



ESTADO DE MATO GROSSO
SECRETARIA DE ESTADO DE CIÊNCIA, TECNOLOGIA E
INOVAÇÃO
UNIVERSIDADE DO ESTADO DE MATO GROSSO
FACULDADE DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CAMPUS UNIVERSITÁRIO DE BARRA DO BUGRES
CURSO DE CIÊNCIA DA COMPUTAÇÃO



LUAN FELIPE DOS SANTOS DA SILVA

**DESENVOLVIMENTO DE UM GPS AGRÍCOLA UTILIZANDO
APLICATIVO ANDROID EM CONJUNTO COM UM SISTEMA EMBARCADO**

BARRA DO BUGRES – MT

2018

LUAN FELIPE DOS SANTOS DA SILVA

**DESENVOLVIMENTO DE UM GPS AGRÍCOLA UTILIZANDO
APLICATIVO ANDROID EM CONJUNTO COM UM SISTEMA EMBARCADO**

**Trabalho de conclusão de curso apresentado
ao Curso de Ciência da Computação da
Universidade do Estado de Mato Grosso –
UNEMAT, como requisito para obtenção do
título de Bacharel em Ciência da
Computação, sob orientação do Professor
Me. Armando da Silva Filho.**

BARRA DO BUGRES – MT

2018

LUAN FELIPE DOS SANTOS DA SILVA

**DESENVOLVIMENTO DE UM GPS AGRÍCOLA UTILIZANDO
APLICATIVO ANDROID EM CONJUNTO COM UM SISTEMA EMBARCADO**

Banca Examinadora

Prof. Me. Armando da Silva Filho
Orientador

Prof. Dr. Fernando Selleri da Silva
Convidado

Prof. Me. Alexandre Bernt
Indicado

Barra do Bugres, 11 de Julho 2018

DEDICATÓRIA

Dedico esse trabalho aos meus pais Jean e Valdirene por estarem sempre ao meu lado e me proporcionarem, através do seu suor, a oportunidade de estudar.

Dedico também a minha irmã Luana por ser exemplo de dedicação e ter me ensinado desde muito nova a sempre ir em busca dos seus sonhos.

Amo vocês!

AGRADECIMENTOS

Agradeço a Deus pela força, sabedoria e paciência para realização dessa etapa em minha vida.

A minha família por todo apoio e por sempre me incentivarem a dar o meu melhor em tudo que faço.

Agradeço a minha namorada Sergiana, pela compreensão, por suportar todos os meus momentos de estresse e sempre sonhar grande junto comigo.

Ao meu orientador e mentor Prof. Armando, por todo aprendizado, e apoio que me encorajaram na realização deste projeto. Pelas suas críticas construtivas e sugestões, bem como pela dedicação e total disponibilidade.

Também a todos os professores da UNEMAT por todo conhecimento passado.

Aos colegas do curso Ciência da Computação que tive a oportunidade de compartilhar bons momentos e construir grandes amizades.

Aos amigos que fiz em Barra do Bugres-MT em especial Elaine e Maria pelos conselhos dados e me ajudado em momentos complicados da minha vida.

Muito obrigado a todos.

“Você nunca sabe que resultados virão da sua ação. Mas se você não fizer nada, não existirão resultados”.

(Mahatma Gandhi)

RESUMO

A produção agrícola no país tem aumentando a cada ano, a utilização da agricultura de precisão tem buscado alcançar uma maior produtividade e, ao mesmo tempo reduzir os custos de produção evitando principalmente o desperdício. No entanto, os custos para implantação da mesma são muito altos, dessa forma, o desenvolvimento de aplicativos móveis para *smartphones* se tornam uma alternativa de baixo custo para pequenos produtores que não tenham condições de adquirir os sistemas convencionais. Este trabalho consiste no desenvolvimento de um aplicativo móvel de localização para a plataforma Android, aliado a um sistema embarcado. O aplicativo permite a condução de máquinas agrícolas em retas AB e linhas paralelas enquanto navega no campo, reduzindo a sobreposição e desperdício de insumos agrícolas, além de ao final da atividade permitir a emissão de relatório referente aos hectares trabalhados, distância percorrida e tempo de trabalho. O *hardware* não apresenta complicações na sua utilização sendo necessário apenas a ligação com uma fonte de energia e realizar a conexão com dispositivo via *Bluetooth*.

Palavras-chave: Agricultura de Precisão, GPS, Aplicativos Android, Dispositivos Móveis.

ABSTRACT

Agricultural production in the country has been increasing every year, the use of precision agriculture has sought to achieve greater productivity while reducing production costs avoiding mainly waste. However, the deployment costs are very high, the development of mobile applications for smartphones becomes a low-cost alternative for small producers who cannot afford conventional systems. This work consists of the development of a mobile location application for the Android platform, coupled to an embedded system. The application allows the driving of agricultural machines in AB lines and parallel lines while navigating in the field, reducing the overlap and waste of agricultural inputs, besides the end of the report of activity issued with hectares worked, distance traveled and working time. The hardware does not present complications in its use being necessary only the connection with a power source and make the connection with the device via Bluetooth.

Keywords: Precision Agriculture, GPS, Android Applications, Mobile Devices.

LISTA DE FIGURAS

Figura 1 – Ciclo da Agricultura de Precisão.....	17
Figura 2 – Constelação de Satélites.....	19
Figura 3 – Funcionamento do monitor de direcionamento em formato de barra de luzes somente com <i>leds</i>	21
Figura 4 – Barra de luzes com <i>leds</i> e tela de visualização da operação.	22
Figura 5 – Mercado global de eletrônicos pessoais por produto (milhões de unidades).....	24
Figura 6 – Diagrama da arquitetura da plataforma Android	29
Figura 7 – Interface do aplicativo Navegador de Campo.....	31
Figura 8 – Interface do aplicativo eFarmer	31
Figura 9 – Interface do aplicativo MachineryGuide.....	32
Figura 10 – Diagrama de Atividade	35
Figura 11 – Diagrama de Caso de Uso	36
Figura 12 – Tela Inicial	37
Figura 13 – Tela Principal de Navegação.....	37
Figura 14 – Botões de ação.....	39
Figura 15 – Tela de Relatório	40
Figura 16 – Tela de Preferências da aplicação	41
Figura 17 – Ativação do GPS Interno	44
Figura 18 – Ativação do GPS Externo	44
Figura 19 – Processo de navegação da conexão <i>Bluetooth</i>	45
Figura 20 – Processo de ligação do dispositivo <i>Bluetooth</i>	45
Figura 21 – Conexão <i>Bluetooth</i>	46
Figura 22 – Pinagem PIC18F4550	47
Figura 23 – Placa Tenorino com o GPS e o módulo <i>Bluetooth</i>	48
Figura 24 – Placa Tenorino	49
Figura 25 – Projeto do circuito da placa Tenorino	50
Figura 26 – Projeto do circuito do display	50
Figura 27 – Visão do módulo ME-1000RW	52
Figura 28 – Módulo Bluetooth HC-05	53
Figura 29 – Funcionamento geral do programa do microcontrolador.....	54

LISTA DE QUADROS

Quadro 1 – Versões do Android e distribuição	26
Quadro 2 – Descrição dos itens da tela principal	38
Quadro 3 – Preferências da aplicação Android	42
Quadro 4 – Especificação do módulo ME-1000RW	51
Quadro 5 – Pinos do conector de saída	52
Quadro 6 – Pinos de ligação do módulo HC-05	53
Quadro 7 – Comandos AT do módulo HC-05.....	54

LISTAGEM DOS CÓDIGOS

Listagem 1 – Trecho do código das preferências	42
Listagem 2 – Trecho do código da SettingsActivity	43

LISTA DE TABELAS

Tabela 1: Vendas de smartphones em todo o mundo para usuários finais por sistema operacional em 2017 (milhares de unidades)	25
---	----

LISTA DE SIGLAS E ABREVIATURAS

AP – Agricultura de Precisão

API – *Application Program Interface*

ART – *Android Runtime*

FAB – *Floating Action Button*

GC – *Garbage Collector*

GPS – *Global Positioning System*

HAL - *Hardware Abstraction Layer*

IDE – *Integrated Development Environment*

IU - Interface do Usuário

NAVSTAR-GPS – *Navigation Satellite with Time And Ranging*

OHA – *Open Handsent Alliance*

SIG – Sistema de Informação Geográfica

SMS – *Short Message Service*

SO – Sistema Operacional

Wifi – *Wireless Fidelity*

XML - *Extensible Markup Language*

JDK – *Java Development Kit*

SUMÁRIO

INTRODUÇÃO	15
CAPÍTULO I	16
1 AGRICULTURA DE PRECISÃO E O SISTEMA DE POSICIONAMENTO	
GLOBAL	16
1.1 Agricultura de Precisão	16
1.1.1 Influência da velocidade de deslocamento na agricultura de precisão	18
1.2 Sistema de Posicionamento Global - GPS	19
1.2.1 Importância do GPS na Agricultura de Precisão	20
CAPÍTULO II.....	23
2 TECNOLOGIAS MÓVEIS	23
2.1 Computação Móvel.....	23
2.2 Popularização dos <i>Smartphones</i> e <i>Tablets</i>	23
2.3 Plataformas Móveis	24
2.3.1 Android.....	25
2.3.1.1 Arquitetura do Android	27
2.4 Google Play Store	30
2.5 Trabalhos Relacionados	30
2.5.1 Navegador de Campo	30
2.5.2 eFarmer.....	31
2.5.1 MachineryGuide (Demo).....	32
CAPÍTULO III	33
3 DESENVOLVIMENTO.....	33
3.1 Ambiente de desenvolvimento	33
3.2 Dispositivo de Testes.....	34
3.3 Identificação dos requisitos e modelagem	35
3.4 Aplicação Android.....	36
3.4.1 Atividade Inicial	36
3.4.2 Atividade Navegação.....	37
3.4.3 Atividade de Relatório.....	40
3.4.4 Preferências da aplicação	40
3.4.5 Ativação do Sistema de Localização	43
3.5 Dispositivo Bluetooth GPS Externo	47

3.5.1 Placa de prototipagem Tenorino	48
3.5.2 Módulo GPS ME-1000RW	51
3.5.3 Módulo Bluetooth HC-05	52
3.5.4 Programação do microcontrolador	54
CONSIDERAÇÕES FINAIS.....	55
REFERÊNCIAS	57
APÊNDICE A - Documento de Levantamento de Requisitos	61
APÊNDICE B – Código XML do Layout da Figura 13 (A)	63
APÊNDICE C – Código Fonte Sistema Embarcado	67

INTRODUÇÃO

A agricultura brasileira, ano após ano, alcança novos patamares de produtividade na produção de grãos. Segundo a CONAB (2017), na safra brasileira de grãos a estimativa foi de 234,33 milhões de toneladas, na área plantada o crescimento foi superior a estimativa de 60,49 milhões de hectares. O estado de Mato Grosso se destaca como maior produtor nacional, correspondendo a 26,2% da produção do país (IBGE, 2017).

Em vista da importância do estado de Mato Grosso na produção de grãos, a adoção de novas tecnologias se faz necessária para melhor utilização de insumos e equipamentos agrícolas otimizando a produtividade. Com extensas planícies e amplos planaltos existe uma dificuldade por parte do tratorista em manter o alinhamento do implemento agrícola devido à falta de referência na lavoura, o uso de tecnologias de geolocalização pode contribuir com a minimização de erros de sobreposição de insumos ou falhas de aplicação ou erros de alinhamento do implemento agrícola. No entanto, as tecnologias disponíveis no mercado apresentam altos custos em razão de serem todas importadas.

Diante disso, este trabalho teve como objetivo o desenvolvimento de uma aplicação móvel aliada a um sistema embarcado de baixo custo para orientação de veículos agrícolas baseado no sistema de posicionamento global. Para tal, foi desenvolvido um aplicativo de localização para dispositivos Android em um *hardware* como GPS externo com comunicação via *Bluetooth*.

A escolha do sistema operacional Android se deve ao fato de ser uma plataforma *open source*, dominar 93% do mercado nacional de dispositivos móveis (KANTAR, 2017), o preço dos aparelhos tem custo em média inferior ao de outros aparelhos que possuem outros sistemas operacionais e 65% dos produtores rurais utilizarem dessa plataforma (SPARK, 2016 apud STARTAGRO, 2016).

Este trabalho está estruturado em três capítulos. O primeiro capítulo apresenta os conceitos sobre Agricultura de Precisão e o Sistema de Posicionamento Global. Enquanto que, no capítulo seguinte, aborda-se sobre tecnologias móveis, plataforma Android e as aplicações existentes. Por fim, no último capítulo, estão descritos os aspectos relacionados ao projeto e desenvolvimento do aplicativo e *hardware* proposto.

CAPÍTULO I

1 AGRICULTURA DE PRECISÃO E O SISTEMA DE POSICIONAMENTO GLOBAL

Este capítulo apresenta os principais conceitos e definições relacionados a Agricultura de Precisão e o Sistema de Posicionamento Global.

1.1 Agricultura de Precisão

A Agricultura de Precisão (AP), utiliza de ferramentas e tecnologias que quando empregadas fornecem ao produtor informações detalhadas sobre as condições das áreas de cultivo, permitindo que sejam adotadas estratégias para melhor utilização dos insumos agrícolas (RESENDE et al., 2010).

Normalmente associa-se o conceito de agricultura de precisão à utilização de equipamentos de alta tecnologia de *hardware* e *software*, com o objetivo de monitorar e analisar as condições em uma determinada parcela de terreno, aplicando depois os diversos fatores de produção conforme houver necessidade (COELHO; SILVA, 2009).

Segundo Kirschner (2013), a AP tem como característica principal a aplicação de insumos agrícolas nas quantidades corretas, no local desejado e na hora certa, a fim de se ter áreas mais homogêneas.

Conforme a definição proposta por Coelho e Silva (2009, p.4):

A Agricultura de Precisão envolve a aplicação diferenciada e à medida dos factores de produção, tendo em conta a variação espacial e temporal do potencial produtivo do meio e das necessidades específicas das culturas, de forma a aumentar a sua eficiência de utilização e, assim, melhorar o rendimento económico e reduzir o impacte ambiental da atividade agrícola.

Os fundamentos para a agricultura de precisão moderna, da forma que conhecemos hoje, teve início no século XX, mas só se tornou viável na década de 1980, com o desenvolvimento de novas tecnologias como: microcomputadores, sensores e *softwares* (LAMPARELLI, 2012).

No início dos anos de 1990 surgiram no Brasil, as primeiras iniciativas no emprego da agricultura de precisão, por meio da utilização de máquinas agrícolas dotadas de receptores GPS e *softwares* específicos para obtenção e processamento de dados georreferenciados. (RESENDE et al., 2010).

A agricultura de precisão se apresenta em forma de um ciclo, em virtude da ocorrência de uma série de acontecimentos que se sucedem em uma ordem determinada. A Figura 1 apresenta uma representação gráfica desse conceito.

Figura 1 – Ciclo da Agricultura de Precisão



Fonte: Extraído Agrocompany (2018)

Como é possível observar na Figura 1, a primeira etapa de “Preparação do Solo” é onde são identificadas as carências de nutrientes do solo através de análise do solo, com base nisso é feito a aplicação desses nutrientes a fim de ajustar até os níveis apropriados para o desenvolvido da cultura.

A segunda etapa se concentra nas técnicas de plantio, para que o solo seja preparado de forma correta e que a ação de lançar a semente na terra seja feita de forma que ela germine, cresça e tenha uma boa produtividade.

A etapa de “Acompanhamento da Lavoura” é uma extensão da preparação do solo, mas envolve cuidados relacionado ao desenvolvimento das plantas. Com aplicação de defensivos agrícolas a taxas variáveis e assim eliminar os problemas diretamente nos locais em que ocorrem.

A última etapa de colheita envolve várias técnicas e ferramentas para geração de mapa de colheita que contribui para obter informações sobre a variabilidade espacial das lavouras.

A agricultura de precisão fornece inúmeras ferramentas úteis para o agronegócio que permite a redução e racionalização dos custos, maximizar a produtividade e eficiência operacional, além de proteger o meio ambiente e gerar melhores resultados econômicos.

