobile application validates the beacons and displays the result correctly for the users.

Keywords: Smart Agriculture, Google Android, Internet Of Things, Beacons, Bluetooth Low Energy, Received Signal Strength Indication.

LISTA DE FIGURAS

Figura 1: Sensores IoT aplicados em máquinas agrícolas.....	20
Figura 2: Troca de dados entre equipamentos agrícolas e escritório.....	21
Figura 3: RFID utilizada para autenticação em eventos.....	27
Figura 4: NFC utilizada para realizar pagamentos.....	28
Figura 5: Logo oficial do Google Android.....	31
Figura 6: Ambiente de desenvolvimento Android Studio.....	34
Figura 7: Camadas da plataforma Android.....	36
Figura 8: Topologia broadcast de Bluetooth LE.....	40
Figura 9: Topologia Connected de Bluetooth LE.....	41
Figura 10: Arquitetura do ambiente.....	44
Figura 11 - Diagrama de Casos de Uso – Administrador.....	46
Figura 12 - Diagrama de Casos de Uso – Usuário.....	47
Figura 13 – Código para validar o IMEI do dispositivo.....	48
Figura 14 – Protótipo da interface da aplicação do usuário	49
Figura 15 – Requisição dos dados do servidor através da API OkHttp.....	50
Figura 16 – Conversão de JSON para classe JAVA.....	51
Figura 17 – Métodos utilizados para receber e enviar mensagens para o beacon...52	
Figura 18 – Código para a detecção e captura de RSSI do beacon encontrado.....	52
Figura 19 – Protótipo da interface da aplicação do administrador.....	53

LISTA DE TABELAS

Tabela 1: Principais bibliotecas da camada Libraries.....	37
---	----

LISTA DE ABREVEATURAS E SIGLAS

ADT – *Android Development Tools*

API – *Application Programming Interface*

BLE – *Bluetooth Low Energy*

IOT – *Internet Of Things*

NFC – *Near Field Communication*

OHA - *Open Handset Alliance*

PIB – *Produto Interno Bruto*

RFID – *Radio Frequency Identification*

RSSI – *Received Signal Strength Indicator*

SDK – *Software Development Kit*

SUMÁRIO

1. INTRODUÇÃO	12
1.1 OBJETIVOS	14
1.2 JUSTIFICATIVA	15
1.3 MOTIVAÇÃO	15
1.4 ESTRUTURA DO TRABALHO	16
2. AGROINDÚSTRIA E SMART AGRICULTURE	17
2.1 EVOLUÇÃO NA AGROINDÚSTRIA.....	17
2.2 ESTATÍSTICAS NA AGROINDÚSTRIA	18
2.3 FUNDAMENTOS DE SMART AGRICULTURE	19
2.4 DESAFIOS E OPORTUNIDADES.....	21
2.5 APLICAÇÕES DE SMART AGRICULTURE	22
3. INTERNET OF THINGS	24
3.1 ORIGEM DE IOT	24
3.2 CONCEITOS DE IOT	25
3.3 PRINCIPAIS TECNOLOGIAS	26
3.3.1 SENSORES.....	26
3.3.2 RFID	26
3.3.3 NFC	27
3.3.4 BLUETOOTH.....	28
3.3.5 BEACONS	29
4. DISPOSITIVOS MÓVEIS E GOOGLE ANDROID	30
4.1 DISPOSITIVOS MÓVEIS	30
4.2 PLATAFORMA GOOGLE ANDROID	30
4.3 OPEN HANDSET ALLIANCE	32
4.4 SISTEMA OPERACIONAL LINUX	32
4.5 AMBIENTE DE DESENVOLVIMENTO	33
4.5.1 INSTANT RUN.....	34
4.5.2 EDITOR DE CÓDIGO.....	35

4.5.3 EMULADOR	35
4.6 ANDROID SDK.....	35
4.6.1 CAMADA APPLICATION.....	36
4.6.2 CAMADA APPLICATION FRAMEWORK	37
4.6.3 CAMADA LIBRARIES.....	37
4.6.4 CAMADA RUNTIME	38
4.6.5 CAMADA LINUX KERNEL.....	38
5. BLUETOOTH LOW ENERGY E RSSI.....	39
5.1 FUNDAMENTOS DE BLUETOOTH LOW ENERGY (BLE)	39
5.2 FUNDAMENTOS DE RECEIVED SIGNAL STRINGTH INDICATION (RSSI) ..	41
5.3 APLICAÇÕES E USOS DE BLE E RSSI	41
6. PROPOSTA E DESENVOLVIMENTO DO TRABALHO	43
6.1 DEFINIÇÃO DO PROBLEMA.....	43
6.2 ARQUITETURA DO AMBIENTE	44
6.3 EXPERIMENTO	45
6.3.1 MODELAGEM DO NEGÓCIO	46
6.3.2 AMBIENTE FÍSICO.....	47
6.3.3 APLICAÇÃO CLIENTE	48
6.3.3.1 APLICAÇÃO DO USUÁRIO.....	48
6.3.3.2 APLICAÇÃO DO ADMINISTRADOR.....	53
7. CONCLUSÃO	54
7.1 TRABALHOS FUTUROS	54
REFERÊNCIAS.....	55

1. INTRODUÇÃO

Com o grande aumento da população mundial ocorreu, conseqüentemente, um aumento na demanda por alimentos. Neste contexto, a agricultura possui grande importância na produção de alimentos ao redor do mundo. No Brasil, a agricultura alcançou grandes avanços e hoje as exportações agrícolas representam grande parcela do PIB nacional (SILVÉRIO, 2016).

De acordo com o Ministério da Agricultura (2010) em uma análise histórica de produção, em 2010 o Brasil atingiu um aumento na produção agrícola de 174% quando comparado com o ano de 1960. Segundo essa análise, sem o avanço das tecnologias utilizadas nos meios rurais, o Brasil deveria disponibilizar cerca de 145 milhões de hectares para atingir a produção alcançada em 2010. No entanto, graças ao grande avanço tecnológico, a produção agrícola atingiu cerca de 150,8 milhões de toneladas em apenas 47,5 milhões de hectares.

No âmbito mundial, a agricultura passará por grandes desafios nas próximas décadas devido aos 9,6 bilhões de pessoas habitando o planeta até 2050. Prevê-se que a produção alimentícia deve atingir um aumento de 70% até o ano previsto, um grande desafio para a agricultura visto que problemas como a disponibilidade de terras aráveis ser limitada, bem como a crescente necessidade de água doce, chegando a atingir 70% do abastecimento mundial voltados para a agricultura (FORBES, 2015).

Com os desafios citados, torna-se relevante a utilização das mais avançadas tecnologias no meio agrícola. Com isso surgiu a *Smart Agriculture*, ou agricultura de precisão que vem ganhando a confiança dos agricultores. O termo pode ser entendido no mesmo conceito de *Internet of Things*, no entanto, seu foco é interconectar objetos do meio agrícola com o mundo por meio da Internet, com o objetivo de tornar as fazendas inteligentes, conectando os mais diversos equipamentos agrícolas a fim de gerar informações consistentes a todo tempo.

Em outras palavras, o termo está totalmente relacionado com a integração de objetos físicos ou virtuais de modo que, conectados à Internet, coletam dados cruciais como, por exemplo, a temperatura de uma caldeira industrial, troquem esses dados com dispositivos

a fim de gerar, através do processamento e análise dos dados, serviços e outras informações em escalas inimagináveis (ALMEIDA, 2015).

Para Zanella et al. (2014) o conceito visa tornar a Internet cada vez mais presente no cotidiano devido à facilidade de acesso, permitindo a interligação entre inúmeros dispositivos ou objetos, como câmeras, sensores e veículos.

Com o avanço da tecnologia, *Smart Agriculture* se tornou a grande responsável pela inserção dos mais avançados dispositivos tecnológicos nos meios de produção agrícola. Segundo Leal (2013), o computador de mão, *Personal Digital Assistant*, utilizado nas áreas agrícolas vem se tornando obsoleto, visto que está perdendo seu posto para crescente evolução dos dispositivos como *smartphones* e *tablets*.

O uso de dispositivos móveis no meio agrícola possui maior aceitação dos usuários se comparado com outros dispositivos, devido à sua constante evolução e inserção no mercado. Devido a este motivo, torna-se indispensável o desenvolvimento de aplicações *mobile* com intuito de auxiliar os usuários nas suas tarefas realizadas no ambiente agrícola (AMADO; GIOTTO, 2009).

Segundo Tonin et al. (2014), o crescimento de dispositivos móveis resulta em um aumento significativo no número de aplicativos desenvolvidos devido à portabilidade e mobilidade dos dispositivos.

Uma tecnologia pouco explorada no meio agrícola e que vem ganhando espaço no mundo da *Internet of Things* nas áreas de automação comercial, *Smart outdoors* e localização em ambientes internos, *indoor position*, é *Bluetooth Low Energy* (BLE) ou *Blueetooth Smart*. A referida tecnologia surgiu em um passado recente e está sendo utilizada pela *Internet of Things* com a proposta de uma alternativa atraente, de baixo custo e consumo de energia para a detecção de outros dispositivos em uma região limitada (RAZA; MISRA; HE; VOIGT, 2015).

Segundo Menegotto (2015), *Bluetooth Low Energy* é uma tecnologia que possibilita a emissão e recepção de sinais utilizando baixas quantidades de energia. O desenvolvimento do BLE possibilitou o surgimento de dispositivos emissores pequenos e econômicos. Muito utilizado em projetos de localização em ambientes internos. Tal detecção ocorre por meio de um indicador da intensidade de sinal recebido pelo dispositivo a ser detectado (*Received*

Signal Strength Indication). Segundo Costa (2015), *Received Signal Strength Indication (RSSI)* é um indicador da potência de sinal de rádio recebido pelo aparelho receptor no qual seu número indicador é representado com valores negativos que podem estar entre 0 e -100.

De acordo com os fabricantes da tecnologia, a principal característica que diferencia BLE com sua versão antecessora é a permissão de conexão entre dispositivos sem a necessidade de pareamento, permitindo assim tomadas de ações passivas e contínuas pelo dispositivo ou aplicativos (SMART CARD ALLIANCE MOBILE AND NFC COUNCIL, 2015).

A possibilidade de desenvolvimento de uma aplicação envolvendo *Smart Agriculture* e a tecnologia *Bluetooth LE* para dispositivos móveis na plataforma *Android* permitirá usufruir dos benefícios da plataforma junto com a eficácia dos métodos de detecção via Bluetooth LE, possibilitando a criação de um sistema de baixo custo.

1.1 OBJETIVOS

O objetivo geral deste projeto é o de pesquisar ferramentas e arquiteturas necessárias para o desenvolvimento de componentes que contribuirão com a técnica de sensoriamento proposta, para a verificação da presença de smartphones em determinada região e o envio de informações filtradas para uma aplicação *mobile* detectada e validada pela arquitetura proposta, bem como apresentar os conceitos ao redor de *Smart Agriculture* e *Internet of Things*.

A partir dos artefatos deste projeto, será possível aplicar os conceitos para a concepção e modelagem de uma arquitetura de sensor *Bluetooth Low Energy* aplicada junto à plataforma móvel *Android*. Assim, espera-se avaliar experimentalmente a aplicabilidade da tecnologia na área agrícola e contribuir para criação de novas técnicas baseados na mesma arquitetura.

1.2 JUSTIFICATIVAS

Tendo em vista o grande crescimento da população mundial, o conceito de *Smart Agriculture* tem sido aplicado como excelente ferramenta auxiliar no meio agrícola para o aumento da produtividade e na precisão, redução dos custos a fim de produzir melhores resultados, pois dispensa o excesso de recursos físicos, do uso de mão de obra além da redução do tempo de execução das tarefas e do tempo de transmissão das informações do campo até os escritórios das fazendas.

Além disso, devido ao grande aumento do uso de smartphones na plataforma *Google Android* por empresas agrícolas, o desenvolvimento de aplicativos voltados para a gestão agrícola e integração com dispositivos da *Smart Agriculture* torna-se extremamente importante para agilizar a busca de informações mesmo fora dos escritórios.

1.3 MOTIVAÇÃO

O desenvolvimento deste projeto de pesquisa consiste no fato de que *Smart Agriculture* é um tema com muito a ser explorado e pode contribuir com a qualidade e gestão da produção em geral.

Em busca de ajudar a obtenção de melhores resultados na produção e gestão agrícola, soluções envolvendo *Smart Agriculture* podem ser implantadas nas mais diversas áreas tais como plantio, adubação, irrigação, colheita, gestão de propriedades, gestão e localização de maquinários, entre outras áreas que competem à agricultura.

Aplicar os conceitos estudados para o desenvolvimento do sistema na plataforma *Google Android* agregará muito valor para os agricultores pois assim poderão receber quaisquer tipos de informações diretamente em seu dispositivo móvel fazendo com que sejam reduzidos os custos e a perda de tempo na busca de uma informação que antes só poderia ser acessada em um computador fora do campo.

Outra motivação são as consideráveis chances de o mercado de trabalho necessitar de profissionais com conhecimentos e experiências na linha do tema abordado nessa pesquisa, uma vez que a agricultura é responsável pela maior produção de alimentos,

demanda que aumenta paralelamente com a população mundial, necessitando do auxílio de tecnologias para que essa produção mínima seja alcançada de forma rápida e precisa.

1.4 ESTRUTURA DO TRABALHO

Este trabalho será estruturado nas seguintes partes:

- **Capítulo 1 – Introdução:** Este capítulo contextualizará a área de estudo apresentará os objetivos, justificativas e motivações para o desenvolvimento da pesquisa.
- **Capítulo 2 – Agroindústria e Smart Agriculture:** Neste capítulo pretende-se apresentar conceitos e regras sobre a agroindústria e Smart Agriculture.
- **Capítulo 3 – Internet of Things (IoT):** Neste capítulo será apresentado os conceitos necessários sobre Internet of Things, bem como seus benefícios e desafios.
- **Capítulo 4 – Plataforma *Google Android*:** Este Capítulo apresentará os benefícios do uso de dispositivos móveis bem como a importância e crescimento da tecnologia *Google Android*.
- **Capítulo 5 – Bluetooth Low Energy e RSSI:** Neste capítulo pretende-se abordar os conceitos relacionados à RSSI bem como os benefícios, conceitos e plataformas relacionadas à tecnologia BLE.
- **Capítulo 6 – Proposta e desenvolvimento do Trabalho:** Este capítulo apresentará a arquitetura proposta baseada nos fundamentos de Bluetooth Low Energy e RSSI, bem como um experimento com intuito de validar a implementação da arquitetura estudada
- **Capítulo 7 – Conclusão:** Neste capítulo serão apresentados as vantagens e desvantagens dos componentes implementados na tecnologia BLE.
- **Referências**

2. AGROINDÚSTRIA E SMART AGRICULTURE

Neste capítulo serão apresentadas as definições fundamentais e básicas para o entendimento da agroindústria, bem como seus avanços e necessidades no decorrer dos anos. Encontram-se também definições e conceitos sobre *Smart Agriculture*, que visa atender e implementar as necessidades do meio agrícola através de dispositivos inteligentes.

2.1 EVOLUÇÃO DA AGROINDÚSTRIA

A agroindústria rural se refere a todas as atividades relacionadas à transformação e beneficiamento, de forma sistemática, dos produtos provindos da agricultura, pecuária, aquicultura ou silvicultura a partir de matéria-prima desde que a destinação final tenha origem do produtor rural. O intuito da agroindústria é prolongar a disponibilidade das matérias-primas, aumentar os prazos de validade bem como agregar valor aos alimentos *in natura*, buscando manter suas características. O grau de transformação pode variar em função dos objetivos dos produtores. Para cada matéria-prima, a agroindústria se torna um segmento que se estende desde o fornecimento de insumos até o consumidor final (IBGE, 2007; RURALTINS, 2017).

No Brasil, o processo do agronegócio iniciou-se durante a ocupação de seu território, com a exploração da madeira do Pau Brasil. Em meados de 1532, com a extinção do Pau Brasil, se deu o início do ciclo de cana-de-açúcar, que se tornara reconhecida no mercado por volta de 1910 com a chegada do automóvel flex e à mistura de etanol com gasolina, que ainda no século XX fez-se a obrigatoriedade do uso de 5% de etanol nacional como aditivo através do decreto 19.717 (UNICA, 2017). Segundo o Ministério da Agricultura (2010), o país se tornou um dos maiores produtores e exportadores de produtos derivados da indústria canavieira.

Devido à grande variedade climática e condições ambientais do país, houveram regiões em que a produção da cana-de-açúcar não foi satisfatória, tornando necessário o desenvolvimento de outro segmento do agronegócio. Um exemplo é a região do sertão,

onde a partir do problema apresentado anteriormente, surgiu a prática da atividade pecuária. Desde então, o crescimento agropecuário não parou de crescer ao redor do país, o tornando um dos grandes produtores de animais e o maior de gado do mundo, representando grande parte da produção mundial (SILVÉRIO, 2016).

O Brasil é credenciado a ser o grande celeiro do mundo, fazendo com que a identificação, interpretação e análise dos problemas e oportunidades sejam de suma importância para o crescimento e destaque no setor do agronegócio. Nos últimos anos, o agronegócio brasileiro se desenvolveu de forma rápida devido ao grande avanço da tecnologia voltada para o setor. Tal avanço teve importante participação no desenvolvimento econômico e social do país, reduzindo a taxa de desemprego e melhorando na condição de vida da população (FERREIRA, 2009).

2.2 ESTATÍSTICAS NA AGROINDÚSTRIA

De acordo com o Ministério da Agricultura (2010), o Brasil possui grande importância nas exportações agrícolas, com mais de US\$ 76 bilhões em exportações, sendo um dos maiores do mundo no setor. Entretanto, em média o país utiliza cerca de 70% de sua produção agrícola para comercialização interna, como por exemplo o etanol onde aproximadamente 90% de sua produção é destinada ao consumo interno do país.

Segundo IBGE (2016), o crescimento do setor da agroindústria em 2015 ocorreu devido ao satisfatório desempenho da agricultura. Os aumentos da produção de alguns produtos puderam ser destacados para tal crescimento, como por exemplo a lavoura de soja, com aumento de 11,9 % e milho com 7,3%. O setor canavieiro teve um crescimento de 4,2%. Segundo a ministra da Agricultura, Pecuária e Abastecimento, Kátia Abreu, o setor agroindustrial traz cada vez mais resultados diferenciados e satisfatórios, acentuando o destaque do setor na economia brasileira.

Segundo estimativas da CNA, de 2015 para 2016 o PIB do agronegócio apresentou alteração de 21,5% para 23%. Para o ano de 2017, a tendência é de que o percentual de participação do agronegócio na economia continue a crescer, podendo apresentar uma expansão de aproximadamente 2%. Ainda de acordo com CNA, o segmento sucroenergético também apresentará expansão devido aos aumentos dos preços do

açúcar e etanol. Já para outros segmentos, a expansão dependerá da recuperação da economia do país, para que as pessoas possuem renda para comprar (CNA, 2016).

Devido à grande importância da agricultura na economia, torna-se necessário a utilização de grandes áreas de terras. Segundo Análise realizada pela FAO, estima-se que 38% das terras são utilizadas para a prática da agricultura. Além das terras, grande parte do consumo de água é destinado à agricultura, onde anualmente são utilizados cerca 3.800 KM³ de água, representando aproximadamente 70% de água de toda água doce do mundo (FAO, 2013). Com a utilização das grandes áreas de terras e o elevado consumo de água no setor, fica evidente a suma importância da agricultura para a economia mundial.

2.3 FUNDAMENTOS DE SMART AGRICULTURE

Após muito tempo utilizando de serviços manuais para a produção agrícola, a agricultura atual encontra-se em um período considerado Agricultura Moderna, onde máquinas agrícolas de tecnologia avançada, capazes de gerar e trocar informações, são cada vez mais importantes nas lavouras, auxiliando a reduzir custos e aumentar a produtividade (SILVÉRIO, 2016). A Figura 1 representa a aplicação de sensores em um equipamento agrícola.



Figura 1 - Sensores IoT aplicados em máquinas agrícolas (In: STECON, 2017)

De acordo com Kaloxyllos (2012), em meados dos anos 2000 os agricultores passaram a utilizar computadores e *softwares* para acompanhar suas transações, organizar seus dados bem como monitorar suas culturas de forma eficaz. Embora o setor agrícola venha se tornando uma intensiva indústria de dados, os agricultores ainda necessitam realizar a coleta e avaliação dessa enorme onda de informações capturadas por sensores a fim de se tornarem mais eficientes nas suas produções (Csótó, 2010).

Ultimamente a tecnologia está cada vez mais presente na realidade do agricultor, de modo a auxiliá-lo em suas tomadas de decisões voltadas às suas lavouras. Tecnologias como *RFID*, *Bluetooth*, *drones* e sensores têm contribuído para a informatização da produção rural, transformando as propriedades rurais em fazendas inteligentes (TANG, 2002).

Com o surgimento de *Smart Agriculture* e com a constante evolução da agricultura, podemos encontrar diversas formas de se aplicar o conceito. É possível através de dispositivos inteligentes e da grande massa de dados coletados, analisar e tomar a melhor decisão possível, facilitando as tarefas para o produtor. O uso da tecnologia por agricultores está em constante crescimento. Sensores, *wearables* nos animais, *drones*, mapeamento da saúde das lavouras, entre outros, se tornaram meios essenciais para a produção sustentável e rentável (Powell, 2016). A Figura 2 apresenta a conexão entre equipamentos agrícolas no campo e a sede de uma fazenda.

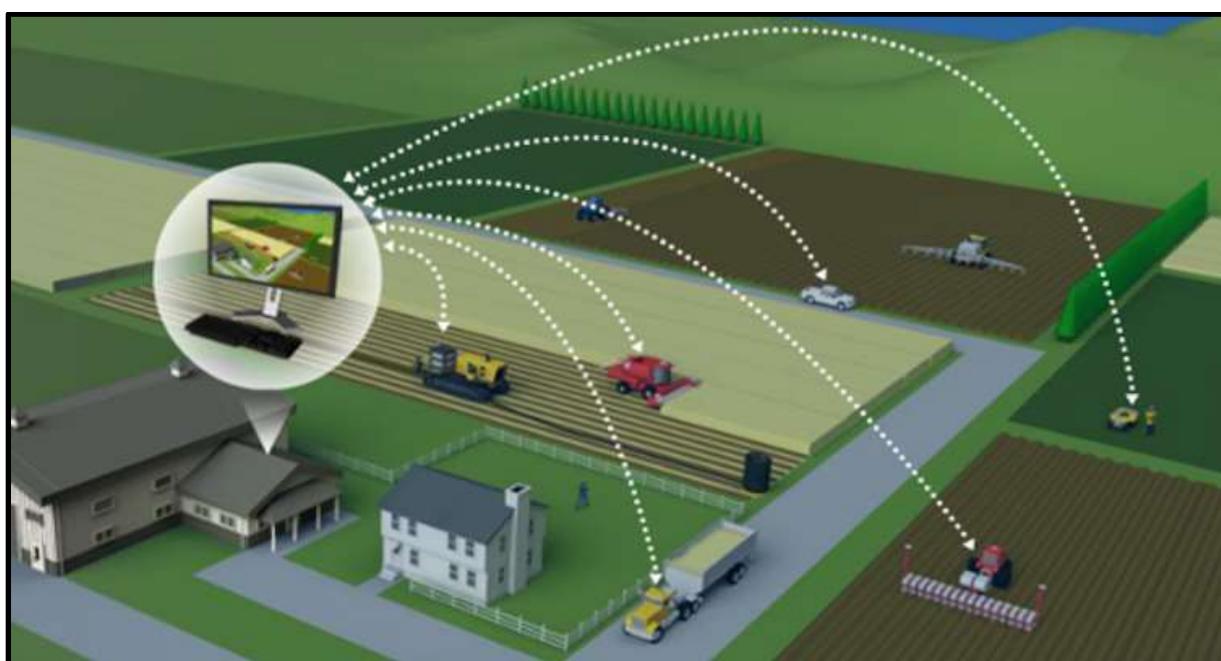


Figura 2 - Troca de dados entre equipamentos agrícolas e escritório. (In: HYDRONAV, 2017)

2.4 DESAFIOS E OPORTUNIDADES

As evoluções da agroindústria, juntamente com o surgimento de *Smart Agriculture*, proporcionaram diversos benefícios e oportunidades para a área. Por meio da grande massa de dados, sensores, tags e dispositivos inteligentes tornou-se possível analisar, prever e tomar decisões importantes, assim auxiliando os agricultores, podendo gerar resultados significantes para o meio ambiente, bem como para a área financeira. Além destes, tal evolução também reflete diretamente na economia brasileira, devido ao país ser

considerado um importante exportador de produtos agrícolas, tendo alcançado cerca de US\$ 76 bilhões em exportações (MINISTÉRIO DA AGRICULTURA, 2010; SILVÉRIO, 2016).

Além dos benefícios citados, a utilização de ferramentas de *Smart Agriculture* também possibilita a redução de custos com mão de obra. Um exemplo é a pecuária leiteira, onde a automação de tarefas possibilitou a redução do número de colaboradores responsáveis pelo manejo do rebanho, bem como possibilita a gestão de operações à distância, permitindo o foco dos gestores em tarefas com maior grau de importância e valor para a empresa. A redução de falhas humanas também pode ser considerada um destaque, onde equipamentos inteligentes e dispositivos eletrônicos embarcados em máquinas e equipamentos agrícolas permitem a redução da dependência da habilidade de seus operadores (PIVOTO; MORAES; SILVA; KAWANO; TALAMINI, 2016).

Para os desafios enfrentados pela agroindústria, podemos destacar a limitação da produção. Segundo FAO (2012), a quantidade de recursos produzidos para cada pessoa no mundo está se tornando limitada, podendo gerar um conflito entre recursos disponíveis e a população mundial. A estimativa é que para o ano de 2050 serão necessários aproximadamente 3 bilhões de toneladas de recursos agroindustriais para atender toda a população, prevista para cerca de 9,15 bilhões de pessoas. A cada dia que se passa existem mais e mais pessoas no mundo e tornando os recursos para cada pessoa limitado.

2.5 APLICAÇÃO DE SMART AGRICULTURE

A disseminação de tecnologias de informação e comunicação no meio rural vem aumentando nos últimos anos com grandes investimentos em máquinas e implementos agrícolas por empresas do setor. Sensores eletrônicos e atuadores estão cada vez mais acessíveis e presentes no dia a dia dos produtores rurais. Diversas são as oportunidades de aplicação, das quais estão a comparação do desempenho de máquinas e operadores, avaliação de erros na aplicação de insumos, sementes e outros produtos de diferentes naturezas, automatização de máquinas e implementos agrícolas, detalhes importantes sobre custos, gastos, erros e perdas de operações, bem como o compartilhamento de

informações entre os agentes da cadeia de produção (PIVOTO; MORAES; SILVA; KAWANO; TALAMINI, 2016).

No Brasil, os investimentos em *Smart Agriculture* também estão presentes e pode ser citada a cultura de cana-de-açúcar, onde destacam-se o uso de *Smart Agriculture* para a gestão de frota e comunicação com a indústria processadora para controle de entrada de cana-de-açúcar; monitoramento e análise das áreas de plantio por meio da utilização de veículos aéreos não tripulados (VANT), bem como sistemas de piloto automático para máquinas de plantio e colheita (PIVOTO; MORAES; SILVA; KAWANO; TALAMINI, 2016).

A aplicação de *Smart Agriculture* possibilita a inserção de novos agentes profissionais no ramo do agronegócio, como as empresas de consultoria e startups especializadas em análise e gestão de dados, onde pode-se citar a DonEng, que utiliza VANT's para a análise de falhas no plantio, colheita, bem como a saúde da lavoura.