O Brasil como grande produtor agrícola, vem apresentando grandes avanços em pesquisas voltadas para agricultura de precisão. No entanto, o desenvolvimento de tecnologias nacionais não apresenta o mesmo crescimento.

1.1.1 Influência da velocidade de deslocamento na agricultura de precisão

O desenvolvimento da agricultura de precisão e do manejo localizado teve como premissa a atuação precisa das máquinas agrícolas. O controle da velocidade de deslocamento dessas máquinas, como as semeadoras e colheitadeiras é um dos fatores que têm grande influência em diversos aspectos na produtividade da lavoura, entre eles o consumo de combustível (FURLANI; LOPES; SILVA, 2005; MARTINS et al., 2017), a capacidade operacional (BRANQUINHO et al., 2004) e a qualidade de semeadura (DIAS et al., 2009; MELLO et al., 2007).

De acordo com Santos et al. (2011), na semeadura mecanizada o aumento da velocidade interfere no estabelecimento de plantas no campo, pois influencia negativamente na quantidade de espaçamentos adequados e eleva o número de falhas durante a semeadura. Para Garcia et al. (2011), a velocidade de operação do conjunto tratorizado irá influenciar na velocidade do disco dosador de sementes dentro do reservatório de sementes e na patinagem da roda motriz da semeadora, ocasionando alterações na uniformidade de distribuição e na colocação adequada da semente no solo, interferindo na densidade ideal de plantio.

No processo de colheita, a velocidade de deslocamento da colheitadeira influencia diretamente nas perdas em conjunto com o estado da lavoura e a capacidade da máquina. Segundo Mesquita et al. (2002 apud CAMARGO; SCHIEBELBEIN, 2011), tais perdas poderiam ser evitadas tomando alguns cuidados como: monitoramento rigoroso das velocidades de trabalho da colhedora e a aferição regular dos mecanismos de trilha.

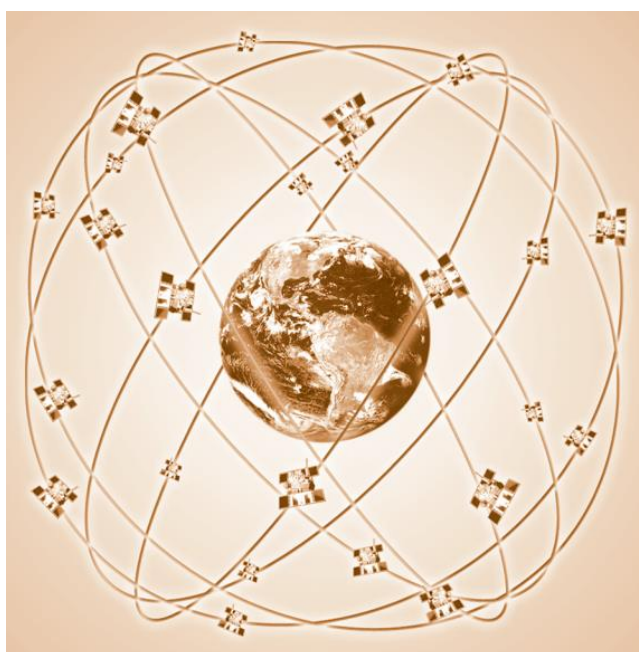
Neste sentido, pode-se afirmar que na grande maioria das situações e em todos os diferentes sistemas de semeadura e colheita é necessário um controle correto da velocidade de deslocamento dos implementos agrícolas. A utilização do sistema GPS permite que seja feito

um monitoramento da velocidade devido ao deslocamento do implemento, possibilitando que o operador tenha um parâmetro preciso da velocidade.

1.2 Sistema de Posicionamento Global - GPS

O Sistema de Posicionamento Global, conhecido por GPS (Global Positioning System) ou NAVSTAR-GPS (Navigation Satellite with Time And Ranging), é um sistema de rádio navegação desenvolvido pelo Departamento de Defesa dos Estados Unidos da América, que tinha como objetivo inicial ser utilizado exclusivamente para fins militares e o principal sistema de navegação do exército americano, mas em 1995 ele ficou disponível para uso civil e fins comerciais (COELHO; SILVA, 2009; CARVALHO; ARAÚJO, 2009). A Figura 2 ilustra o sistema GPS composto por uma constelação de satélites.

Figura 2 – Constelação de Satélites



Fonte: (Adaptado de CARVALHO; ARAÚJO, 2009)

A Figura 2 ilustra o GPS como um sistema composto por uma constelação de 24 satélites, que orbitam o globo terrestre a cada 12 horas, distribuídos de tal forma que em qualquer lugar do planeta exista pelo menos quatro satélites visíveis que fornecem aos aparelhos receptores móveis a sua posição em campo (COELHO; SILVA, 2009).

O funcionamento do sistema de GPS fundamenta-se na determinação da distância entre um ponto denominado receptor e os pontos de referência, no caso, os satélites. Cada satélite

emite um sinal codificado que o identifica, o receptor ao receber o sinal enviado calcula a distância até o satélite que emitiu o sinal (CARVALHO; ARAÚJO, 2009).

Os autores ainda explicam que, o cálculo da distância é realizado em função do tempo que o sinal leva para chegar ao receptor, levando em consideração que o sinal viaja na velocidade da luz ($\text{tempo} \times 300.000 \text{ km/h} = \text{distância}$). Sabendo a distância que o receptor está dos satélites e conhecendo suas órbitas, ele é capaz a partir da intersecção de três satélites realizar a triangulação e encontrar a posição em duas dimensões (latitude e longitude). Com a intersecção de quatro satélites é possível obter a posição em três dimensões (latitude, longitude e altitude), mesmo que possa ocorrer erros nas medições de tempo efetuadas, ele é facilmente diluído no conjunto das quatro medições.

1.2.1 Importância do GPS na Agricultura de Precisão

Segundo Kirschner (2013), o emprego da Agricultura de Precisão só foi possível com o desenvolvimento de tecnologias que permitissem obter informações mais detalhadas sobre as condições da propriedade agrícola. A integração de diferentes tipos de tecnologias, como: Sistema de Posicionamento Global (GPS), Sistema de Informação Geográfica (SIG), sensores eletrônicos, *softwares* e computadores com mais poder computacional, foram de extrema importância para o desenvolvimento da mesma.

O uso do GPS, possibilita o mapeamento de toda propriedade agrícola, realizar o levantamento de solo e uso da terra, analisar os diferentes níveis de produção em diversos pontos da área agrícola, permite a junção da coleta de dados em tempo real com informações precisas de posição, tendo assim, uma eficiente manipulação e análise de grandes quantidades de dados georreferenciados (COELHO; SILVA, 2009).

O emprego de GPS por máquinas agrícolas também contribui para uma maior produtividade. Conforme Silva et al., (2010),

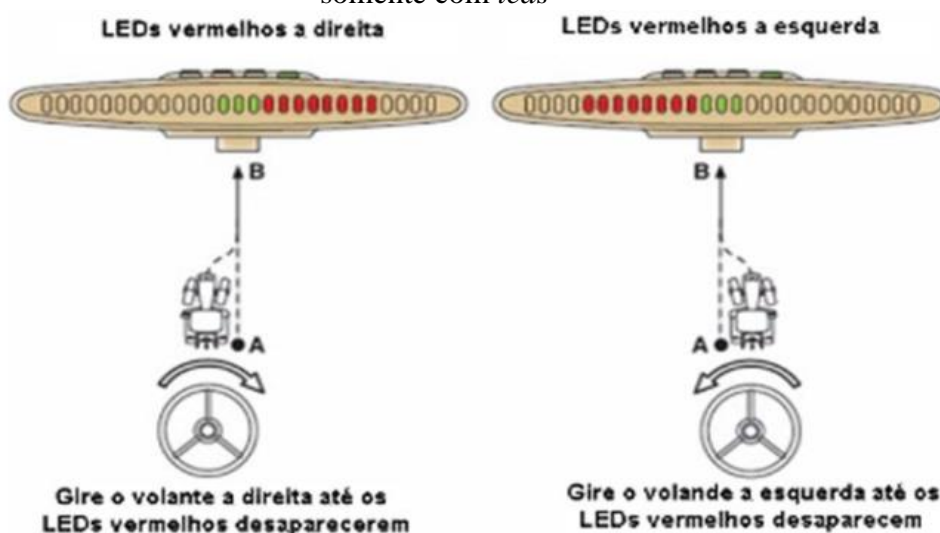
[..] o deslocamento preciso de máquinas (ex: semeadoras, pulverizadores e colhedoras), contribuindo para maior rendimento operacional e eficiência nas operações mecanizadas de semeadura, tratos culturais e colheita. Como resultado, criam-se oportunidades para otimização da frota agrícola, economia de tempo, combustível, redução do desperdício de defensivos e do trânsito de máquinas nas lavouras, amenizando os problemas de compactação do solo e de contaminação ambiental. (apud Resende et al., 2010, p. 5).

Atualmente no mercado existem diversos equipamentos que auxiliam o operador de máquinas agrícolas na orientação de direcionamento para aplicação de defensivos, adubos e

corretivos. Dentre as tecnologias existentes, a mais difundida é a orientação de máquinas por barra de luzes. Sendo a adoção dessa tecnologia de simples instalação e sua utilização considerada praticamente intuitiva, visto que o uso de tecnologias de geolocalização está cada vez mais inserida no cotidiano das pessoas.

O monitor de direcionamento por barra de luz (Figura 3) foi uma das primeiras ferramentas disponíveis no mercado a utilizar o sistema de navegação por satélite e consiste em um conjunto de sinais luminosos dispostos à frente do operador da máquina ou veículo, ligado a um processador que recebe a informação de posicionamento de um receptor GPS (BAIO; ANTUNIASSI, 2008).

Figura 3 – Funcionamento do monitor de direcionamento em formato de barra de luzes somente com *leds*



Fonte: (JOHN DEERE, 2013 apud FERNANDES, 2013)

Conforme ilustrado na Figura 3, o monitor de direcionamento de barra de luz permite que o operador consiga manter um alinhamento e, posteriormente, depois de determinada a largura de trabalho manter o paralelismo do implemento agrícola, evitando assim, falhas ou sobreposições nas operações de calagem, preparo de solo, semeadura, pulverização e colheita (SENAR, 2012).

Na Figura 4 são apresentados monitores de sistemas de condução em que se observa a localização do trator relativamente à linha de trajetória e na parte superior as barras de luzes em que os *leds* coloridos indicam o alinhamento do trator com o percurso a ser seguido.

Figura 4 – Barra de luzes com *leds* e tela de visualização da operação.



Fonte: Conceição (2014)

Os modelos da Figura 4 que apresentam uma interface gráfica e permite uma maior interação com o equipamento através de botões e telas sensível ao toque são os mais comumente encontrados em máquinas agrícolas, entretanto são produtos que possuem um alto custo com valores entre R\$7.000,00 reais até R\$20.000,00.

De acordo com Baio et al. (2001), tendo em vista, os altos custos de implantação dos sistemas de orientação por GPS a aplicação de insumos agrícolas é frequentemente realizada por métodos convencionais que incluem marcadores de espuma, orientação pelas fileiras de plantio, riscadores de solo, dentre outros. No entanto, a utilização dessas técnicas implica em custos adicionais, além de danos à cultura e ao ambiente.

Diante disso, o desenvolvimento de tecnologias de baixo custo e mais acessíveis permite que produtores que não tenham condições de adquirir os sistemas atualmente disponíveis no mercado possam ter acesso a este tipo de tecnologia que contribui para um aumento na produtividade e melhor rendimento da lavoura.

CAPÍTULO II

2 TECNOLOGIAS MÓVEIS

Nesse capítulo são apresentados os conceitos de computação móvel, a popularização dos *smartphones*, plataformas móveis e aplicativos móveis que são relevantes no contexto do presente trabalho.

2.1 Computação Móvel

A Computação Móvel teve início por volta da década de 1980, quando à miniaturização de dispositivos e a comunicação sem fio permitiu que usuários pudessem carregar seus dispositivos e manter certa conectividade (COULOURIS; DOLLIMORE; KINDBERG, 2007). Segundo Figueiredo e Nakamura (2003, p. 16),

Computação móvel pode ser representada como um novo paradigma computacional que permite que usuários desse ambiente tenham acesso a serviços independentemente de sua localização, podendo inclusive, estar em movimento. Mais tecnicamente, é um conceito que envolve processamento, mobilidade e comunicação sem fio. A idéia é ter acesso à informação em qualquer lugar e a qualquer momento.

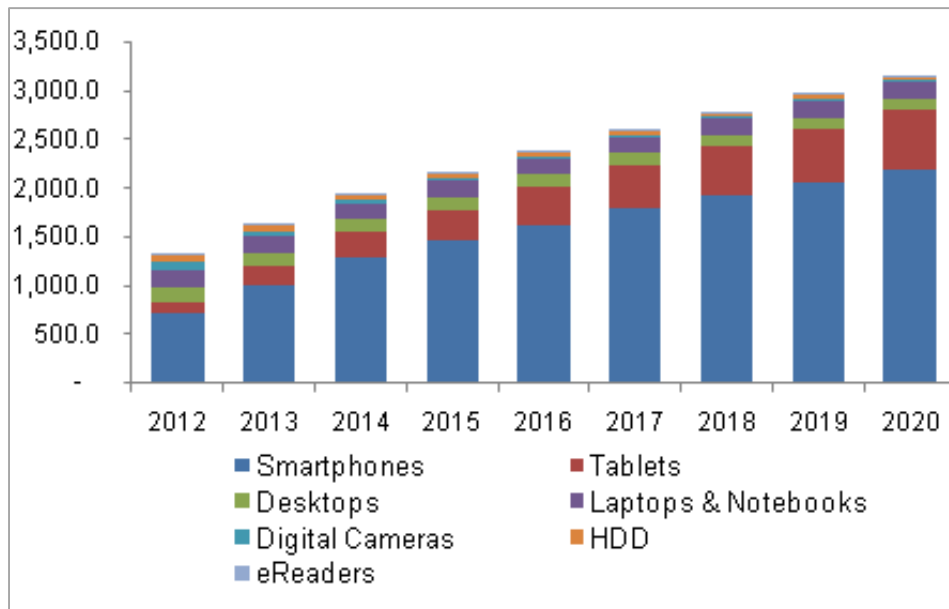
Os avanços tecnológicos que houveram nos últimos anos contribuíram cada vez mais para o crescimento da computação móvel, com o desenvolvimento de componentes eletrônicos cada vez menores que permitiram dispositivos mais leves, com maior capacidade de processamento, memória e bateria, além de novas tecnologias de comunicação sem fio, como o *Wifi*, *Bluetooth*, 3G/4G, entre outras, fazendo com que esses aparelhos ganhassem bastante popularidade e fossem utilizados nas mais diversas áreas.

2.2 Popularização dos *Smartphones* e *Tablets*

Com o surgimento dos *smartphones* e *tablets* os aparelhos como: *notebooks*, *netbooks*, câmeras digitais e telefones móveis começaram a perder espaço, pois estes foram absorvidos pelos *smartphones* que conseguem englobar características encontradas anteriormente em diferentes dispositivos. (CNN MONEY, 2010). Espera-se que os *smartphones* representem mais de 60% da receita total do mercado global de eletrônicos até 2020, isso pode ser atribuído ao alto grau de convergência digital e iniciativas de inovação dos principais fabricantes

(GRAND VIEW RESEARCH, 2015). A Figura 5 mostra a comparação das vendas de eletrônicos pessoais no mundo e a previsão até 2020.

Figura 5 – Mercado global de eletrônicos pessoais por produto (milhões de unidades).



Fonte: Grand View Research (2015)

Conforme é possível visualizar na Figura 5 o aumento nas vendas de dispositivos eletrônicos, principalmente dos *smartphones* e *tablets* apresentou uma grande expansão durante os anos e isso tende a continuar. A utilização de *smartphones* por parte dos brasileiros atualmente também vêm apresentando crescimento, o Brasil conta com 198 milhões de aparelhos ativos e a estimativa é que chegará a 236 milhões em até dois anos (MEIRELLES, 2017). Esse crescimento também se reflete no campo, segundo um estudo realizado com 2.835 produtores rurais de 15 Estados 61% dos entrevistados afirmaram possuir pelo menos um *smartphone* e 42% fazem uso da internet (ABMR&A, 2017 apud UNIVERSOAGRO, 2017).