3. INTERNET OF THINGS

Com o rápido avanço e popularização de dispositivos inteligentes, estima-se que até 2020 cerca de 26 bilhões de dispositivos inteligentes estarão instalados pelo mundo, movimentando uma receita superior a US\$ 300 bilhões que resultará em um valor econômico agregado de US\$ 1,9 trilhão (STAMFORD, 2013)

Atualmente vivemos um momento de grande avanço tecnológico onde a maximização do poder de processamento dos dispositivos ocorre mesmo com a miniaturização dos mesmos. Tal avanço não só altera a natureza dos dispositivos como também possibilita inúmeras novas oportunidades. O acoplamento de sensores e atuadores aos dispositivos tornando objetos inteligentes estão aumentando em grande escala e estes quando conectados à Internet, permitem a capacidade de calcular, monitorar, analisar e agir com base nos resultados. O conceito de interconectar objetos e dispositivos inteligentes denomina-se *Internet of Things* (COETZEE; EKSTEEN, 2011).

Neste capítulo serão descritos os conceitos de *Internet of Things* (IoT), sua evolução e as principais tecnologias relacionadas, além de um exemplo da arquitetura.

3.1 ORIGEM DE IOT

O termo *Internet of Things* já é bastante antigo, embora seu grande destaque tenha ocorrido nos últimos anos. O relato da primeira utilização do termo foi feito pelo pesquisador britânico Kevin Ashton em 1999, porém não se pode garantir que tenha sido a origem da ideia de IoT. No relato foi proposto que o conceito que *Internet of Things* referisse à revolução tecnológica cujo objetivo é interconectar objetos e dispositivos do dia a dia à rede mundial (ROCHA, 2015).

De acordo com Zambarda (2014) a limitação de tempo fará com que pessoas busquem outras formas de se conectar com o mundo. Com os diversos objetos inteligentes conectados, como geladeiras, óculos, pulseiras, e automóveis será possível acumular

dados cada vez mais precisos possibilitando a redução, otimização e economia de recursos.

3.2 CONCEITOS DE IOT

Internet of Things (IoT) pode ser considerada uma revolução tecnológica nos campos da computação e comunicação. Seu conceito é definido pela conexão entre diversos objetos inteligentes com a rede mundial de computadores, onde é possível realizar a comunicação com pessoas e até mesmo entre outros objetos.

A proposta de *Internet of Things* é de que quaisquer objetos do meio físico possam ser virtualmente conectados à Internet. A variedade dos objetos à serem conectados não está restrita apenas aos equipamentos eletrônicos ou produtos com alto valor agregado, mas também objetos simples do cotidiano como roupas, sapatos, acessórios, e outros infinitos utilizados no dia a dia (BERGAMASCHI, 2010).

Segundo Vasseur e Dunkels (2010) para a concretização da *Internet of Things* faz-se necessário a utilização de objetos acoplados a sensores, atuadores, pequenos microprocessadores, dispositivo de comunicação e uma fonte de alimentação.

A *Internet Of Things* terá grande impacto nos mais diversos processos de nossas vidas, podendo influenciar em nosso comportamento, decisões e valores. A aplicação de tecnologias de IoT permitirá infinitudes de aplicações inovadoras que, conseqüentemente, proporcionarão melhorias na qualidade de vida das pessoas, redução de custos e recursos bem como novas oportunidades inovadoras.

3.3 PRINCIPAIS TECNOLOGIAS

3.3.1 SENSORES

Dispositivos responsáveis por capturar estímulos do meio físico e transformá-los em pulsos elétricos mensuráveis. O principal objetivo de um sensor é o de monitorar as mais diversas alterações no ambiente físico como umidade, níveis de oxigênio, pressão atmosférica, temperatura, presença de luz, movimentos, orientação, proximidade, vibrações e outras mais possibilidades. Cada um dos sensores possui sua implementação própria, destinando-se para distintas situações com intuito de obter a maior quantidade de informações possível. Devido à grande diversidade de sensores e de suas aplicações, bem como a possibilidade de utilização de múltiplos para a captura de dados do meio físico, estes dispositivos vem sendo grande alvo para pesquisas e investimentos (ZHU, 2010; CAMAROTTO, 2014).

3.3.2 RFID

RFID é a abreviação de *Radio Frequency Identification* ou Identificação por Radiofrequência. Trata-se de uma rede de comunicação à distância e sem fio, que funciona pela através da identificação de frequências de rádio. A referida tecnologia é composta por dispositivos denominados etiquetas e leitores de rádio identificação, onde as etiquetas podem ser acopladas à produtos, seres vivos ou qualquer objeto a fim de identificação e os leitores são responsáveis por receber as informações provindas da etiqueta. De acordo com Duroc e Kaddour (2012), a comunicação entre os dispositivos ocorre através da etiqueta, cujo objetivo é enviar sinais à um determinado leitor capaz de converte-los em informações significativas.

Para o funcionamento do RFID são possíveis dois modos: Ativo e passivo. No primeiro, a etiqueta fica responsável pela geração de sua energia, através de uma fonte de alimentação, com capacidade de envio de dados ao leitor de forma autônoma. Já no segundo modo, toda a energia do sistema é fornecida pelo leitor, que fica responsável pela detecção e autenticação da etiqueta. Normalmente tags RFID são utilizadas com intuito de

identificação, rastreamento de objetos em supermercados e setores logísticos, bem como transporte de cargas. Uma das vantagens de sua utilização é a capacidade de captura de múltiplos dados em uma única vez realizada pelo leitor, não havendo necessidade de contato físico ou visual com a etiqueta (FREIBERGER; BEZERRA, 2010). A Figura 3 demonstra a utilização da tecnologia RFID para a autenticação de participantes em eventos.



Figura 3 - RFID utilizada para autenticação em eventos.

3.3.3 NFC

Near Field Communication (NFC) é uma tecnologia de comunicação entre dispositivos eletrônicos, de curto alcance, baseada no princípio de identificação por radiofrequência (RFID). Trata-se de uma comunicação *peer-to-peer*, de duas vias, entre dois dispositivos (Iniciador e alvo), habilitados para esta tecnologia, capazes de enviar e receber informações. Para garantir uma segurança quase intrínseca, a transmissão das informações ocorre quando ambos os dispositivos são colocados a poucos centímetros de distância (SPINSANTE; GAMBI, 2015).

De acordo com Gordilho (2012), NFC pode operar de duas maneiras: Ativa ou passiva. Na primeira, ambos os dispositivos, iniciador e alvo, geram seus próprios campos de

radiofrequência, podendo compartilhar informações entre si. Na segunda, o iniciador gera o campo de radiofrequência e o alvo utiliza deste para se comunicar.



Figura 4 - NFC utilizada para realizar pagamentos.

3.3.4 BLUETOOTH

Bluetooth consiste em uma tecnologia sem fio criada por Ericsson em 1994. O objetivo da tecnologia é de comunicação entre dispositivos a curta distância, usando transmissão de rádio podendo operar entre 2.4 e 2.485 GHz. Pode ser aplicado em diversos dispositivos, como celulares, fones de ouvido, rádios, equipamentos da área da saúde entre outros mais, permitindo a troca de dados entre os dispositivos pareados. Para que ocorra o pareamento, a tecnologia conta com a junção de hardware e software, utilizando-se de um chip para a realização da detecção e uma interface programada para a realização do pareamento (BLUETOOTH, 2017).

3.3.5 BEACONS

Beacons consistem em dispositivos que podem utilizar a tecnologia *Bluetooth Low Energy (BLE)* para a localização de outros dispositivos em ambientes internos, permitindo que aplicativos emitam sinais ao usuário. A comunicação de um *beacon* BLE ocorre por meio do envio de pequenos pacotes, transmitidos em intervalos regulares, contendo o número de identificação do *beacon* transmissor. Quando uma aplicação capaz de captar esses sinais e seus pacotes, uma ação ou conteúdo vinculado com o número de identificação do *beacon* é exibida ao usuário, sem necessidade de intervenção do mesmo (KONTAKT, 2017)

4. DISPOSITIVOS MÓVEIS E GOOGLE ANDROID

O objetivo desta sessão é apresentar as definições fundamentais e básicas para o entendimento da tecnologia Google Android, bem como sua arquitetura. Encontram-se também os avanços no mercado dos dispositivos móveis.

4.1 DISPOSITIVOS MÓVEIS

O mercado da mobilidade está crescendo cada vez mais. Com o passar dos anos dispositivos móveis estão ganhando o mercado devido à grande evolução na capacidade de processamento, armazenamento, integrações, além de se tornarem mais acessíveis aos seus consumidores (PEREIRA; SILVA, 2009).

A popularidade dos dispositivos móveis tomou tamanha proporção que no ano de 2015 cerca de 7,2 bilhões de celulares circularam pelo mundo, quase igualando a estimativa realizada pela ONU, onde projetou-se cerca de 7,3 bilhões de habitantes no mesmo ano (UN, 2015; TELECO, 2016).

4.2 PLATAFORMA GOOGLE ANDROID

Atualmente a procura por dispositivos móveis encontra-se em constante crescimento. Aparelhos com funcionalidades mais comuns estão se extinguindo, o que aquece a busca por dispositivos móveis mais inteligentes e com funcionalidades que facilitem ainda mais o cotidiano de seus usuários. Nesse mercado competitivo, usuários comuns tendem a buscar por aparelhos com um visual elegante, moderno, com navegação intuitiva bem como infinitos aplicativos e recursos. O *Android* foi criado justamente para agradar os seus usuários, possibilitando a utilização de diversos recursos em um único aparelho. (LECHETA, 2015)

Além da grande procura desses dispositivos por usuários comuns, o meio corporativo está buscando incorporar aplicações móveis em seus ambientes com intuito de agilizar

processos e integrar suas aplicações. O foco principal de uma empresa é o lucro; portanto a utilização de dispositivos móveis se torna indispensável em um mundo onde a palavra “mobilidade” está cada vez mais conhecida (ABLESON; COLLINS; SEM, 2009).

O *Android* é o sistema operacional móvel da *Google*, lançado no ano de 2007. Seu sucesso se deu devido à união entre *Google* e um grupo de gigantes do mercado de mobilidade denominado *Open Handset Alliance* (OHA). A plataforma consiste em uma pilha de *softwares* com foco no desenvolvimento de aplicativos para diversos dispositivos móveis, baseado no núcleo do sistema operacional *Linux*, garante estabilidade e confiabilidade, além de fornecer recursos essenciais de segurança, como o modelo de permissões baseado nos usuários, isolamento de processos e remoção de funcionalidades desnecessárias ou potencialmente inseguras. Também possui diversas aplicações pré-instaladas e um poderoso ambiente de desenvolvimento (ABLESON; COLLINS; SEM, 2009; LECHETA, 2015; ANDROID, 2017). A Figura 8 ilustra o logo oficial escolhido para a plataforma *Google Android*.

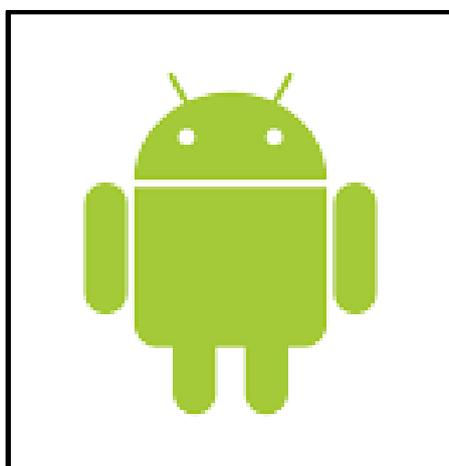


Figura 5 - Logo oficial do Google Android (In: ANDROID, 2017)

A plataforma da *Google* é a primeira do segmento que possui o código completamente livre e *open source*, o que representa grande vantagem na sua evolução uma vez que toda a comunidade pode contribuir para melhorar a plataforma. Além disso, fabricantes de celulares podem fazer uso do sistema operacional bem como alterar o seu código-fonte

para criação de produtos customizados, sem custo alguma e sem a necessidade de compartilhar as alterações (LECHETA, 2015).

Android é considerada uma plataforma de *software* completa, adaptável para qualquer configuração, desde a inicialização das configurações e todo o caminho até as aplicações. Com os diversos dispositivos *Android* no mercado, provou-se que tem potencial para realmente competir na área da mobilidade (PEREIRA; SILVA, 2009).

4.3 OPEN HANDSET ALLIANCE

Para o desenvolvimento da plataforma *Android* diversas empresas líderes de mercado de telefonia como a *Motorola*, *LG*, *Samsung*, *Sony Ericsson* e muitas outras se uniram, juntamente com a *Google*, formando a *Open Handset Alliance* (OHA). O objetivo desta aliança era padronizar uma plataforma de código aberto e livre para celulares, justamente para atender a todas as expectativas e tendências do mercado atual, deixando os consumidores mais satisfeitos com o produto final (PEREIRA; SILVA, 2009).

A *Open Handset Alliance* (OHA) é um grupo formado empresas líderes do ramo da tecnologia móvel, que compartilham a visão de enriquecer a vida de inúmeras pessoas ao redor do mundo através da mudança de suas experiências com a mobilidade (ALLIANCE, 2017).

De acordo com LECHETA et al. (2015) o objetivo do grupo é definir uma plataforma única e aberta para celulares, com intuito de satisfazer os consumidores com o produto final. Outro objetivo dessa aliança é criar uma plataforma moderna e flexível para o desenvolvimento de aplicações corporativas

4.4 SISTEMA OPERACIONAL LINUX

O sistema operacional do *Android* é baseado no *Kernel do Linux*. Usar um *Kernel do Linux* garante estabilidade e confiabilidade no sistema, além de aproveitar de principais recursos de segurança, como o modelo de permissões baseado nos usuários, isolamento de

processos e remoção de funcionalidades desnecessárias ou potencialmente inseguras. Além disso, permite o desenvolvimento de drivers de hardware para um Kernel conhecido (ANDROID, 2017).

Cada aplicativo no *Android* dispara um novo processo no sistema operacional. Diversos processos podem ser executados simultaneamente, enquanto o *Kernel* do sistema operacional é o responsável por realizar todo o controle da memória. Caso necessário, cabe ao sistema operacional decidir encerrar algum processo para liberar espaço na memória, podendo ou não o reiniciar posteriormente. Para cada aplicação executada no *Android*, um único processo é disparado ao sistema operacional, e cada processo possui uma *thread* dedicada. Para garantir a segurança da aplicação instalada no dispositivo, o sistema operacional cria um usuário que terá acesso apenas a sua estrutura de diretórios. (FARTO, 2010; LECHETA, 2015).

4.5 AMBIENTE DE DESENVOLVIMENTO

Para aplicações *Android* é possível utilizar a linguagem de programação *Java* e um ambiente de desenvolvimento como *Eclipse* e *NetBeans*. No ambiente de desenvolvimento *Eclipse* é possível utilizar o *plug-in Android Development Tools*, ADT, que é um conjunto de ferramentas para o auxílio no desenvolvimento *Android* (FARTO, 2010).

De acordo com Android (2017), até meados de 2013 o ambiente *Eclipse* era o preferido pela *Google* tratando-se de desenvolvimento *Android*. No entanto, foi anunciado o ambiente *Android Studio*, passando a ser a *IDE* oficial da plataforma incluindo ferramentas mais rápidas necessárias para o desenvolvimento de aplicativos *Android*, bem como recursos de depuração, desempenho e sistema de compilação flexível. A Figura 6 demonstra como é a Interface de desenvolvimento do *Android Studio* e sua estrutura de projetos.

4.5.2. EDITOR DE CÓDIGO

O editor inteligente auxilia no desenvolvimento rápido e no aumento da produtividade oferecendo sugestões inteligentes de códigos conforme o desenvolvedor digita, além de ser uma poderosa ferramenta para refatoração e análise de código (ANDROID, 2017).

4.5.3. EMULADOR

Android Studio conta com um emulador capaz de criar e testar aplicativos em qualquer versão de dispositivos *Android*. Com o *Android Emulator* é possível iniciar aplicativos mais rapidamente do que em dispositivos físicos, além de poder simular diversos recursos de hardware, como GPS, velocidade de rede e entrada de toques na tela (ANDROID, 2017).

4.6 ANDROID SDK

O *Android SDK* é o conjunto de softwares utilizado para o desenvolvimento no *Android*. Segundo Lecheta (2015), o *Android SDK* contém ferramentas utilitárias como recursos de depuração e desempenho, um emulador para a simulação de um dispositivo móvel bem como uma *API* completa para a linguagem *Java*, contendo todas as classes necessárias. A Figura 7 representa a estrutura em camadas da plataforma *Google Android*. A seguir será feita uma descrição de cada camada a fim de ajudar no entendimento do funcionamento dos processos bem como no desenvolvimento de aplicações mais eficientes.

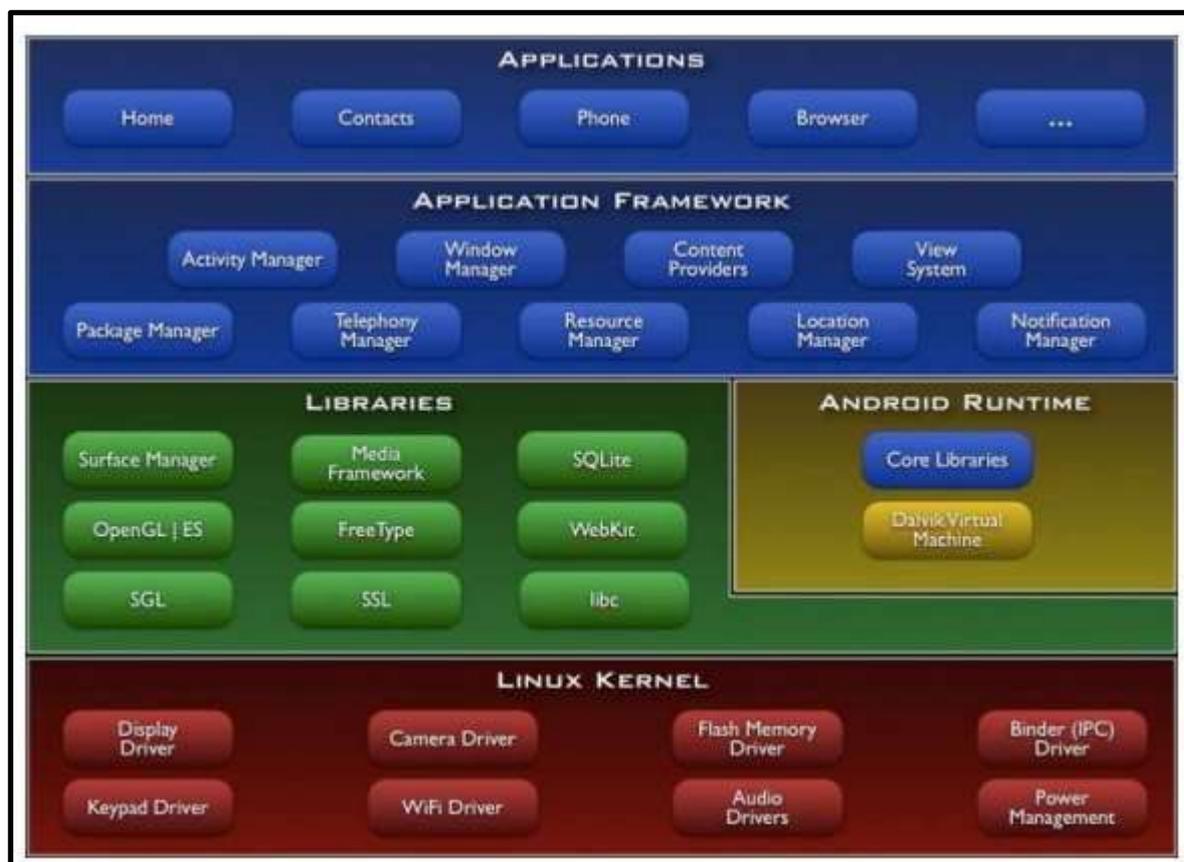


Figura 7 - Camadas da plataforma *Android*. (In: FARTO, 2010).

4.6.1 CAMADA *APPLICATIONS*

A camada *applications* contém o conjunto dos aplicativos instalados no sistema *Android*, tais como calendários, gerenciadores de e-mail, SMS, navegadores de Internet, contatos, entre outros. Os aplicativos inclusos na plataforma não possuem um status diferenciado em relação aos aplicativos terceirizados, podendo estes se tornar um aplicativo padrão do usuário. (ANDROID, 2017).

4.6.2 CAMADA *APPLICATION FRAMEWORK*

É a camada responsável por disponibilizar, por meio de APIs, o conjunto completo de recursos do Sistema Operacional *Android* necessários para reutilização de componentes e desenvolvimento simplificado de aplicativos *Android* (ANDROID, 2017).

4.6.3 CAMADA *LIBRARIES*

O Android inclui um conjunto de bibliotecas C/C++ que podem ser utilizadas diretamente no código nativo por meio do Android NDK. As funcionalidades de algumas dessas bibliotecas são expostas aos desenvolvedores por meio das *Java Framework APIs*. Na Tabela 1, é possível visualizar as principais bibliotecas da plataforma (FARTO, 2010; ANDROID, 2017).

Biblioteca	Descrição
System C Library	Uma implementação derivada da biblioteca C padrão sistema (libc) do BSD sintonizada para dispositivos rodando Linux
Media Libraries	As bibliotecas de mídia suportam os mais populares formatos de áudio, vídeo e imagem
Surface Manager	Subsistema de exibição, bem como múltiplas camadas de aplicações 2D e 3D
LibWebCore	Um <i>Web Browser Engine</i> utilizado tanto no <i>Android Browser</i> quanto para exibições <i>Web</i>
SGL	<i>Engine</i> de gráficos 2D
3D libraries	Implementação baseada no <i>OpenGL</i> As bibliotecas empregam aceleração 3D via <i>hardware</i> (quando disponível) ou o <i>software</i> de renderização 3D altamente otimizado
FreeType	Renderização de fontes <i>Bitmap</i> e <i>Vector</i>
SQLite	Um poderoso e leve <i>Engine</i> de banco de dados relacional disponível para todas as aplicações <i>Android</i>

Tabela 1 – Principais bibliotecas da camada *Libraries*. (In: FARTO, 2010)

4.6.4 CAMADA *RUNTIME*

Toda aplicação *Android* roda seu próprio processo em uma instancia do *Android Runtime*, ART, projetado para executar diversas máquinas virtuais em dispositivos de baixa memória. Antes do *Android 5.0*, cada aplicação rodava seu processo em uma instancia da máquina virtual *Dalvik*. A vantagem do ART está na utilização de arquivos em formato de *bytecode*, projetados especialmente para *Android*, otimizados para menor consumo de memória, além de possuir recursos como compilação *ahead-of-time* e *just-in-time*, *Garbage Coletor* otimizado, bem como melhor compatibilidade de depuração, exceções detalhadas e relatório de erros (ANDROID, 2017).

4.6.5 CAMADA *LINUX KERNEL*

A plataforma *Google Android* utiliza o *Kernel* do *Linux* para os serviços centrais do sistema, tais como gerenciamento e encadeamento de memória de baixo nível, segurança, além de permitir aos fabricantes de dispositivos desenvolver de drivers de hardware para um *Kernel* conhecido. O kernel também funciona como um middleware entre as camadas da plataforma, atuando como uma camada de abstração entre hardware e o software (FARTO, 2010; ANDROID, 2017).

5. BLUETOOTH LOW ENERGY E RSSI

O objetivo desta sessão é apresentar as definições fundamentais e básicas para o entendimento das tecnologias baseadas em RSSI e Bluetooth Low Energy bem como as aplicações e usos das mesmas.

5.1 FUNDAMENTOS DE BLUETOOTH LOW ENERGY (BLE)

A tecnologia *Bluetooth Low Energy*, ou *Bluetooth Smart*, foi introduzida pela SIG com a proposta de uma versão *Bluetooth* de baixa energia construída para IoT. Sua eficiência energética torna-o destaque para dispositivos que funcionam por longos períodos de tempo sob fonte de energia. *Bluetooth Low Energy* (BLE) é uma tecnologia desenvolvida como complemento ao *Classic Bluetooth*, no entanto a proposta é de se ter a menor potência possível em uma tecnologia de comunicação sem fio. A tecnologia foi projetada para o mais baixo consumo de energia possível, conseqüentemente não obterá altas taxas de dados, bem como uma conexão por muitas horas ou dias (TOWNSED; CUFÍ; DAVIDSON, 2014).

Segundo as especificações da SIG, *BLE* opera com banda de 2,4 GHz empregando um transceptor para o combate de interferências e oscilações. A operação a taxa básica utiliza uma modulação de frequência binária para minimizar a complexidade do transceptor. A taxa de bits suportada pode chegar a 1 megabit por segundo. BLE emprega esquema de acesso múltiplo conhecido como acesso múltiplo por divisão de frequência que possui 40 canais físicos onde 3 são utilizados para publicidade e 37 são utilizados para transferência de dados (BLUETOOTH, 2017).

Um dispositivo *Low Energy* pode se conectar à um dispositivo por duas maneiras: *Broadcasting* ou *Connections*. No caso de *broadcasting*, uma estrutura é estabelecida onde é possível um dispositivo emissor enviar dados de maneira unidirecional para os demais dispositivos com escuta ativa conhecidos como observadores. Neste caso existem dois tipos de dispositivos específicos: os *broadcaster* e os *observers*. O *broadcaster* é responsável por enviar periodicamente pacotes não-conectáveis a qualquer dispositivo disposto a receber, enquanto os *observers* buscam em frequência pré-definida, quaisquer

pacotes não-conectáveis transmitidos no momento. O entendimento de Broadcasting é importante devido ser a única maneira de transmitir informações para mais de um dispositivo em mesmo instante de tempo. No entanto, nessa forma de comunicação não existe a confirmação de entrega. A Figura 8 demonstra o funcionamento do mecanismo da topologia *broadcast*, onde é permitido o envio de dados de um BLE para vários dispositivos capazes de receber a informação transmitida (TOWNSED; CUFÍ; DAVIDSON, 2014)

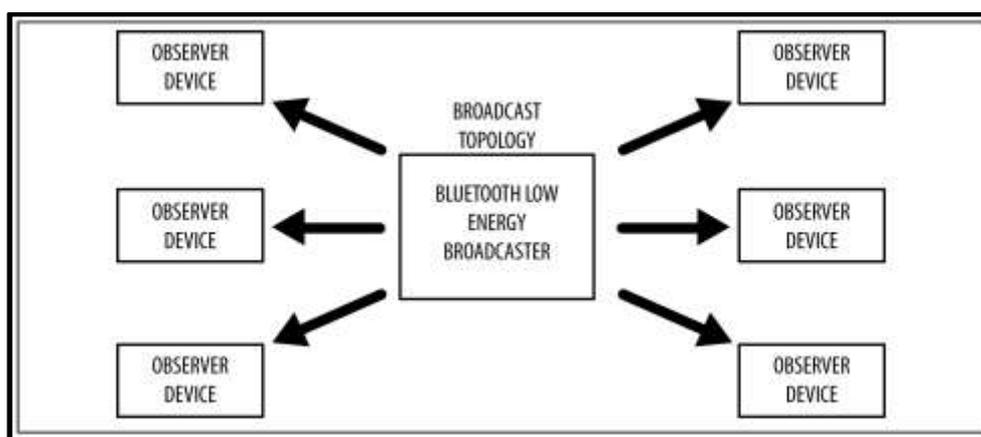


Figura 8 - Topologia broadcast de Bluetooth LE. (In: TOWNSED; CUFÍ; DAVIDSON, 2014)