Baseado nestas pesquisas é possível identificar o quanto é promissor o mercado de desenvolvimento de *softwares* para *smartphones* e *tablets*.

2.3 Plataformas Móveis

As plataformas de dispositivos móveis é um conjunto de tecnologias que envolvem sistemas operacionais, linguagens de programação e ferramentas de desenvolvimento. Ela é a responsável por gerenciar os recursos do aparelho como banco de dados, câmera, GPS, sensores e periféricos (RECH, 2013).

Entre as principais plataformas de dispositivos móveis atualmente, temos: Android e iOS que juntas dominam 99,9% do mercado de *smartphones* vendidos no ano de 2017 (GARTNER, 2018). As características do Android são apresentadas a seguir, por ser a plataforma utilizada neste trabalho.

2.3.1 Android

O Android é uma plataforma para aplicativos móveis, baseado em *Linux* e é muito utilizado em *smartphones* e *tablets* de diversos fabricantes (SCHEMBERGER et al., 2009 apud KIRSCHNER, 2013). Foi desenvolvido pela Google e lançado junto com a OHA (*Open Handsent Alliance*), aliança de diversas empresas de *hardware*, *software* e telecomunicações que se uniram com a intenção de desenvolver a indústria de dispositivos móveis, dentre essas empresas, estão: Samsung, LG, Telefônica, Dell, Motorola, Nvidia, Qualcomm (CAFÉ, 2012).

O sistema operacional Android é *open source* sob licença *Apache 2.0*, é um sistema que se mostra bastante estável, versátil e completo, possuindo a capacidade de reproduzir gráficos 2D e 3D, suporte para arquivos de áudio, vídeo e imagem e a possibilidade de desenvolver aplicações que utilizam GPS nativamente do sistema operacional (CAFÉ, 2012).

Com o crescimento nas vendas de *smartphones*, o Android teve grande crescimento no mercado se tornando líder dois anos depois de lançado. Conforme mostrado na Tabela 1, a plataforma Android vem dominando o mercado em relação aos seus concorrentes.

Tabela 1: Vendas de smartphones em todo o mundo para usuários finais por sistema operacional em 2017 (milhares de unidades)

Operating System	2017 Units	2017 Market Share (%)	2016 Units	2016 Market Share (%)
Android	1,320,118.1	85.9	1,268,562.7	84.8
iOS	214,924.4	14.0	216,064.0	14.4
Other OS	1,493.0	0.1	11,332.2	0.8
Total	1,536,535.5	100.0	1,495,959.0	100.0

Fonte: Adaptado de Gartner (2018)

Desde o seu lançamento em 2008, o Android recebeu uma série de atualizações e com elas vieram novas funcionalidades e correções de erros. As versões do Android receberam codinomes de doces e foram lançadas em ordem alfabética.

No Quadro 1 estão especificadas todas as versões do Android desde o seu primeiro lançamento oficial, seus codinomes, a data de lançamento, o nível da API e a distribuição de dispositivos que utilizam cada versão.

Quadro 1 – Versões do Android e distribuição

Codiname	Versão	API	Data	Distribuição*
Alpha	1.0	1	23 de setembro 2008	<0,1%
Beta	1.1	2	9 de fevereiro 2009	<0,1%
Cupcake	1.5	3	30 de abril de 2009	<0,1%
Donut	1.6	4	15 de setembro de 2009	<0,1%
Eclair	2.0	5	26 de outubro de 2009	<0,1%
Froyo	2.2 – 2.2.3	8	20 de maio de 2010	<0,1%
Gingerbread	2.3 - 2.3.7	10	09 de fevereiro 2011	0.3%
Honeycomb	3.0	11	22 de fevereiro de 2011	<0,1%
Ice Cream Sandwich	4.0	15	16 de dezembro de 2011	0.4%
Jelly Bean	4.1.x	16	9 de julho de 2012	1.7%
	4.2.x	17	13 de novembro de 2012	2.2%
	4.3	18	23 de julho de 2013	0.6%
KitKat	4.4	19	31 de outubro de 2013	10.5%
Lollipop	5.0	21	12 de novembro de 2014	4.9%
	5.1	22	10 de março de 2015	18.0%
Marshmallow	6.0	23	05 de outubro de 2015	26.0%
Nougat	7.0	24	22 de agosto de 2016	23.0%
	7.1	25	5 de dezembro de 2016	7.8%
Oreo	8.0	26	21 de agosto de 2017	4.1%
	8.1	27	05 de dezembro de 2017	0.5%

*Dados de distribuição coletados durante um período de 7 dias encerrado em 08/01/2018

Fonte: Adaptado de Android Developers (2018)

O Google disponibiliza dados estatísticos sobre os dispositivos dos seus usuários, esses dados compartilham de determinada característica, como a versão do Android ou o tamanho da tela, assim, contribui para que o desenvolvedor saiba priorizar os perfis de dispositivos para os quais ele quer otimizar o aplicativo e oferecer compatibilidade com dispositivos diferentes. (ANDROID DEVELOPERS, 2018).

A principal característica responsável pelo sucesso do Android é as constantes atualizações, recebendo sempre novas tecnologias, funcionar em dispositivos de diversos fabricantes permitindo que os usuários tenham mais opções, ser de código fonte aberto atraindo desde grandes empresas até desenvolvedores fortalecendo a comunidade e incentivando ainda mais a criação de aplicativos para a plataforma.

Os aplicativos nativos para o sistema operacional Android são desenvolvidos na linguagem de programação *Java*, utilizando a IDE (*Integrated Development Environment*) Android Studio. Atualmente o Android Studio é a ferramenta oficial do Android e encontra-se na versão 3.1.3 (ANDROID STUDIO, 2018).

2.3.1.1 Arquitetura do Android

A arquitetura da plataforma Android é composta por cinco camadas, sendo elas: *Linux Kernel*, *Hardware Abstraction Layer (HAL)*, *Native C/C++ Libraries*, *Android Runtime*, *Java API Framework*, *System Apps*, conforme apresentado na Figura 6. Essas camadas têm suas características próprias e juntas dão sentido ao montante das aplicações para execução, as quais serão detalhas a seguir conforme consta na documentação do Android:

Linux Kernel: Esta camada é responsável por gerenciar os processos, *threads*, além de permitir que o Android aproveite os principais recursos de segurança e que os fabricantes de dispositivos desenvolvam *drivers* de *hardware* para o *kernel*.

Camada de abstração de hardware (Hardware Abstraction Layer - HAL): A camada de abstração de *hardware* fornece interfaces padrão que expõem os recursos de *hardware* do dispositivo para a estrutura da API Java de maior nível. A HAL consiste em vários módulos de biblioteca, que implementam uma interface para um tipo específico de componente de *hardware*, como a câmera ou módulo *bluetooth*. Quando uma *Framework API* faz uma chamada para acessar o *hardware* do dispositivo, o sistema Android carrega o módulo da biblioteca para este componente de *hardware*.

Android Runtime (ART): Para dispositivos com Android versão 5.0 (API nível 21) ou superior, cada aplicativo executa o próprio processo com uma instância própria do *Android Runtime*.

O ART é projetado para executar várias máquinas virtuais em dispositivos de pouca memória executando arquivos DEX, um formato de *bytecode* projetado especialmente para o

Android, otimizado para ocupar pouco espaço na memória. Antes do Android versão 5.0 (API nível 21), o *Dalvik* era a máquina virtual do Android (ANDROID DEVELOPERS, 2018).

Alguns dos recursos principais do ART são:

- Compilação “*ahead-of-time*” (AOT) e “*just-in-time*” (JIT);
- Coleta de lixo otimizada (GC);
- Melhor suporte de depuração, inclusive um gerador de perfil de amostragem, exceções de diagnóstico detalhadas e relatórios de erros, além da capacidade de definir pontos de controle para monitorar campos específicos.

Bibliotecas C/C++ Nativas (*Native C/C++ Libraries*): Vários componentes e serviços principais do sistema Android, como ART e HAL, são implementados em código nativo que exige bibliotecas nativas programadas em C e C++. A plataforma Android fornece as *Java Framework APIs* para expor a funcionalidade de algumas dessas bibliotecas nativas aos aplicativos. Por exemplo, é possível acessar *OpenGL ES* pela *Java OpenGL API* da estrutura do Android para adicionar a capacidade de desenhar e manipular gráficos 2D e 3D no aplicativo.

Java API Framework: O conjunto completo de recursos do SO Android está disponível pelas APIs programadas na linguagem *Java*. Essas APIs formam os blocos de programação que é preciso para criar os aplicativos Android simplificando a reutilização de componentes e serviços de sistema modulares e principais, incluindo:

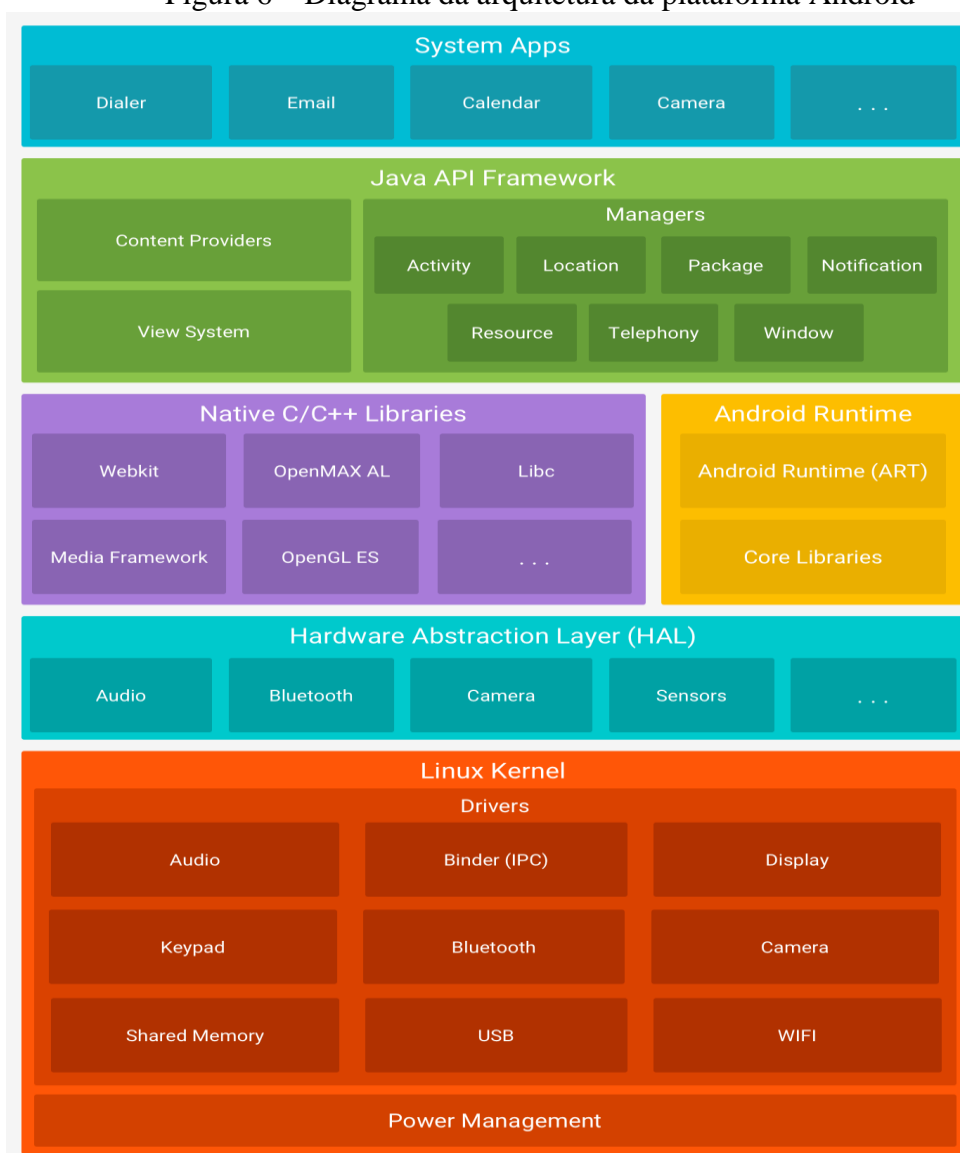
- Um sistema de visualização (*View System*) rico e extensivo, útil para programar a Interface do Usuário (IU) de um aplicativo, com listas, grades, caixas de texto, botões e até mesmo um navegador web incorporado;
- Um gerenciador de recursos (*Resource Manager*), fornecendo acesso a recursos sem código como *strings* de localização, gráficos e arquivos de layout;
- Um gerenciador de notificação (*Notification Manager*) que permite que todos os aplicativos exibam alertas personalizados na barra de status;
- Um gerenciador de atividade (*Activity Manager*) que gerencia o ciclo de vida dos aplicativos e fornece um *navigation back stack*;
- Provedores de conteúdo (*Content Providers*) que permite que aplicativos acessem dados de outros aplicativos, como o aplicativo Contatos, ou compartilhem os próprios dados.

Os desenvolvedores têm acesso completo às mesmas *Framework APIs* que os aplicativos do próprio sistema Android utilizam.

Aplicativos do sistema (*System Apps*): O Android vem com um conjunto de aplicativos principais para e-mail, envio de SMS, calendários, navegador de internet, contatos, entre outros. Os aplicativos inclusos na plataforma não têm status especial entre os aplicativos que o usuário escolhe instalar. Portanto, um aplicativo de terceiros pode se tornar o navegador web padrão do usuário, o aplicativo de envio de SMS ou até mesmo o teclado padrão do usuário (existem algumas exceções, como o aplicativo Configurações do sistema).

Os aplicativos do sistema funcionam como aplicativos para os usuários e fornecem recursos importantes que os desenvolvedores podem acessar pelos próprios aplicativos. Por exemplo: se o aplicativo deseja enviar uma mensagem SMS, não é necessário programar essa funcionalidade; em vez disso, é possível invocar o aplicativo de SMS que já está instalado para enviar uma mensagem ao destinatário que o usuário especificar.

Figura 6 – Diagrama da arquitetura da plataforma Android



Fonte: Android Developers (2018)

2.4 Google Play Store

Todos os aplicativos desenvolvidos para a plataforma Android podem ser disponibilizados de forma gratuita ou paga na loja oficial, a *Google Play Store*. Inaugurada em 2008 com o nome de *Android Market* teve seu nome alterado para *Google Play* em 2012 e foi unificada com a loja de músicas, filmes e livros (RECH, 2013).

Qualquer desenvolvedor pode expor seu aplicativo na *Google Play* desde que tenha uma conta no Google que pode ser criada gratuitamente, após isso ele precisa efetuar um cadastro como desenvolvedor que exige que pague uma taxa única de \$25,00, que permite publicar um número ilimitado de aplicativos.

2.5 Trabalhos Relacionados

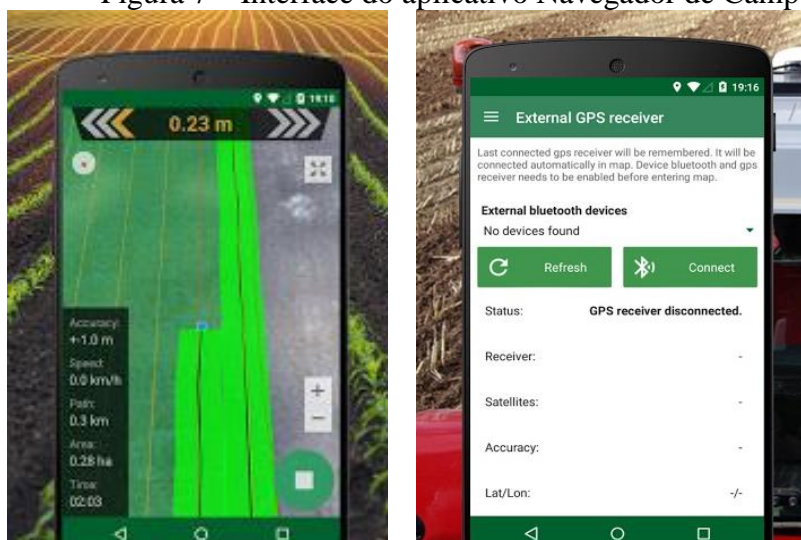
Alguns aplicativos, tanto pagos como gratuitos, já foram desenvolvidos com características semelhantes ao que está sendo desenvolvido neste trabalho e já se encontram para uso.