Em caso das *Connections*, ou conexões, um dispositivo mestre inicia uma busca por dispositivos que estejam habilitados para conexões. A conexão é estabelecida através do envio e da resposta de um pacote de requisição enviado do dispositivo mestre para o dispositivo detectado. Neste caso, existem dois tipos de dispositivos específicos: Central (*master*) e periféricos (*slave*). Dispositivos centrais fazem repetidamente uma varredura nas frequências pré-definidas em busca de pacotes conectáveis, inicia uma conexão em momento adequado e gere o tempo para trocas periódicas de dados. Já os dispositivos periféricos enviam pacotes conectáveis periodicamente e também aceita uma conexão de entrada. Uma vez que a conexão é estabelecida, o periférico respeita o cronograma estabelecido pelo dispositivo central e troca regularmente informação com este. A Figura 9 demonstra o funcionamento do mecanismo da topologia *connection*, onde é permitida a conexão entre diversos dispositivos periféricos com um dispositivo central, capazes de trocar informações entre si (TOWNSED; CUFÍ; DAVIDSON, 2014)

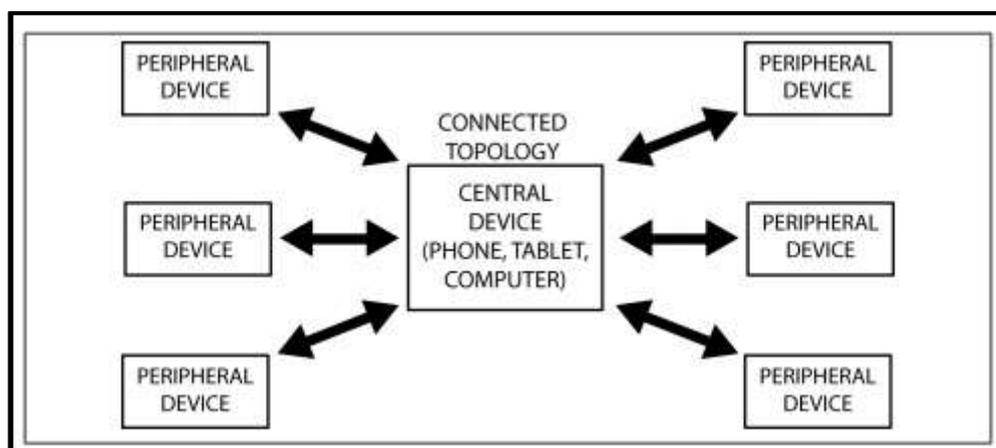


Figura 9 - Topologia Conectada de Bluetooth LE. (In: TOWNSED; CUFÍ; DAVIDSON, 2014)

5.2 FUNDAMENTOS DE RECEIVED SIGNAL STRENGTH INDICATOR (RSSI)

Received Signal Strength Indicator é a medição da potência de sinal de radiofrequência recebido em uma rede sem fio, onde geralmente é utilizado para cálculos de distâncias entre dispositivos. Vale ressaltar que *RSSI* não está relacionado à qualidade de sinal, e sim à força na qual o sinal é recebido (MOATAMED; ARJUN; SHAHMOHAMMADI; RAMEZANI; NAEIM; SARRAFZADEH, 2016).

De acordo com Presa (2014), *RSSI* consiste na análise da intensidade de sinal recebido pelo receptor, baseados em princípios físicos de modo a analisar a propagação do sinal. Para obtenção de uma resposta mais precisa e eficiente, torna-se indispensável a realização da calibragem dos dispositivos, sendo necessário possuir base de dados completa sobre as características do ambiente.

5.3 APLICAÇÕES E USOS DE BLE E RSSI

Com o passar dos anos, a popularidade crescente sobre a presunção da informação de posição, tem conduzido uma quantidade considerável de estudos. As informações de

localização destinam-se a diversos tipos de aplicações, tais como monitoramento de crianças, pacientes em hospitais, propagandas específicas dentro de um estabelecimento, dentre outras. No entanto, a localização em estabelecimentos internos se torna um grande desafio visto que os dispositivos não conseguem obter sinais suficientes do GPS para isso. Para tal desafio, uma das soluções encontradas foi a utilização de BLE juntamente com seu RSSI para a detecção de distância de dispositivos em ambientes internos ().

Um exemplo de utilização de BLE juntamente com RSSI é a empresa Estimote, que utiliza beacons com transmissores BLE, transformando os locais em ambientes mais interativos e objetos comuns em comunicadores com telefones e outros dispositivos inteligentes. Um aplicativo vinculado recebe o sinal transmitido pelo beacon e busca por algum conteúdo vinculado ao perfil do usuário ou à microlocalização do beacon. Se houver conteúdo, o mesmo é mostrado ao usuário como notificação ou diretamente na aplicação (ESTIMOTE, 2017)

Atualmente, grandes empresas como HAMAD INTERNATIONAL AIRPORT, HIA, e F.C.BARCELONA utilizam beacons em suas dependências. No caso do aeroporto HIA, os beacons vem sendo utilizados para permitir uma navegação simples e precisa, conduzindo seus passageiros aos seus destinos, permitindo que seus passageiros se sintam mais à vontade e ajudando-os a identificar sua localização em tempo real dentro do aeroporto. Já o FC BARCELONA passou a utilizar os beacons para aumentar a experiência de seus fãs dentro de suas companhias. Seu aplicativo contém informações sobre os jogadores e profissionais do time, bem como dispara promoções aos visitantes de acordo com a sua localização no estabelecimento (MOHAMMED, 2016; ESTIMOTE, 2017).

6. PROPOSTA E DESENVOLVIMENTO DO TRABALHO

Neste capítulo serão apresentados a especificação e a implementação do modelo proposto por este Trabalho, bem como a definição do problema a ser abordado. O principal objetivo constitui em desenvolver uma arquitetura apoiada por Bluetooth Low Energy, capaz de detectar dispositivos móveis em uma fazenda, a fim de fornecer, em tempo real, informações sobre determinadas áreas de uma propriedade.

6.1 DEFINIÇÃO DO PROBLEMA

A definição do problema constituiu-se em um experimento sobre a implantação de aplicações baseadas na tecnologia *Bluetooth* e na arquitetura *Google Android*. Para o desenvolvimento do experimento, foram utilizados conceitos da comunicação *Bluetooth 4.0*, uma placa *Arduíno*, um módulo BLE, bem como ferramentas para implementar uma aplicação na arquitetura *Google Android* capaz de comunicar com a arquitetura proposta. Este desenvolvimento foi aplicado em um ambiente rural, permitindo que os administradores cadastrem informações e que os usuários obtenham essas informações na aplicação *Mobile*. O foco principal do problema foi implementar uma arquitetura capaz de detectar dispositivos inteligentes com intuito capturar informações de uma base de dados para que possam ser visualizadas na aplicação *Android*. A fim de facilitar o desenvolvimento, o mesmo foi dividido em três módulos.

1º Módulo: Ambiente Físico

Neste módulo será desenvolvido uma arquitetura responsável por detectar os dispositivos inteligentes, a fim de validar o perfil do dispositivo e realizar a transferência de dados referentes a microlocalização da arquitetura e ao perfil do dispositivo detectado.

2º Módulo: Interface do administrador

Neste módulo serão desenvolvidos os componentes responsáveis pelo gerenciamento das informações contidas no sistema, bem como o gerenciamento dos dispositivos autorizados a serem detectados pelo ambiente físico.

3º Módulo: Interface do usuário

Neste módulo serão desenvolvidos os componentes responsáveis pela conexão com o ambiente físico, buscar e receber informações, bem como criar uma interface que permitirá a interação do usuário.

6.2 ARQUITETURA DO AMBIENTE

A modelagem do problema a ser abordado no experimento é ilustrada na figura 10. A imagem ilustra a forma como é arquitetado o sistema.

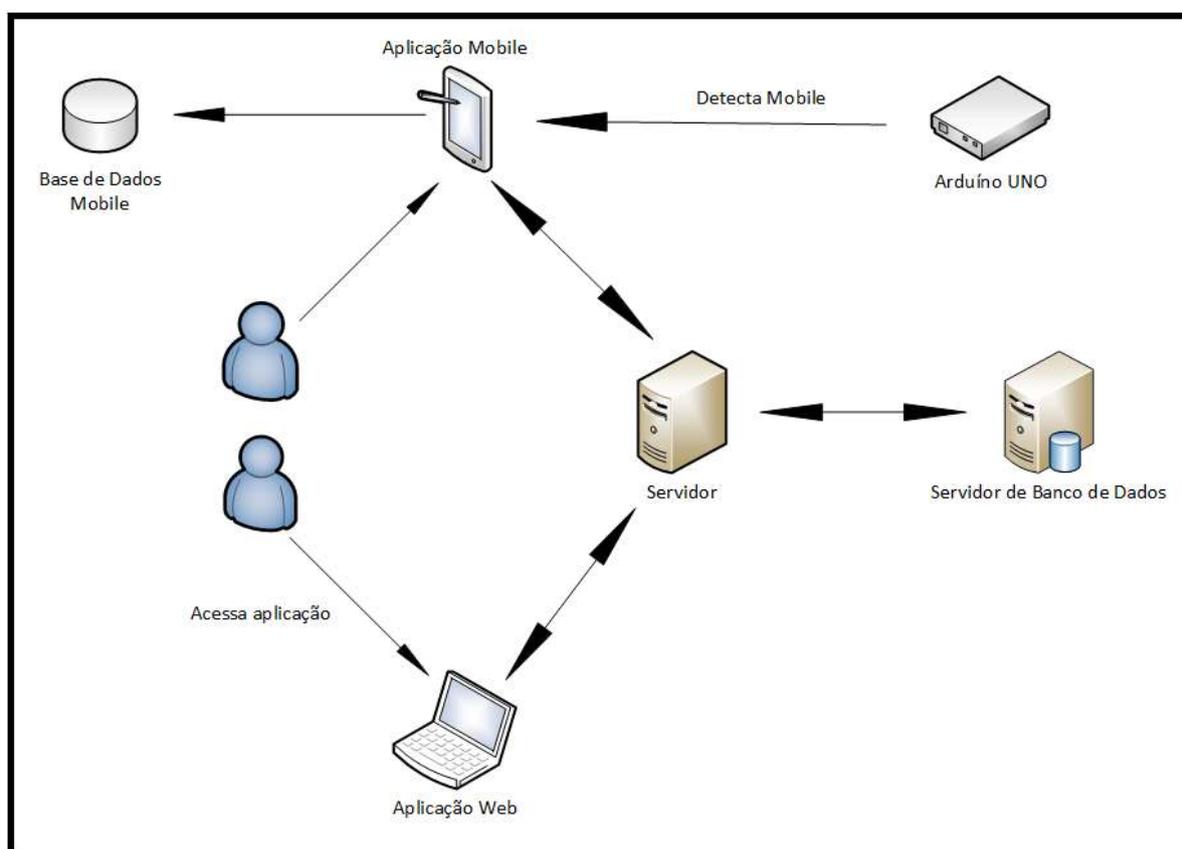


Figura 10 - Arquitetura do ambiente

O modelo arquitetado conta com uma estrutura em *Arduíno*, juntamente com um módulo *Bluetooth* 4.0, que será responsável pela detecção dos dispositivos em determinado raio de distância. Além disso, pode-se observar as entidades usuárias, administrador e usuário. A entidade administrador será responsável por gerenciar os dispositivos que poderão ser detectados, assim garantindo a segurança para que dispositivos não autorizados recebam

informações. A entidade usuário utilizará um aplicativo Mobile, desenvolvido na plataforma *Google Android*, para receber informações e interagir com o sistema.

O ambiente conta com uma base de dados, que é responsável por armazenar a identificação dos dispositivos autorizados, bem como informações referentes às microrregiões onde se encontrarão a arquitetura. Uma vez que o dispositivo é autorizado pela entidade administrador, o mesmo pode ser detectado e passa a receber informações por meio da arquitetura.

6.3 EXPERIMENTO

O experimento consistirá na implementação de um ambiente agrícola, permitindo que a arquitetura reconheça os dispositivos móveis em um determinado raio de alcance; administradores realizem o cadastro de informações das fazendas, bem como de seus dispositivos por meio de uma interface web, e também que usuários obtenham informações por meio de um aplicativo móvel, resultando em duas aplicações cliente.

A aplicação Web será acessada pelos administradores que deverão ser devidamente autenticados no sistema por meio de uma tela de *Login*. A partir do acesso ao sistema, o administrador poderá gerenciar os dispositivos móveis, tipos de culturas, equipamentos, as áreas da fazenda, bem como as operações que nelas podem ser realizadas (plantio, colheita, aplicação de insumos, herbicidas, dentre outras).

A aplicação móvel tem como objetivo fornecer mecanismos para que, ao serem detectados pela arquitetura proposta, exiba informações referentes à região na qual se encontra, bem como oferecer opções para a solicitação da realização de alguma das operações possíveis para a região sem a dependência de quaisquer outros recursos além de um celular ou *tablet*, e conexão com a *Internet*. Além de agilizar a tomada de ações em determinadas regiões, também será possível visualizar as informações adicionais sobre a região na qual o usuário se encontra, permitindo uma visão moderna para essa tarefa. No momento da solicitação de alguma operação, os dados serão transmitidos para a fazenda assim que uma conexão com a Internet for estabelecida, caso contrário, a solicitação ficará armazenada no aplicativo até que isso ocorra.

6.3.1 MODELAGEM DO NEGÓCIO

Para um melhor entendimento do funcionamento do sistema, serão apresentados os diagramas de casos de uso das entidades administrador e usuário. O administrador é o responsável pelo gerenciamento dos dispositivos autorizados pelo sistema, bem como pelas informações persistidas na base de dados como equipamentos, regiões, tipos de cultura e operações.

A figura 11 representa o diagrama de casos de uso para a entidade administrador.

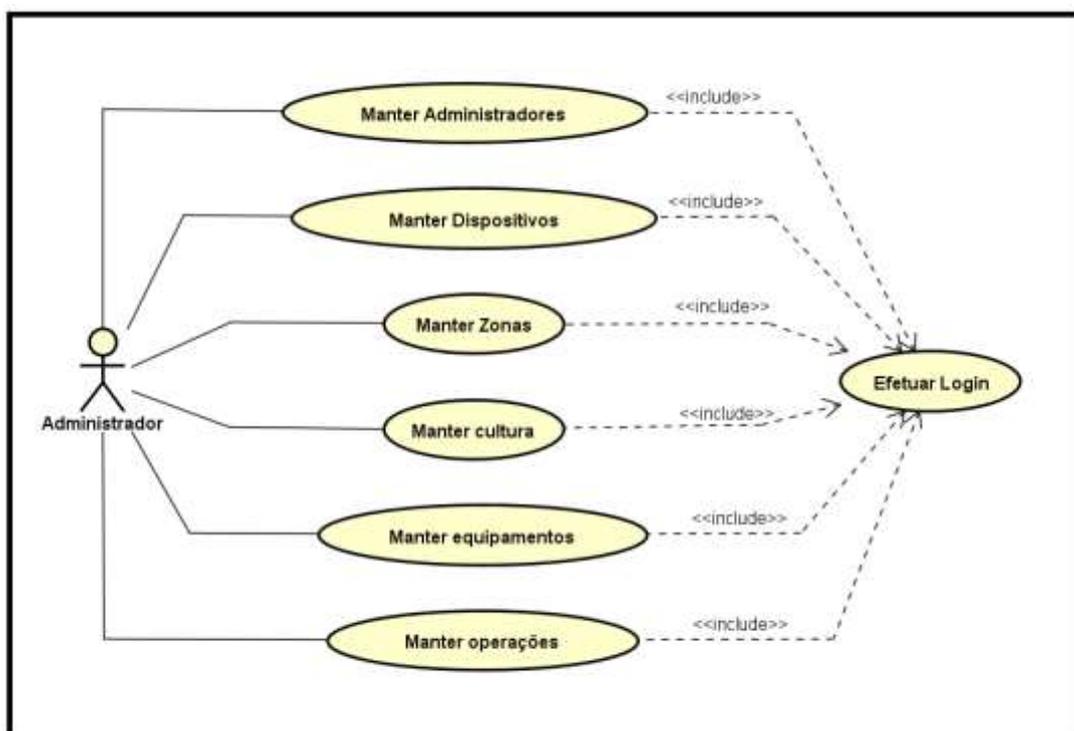


Figura 11 - Diagrama de Casos de Uso – Administrador

A entidade usuário é capaz de visualizar as informações da região na qual seu dispositivo foi detectado, além de poder interagir com o sistema, solicitando a realização de algumas operações, tais como solicitar plantio, solicitar colheita e solicitar aplicação de produtos. A figura 12 representa o diagrama de casos de uso para o usuário.

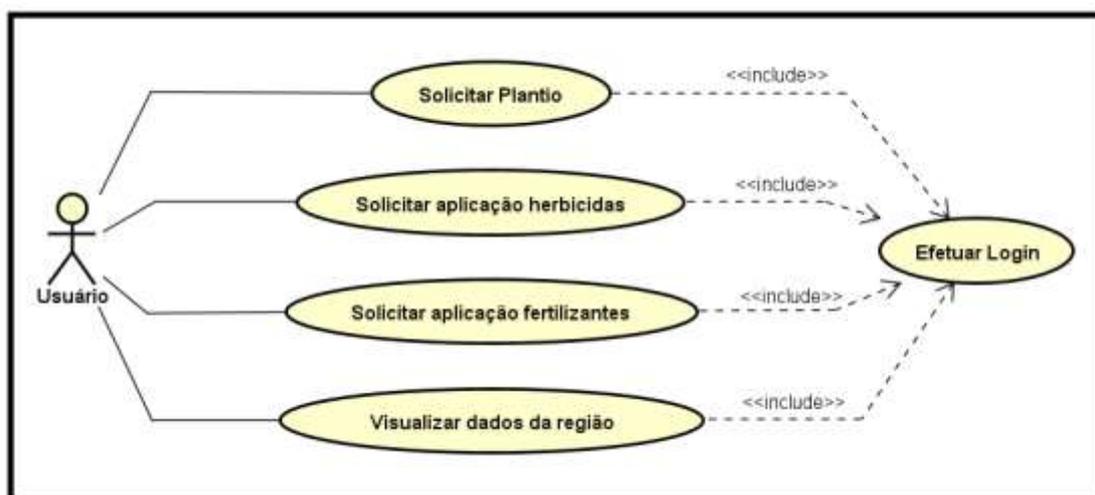


Figura 12 - Diagrama de Casos de Uso - Usuário

Para que qualquer atividade do sistema seja efetivada é necessário que a entidade se autentique no sistema através de um login e senha, assegurando assim a veracidade das informações e restringindo o acesso às pessoas que possuem devida autorização.

6.3.2 AMBIENTE FÍSICO

O ambiente físico é composto por uma placa *Arduíno UNO* e por um módulo de *Bluetooth* responsáveis por gerenciar toda a detecção dos dispositivos, bem como transferir as informações para a aplicação *mobile*. Se encontra no ambiente também o servidor de banco de dados, usado como repositório das informações referentes aos dispositivos registrados como autorizados, bem como as informações que serão visualizadas e manipuladas pela entidade usuário. A figura 13 apresenta parte do código utilizado para responder à uma solicitação da aplicação *mobile*.

```
comando = "";
while(blueetooth.available()){
    char caractere = (blueetooth.read());
    comando += caractere;

    delay(100);
}

if(comando.indexOf("355691062190266"){
    blueetooth.print("");
    blueetooth.print("'code': '1', 'descricao': 'aprovado'");
    blueetooth.print("");
}else{
    blueetooth.print("");
    blueetooth.print("'code': '2', 'descricao': 'não aprovado'");
    blueetooth.print("");
}
}
```

Figura 13 – Código para validar o IMEI do dispositivo.

6.3.3 APLICAÇÃO CLIENTE

O foco principal do projeto será a implementação da arquitetura proposta neste trabalho, utilizando Arduino juntamente com BLE e os conceitos de Beacon e RSSI. Sendo assim, algumas funcionalidades descritas serão utilizadas apenas como complemento e não como ponto principal do projeto.

6.3.3.1 APLICAÇÃO DO USUÁRIO

A aplicação do usuário envolve a implementação de uma interface a fim de permitir a interação da entidade usuário com o ambiente físico do sistema. Essa aplicação consiste em um aplicativo *Mobile* na plataforma *Google Android*, também implementada com Java, comunicando-se com o ambiente físico através da comunicação Bluetooth para o recebimento das informações. A figura 14 demonstra como o protótipo da aplicação do usuário.



Figura 14 – Protótipo da interface da aplicação do usuário

Inicialmente, a aplicação móvel faz uma requisição ao servidor em busca das informações sobre os beacons (qual a localização do beacon, a qual fazenda o mesmo pertence).

Para tornar a sincronização entre o aplicativo e o servidor Web simplificada, foram utilizadas algumas API não nativas do *Android*, tais como *Retrofit*, *OkHttp*, responsáveis por facilitar a recuperação de dados através de requisições HTTP em servidores baseados na arquitetura REST (SQUARE, 2017). Outra API utilizada foi a GSON desenvolvida pela empresa *Google*, responsável por converter objetos Java em sua representação JSON, bem como converter uma representação JSON para um objeto Java (GOOGLE DEVELOPERS, 2017).

Para a implementação da API do *Retrofit* e *OkHttp* foi necessário utilizar uma classe responsável por conter as configurações da API, bem como o endereço base para a conexão com o servidor e também o conversor padrão a ser utilizado pela API. A figura 15

representa a classe que contém as informações básicas para a implementação das API acima citadas.

```
public class ServiceGenerator {

    public static final String API_BASE_URL = "http://192.168.0.105/projetotccweb/index.php/";
    private static final String STANDARD_FORMS1 = "yyyy-MM-dd'T'HH:mm:ss.SSSZ";
    private static final String STANDARD_FORMS2 = "yyyy-MM-dd'T'HH:mm:ss.SSS'Z'";
    private static final String STANDARD_FORMS3 = "EEE, dd MMM yyyy HH:mm:ss zzz";
    private static final String STANDARD_FORMS4 = "yyyy-MM-dd";

    private static OkHttpClient.Builder httpClient = new OkHttpClient.Builder();

    static final Gson gson = new GsonBuilder()
        .setDateFormat(STANDARD_FORMS1)
        .create();

    private static Retrofit.Builder builder =
        new Retrofit.Builder()
            .baseUrl(API_BASE_URL)
            .addConverterFactory(GsonConverterFactory.create(gson));

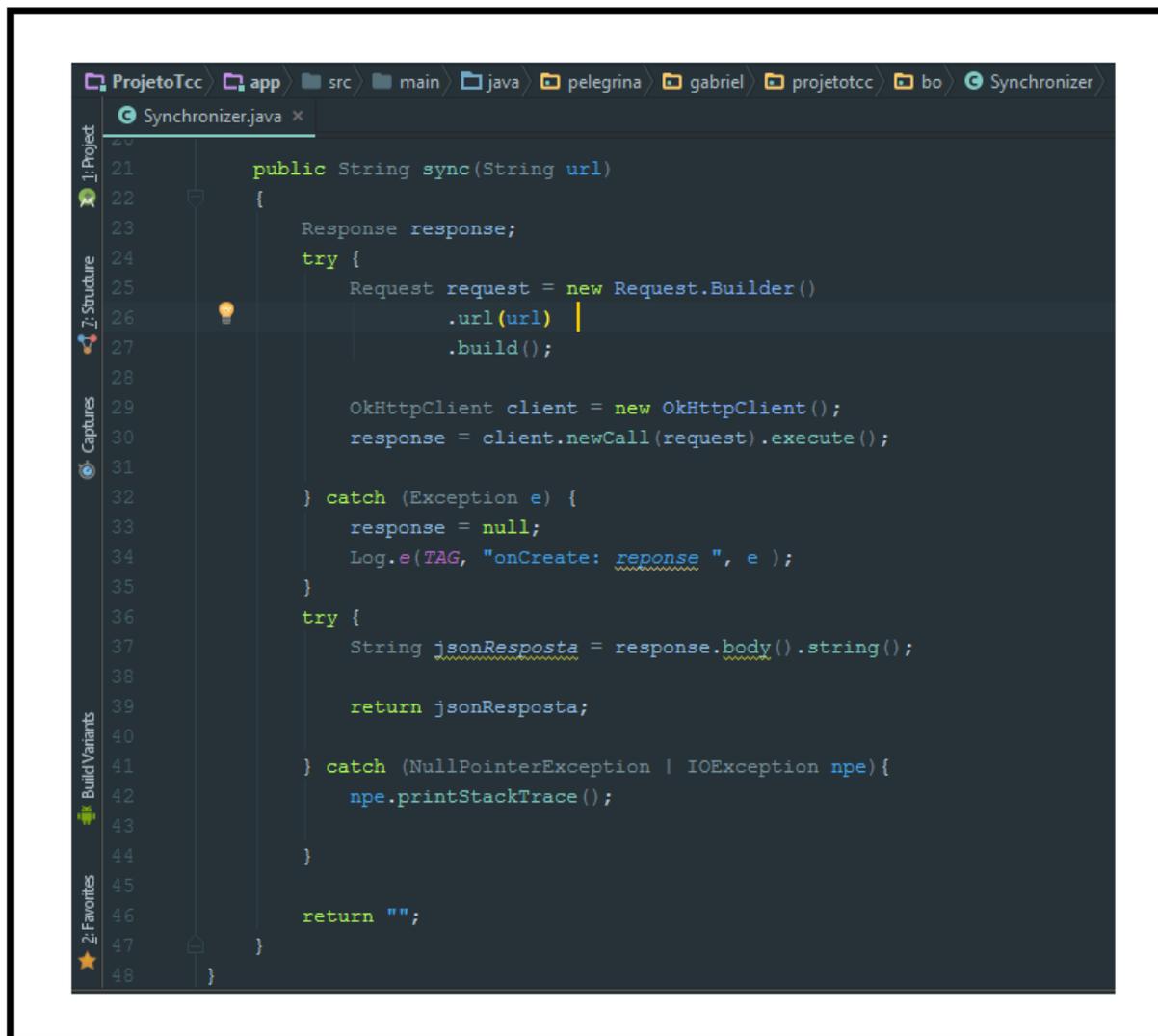
    public static <S> S createService(Class<S> serviceClass) {
        Retrofit retrofit = builder.client(httpClient.build()).build();
        return retrofit.create(serviceClass);
    }

}
```

Figura 15 – Configurações básicas da API Retrofit

Para o recebimento de informações do servidor na aplicação *mobile* foi necessário criar um objeto *Request* passando a URL a qual será requisitada as informações. Em seguida, através do *OkHttpClient* é possível criar uma nova chamada HTTP. Após receber a resposta do servidor é possível recuperar o corpo da solicitação, utilizando o comando *response.body.string()*.

A figura 16 demonstra o código utilizado para o recebimento de informações do servidor na aplicação *mobile* por meio do *OkHttp*.

The image shows a screenshot of an IDE window titled 'Synchronizer.java'. The code is in Java and implements a method 'sync' that takes a 'String url' as input. It uses OkHttp to send a request to the specified URL. The code includes a try-catch block to handle exceptions. If the request is successful, it returns the response body as a string. If there is a NullPointerException or IOException, it prints the stack trace. The code is as follows:

```
21 public String sync(String url)
22 {
23     Response response;
24     try {
25         Request request = new Request.Builder()
26             .url(url)
27             .build();
28
29         OkHttpClient client = new OkHttpClient();
30         response = client.newCall(request).execute();
31
32     } catch (Exception e) {
33         response = null;
34         Log.e(TAG, "onCreate: reponse ", e );
35     }
36
37     try {
38         String jsonResposta = response.body().string();
39
40         return jsonResposta;
41     } catch (NullPointerException | IOException npe){
42         npe.printStackTrace();
43     }
44
45     return "";
46 }
47
48 }
```

Figura 16 – Requisição dos dados do servidor através da API OkHttp.