Nas pesquisas realizadas na *Google Play* foram encontrados 5 aplicativos que atendem a demanda de facilitar com que o operador mantenha o alinhamento do implemento agrícola, que estejam no idioma Português. Entretanto, todos os aplicativos analisados foram desenvolvidos por empresas estrangeiras, por isso apresentam vários erros de tradução e também não existe a tradução completa do aplicativo. Alguns aplicativos só funcionam na versão paga ou exigem um *hardware* externo com um alto custo. Nas seções seguintes são apresentados os três aplicativos com maior número de avaliações positivas na *Google Play*.

2.5.1 Navegador de Campo

A aplicação Navegador de Campo é um dos mais populares aplicativos de direção paralela para agricultura de precisão. Para a utilização do aplicativo não é necessário qualquer equipamento adicional. O aplicativo permite o uso de retas AB e linhas paralelas enquanto navega no campo. No entanto, apesar do aplicativo ser um dos mais utilizados ele apresenta várias partes sem tradução. A Figura 7 a exibe interface do aplicativo.

Figura 7 – Interface do aplicativo Navegador de Campo

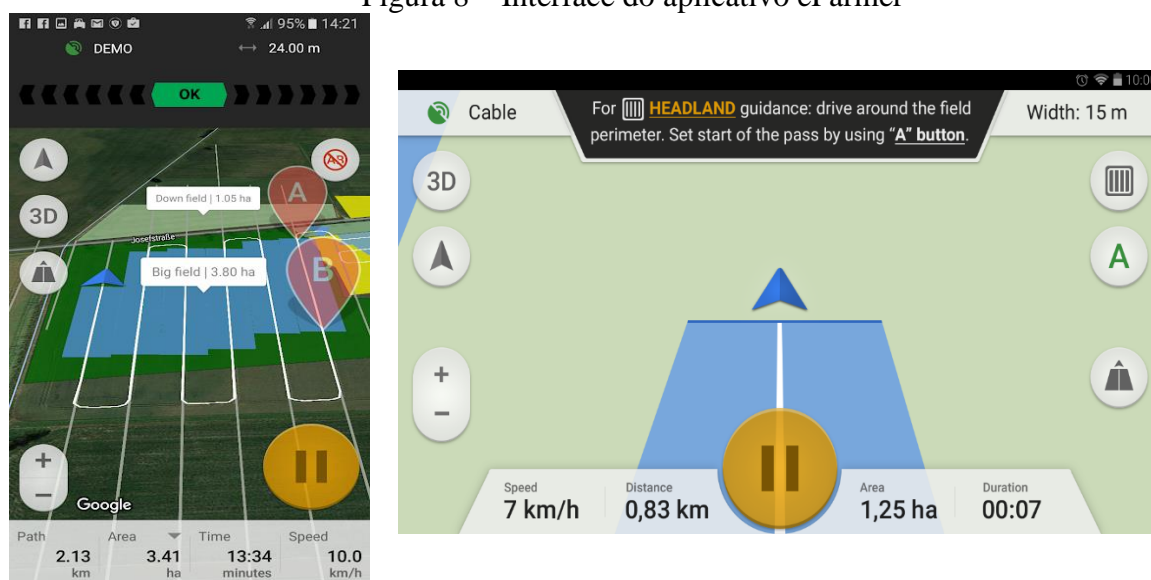


Fonte: Adaptador de Google Play (2018)

2.5.2 eFarmer

O aplicativo *eFarmer* (Figura 8) possui uma versão grátis e a versão paga. A licença para uso da versão paga é de €120,00 ou aproximadamente R\$523,00. Para utilizar o aplicativo é necessário realizar um cadastro ou se já estiver registrado efetuar o login, caso o usuário não tenha conexão com a internet ele não conseguirá utilizar a aplicação.

Figura 8 – Interface do aplicativo eFarmer



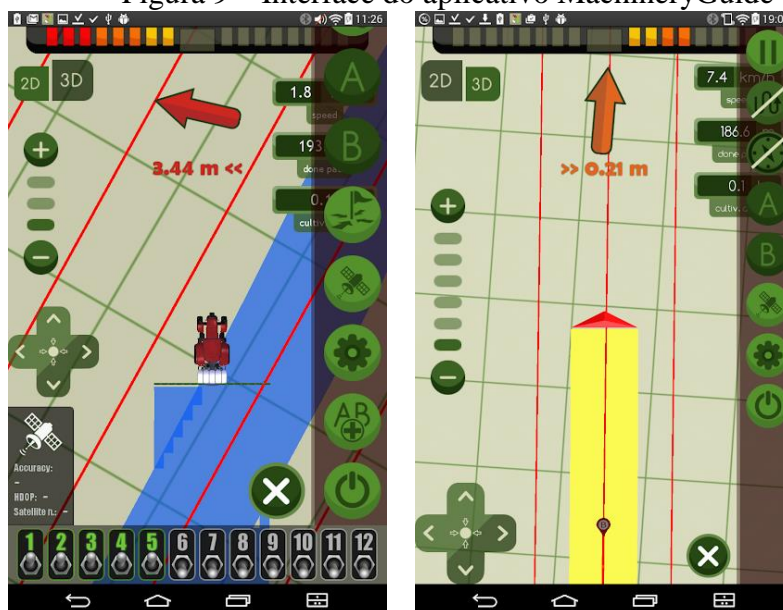
Fonte: eFarmer (2018)

2.5.1 MachineryGuide (Demo)

O aplicativo *MachineryGuide*, demonstrado na Figura 9, é mais um dos aplicativos disponíveis para download na *Google Play*. Essa versão disponível na loja é de demonstração, a versão completa só é possível mediante a compra da licença e baixar diretamente do site oficial. A versão demo possui quase todas as funções da versão paga, mas o usuário não consegue se conectar a um GPS real.

Este aplicativo também funciona com um hardware externo que melhora a precisão dos dados GPS, mas há relatos nos comentários do aplicativo na *Google Play* de usuários que adquiriram o *hardware* e mesmo assim não teve boa funcionalidade com a aplicação e também há relatos de o aplicativo possuir a opção para o Português, porém não ser feita a tradução integral.

Figura 9 – Interface do aplicativo MachineryGuide



Fonte: MachineryGuide (2018)

CAPÍTULO III

3 DESENVOLVIMENTO

Neste capítulo é apresentado o resultado deste trabalho, as ferramentas utilizadas para o desenvolvimento, incluindo ambiente de programação e teste, as características de modelagem, através de diagramas, e apresentada a interface do aplicativo com sua descrição.

3.1 Ambiente de desenvolvimento

O aplicativo foi desenvolvido em um notebook pessoal, com as seguintes configurações de *hardware* e *software*:

- **Processador:** Intel® Core™ i7-4500U CPU @ 1.80Ghz x 2;
- **Tipo de Sistema Operacional:** 64-bits;
- **Memória RAM:** 8 GB;
- **Disco Rígido:** 1 TB
- **Sistema Operacional:** Ubuntu 16.04 LTS (Xenial Xerus).

Além disso, foi necessário preparar o ambiente de desenvolvimento, que conta com alguns *softwares* voltados para criação desse tipo de aplicação. Os programas utilizados foram:

- **JDK 8:** O JDK (*Java Development Kit*) é um ambiente de desenvolvimento para a criação de aplicativos e componentes usando a linguagem de programação *Java*.
- **Android Studio 2.3.3:** O Android Studio é um ambiente de desenvolvimento integrado oficial da Google para criar aplicações Android. Ele conta com diversos componentes que dão suporte ao desenvolvimento de aplicativos, tanto para *smartphones* e *tablets*, quanto para *Android Wear* e outras plataformas.
- **Git:** O Git é um sistema de controle de versão para desenvolvimento de *softwares*. Ele tem como características principais a velocidade, integridade dos dados e suporte para fluxos de trabalho distribuídos e não-lineares.
- **Trello:** O Trello é uma ferramenta de gerenciamento de projetos em listas extremamente versátil baseado no sistema de Kanban. Ele foi utilizado para gerenciar as funcionalidades e etapas do desenvolvimento.

Apesar dos programas utilizados para o desenvolvimento do aplicativo funcionarem nos sistemas operacionais *Linux* e *Windows* a escolha por um sistema operacional *Linux* se deu devido ao fato de ser um sistema que oferece praticidade na instalação e configuração do ambiente, além de utilizar menos recursos de máquina para seu gerenciamento possibilitando que os *softwares* consigam ter um melhor desempenho.

No entanto, para o desenvolvimento do sistema embarcado foi necessário a mudança para o sistema operacional *Windows*. Pois os *softwares* necessários para a programação e prototipagem do *hardware* não são compatíveis com o sistema operacional *Linux*, por isso utilizou-se o SO *Windows 8.1 Single Language*. Os programas utilizados no desenvolvimento do sistema embarcado estão descritos a seguir:

Proton IDE: O algoritmo do dispositivo embarcado foi feito utilizando o Proton IDE versão 2.0.1.0 e o compilador Proton Basic Compiler versão 3.5.2.7. Desenvolvido pela Mecanique UK, com ele é possível escrever, depurar e compilar o código implementado no sistema operacional *Windows*. Por ser compatível com o PIC18F4550 foi escolhido por ser um ambiente simples e de fácil utilização.

Linguagem de programação BASIC: A linguagem BASIC é imperativa e de baixo nível, criada originalmente para fins didáticos, ela é simples e de fácil aprendizagem. Oferece uma grande vantagem em relação a outras linguagens por não encapsular o acesso aos registradores e, deste modo, permite maior liberdade de trabalho. Também possui alguns conjuntos de bibliotecas que facilitam o trabalho durante a implementação do código.

Simulador Proteus: O Proteus é um *software* que integra um ambiente para projetar e simular os circuitos eletrônicos. Seu ambiente de simulação suporta uma grande lista de componentes. Foi desenvolvido pela empresa *Labcenter Eletronics*, este disponibiliza uma versão de avaliação para estudantes.

3.2 Dispositivo de Testes

Inicialmente, para os testes do aplicativo utilizou-se do próprio emulador nativo do Android Studio que é bem completo, porém devido ao seu alto consumo de memória e com a inclusão de funcionalidades na aplicação como a comunicação *Bluetooth* e a utilização do serviço de localização foi necessário utilizar um dispositivo real.

O dispositivo real foi um *smartphone* Asus Zenfone 3 Max tela 5.5 com Android versão 7.1.1 Nougat.

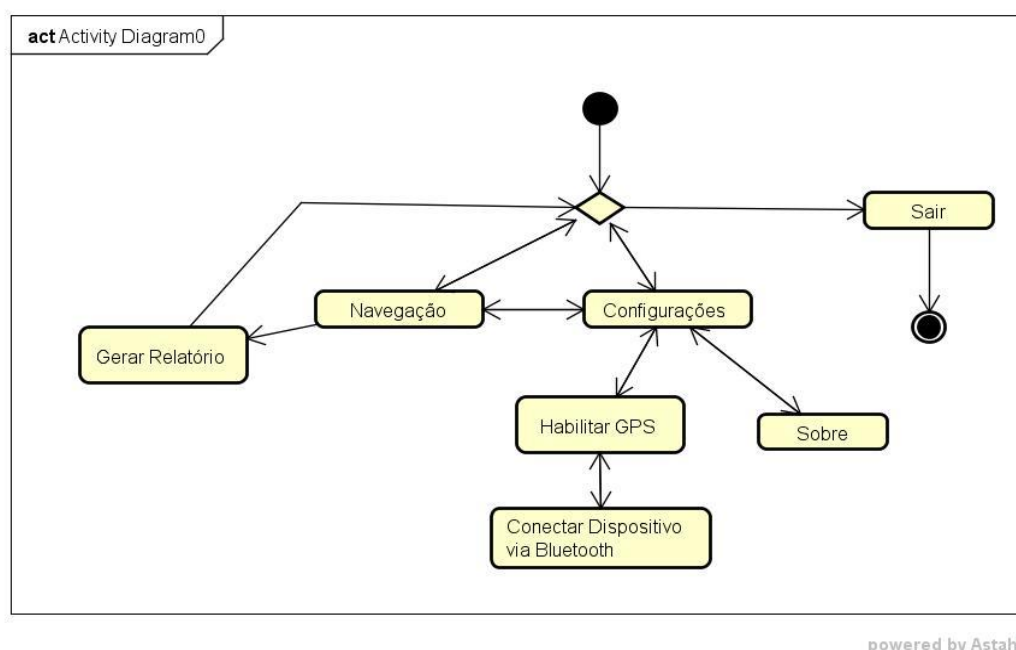
3.3 Identificação dos requisitos e modelagem

Este trabalho tem como objetivo desenvolver um aplicativo que auxilie o motorista do implemento agrícola a manter o alinhamento e paralelismo na hora do trabalho. A proposta de desenvolver este trabalho surgiu da necessidade de se ter uma ferramenta de baixo custo e que fosse acessível aos produtores de baixa renda. As informações apresentadas ao usuário da aplicação Android desenvolvida foram escolhidas com base no levantamento de requisitos realizado a partir de outras aplicações já existentes.

O diagrama de atividade demonstrado na Figura 10 mostra o fluxo de atividades dentro do aplicativo e o diagrama de caso de uso, que mostra as funcionalidades que o usuário terá com o aplicativo

A primeira etapa para o desenvolvimento do aplicativo foi verificar as funcionalidades necessárias para uma boa interação com o usuário. Foi feito a especificação dos requisitos conforme o documento gerado no Apêndice A, neste estão listados os requisitos funcionais, ou seja, as funcionalidades desejadas no sistema. Também estão os requisitos não funcionais como qualidade e restrições.

Figura 10 – Diagrama de Atividade

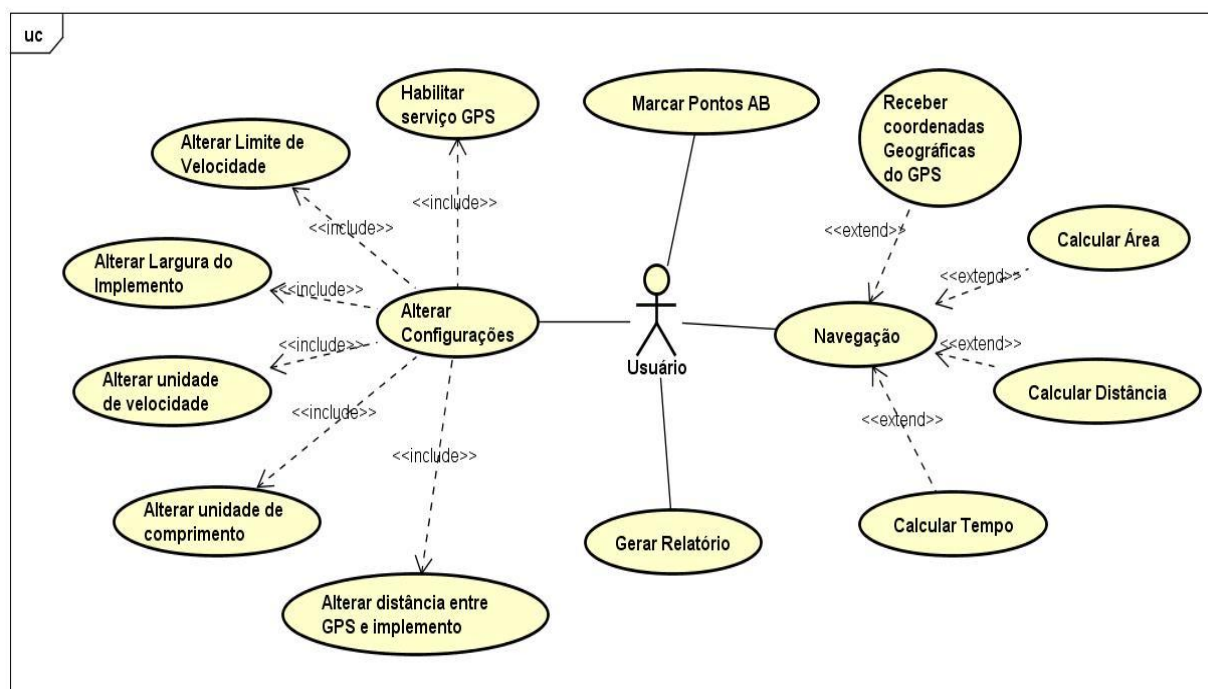


powered by Astah

Fonte: Elaborado pelo Autor (2018)

A partir das funcionalidades levantadas para a aplicação, foi possível criar o diagrama de caso de uso, conforme apresentado na Figura 11. Este diagrama mostra quais são as funcionalidades que o usuário pode ter.