Em seguida, após o recebimento das informações através da requisição HTTP é necessário converter a representação JSON para um objeto Java com intuito de manipulá-lo e persisti-lo na base de dados. Tal processo é realizado pelo conversor *Gson*, que após receber a *String* de resposta do servidor transforma a mesma em um objeto *JSONArray*, contendo cada objeto, representado no formato de JSON. Para a conversão de cada elemento, é utilizado o método *fromJson* da classe *Gson*, passando como parâmetros o elemento e a Classe a qual esse elemento será convertido. A figura 17 demonstra o código utilizado para a conversão da representação JSON e a inserção dos dados na base de dados.

```

45     @Override
46     protected String doInBackground(ParamSincronizacao... params) {
47         this.clz = params[0].getClz();
48
49         return new Synchronizer().sync(params[0].getUrl());
50     }
51
52     @Override
53     protected void onPostExecute(String json) {
54
55         JSONArray trade = new JsonParser().parse(json).getAsJSONArray();
56
57         try {
58             List<Object> lista = new ArrayList<>();
59             if (trade.size() > 0) {
60                 for (JsonElement el : trade) {
61                     Object obj = g.fromJson(el, this.clz);
62                     try {
63                         bo.save((ValueObject) (obj));
64                     } catch (SQLException | ClassCastException se) {
65                         Log.e(TAG, "onPostExecute: error Save " + clz.getSimpleName(), se);
66                     }
67                 }
68             }
69         } catch (JsonSyntaxException e) {
70             e.printStackTrace();
71         }
72     }

```

Figura 17 – Conversão de JSON para classe JAVA.

Após as etapas descritas acima a aplicação mobile faz uma busca por dispositivos BLE próximos. Caso ocorra a detecção, um pacote com as informações do smartphone é enviado para o beacon, afim de validar a permissão do dispositivo. O beacon por sua vez, valida os dados recebidos do smartphone e retorna um pacote com o resultado da validação. Ao ser validado, o smartphone então exibe as informações referentes ao local onde o beacon foi detectado, por exemplo, se o beacon se encontra na Fazenda A, informações sobre a Fazenda A serão exibidas ao usuário. Caso não ocorra a validação, o usuário é redirecionado para a tela inicial da aplicação, afim de garantir a segurança da aplicação *mobile*. As figuras 18 e 19 representam os métodos utilizados para a detecção do beacon, bem como os métodos criados para o envio e recebimento de pacotes entre a aplicação *mobile* e o beacon.

```

ProjetoTcc app src main java pelegrina gabriel projetotcc bluetooth ComunicadorBluetoothThread
ComunicadorBluetoothThread.java x
67 public void run() {
68     mmmBuffer = new byte[1024];
69     int numBytes;
70     while (true) {
71         try {
72             numBytes = mmmInStream.read(mmmBuffer);
73
74             String dados = new String(mmmBuffer, 0, numBytes);
75
76             mHandler.obtainMessage(MESSAGE_READ, numBytes, -1, dados).sendToTarget();
77
78         } catch (IOException e) {
79             Log.d(TAG, "Input stream was disconnected", e);
80             break;
81         }
82     }
83 }
84
85 public void enviar(String output) {
86     try {
87
88         byte[] mensagem = output.getBytes();
89         mmmOutputStream.write(mensagem);
90
91     } catch (IOException e) {
92         Log.e(TAG, "Erro ao enviar para o arduino", e);
93     }
94 }

```

Figura 18 – Métodos utilizados para receber e enviar mensagens para o beacon

```

@Override
public void onReceive(Context context, Intent intent) {
    String action = intent.getAction();

    if (BluetoothDevice.ACTION_FOUND.equals(action)) {
        String name = intent.getStringExtra(BluetoothDevice.EXTRA_NAME);
        BluetoothDevice bluetoothDevice = intent.getParcelableExtra(BluetoothDevice.EXTRA_DEVICE);

        int rssi = intent.getShortExtra(BluetoothDevice.EXTRA_RSSI, Short.MIN_VALUE);

        double distance = getDistance(rssi);

        if (distance > CINCO_METROS) {
            startActivity(new Intent(context, ExibeInformacaoFazenda.class));
        }
    }
}
}

```

Figura 19 – Código para a detecção e captura de RSSI do beacon encontrado.

Por fim, ao realizar uma operação, a aplicação salva as informações preenchidas pelo usuário no banco de dados embarcado do celular. Assim, ao retornar ao escritório, a aplicação móvel é integrada com a aplicação Web, atualizando os dados do servidor com os dados capturados pela visita no campo. Para tal processo, mais uma vez foi utilizado a API do *Retrofit*. Após o usuário realizar a submissão do formulário, a aplicação mobile tentará submeter os dados ao servidor. Para isso, foi necessário criar uma interface *EndPoint*, a qual ficou responsável por definir o método HTTP da requisição, o endereço final da (o endereço base foi definido na classe de configuração do *Retrofit*), bem como o corpo da requisição. A figura 20 demonstra o código da interface *EndPoint* descrita acima.



```
EndPoint.java x
EndPoint
1 package pelegrina.gabriel.projetoctcmobile.interfaces;
2
3 import ...
10
11 /**...*/
14
15 public interface EndPoint {
16
17     @POST("operacoes/insert")
18     Call<ResponseBody> createService(@Body List<OperacaoVO> list);
19 }
```

Figura 20 – Interface Endpoint.

Além disso também foi necessário criar uma classe responsável por realizar a chamada HTTP, que implementa a interface *CallBack* do *Retrofit* responsável por gerenciar os retornos do servidor. Caso o servidor responda à requisição da chamada é possível manipular a resposta sobrescrevendo o método *onResponse* do callback. Caso o servidor não responda à requisição da chamada é possível manipular a falha por meio do método *onFailure*. As figuras 21 e 21 demonstram os códigos em caso de sucesso ou falha na comunicação com o servidor.

```

FragmentRealizarOperacao.java x Sender.java x
Sender CallbackOperacao onResponse()
63 @Override
64 public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {
65     if (response == null || !response.isSuccessful()) {
66         notResponse();
67         return;
68     }
69     String sBody = getStringBody(response);
70     if (sBody == null || sBody.isEmpty()) {
71         notStringBody();
72         return;
73     }
74     List<OperacaoVO> listaResposta = getConvertedListResponse(sBody);
75
76     if (listaResposta == null || listaResposta.isEmpty()) {
77         nullList();
78         return;
79     }
80     boolean allSend = saveAllFromServer(listaResposta);
81
82     String title = "Aviso";
83     String content = "Erro ao enviar alguma(s) operação(ões)!";
84
85     if (allSend) {
86         content = "Apontamentos Enviados!";
87     }
88     showDialogSuccess(title, content);
89 }

```

Figura 21 – Método onResponse do callback.

```

FragmentRealizarOperacao.java x Sender.java x
Sender CallbackOperacao onFailure()
91 @Override
92 public void onFailure(Call<ResponseBody> call, Throwable t) {
93     String title = "Conexão com servidor!";
94     String content = "Erro ao conectar com o servidor. As Operações não foram enviadas!";
95
96     Log.e(TAG, "onFailure: ", t);
97     showDialogError(title, content);
98 }

```

Figura 22 – Método onFailure do callback.

6.3.3.2 APLICAÇÃO DO ADMINISTRADOR

A aplicação do administrador envolve a implementação de uma interface com intuito de facilitar o gerenciamento de informações necessárias para o funcionamento do sistema. A figura 19 demonstra como o protótipo da aplicação do administrador.

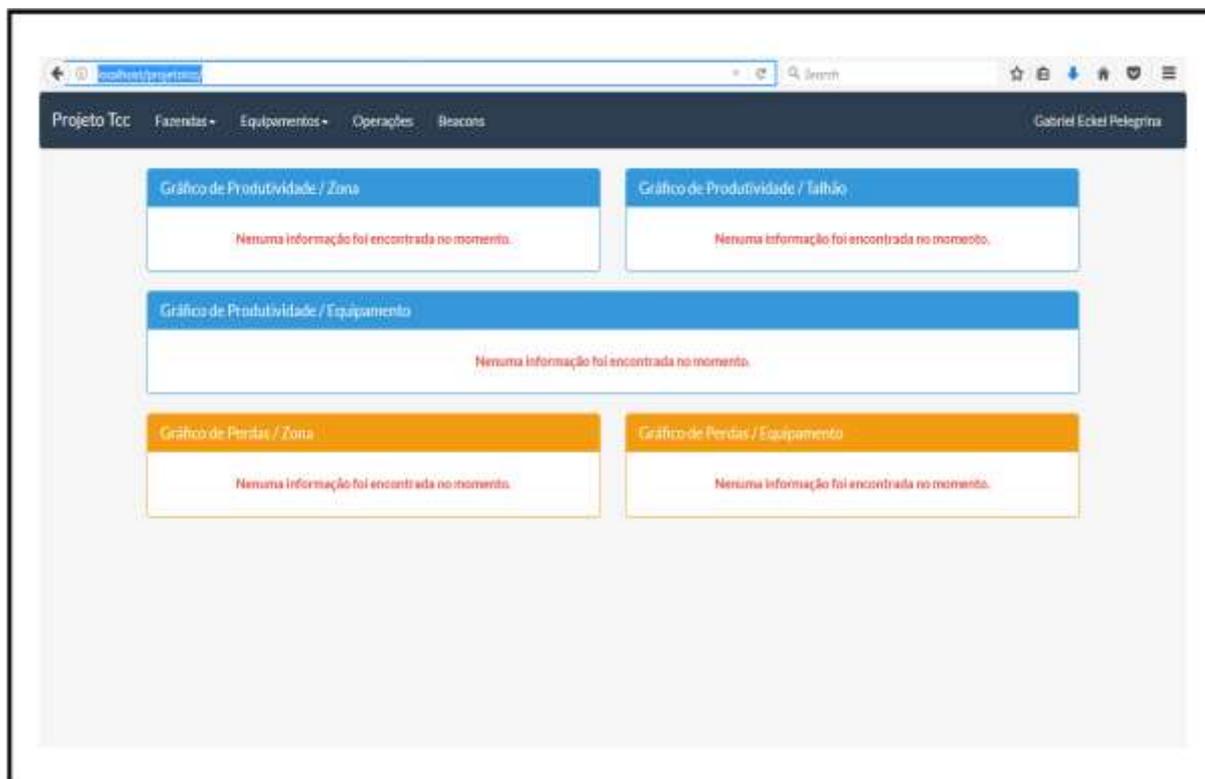


Figura 19 – Protótipo da interface da aplicação do administrador

Essa interface consiste em uma aplicação *Web* desenvolvida na linguagem PHP com o auxílio do framework CodeIgniter que utiliza o padrão MVC para o desenvolvimento de software, permitindo a criação de novas entidades de forma rápida e bem estruturada. Nesse padrão é encontrado os controladores, responsáveis por receber os dados dos usuários e gerenciar qual ação deve ser tomada pela aplicação (GABARDO, 2015). Para a comunicação entre o controller e a aplicação mobile, foi criado dois métodos na classe Controller. O primeiro, chamado de *service()*, ficou responsável por retornar uma representação JSON dos objetos salvos no servidor. Para isso foi carregado a lista do banco de dados e em seguida, por meio do método *json_encode* foi gerada a representação JSON a ser enviada à aplicação mobile. O segundo método, chamado de *insert()* ficou responsável por receber a requisição da aplicação mobile por meio do método POST do HTTP, decodificar o JSON para objeto e então persistir no banco de dados do servidor web. A figura 23 representa os métodos *service()* e *insert()* do Controller responsável pelas operações da aplicação.

```

11 public function service(){
12     $this->load->model("operacao_model");
13     $operacoes = $this->operacao_model->lista();
14
15     $data['operacoes'] = json_encode($operacoes, JSON_PRETTY_PRINT | JSON_UNESCAPED_UNICODE);
16 }
17
18 public function insert()
19 {
20     $this->load->model('operacao_model');
21     $this->load->model('status_model');
22
23     $post = file_get_contents($_SERVER['REQUEST_URI']);
24
25     $operacoes = json_decode($post, true);
26
27     $operacoes_atualizadas = array();
28
29     foreach ($operacoes as $op) :
30         $op['it_status'] = $this->status_model->status_enviado();
31
32         $this->operacao_model->salva($op);
33
34         array_push($operacoes_atualizadas, $op);
35     endforeach;
36 }

```

Figura 23- Métodos Service e Insert do Controller Operações

Além das classes Controllers também são encontradas as classes Models ou Modelos e as Views ou visualizações. As classes modelos ficam responsáveis por toda a abstração dos dados no banco de dados da aplicação. Já as Views são responsáveis por criar as interfaces gráficas que serão exibidas aos usuários (GABARDO, 2015). Na classe modelo das operações foi criado o método listar() que retorna os dados da base de dados. Já para a view, foi criado o arquivo service.php que retorna esses dados representados por um JSON, a ser recebido pela aplicação mobile. As figuras 24 e 25 demonstram um exemplo do código de uma classe model e da view service.php das operações.

```
operacao_model.php x
\operacao_model
9 class operacao_model extends CI_Model
10 {
11     public function lista()
12     {
13         $sql = "SELECT *
14             ,o.lat_centro lat_centro
15             ,o.lng_centro lng_centro
16             FROM modelo m
17             ,cultura c
18             ,bloco b
19             ,zona z
20             ,equipamento e
21             ,talhao t
22             ,tipo_operacao tpo
23             ,operacao o
24             WHERE o.fk_tipo_operacao = tpo.cd_tipo_operacao
25             AND o.fk_talhao = t.cd_talhao
26             AND o.fk_equipamento = e.cd_equipamento
27             AND t.fk_zona = z.cd_zona
28             AND z.fk_bloco = b.cd_bloco
29             AND t.fk_cultura = c.cd_cultura
30             AND e.fk_modelo = m.cd_modelo";
31         return $this->db->query($sql)->result_array();
32     }
}
```

Figura 24 – Demonstração do método listar da classe model das operações.

```
service.php x
1 <?= $operacoes ?>
2
```

Figura 25- Demonstração da view Service.php das operações.

7. CONCLUSÃO

O interesse pelo tema do trabalho se deve ao fato do grande avanço tecnológico na área de *Internet of Things*, proposta que vem sendo aceita entre pessoas e grupos corporativos, bem como empresas do setor agrícola, possibilitando ideias inovadoras para disponibilizar informações e melhorias em atividades realizadas no cotidiano.

A proposta do projeto foi a de desenvolver os estudos teóricos, com a finalidade de adquirir os conhecimentos necessários sobre a arquitetura proposta, bem como a sua implantação por meio de uma comunicação via *Bluetooth*. Juntamente com a pesquisa foi elaborado um experimento que permitiu aplicar os conhecimentos obtidos em um ambiente baseado na comunicação *Bluetooth Low Energy*, bem como a implementação de uma aplicação na plataforma *Google Android*, para interagir com o dispositivo *Bluetooth* e com o usuário final.

Como proposta final deste projeto, foi desenvolvido um ambiente de comunicação baseado na tecnologia *Bluetooth Low Energy*, bem como no RSSI, responsáveis por determinar a localização e a distância entre o *beacon* e o dispositivo móvel. Com a aplicação *Mobile* tornou-se possível validar o beacon localizado e assim exibir as informações diretamente na tela do *smartphone*. Como complemento, foi desenvolvido uma aplicação Web responsável pela parte lógica do negócio de uma fazenda, expor as informações que serão consumidas pela aplicação *Mobile*, bem como oferecer uma visualização gráfica das operações realizadas em determinada região.

7.1. TRABALHOS FUTUROS

A partir da arquitetura proposta é possível dar continuidade no desenvolvimento, complementando com novas funcionalidades a fim de possibilitar maior flexibilidade aos usuários. Além disso, pretende-se adicionar novos sensores à arquitetura, com intuito de aumentar a quantidade de informações visualizadas pelo usuário, bem como tomar decisões automatizadas, sem a necessidade de intervenção humana. Pretende-se também utilizar a arquitetura proposta em outros contextos.

REFERÊNCIAS

ABLESON, W. F.; COLLINS, Charlie; SEN, Robi. **Unlocking Android**. A Developer's Guide. 1. ed. Greenwich: Manning, 2009.

ALLIANCE, Open H. **Overview**. Disponível em: <http://www.openhandsetalliance.com/oha_overview.html>. Acesso em 22 jan. 2017.

ANDROID, Source. **System and Kernel security**. Disponível em: <<https://source.android.com/>> Acesso em 22 jan. 2017

ALMEIDA, Hygoo. Tudo Conectado. In: **Revista da Sociedade Brasileira de Computação**, No. 29, Abril, 2015.

AMADO, T.J.C.; GIOTTO, E. **A sua lavoura na tela**. A Granja, São Paulo, v 732, p.38-42, dez. 2009.

BERGAMASCHI, Eloisio A. **Um modelo de arquitetura para a próxima geração de aplicações e serviços de Realidade Aumentada**. 2010. Monografia de Conclusão de Curso – Universidade Federal do Rio Grande do Sul, Rio Grande do Sul, Porto Alegre, 2010.

BLUETOOTH SIG. **Discover Bluetooth**. 2017. Disponível em: <<https://www.bluetooth.com/>>. Acesso em: 22 fev. 2017.

CAMAROTTO, Rodolfo Santos. **Internet das coisas: De onde vêm e para onde vai?**. 2010, 63p. Monografia de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis, São Paulo, Assis, 2014.

COETZEE, L.; EKSTEEN, J.; "The Internet of Things-promise for the future: An introduction", In: **IST-Africa Conference Proceedings**, 2011, Maio 2011.

CNA – CONFEDERAÇÃO DA AGRICULTURA E PECUÁRIA. **Agronegócio deve ter crescimento de 2% em 2017**. 2016. Disponível em: <<http://www.brasil.gov.br/economia-e-emprego/2016/12/agronegocio-deve-ter-crescimento-de-2-em-2017>>. Acesso em: 02 mar. 2017

COSTA, Lucas J. S. **Técnica de localização em ambientes fechados utilizando padrões de redes sem fio**. 2015. Dissertação. Escola Politécnica da Universidade de São Paulo. São Paulo. São Paulo, 2015.

CSÓTÓ, M. Information flow in agriculture-through new channels for improved effectiveness. **Journal of Agricultural Informatics**. 2010 Vol. 1, No. 2. 2010.

DUROC, Yvan; KADDOUR, Darine. RFID potential impacts and future evolution for Green projects. **Energy Procedia**. 18, 2012, Valance.

ESTIMOTE. **Create magical experiences in the physical world**. Disponível em: <http://estimote.com>. 2017. Acesso em: 16 mar. 2017.

FARTO, Guilherme de Cleva. Abordagem Orientada a Serviços para Implementação de um Aplicativo Google Android. 2010, 83p. Monografia de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis, São Paulo, Assis, 2010.

FAO - FOOD AND AGRICULTURE ORGANIZATION OF THE UNITED NATIONS. Our Food and Agriculture in Numbers. Disponível em: <http://www.fao.org/assets/infographics/FAO-Infographic-food-ag-en.pdf>. Acesso em: 03 mar. 2017.

FERREIRA, Natália. **O agronegócio no Brasil**. 2009. Disponível em: <http://www.administradores.com.br/artigos/economia-e-financas/o-agronegocio-no-brasil/32808/>. Acesso em: 02 mar. 2017.

FREIBERGER, A.; BEZERRA, M. B. P. **RFID e Seus Impactos na Logística**. Março, 2010. Disponível em: <http://www.logisticadescomplicada.com/rfid-e-seus-impactos-na-logistica/>. Acesso em 22 fev. 2017.

FORBES. **The Future Of Agriculture? Smart Farming**. 2015. Disponível em: <http://www.forbes.com/sites/federicoguerrini/2015/02/18/the-future-of-agriculture-smart-farming/#9790d62337cb>. Acesso em 25 out. 2016.

GOOGLE DEVELOPERS. **JSON - Pluggable streaming library**. 2017. Disponível em: <https://developers.google.com/api-client-library/java/google-http-java-client/json>. Acesso em: 27 jul. 2017.

GORDILHO, J. O. **O guia completo para Near Field Communication (NFC): como funciona, o que faz e muito mais**. 2012. Disponível em: <http://www.showmetech.com.br/guia-completo-sobre-nfc/#ixzz4LUrw6NLk>. Acesso em: 20 fev. 2017.

HYDRONAV. **Trimble Connected Farm**. 2017. Disponível em: <http://www.hydronav.com>. Acesso em 8 mar. 2017.

IBGE – INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Censo agropecuário de 2006**. 2007. Disponível em: www.ibge.gov.br. Acesso: 22 fev. 2017.

IBGE – INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **PIB do agronegócio cresceu 1,8% em 2015**. 2016. Disponível em: <http://www.brasil.gov.br/economia-e-emprego/2016/03/pib-do-agronegocio-cresceu-1-8-em-2015>. Acesso em 02 mar. 2017.

KONTAKT. **What is beacon?**. 2017. Disponível em: <https://kontakt.io/beacon-basics/what-is-a-beacon/>. Acesso em 16 mar. 2017.

KALOXYLOS, A.; EIGENMANN, R.; TEYE, F.; POLITOPOULOU, Z.; WOLFERT, S.; SHRANK, C.; DILLINGER, M.; LAMPROPOULOU, I.; ANTONIOU, E.; PESONEN, L.;

LEAL, Leonan J. P.. **Desenvolvimento de aplicativos para dispositivos móveis para coleta de dados georreferenciados através de reconhecimento de voz**. 2013. 81. Dissertação. Escola Superior de Agricultura “Luiz de Queiroz”. Universidade de São Paulo. Piracicaba. São Paulo, 2013.

LECHETA, Ricardo R. **Google Android – Aprenda a criar aplicações para dispositivos móveis com o Android SDK**. 5. ed. São Paulo: Novatec, 2015.

MENEGOTTO, José Luis. Sensoriamento da edificação: Um sistema de localização baseado em beacons BLE. In: **VII Encontro de Tecnologia de Informação e Comunicação na Construção**, Recife, Brasil 2015. Edificações, Infraestrutura e Cidade: Do BIM ao CIM, novembro, 2015.

MINISTÉRIO DA AGRICULTURA, PECUÁRIA E ABASTECIMENTO. **Agronegócio Brasileiro em números**. 2010. Disponível em: <<http://www.agricultura.gov.br>>. Acesso em: 20 out. 2016.

MOATAMED, Babak; ARJUN; SHAHMOHAMMADI, Farhad; RAMEZANI, Ramin; NAEIM, Arash; SARRAFZADEH, Majid; Low-cost Indoor Health Monitoring System. 2016 In: **Wearable and Implantable Body Sensor Networks (BSN), 2016 IEEE 13th International Conference**. 2016.

MOHAMMED, Badr A. M. **Hamad International Airport launches mobile app with iBeacon features**. 2016. Disponível em: <http://www.passengerselfservice.com/2016/03/hamad/>. Acesso em: 15 mar. 2017.

PIVOTO, Dieisson; MORAES, Giana. V; SILVA, Roberto F.; KAWANO, Bruno R.; TALAMINI, Edson. **Smart Farming pode trazer ganhos ao agronegócio?**. 2016. Disponível em: <https://www.researchgate.net/publication/299616595_Smart_farming_pode_trazer_ganhos_ao_agronegocio>. Acesso em: 16 mar. 2017.

PEREIRA, Lúcio C. O.; SILVA, Michel L. S. **Android para Desenvolvedores**. 1. ed. Rio de Janeiro: Brasport Livros e Multimídia Ltda., 2009.

POWELL, Jennifer K. **Take A Look At How Technology Makes Smart And Sustainable Farming**. 2016. Disponível em: <<https://www.forbes.com/sites/jenniferhicks/2016/12/31/take-a-look-at-how-technology-makes-smart-and-sustainable-farming/#7d659653deb2>>. Acesso em: 02 mar. 2017.

PRESA, Eduardo. **Proposta de um sistema de localização interna para o ambiente universitário**. 2014, 104p. Monografia de Conclusão de Curso – Universidade Federal de Santa Catarina, Santa Catarina, Araranguá, 2014.

RAZA, Shahid; MISRA, Prasant; HE, Zhitao; VOIGT, Thiemo. Bluetooth Smart: An Enabling Technology for the Internet of Things. In: **Eight International Workshop on Selected Topics in Mobile and Wireless Computing**, 2015, Bangalore, India, 2015. p.155-162.

ROCHA, C. **CES 2015: o que é a internet das coisas e como ela entrará na sua vida**. 2015. Disponível em: <http://blogs.estadao.com.br/homem-objeto/o-que-e-a-internet-das-coisas/>. Acesso em: 18 fev. 2017.

RURALTINS. **Agroindústria**. Disponível em: <http://ruraltins.to.gov.br/agroindustria>. Acesso em: 22 fev. 2017.

SILVÉRIO, Lucas A. F. **PLUGINS DE APOIO PARA GIS COM BASE EM INTERNET OF THINGS E SMART AGRICULTURE**. 2016, 54p. Monografia de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis, São Paulo, Assis, 2016.

SMART CARD ALLIANCE MOBILE AND NFC COUNCIL. **Bluetooth Low Energy (BLE) 101: A technology Primer with Example Use Cases**. 2014. Disponível em: <http://www.smartcardalliance.org/publications-bluetooth-low-energy-ble-101-a-technology-primer-with-example-use-cases>. Acesso em 31 out. 2016.

SPINSANTE, Susanna; GAMBI, Ennio. **NFC-Based User Interface for Smart Environments**. 2015. 12p. Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Ancona, Italy. 2015.

SQUARE; **An HTTP & HTTP/2 client for Android and Java applications**. 2017. Disponível em: <http://square.github.io/okhttp/>. Acesso em: 28 jul. 2017.

STAMFORD, Conn. **Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020**. 2013. Disponível em: <http://www.gartner.com/newsroom/id/2636073>. Acesso em: 10 fev. 2017.

STECOM. **Internet of Things and Big Data Collection Smart Farm/Smart Agriculture**. Disponível em: http://www.stecom.com/Case_history/Stecom_Case_history_Smart_agriculture.html. Acesso em: 8 mar. 2017.

TANG, S.; ZHU, Q.; ZHOU, X. A conception of digital agriculture. In: **Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS)**, 2002.

TELECO, Inteligência em Telecomunicações. **Estatísticas de Celular no Mundo**. Disponível em: <http://www.teleco.com.br/pais/celular.asp>. Acesso em: 23 jan. 2017.

TONIN, Graziela Simone. **Tendências em Computação Móvel**. 2012. 17p. Departamento de Ciência da Computação. Universidade de São Paulo. São Paulo. São Paulo, 2012.

TOWNSED, Kevin; CUFÍ, Carles; DAVIDSON, Akiba & Robert. **Getting start with Bluetooth Low Energy**. 1 ed. California: O'Reilly, 2014.

UN, News Centre. **UN projects world population to reach 8.5 billion by 2030, driven by growth in developing countries**. 2015. Disponível em: <http://www.un.org/apps/news/story.asp?NewsID=51526#.WKuI_vkrK01>. Acesso em 23 jan. 2017.

VASSEUR, J .P; DUNKELS, A. **Interconnecting Smart Objects with IP – The Next Internet**. 1.ed. Morgan Kaufmann. 2010.

ZAMBARDA, Pedro. **'Internet das Coisas': entenda o conceito e o que muda com a tecnologia**. 2014. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2014/08/internet-das-coisas-entenda-o-conceito-e-o-que-muda-com-tecnologia.html>>. Acesso em: 20 fev. 2017.

ZANELLA, A.;BUI, N. CASTELLANI, A; VANGELISTA, L.; ZORZI, M. Internet of Things for smart cities. **IEEE Internet of Things Journal**, Vol. 1 No. 1, 2014.

ZHU. Qian. IOT Gateway: BridgingWireless Sensor Networks into Internet of Things. In: **Embedded and Ubiquitous Computing (EUC)**, 8, 2010 Hong Kong.