Figura 11 – Diagrama de Caso de Uso



powered by Astah

Fonte: Elaborado pelo Autor (2018)

O diagrama de Caso de Uso do aplicativo, representado na Figura 11 tem o Usuário como ator, onde o aplicativo recebe as coordenadas geográficas pelo GPS e assim determina sua localidade.

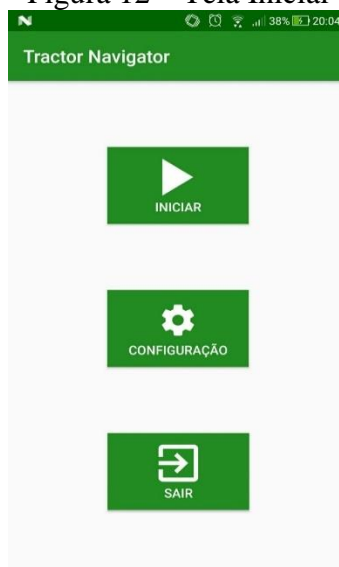
3.4 Aplicação Android

A aplicação Android desenvolvida é responsável por gerir a comunicação com o GPS Externo/*Bluetooth*, obter a posição geográfica do dispositivo e auxiliar o condutor do veículo agrícola a manter o paralelismo do implemento no momento do plantio, colheita ou pulverização. Esta aplicação é composta por um serviço Android e por quatro atividades (“Menu Principal”, “Configurações”, “Navegação” e “Relatório”).

3.4.1 Atividade Inicial

A Figura 12 exibe a tela inicial do aplicativo a partir dela o usuário poderá abrir para a atividade das preferências de modo a configurar alguns parâmetros da aplicação, escolher o serviço de localização ou dar início a atividade de navegação e sair da aplicação.

Figura 12 – Tela Inicial

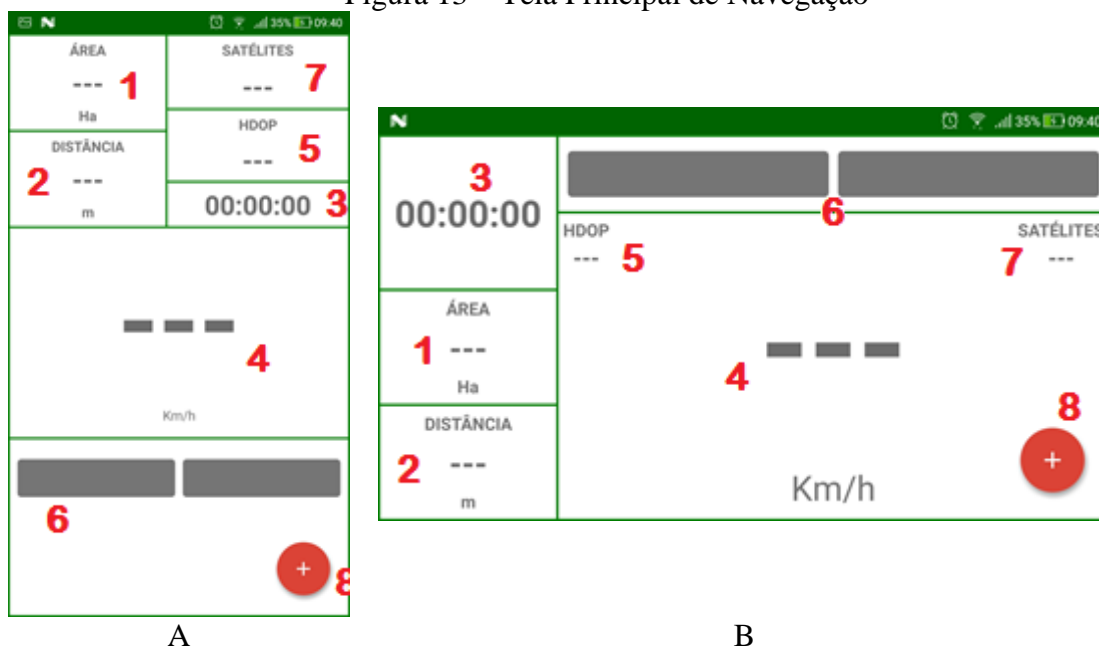


Fonte: Elaborado pelo Autor (2018)

3.4.2 Atividade Navegação

A atividade principal da aplicação (Figura 13) foi criada a partir de dois layouts XML um para cada orientação, isso foi feito para que o usuário pudesse utilizar a aplicação na orientação que mais lhe agradar, e também seguindo as boas práticas de desenvolvimento de aplicativos que sugere que o aplicativo funcione em mais de uma orientação. No Apêndice B é possível visualizar o código XML do layout da Figura 13 (A).

Figura 13 – Tela Principal de Navegação



Fonte: Elaborado pelo Autor (2018)

A tela principal, conforme demonstrada na Figura 13 é composta por:

- 2 Progress Bar que informa ao condutor o quanto ele está saindo do percurso estabelecido em relação ao caminho traçado inicialmente;
- Campos de dados provenientes do veículo via GPS;
- Botão de controle da aplicação;

É possível visualizar os itens com as informações que o condutor do veículo terá no momento que estiver em atividade, os itens são descritos no Quadro 2. Enquanto o usuário não iniciar a atividade os campos de dados ficaram vazios “---” indicando que não foi inicializado.

Quadro 2 – Descrição dos itens da tela principal

Item	Nome	Descrição
1	Área	Informa a área total em hectares trabalhada.
2	Distância	Informa a distância percorrida pelo implemento.
3	Tempo	Informa o tempo que o usuário esteve em atividade.
4	Velocidade	Informa a velocidade que o veículo está trafegando a partir dos dados fornecidos pelo GPS, alterando a cor do campo para vermelho caso o condutor ultrapasse o limite estabelecido nas preferências.
5	HDOP ou Precisão	Informa a precisão horizontal (HDOP), caso o usuário esteja utilizando o GPS externo, ou a Precisão em metros se estiver utilizando o GPS interno do dispositivo.
6	Barra de Luz	Informar para o usuário que ele está saindo do traçado correto, possui a mesma função da barra de luz (barra de <i>leds</i>) nos GPS Agrícolas, a <i>progress bar</i> irá “enchendo” conforme ele se afasta da rota.
7	Satélites	Informa a quantidade de satélites que o GPS está recebendo os dados.
8	Botão de Ação	A partir desse botão o usuário realiza as ações de iniciar, pausar, marcar localização ou finalizar o trabalho.

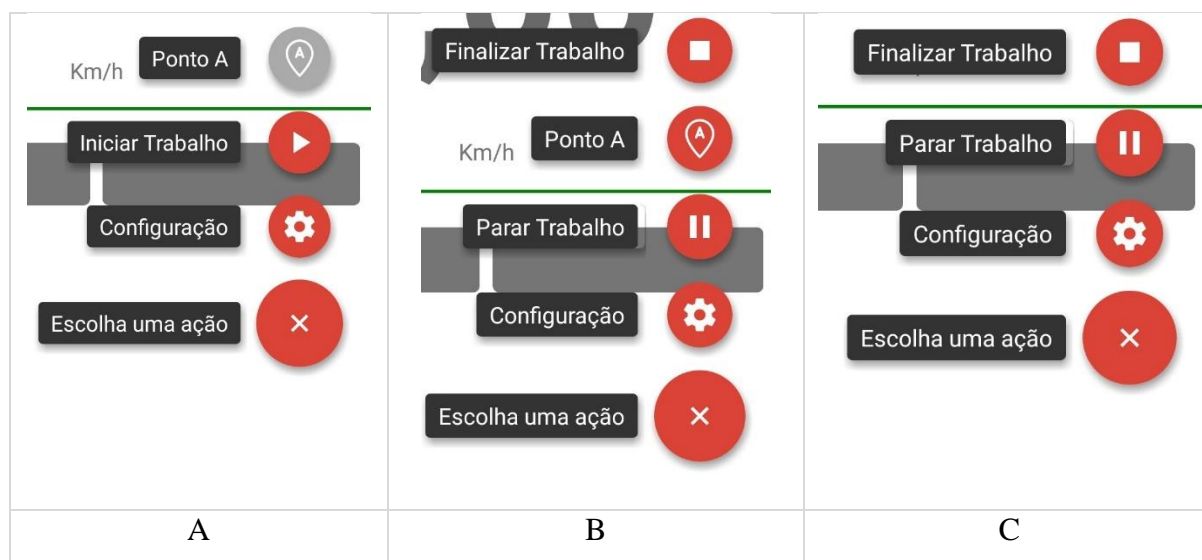
Fonte: Elaborado pelo Autor (2018)

A fim de promover uma interação simples com a aplicação, utilizou-se um *Floating Action Button* (FAB) que permite a criação de um único botão de menu de ação flutuante e

assim uma interface visualmente limpa com poucos itens que o usuário precisa interagir e assim tendo algo mais prático.

A Figura 14 apresenta os demais FAB que ficam ocultos no *Floating Action Menu*. Cada botão é descrito a seguir.

Figura 14 – Botões de ação



Fonte: Elaborado pelo Autor (2018)

Botão de Iniciar: O botão de “Iniciar Trabalho” dará início a atividade que o condutor estiver realizando no momento e também ao percurso;

Botão de Pausar: O usuário poderá pausar sua atividade, caso ocorra algum imprevisto ou necessite fazer outra atividade, bastará clicar em “Parar Trabalho”, para que a aplicação pare de registrar os dados de localização. Quando desejar voltar, bastará apenas clicar novamente em “Iniciar Trabalho” para retornar suas atividades.

Botão de Marcação: O botão de marcação “Ponto A” e “Ponto B” irá armazenar a localização do GPS para que possa ser traçada a reta inicial. A partir desses dois pontos, o aplicativo irá determinar o paralelismo entre a reta inicial e se o condutor está seguindo o percurso definido. Esse botão fica desabilitado sempre que o usuário clicar em “Parar Trabalho”, pois assume que ele não está em atividade. A marcação do “Ponto B” só irá aparecer, caso o usuário já tenha marcado o “Ponto A”.

Botão de Configuração: Este botão levará o usuário diretamente para a tela de preferências, sem a necessidade de retornar a tela inicial, assim ele pode alterar alguma informação em tempo de execução.

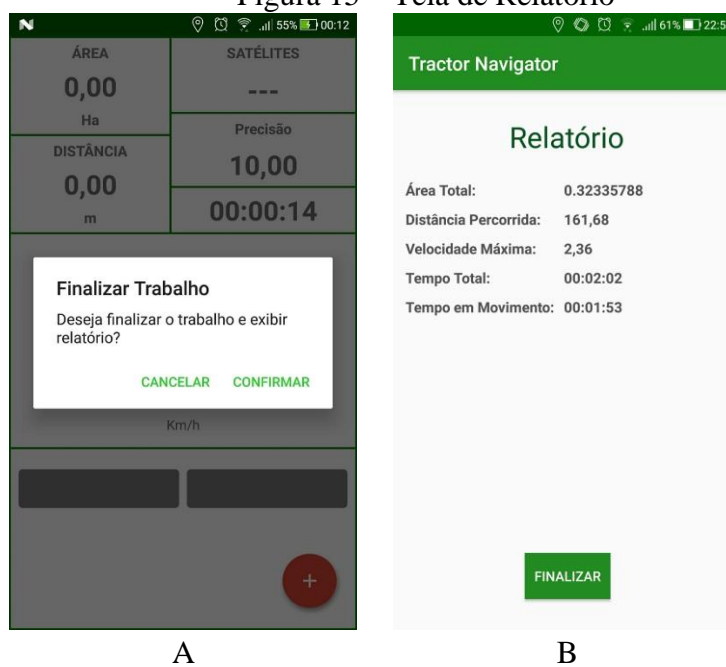
Botão de Finalizar: O botão “Finalizar Trabalho” irá finalizar a atividade e direcionar o usuário para o relatório final.

Para dar início ao registro dos dados, bastará que o usuário esteja com o GPS ligado e que esteja em campo, então ele poderá clicar em “Iniciar Trabalho” para que o GPS busque sua posição atual e a exiba na tela.

3.4.3 Atividade de Relatório

Após concluir o percurso desejado e o usuário desejar encerrar a atividade que está sendo realizada, basta apenas clicar no botão “Finalizar Trabalho” (Figura 14), nesse momento aparece um alerta para o usuário pedindo a confirmação da ação, conforme demonstrado na Figura 15 (A), depois disso é exibida a tela de relatório como podemos ver na Figura 15 (B), que mostra algumas informações referentes à atividade realizada, como: Área total, distância percorrida, velocidade máxima atingida, tempo de trabalho, tempo em que o veículo esteve em movimento.

Figura 15 – Tela de Relatório



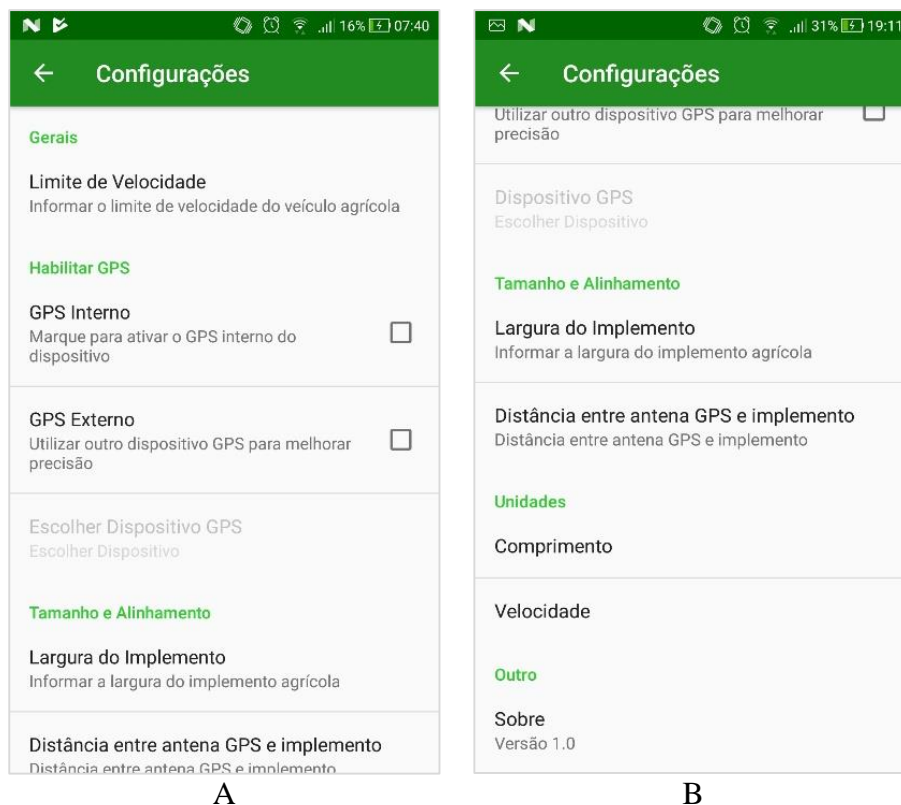
A B
Fonte: Elaborado pelo Autor (2018)

3.4.4 Preferências da aplicação

A aplicação desenvolvida permite ao usuário modificar alguns dos parâmetros necessários ao seu funcionamento. A Figura 16 ilustra a interface de preferências exibida ao

usuário. As opções disponíveis bem como as respectivas descrições estão enumeradas no Quadro 3.

Figura 16 – Tela de Preferências da aplicação



Fonte: Elaborado pelo Autor (2018)

A plataforma Android nos fornece uma forma simples de armazenamento das preferências, atribuindo um par de chave-valor correspondente que é salvo em um arquivo *SharedPreferences* padrão. Quando o usuário altera uma configuração, o sistema atualiza o valor correspondente no arquivo *SharedPreferences*.

A implementação da tela de preferências teve como orientação a documentação do Android. As configurações são criadas por meio de um arquivo XML que contém várias subclasses da classe *Preference*. Cada *Preference* aparece como um item em uma lista e oferece a IU adequada para que o usuário modifique a configuração. É necessário especificar qual o tipo de preferência para cada opção bem como os seus atributos (ANDROID DEVELOPERS, 2018).

O código XML da Listagem 1 exemplifica a definição de uma *EditTextPreference* que permitirá ao utilizador definir o limite de velocidade com o qual o veículo agrícola deverá trafegar, conforme descritos na seção 1.1.1 sobre a importância do controle da velocidade.

Quadro 3 – Preferências da aplicação Android

Opção	Tipo de Preferência	Descrição
Limite de Velocidade	EditTextPreference	Informa o limite de velocidade do veículo agrícola
GPS Interno	CheckboxPreference	Utiliza o GPS do dispositivo
GPS Externo	CheckboxPreference	Utiliza o GPS Externo via <i>Bluetooth</i>
Dispositivo GPS	PreferenceScreen	Subtela para conexão com dispositivo <i>bluetooth</i>
Largura do Implemento	EditTextPreference	Informa a largura do implemento agrícola
Distância entre antena GPS e implemento	EditTextPreference	Informa a distância que a antena do GPS se encontra do implemento agrícola.
Comprimento	ListPreference	Escolhe a unidade de medida da distância (km ou mihas)
Velocidade	ListPreference	Escolhe a unidade de medida da velocidade (Km/h ou mph)
GPS Externo	CheckboxPreference	Habilita para utilizar outro dispositivo GPS.
Sobre	PreferenceScreen	Exibe informações referentes ao aplicativo

Fonte: Elaborado pelo Autor (2018)

Listagem 1 – Trecho do código das preferências

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android">
3
4      <PreferenceCategory android:title="@string/pref_category_general_title">
5
6          <ListPreference android:title="@string/pref_list_format_coordinates_title"
7              android:key="@string/pref_list_format_coordinates_key"
8              android:defaultValue="0"
9              android:entries="@array/pref_list_format_coordinates_entries"
10             android:entryValues="@array/pref_list_format_coordinates_values" />
11     </PreferenceCategory>

```

Fonte: Elaborado pelo Autor (2018)

Em seguida é criada uma classe *SettingsFragment* que estende a classe *PreferenceFragment*. Está irá adicionar as preferências do arquivo XML, anteriormente definido, para o mostrar ao usuário.

Por fim, uma *SettingsActivity* (Listagem 2) é criada e o seu conteúdo é substituído pelo conteúdo do fragmento.

Listagem 2 – Trecho do código da *SettingsActivity*

```

8 public class SettingsActivity extends AppCompatActivity {
9
10
11
12     @Override
13     protected void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15
16         if (getSupportActionBar() != null) {
17             getSupportActionBar().setDisplayHomeAsUpEnabled(true);
18             getSupportActionBar().setDisplayShowHomeEnabled(true);
19         }
20         Fragment fragment = getSupportFragmentManager().findFragmentById(android.R.id.content);
21         if (fragment == null) {
22             getSupportFragmentManager().beginTransaction()
23                 .add(android.R.id.content, SettingsFragment.newInstance())
24                 .commit();
25         } else {
26             getSupportFragmentManager().beginTransaction()
27                 .replace(android.R.id.content, SettingsFragment.newInstance())
28                 .commit();
29         }
30     }

```

Fonte: Elaborado pelo Autor (2018)

3.4.5 Ativação do Sistema de Localização

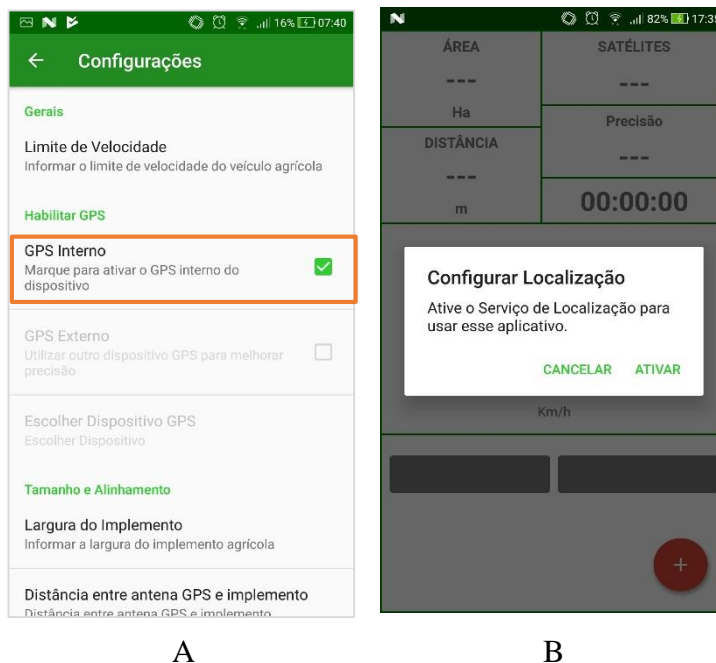
O aplicativo desenvolvido permite que o usuário tenha duas formas de receber a localização GPS uma utilizando um *hardware* externo com comunicação via *Bluetooth*, outra do próprio *smartphone/tablet*, dessa forma o usuário não fica dependente de possuir outro dispositivo para conseguir utilizar o aplicativo, no entanto, a precisão tende a ser menor com o GPS do dispositivo Android.

Para escolher o GPS do próprio *smartphone* ou *tablet*, ele irá até as preferências e no momento que ele escolhe a opção “GPS Interno”, a opção que habilita o GPS externo é desativada, conforme é possível visualizar na Figura 17 (A).

Ao habilitar o primeiro botão “GPS Interno”, se estiver com o GPS do dispositivo desligado, no momento que ir para a tela de navegação aparece um alerta solicitando a ativação do mesmo, conforme Figura 17 (B). Quando clicado em “Ativar”, a aplicação irá nos mostrar a tela de “Configuração do Android”, após ele ativar o *smartphone* leva alguns poucos segundos

para receber esses dados e assim a aplicação conseguir atualizar os campos, a partir disso a aplicação já estará funcionando normalmente.

Figura 17 – Ativação do GPS Interno



Fonte: Elaborado pelo Autor (2018)

Caso o usuário escolha por utilizar o GPS externo via conexão *Bluetooth*, novamente ele irá nas configurações e será preciso desabilitar a opção com o GPS interno para que possa ser habilitado o GPS Externo. Na Figura 18 é possível visualizar esta ação.

Figura 18 – Ativação do GPS Externo

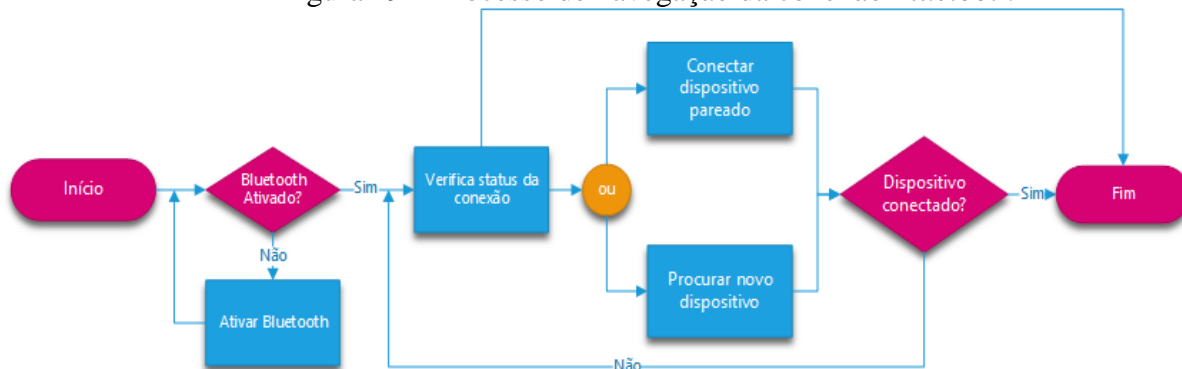


Fonte: Elaborado pelo Autor (2018)

Como é possível visualizar na Figura 18 após desabilitar o uso do GPS interno e clicar em “Conectar Dispositivo GPS” a opção para escolher um dispositivo fica disponível.

Devido ao processo de conexão via *Bluetooth* envolver várias etapas a Figura 19 ilustra o fluxo de navegação para realizar a conexão.

Figura 19 – Processo de navegação da conexão *Bluetooth*

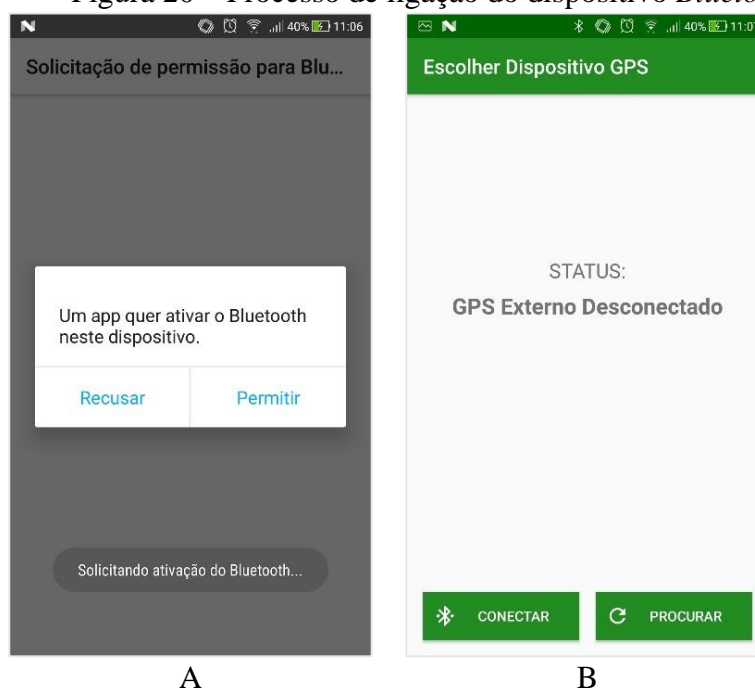


Fonte: Elaborado pelo Autor (2018)

Como é ilustrado na Figura 19, inicialmente é verificado se o *Bluetooth* do *smartphone* ou *tablet* está ligado, após isso é verificado se já está conectado a algum dispositivo. Caso não esteja é possível procurar por um novo dispositivo ou conectar a um dispositivo já pareado, depois de feita a conexão a aplicação já estará recebendo os dados do GPS.

Nas Figuras 20 e 21 é exemplificado o processo de conexão do *smartphone* com o dispositivo *Bluetooth*.

Figura 20 – Processo de ligação do dispositivo *Bluetooth*

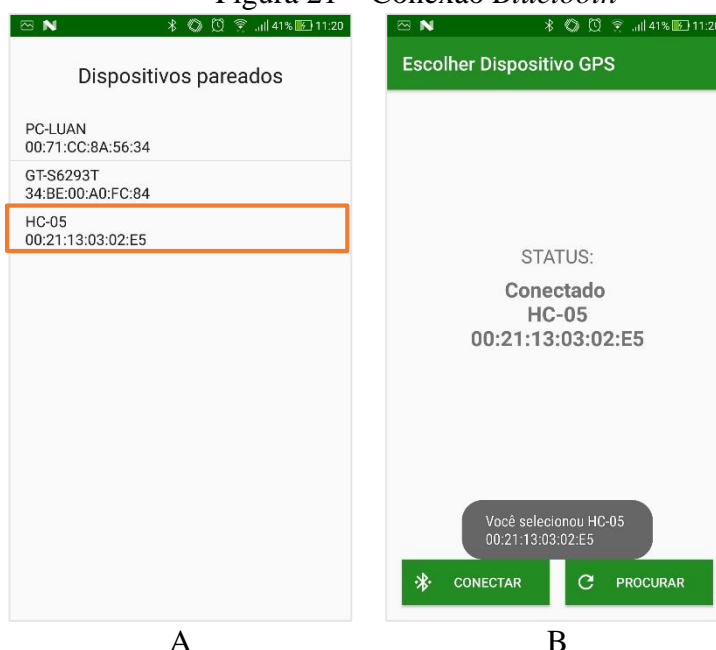


Fonte: Elaborado pelo Autor (2018)

Conforme é demonstrado na Figura 20 (A) primeiramente é verificado se o usuário está com o *Bluetooth* ativado, caso não esteja, uma notificação irá aparecer solicitando que o usuário ligue o *Bluetooth* do *smartphone*. Posteriormente, é exibido (Figura 20 (B)) para o usuário que ele não está conectado a nenhum dispositivo e dois botões de ação para ele realizar a conexão.

- **Botão Conectar:** Quando clicado nesse botão o Android irá buscar no gerenciador *Bluetooth* uma lista com os dispositivos já pareados.
- **Botão Procurar:** Quando o usuário clica nesse botão ele irá iniciar a descoberta de dispositivos *Bluetooth* exibindo uma lista com os dispositivos descobertos.

Figura 21 – Conexão *Bluetooth*



A

B

Fonte: Elaborado pelo Autor (2018)

Nesse exemplo, o dispositivo *Bluetooth*, no caso, o módulo HC-05 já estava pareado e já é exibido na lista dos dispositivos pareados (Figura 21 (A)), após clicar sobre ele é possível visualizar na Figura 21 (B) que ele se conectou com sucesso. Concluída a conexão, o *smartphone* já estará recebendo os dados do módulo *Bluetooth*.

Devido aos erros provenientes da falta de precisão do GPS, quando o usuário está a uma distância inferior a esse erro a barra de luz (Figura 13 e Quadro 2) perde o rumo e não fornece a direção correta. Essa margem de erro dependerá da precisão do GPS.

Como a aplicação requer uma conexão com o serviço GPS, o consumo da bateria do *smartphone* irá aumentar significativamente, por isso no momento da utilização independente se está utilizando o serviço GPS do próprio dispositivo ou utilizando um GPS externo é recomendado que ele esteja conectado a uma fonte de energia.

3.5 Dispositivo Bluetooth GPS Externo

A aplicação Android desenvolvida permite a comunicação com dispositivos *Bluetooth* para obtenção de dados de localização que sejam mais precisas que a localização através do GPS interno do próprio aparelho Android.

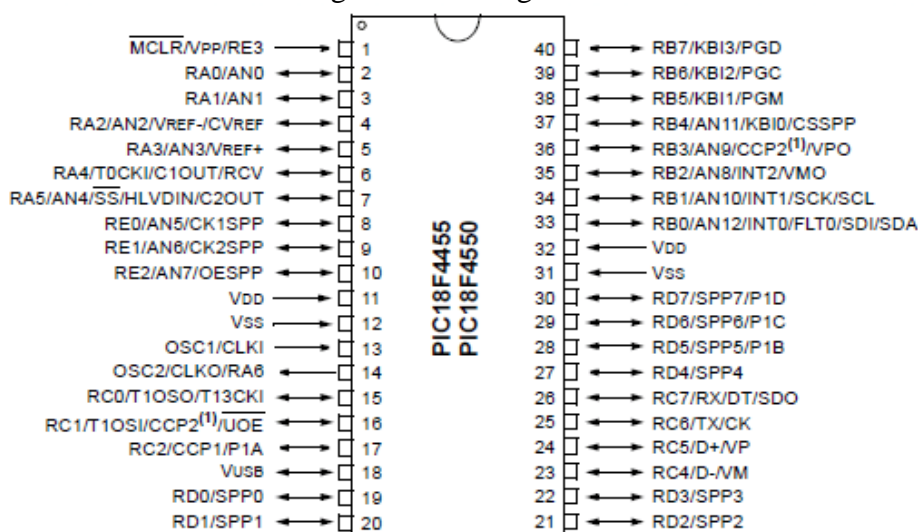
Esta seção pretende demonstrar o desenvolvimento de um dispositivo que integra um GPS e se comunica com o *smartphone* Android através da tecnologia *Bluetooth*.

O dispositivo projetado, é baseado no microcontrolador de 8 bits da Microchip PIC 18F4550. Este foi escolhido para o desenvolvimento do dispositivo dado as suas características, tais como:

- Possui 40 pinos, sendo configuráveis como I/O 35;
- 13 portas configuráveis como canais de entradas analógicas;
- Tensão de alimentação máxima 5.5V;
- Memória Flash de 32 KB;
- Memória RAM de 2048 Bytes;
- Memória EEPROM de 256 Bytes;
- USB 2.0;
- Desempenho de 48 MHz;
- Comunicação SPI, USART e I2C;

Na Figura 22 é apresentada a pinagem do microcontrolador PIC 18F4550 de 40 pinos, sendo até 35 de I/O.

Figura 22 – Pinagem PIC18F4550



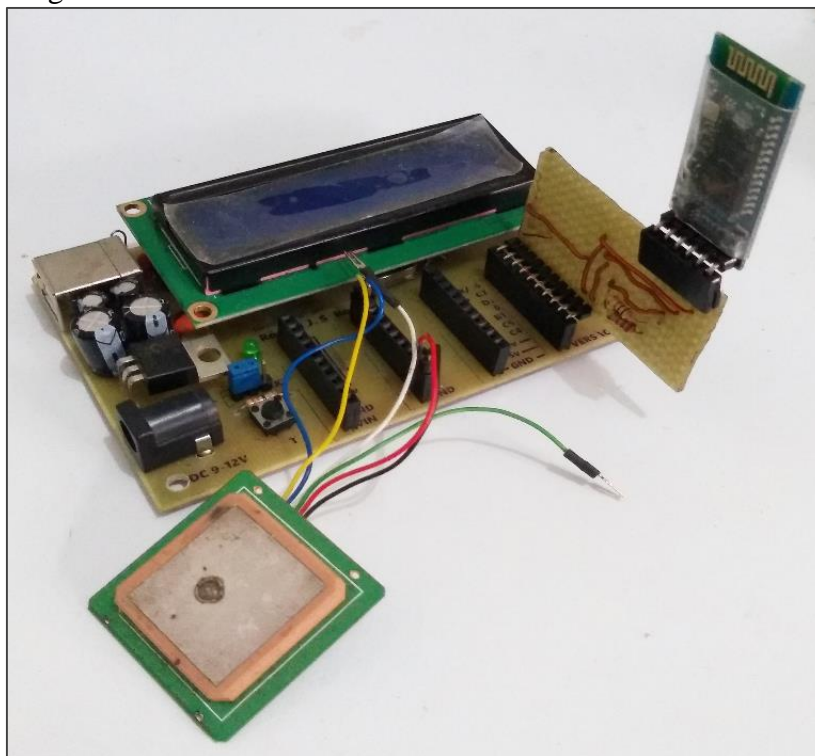
Apesar deste microcontrolador possuir algumas características que não serão necessárias para o desenvolvimento deste dispositivo em particular, o mesmo pode servir como base para outros dispositivos que possam vir no futuro a ser desenvolvidos.

Assim, o dispositivo desenvolvido é composto por:

- Placa de prototipagem com microcontrolador PIC 18F4550;
- Módulo GPS ME-1000;
- Módulo Bluetooth HC-05.

Com base no projeto do circuito inicial, o protótipo foi montado utilizando a placa de prototipagem Tenorino, como pode ser visualizado na Figura 23, pois esta atendia perfeitamente os requisitos do projeto.

Figura 23 – Placa Tenorino com o GPS e o módulo *Bluetooth*



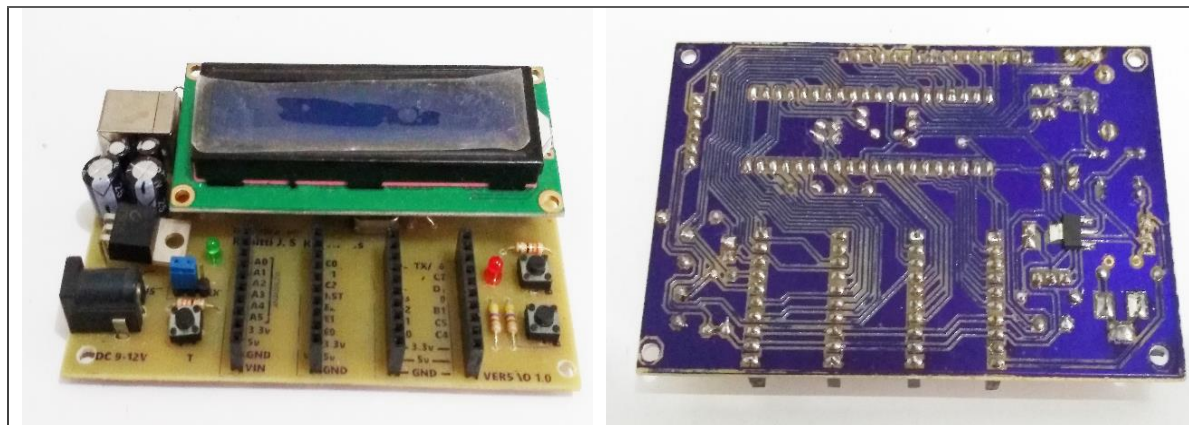
Fonte: Elaborado pelo Autor (2018)

3.5.1 Placa de prototipagem Tenorino

O Tenorino (Figura 24) é uma placa de prototipagem eletrônica desenvolvida na Universidade do Estado de Mato Grosso (UNEMAT). Sua 1ª versão foi desenvolvida pelo acadêmico Michael Henrique Tenório em 2012, tinha como objetivo auxiliar os alunos a construir e customizar seu próprio hardware de acordo com sua necessidade, posteriormente o

acadêmico Ronitti Juner Rodrigues em 2017 produziu uma 2ª versão, mais atualizada e com mais recursos, por isso ela foi escolhida para ser utilizada neste projeto. Essa placa que foi desenvolvida utilizando o microcontrolador PIC 18F4550, se assemelha com as placas de outras plataformas de prototipagem de hardware livre como a plataforma Arduino, que hoje é bem difundida e utilizada por desenvolvedores amadores e estudantes.

Figura 24 – Placa Tenorino



Fonte: Elaborado pelo Autor (2018)

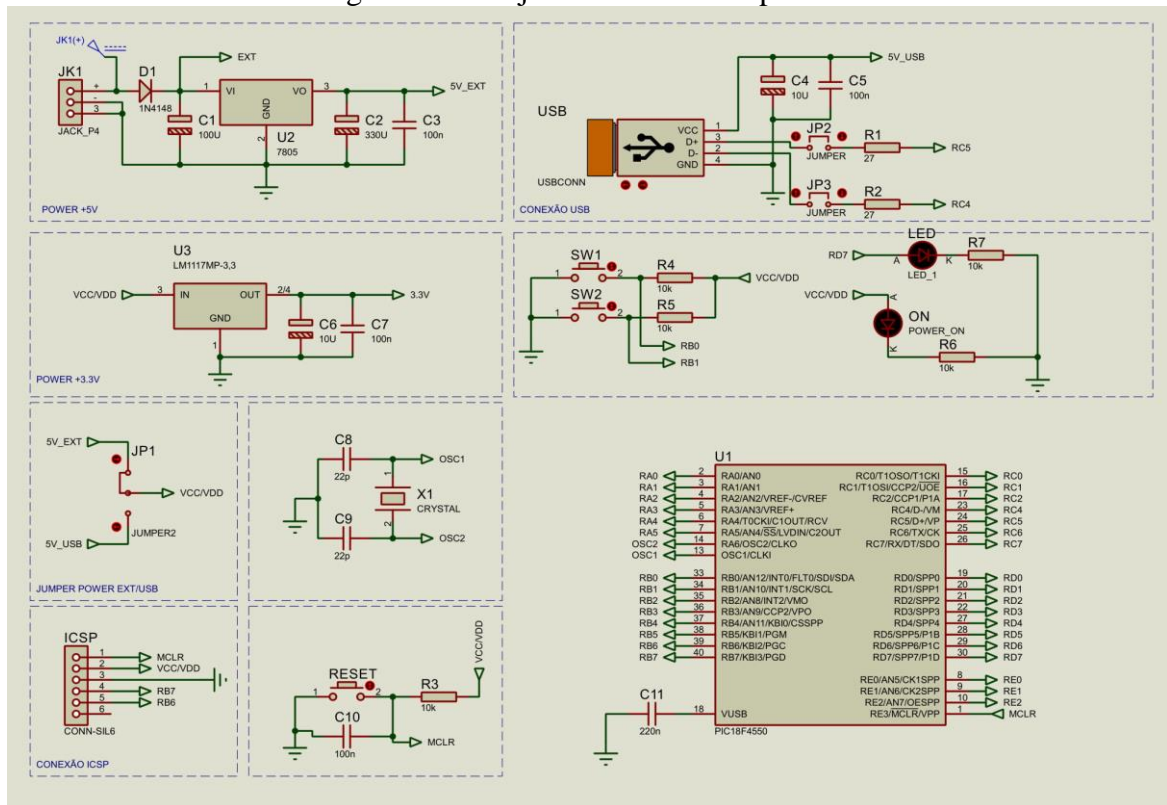
Conforme é possível visualizar na Figura 24, o Tenorino possui características que agilizam no desenvolvimento e prototipagem, tendo em vista, que não é necessário o desenvolvimento de uma placa para uma única finalidade, permitindo que em uma mesma placa seja adaptado vários projetos diferentes.

As principais características do Tenorino são:

- PIC 18F4550;
- Gravador *On-Board* (USB);
- *Display* de LCD 16x2;
- Entradas Analógicas;
- USB 2.0 permitindo que seja feito programas sofisticados comunicando via canal USB 2.0;
- Fonte de Alimentação externa ou USB;
- Tensão de alimentação 3.3V ou 5V;
- Push-Button;
- Reset, circuito *reset* manual para que seja possível inicializar a aplicação no momento em que desejar.

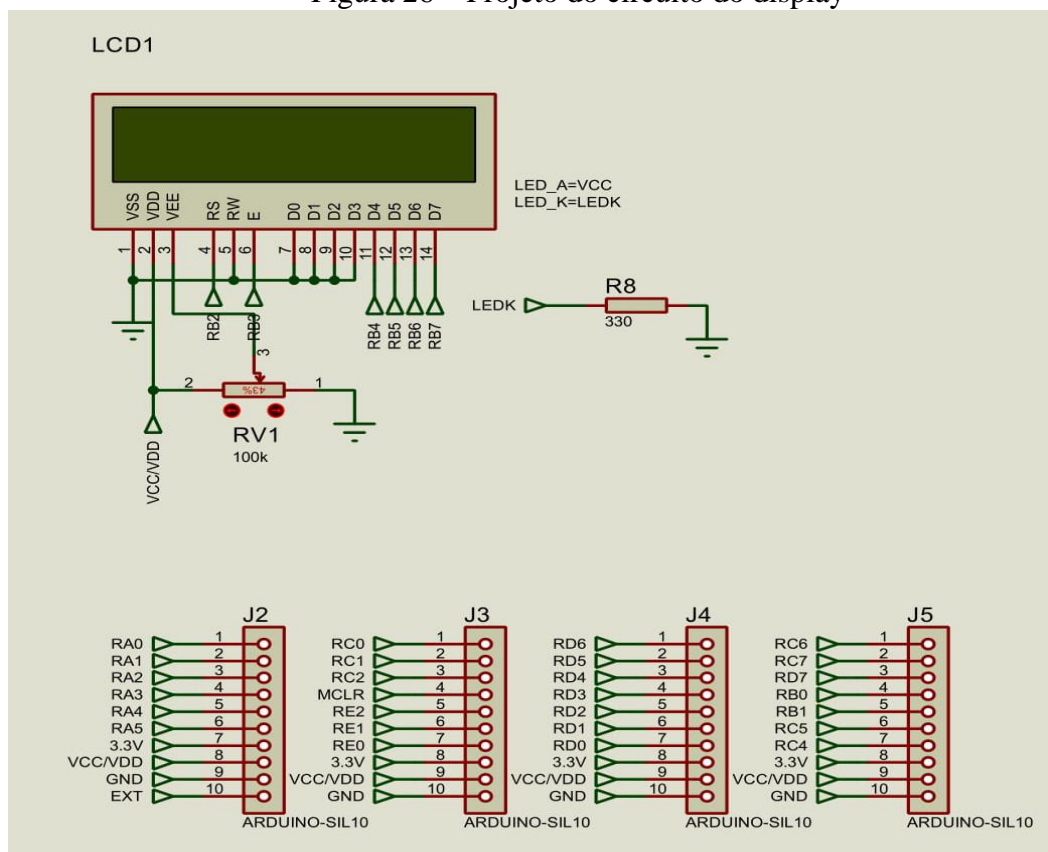
Na Figura 25 e Figura 26 é apresentado o circuito elétrico da placa Tenorino:

Figura 25 – Projeto do circuito da placa Tenorino



Fonte: Elaborado pelo Autor (2018)

Figura 26 – Projeto do circuito do display



Fonte: Elaborado pelo Autor (2018)

O Tenorino conta com um gravador *On-Board* via USB presente na placa, com um Boot Loader previamente gravado permite que ela seja compatível com diversos ambientes desenvolvimento.

3.5.2 Módulo GPS ME-1000RW

O módulo GPS escolhido para o trabalho foi o GPS ME-1000RW, por apresentar um bom desempenho, atender os requisitos do projeto e também por já possuir o mesmo sem a necessidade de compra.

O ME-1000RW possui uma antena acoplada que é conectada ao receptor através de um amplificador de baixo ruído, o receptor tem 51 canais de aquisição e 14 canais de rastreamento que são capazes de receber sinais de até 65 satélites. O equipamento tem um baixo consumo e a faixa de tensão suportada vai de 3.3V – 6.0V e possui um LED de indicação de status. O conector possibilita a saída tanto em nível LVTTTL quanto em nível RS232 (DATASHEET ME-1000RW, 2018). No Quadro 4 estão algumas especificações do módulo ME-1000RW.

Quadro 4 – Especificação do módulo ME-1000RW

Item	Especificação
Canais	65 canais
Sensibilidade	- 161 dBm
Frequência	L1 – 1.575,42 MHz
Tempo de início	Início Frio 29 segundos; Início Intermediário 28 segundos; Início Quente 1 segundo.
Reaquisição de Sinal	< 1 segundo.
Interface Serial	RS232 e LVTTTL
Precisão	Posição 5 metros Velocidade 0,1 m/s Tempo 300 ns
Datum	WGS-84
Taxa de Atualização	1 Hz
Protocolo	NMEA-0183 V3.01
Faixa de Tensão	3.6V a 6V
Sentença de Saída	GPGGA, GPGSA, GPRMC, GPVTG, GPGSV

Fonte: Adaptado de Datasheet ME-1000RW (2018)

O Quadro 5 apresenta como está organizada a comunicação do módulo GPS ME-1000RW, indicando as entradas e saídas VCC, GND, TX (transmissor) e RX (receptor).

Quadro 5 – Pinos do conector de saída

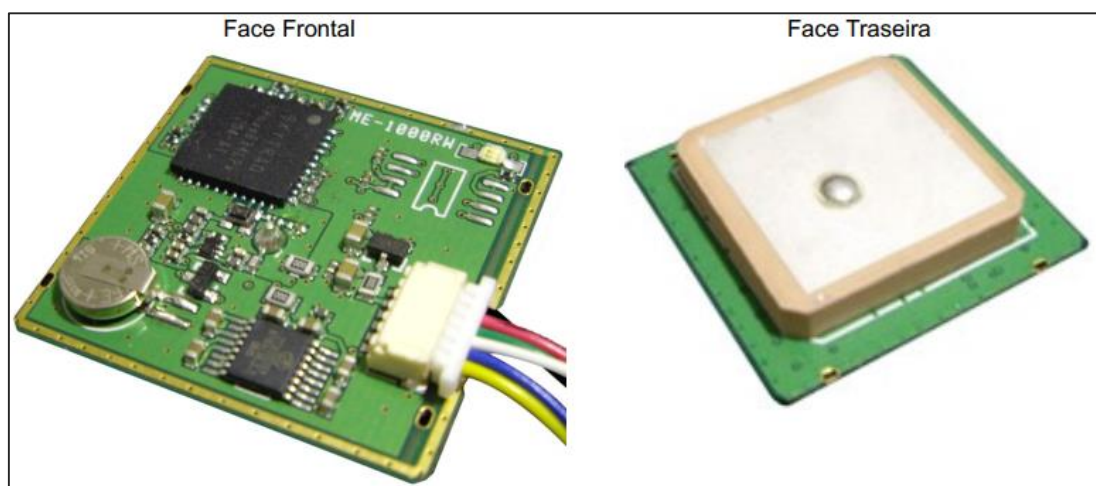
Pino	Descrição	Tipo	Função
1	TX LVTTL	O	Saída de dados serial UART (LVTTL) (amarelo)
2	RX LVTTL	I	Entrada de dados serial UART (LVTTL) (azul)
3	TX RS-232	O	Saída de dados serial RS-232 (branco)
4	RX RS-232	I	Entrada de dados serial RS-232 (verde)
5	VIN	I	Tensão de entrada 3.6 – 6V (vermelho)
6	GND	O	GND – Terra (preto)

I: Input, O: Output

Fonte: Adaptado de Datasheet ME-1000RW (2018)

Na Figura 27 está ilustrada a face frontal e traseira do módulo ME-1000RW.

Figura 27 – Visão do módulo ME-1000RW



Fonte: Adaptado de Datasheet ME-1000RW (2018)

3.5.3 Módulo Bluetooth HC-05

O módulo HC-05 (Figura 28) é um módulo SPP (*Serial Port Protocol*) *Bluetooth* de fácil utilização e baixo custo, possuindo alcance de até 10 metros. Normalmente é utilizado para adicionar conectividade *Bluetooth* a projetos embarcados, tendo em vista, que foi projetado para configuração de conexão serial sem fio. O módulo HC-05 pode funcionar como *Master* (mestre), ou seja, iniciar a conexão ou *Slave* (escravo), não podendo iniciar uma conexão e só se conecta quando o Master desejar (ITEAD STUDIO, 2010).

Em sua placa existe um regulador de tensão que permite alimentar com 3.3V a 5V, bem como um LED que indica se o módulo está pareado com outro dispositivo. Apesar de o módulo

poder ser alimentado com 5V, os pinos de RX/TX trabalham com 3.3V. Por este motivo é altamente recomendado o uso de um conversor de nível lógico, ele realizará as adequações de tensão (tanto de 5V para 3.3V, quanto o inverso) para que a comunicação ocorra corretamente.

Figura 28 – Módulo Bluetooth HC-05



Fonte: Elaborado pelo Autor (2018)

O módulo HC-05 vem pré-configurado pelo fabricante da seguinte forma:

- *Name: HC-05*
- *Baud rate padrão: 9600 bps e suporta baud rate: 9600, 19200, 38400, 57600, 115200, 230400, 460800;*
- *Data bits: 8;*
- *Stop bit: 1;*
- *Parity: No parity;*
- *PINCODE: "1234"*

O módulo HC-05 utilizado possui seis pinos, descritos no Quadro 6.

Quadro 6 – Pinos de ligação do módulo HC-05

Pino	Descrição
STATE	Estado de funcionamento do módulo
EM	Utilizado para habilitar o modo de comandos AT
RX	Entrada de dados (Ligado no pino TX do microcontrolador)
TX	Saída de dados – (Ligado no pino RX do microcontrolador)
GND	GND – Terra
VCC	Alimentação (entre 3.3 volts e 5 volts)

Fonte: Adaptado de Itread Studio (2010)

O Quadro 7 apresenta alguns comandos “AT” suportados pelo módulo HC-05, a resposta esperada e descrição de cada um dos comandos disponíveis.

Quadro 7 – Comandos AT do módulo HC-05

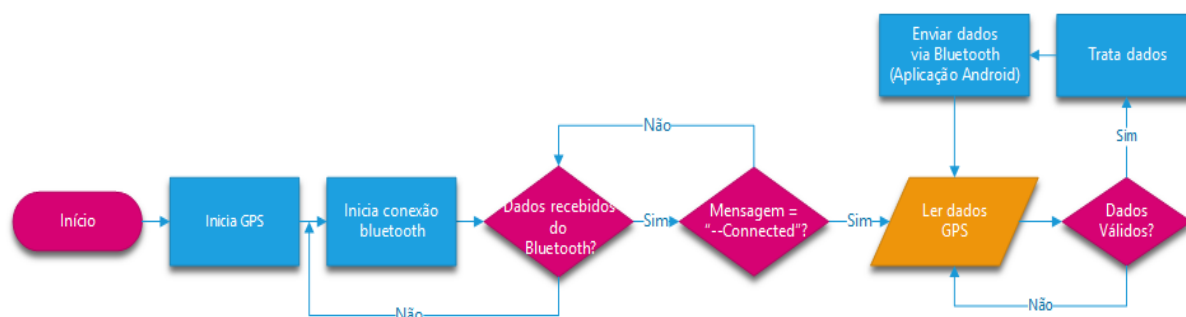
Comandos AT	Resposta esperada	Descrição
AT	OK	Verificar a comunicação
AT+ORGL	OK	Reseta o módulo para configurações originais
AT+VERSION?	+VERSION:<Param>	Versão do <i>firmware</i>
AT+NAME=<Param>	OK	Alteração do nome do módulo
AT+ADDR?	+ADDR:<Param>	Endereço do módulo
AT+PSWD=<Param>	OK	Alteração do PIN
AT+INIT	OK	Inicia a aplicação
AT+UART=<Param>	OK	Altera o BaudRate

Fonte: Adaptado de Itread Studio (2010)

3.5.4 Programação do microcontrolador

O desenvolvimento do programa (disponível para ser analisado no Apêndice C) para o microcontrolador passou pela configuração da conexão entre o módulo *Bluetooth* e a aplicação Android, para após iniciar com a recepção dos dados do GPS. O fluxograma visível na Figura 29 ilustra o funcionamento geral do programa do microcontrolador.

Figura 29 – Funcionamento geral do programa do microcontrolador



Fonte: Elaborado pelo Autor (2018)

A implementação do software embarcado aconteceu por etapas para verificação do funcionamento correto do dispositivo. Inicialmente, foram feitas as rotinas para receber os dados NMEA (nesse trabalho utilizou-se as *strings* GPRMC e GPGLA), posteriormente desenvolveu-se a rotina que verifica a validade dos dados recebidos, essa verificação acontece através de um campo da *string* GPRMC. Somente depois dos dados estarem válidos é que o dispositivo começa a efetuar o tratamento e extração da informação que será utilizada.

Após o desenvolvimento das rotinas responsáveis pela recepção e tratamento dos dados NMEA, foi desenvolvida a rotina principal que integra o módulo *Bluetooth*.

CONSIDERAÇÕES FINAIS

O objetivo deste trabalho foi desenvolver uma aplicação em conjunto com um sistema embarcado de localização que contribua para o avanço tecnológico na agricultura de precisão e possa ser uma ferramenta que irá auxiliar pequenos produtores a ter maior controle das atividades agrícolas em sua propriedade otimizando sua produtividade.

Sendo a primeira solução, o desenvolvimento de uma aplicação móvel destinada a *smartphone* e *tablets* com sistema operacional Android. A segunda solução um hardware composto por um módulo GPS e um módulo *Bluetooth*.

Em relação a utilização da plataforma Android, foi obtido um grande aprendizado nessa plataforma, a mesma se mostrou complexa, porém com uma grande documentação disponível e uma comunidade de desenvolvedores bem ativa.

A aplicação revelou-se funcional, no entanto, é necessário a realização de testes com usuário final em situação real a fim de se ter uma análise de usabilidade e se a aplicação atende todas as necessidades do condutor do implemento. Devido aos erros de precisão provenientes do GPS percebeu-se a necessidade de desenvolver algum algoritmo de correção que ajude a diminuir essa margem, mas pelo pouco tempo isso não foi possível.

No desenvolvimento do *hardware* não foi encontrada grandes dificuldade, em vista, que a utilização da placa de prototipagem Tenorino evitou a necessidade de ter conhecimentos específicos em eletrônica. A maior dificuldade encontrada foi com o desenvolvimento do código que envolveu a programação de várias rotinas de tratamento de dados e o desenvolvimento de um protocolo de comunicação entre *hardware* e *smartphone* para que se soubesse quando estava conectado.

Por se tratar de um protótipo, o sistema embarcado ainda precisa de algumas melhorias para transforma-se em um produto real. Primeiramente, o seu tamanho pode ser reduzido utilizando uma placa desenvolvida especificamente para essa finalidade e com algum microcontrolador com *Bluetooth Low Energy* (*Bluetooth* de baixo consumo de energia) no mesmo circuito integrado. Além da utilização de um GPS de melhor qualidade com uma antena externa que permita ao usuário fixa-la em cima do veículo.

O aplicativo possibilita ao usuário fazer o controle da semeadura, colheita ou pulverização, porém há recursos que poderão ser implementados para melhorar ainda mais o controle e a gestão de sua propriedade.

O desenvolvimento do aplicativo possuiu vários desafios para que pudesse ser criado, por este motivo a implementação de uma tela gráfica com animação do trajeto percorrido ou de

um banco de dados para armazenar as atividades realizadas não pode ser concretizada, todavia pode ser acrescentado em versões futuras do aplicativo, sem que haja problemas para o usuário.

Algumas das funcionalidades que podem ser acrescentadas em versões futuras:

- Banco de dados: A aplicação conta com a geração de relatório ao final do trabalho, porém o próximo passo é ele salvar esse relatório em uma base de dados para conseguir consultar posteriormente;
- Mapeamentos georreferenciados na tela: O usuário no momento da atividade ele não consegue visualizar por onde passou;
- Visualização do trajeto percorrido: Armazenar o trajeto percorrido pelo usuário e ao fim do trabalho conseguir exportar para ser visualizado em outros *softwares* de mapas como o Google Earth.
- Possibilidade de efetuar o trabalho em linhas curvas: A aplicação desenvolvida só permite que o usuário trabalhe em linhas retas para manter o paralelismo, acrescentar a funcionalidade de as linhas paralelas serem feitas também em curvas.

REFERÊNCIAS

- ANDROID DEVELOPERS. **Documentation for app developers**. jun 2018. Disponível em: <<https://developer.android.com/docs/?hl=pt-br>>. Acesso em: 25 jun. 2018.
- ANDROID STUDIO. **Android Studio: O IDE oficial do Android**. jun. 2018. Disponível em: <<https://developer.android.com/studio/?hl=pt-br>>. Acesso em: 25 jun. 2018.
- BAIO, F.H.R. et al. Avaliação da acurácia de uma barra de luz utilizada na agricultura de precisão em relação ao marcador de espuma. **Revista Brasileira de Engenharia Agrícola e Ambiental**, Campina Grande, v.5, n.2, p.357-360, 2001.
- BAIO, F.H.R.; ANTUNIASSI, U.R. Ensaio comparativo da acurácia de um sistema de direcionamento via satélite e por cabo de aço na orientação de máquinas agrícolas. **Energia na Agricultura**, Botucatu, v.23, p.60-73, 2008.
- BRANQUINHO, K. B. et al. Desempenho de uma semeadora-adubadora direta, em função da velocidade de deslocamento e do tipo de manejo da biomassa da cultura de cobertura do solo. **Revista Engenharia Agrícola**, v. 24, n. 02, p. 374-380, 2004.
- CAFÉ, Adriel Almeida. **Desenvolvimento de Cross-Platform Mobile Apps Utilizando o Titanium Mobile**. (2012). 19 f. Monografia (Graduação) – Curso de Sistemas de Informação, Faculdade Zacarias de Góes (fazag), Valença – BA, 2012. Disponível em: <<http://adrielcafe.com/images/adrielcafe/artigos/2012/12/tcc/adriel-tcc.pdf>>. Acesso em: 09 jan. 2018.
- CAMARGO, J. L. D.; SCHIEBELBEIN, L. M. Perdas na colheita mecanizada de milho em função da velocidade de deslocamento e rotação do cilindro. **Revista Scientia Rural**, Campos Gerais-MG, v. 2, 4. ed. jul./dez. 2011.
- CARVALHO, E. A.; ARAÚJO P. C. **Noções básicas de sistema de posicionamento global GPS**. Disciplina: Leituras Cartográficas e Interpretações Estatísticas II, Universidade Federal do Rio Grande do Norte, Natal, 2009, v. 12.
- CNN MONEY. **The end of the desktop PC**. 2010. Disponível em: <http://money.cnn.com/2010/07/20/technology/desktop_PC_death/index.htm>. Acesso em: 29 abr. 2018.
- COELHO, J. P. C.; SILVA, J. R. M. (Ed). **Agricultura de precisão**. 1ª. ed. Lisboa: Associação dos Jovens Agricultores de Portugal, 2009. Disponível em: <http://agrinov.ajap.pt/images/manuais/Manual_Agricultura_de_Precisao.pdf>. Acesso em: 14 abr. 2018
- CONAB, COMPANHIA NACIONAL DE ABASTECIMENTO. **Acompanhamento Safra brasileira Grãos: v.4 Safra 2016/17 – Nono levantamento**. Brasília: CONAB, p.1-161. jun. 2017. Disponível em: <http://www.conab.gov.br/OlalaCMS/uploads/arquivos/17_06_08_09_02_48_boletim_graos_junho_2017.pdf>. Acesso em: 03 dez. 2017.

CONCEIÇÃO, L. A. Máquinas agrícolas: Sistemas de apoio à condução por GPS. **Revista Abolsamia**, v 21, n. 92, p. 24-28. jul./ago. 2014.

COULOURIS, George; DOLLIMORE, Jean; KILDBERG, Tim. **Sistemas Distribuídos: Conceito e Projeto**. 4. ed. Porto Alegre: Bookman, 2007. p.790.

DATASHEET ME-1000RW. **Módulo GPS com antena acoplada ROM ME-1000RW**. 2018. Disponível em: <<http://www.mecomp.com.br/rumo/ME-1000RW.pdf>> Acesso em: 06 jun. 2018.

DIAS, O. V. et al. Distribuição de sementes de milho e soja em função da velocidade e densidade de semeadura. **Ciência Rural**, v. 39, n. 06, p. 1721-1728, 2009.

FERNANDES, F. **Análise comparativa de sistemas de direcionamento na operação de pulverização terrestre**. 2013. Dissertação (Mestrado em Agricultura de Precisão) – Universidade Federal de Santa Maria, Centro de Ciências Rurais, Programa de Pós – Graduação em Agricultura de Precisão, Santa Maria - RS, 2013.

FURLANI, C. E. A.; LOPES, A. SILVA, R. P. Avaliação de semeadora-adubadora de precisão trabalhando em três sistemas de preparo do solo. **Revista Engenharia Agrícola**, Jaboticabal, v. 25, n. 02, p. 458-464, 2005.

FIGUEIREDO, Carlos Maurício Seródio; NAKAMURA, Eduardo. **Computação Móvel: Novas Oportunidades e Novos Desafios**. T&C Amazônia, [S.I]: FUCAPI, Ano 1, nº 2, p. 16-28, jun. 2003. Disponível em: <https://issuu.com/revistatec/docs/revista_tec_ed02>. Acesso: 03 dez. 2017.

GARCIA, R. F. et al. Influência da velocidade de deslocamento no desempenho de uma semeadora-adubadora de precisão no Norte Fluminense. **Acta Scientiarum Agronomy**, Maringá, v. 33, nº 3, p. 417-422, jul. 2011. Disponível em: <<http://periodicos.uem.br/ojs/index.php/ActaSciAgron/article/view/6085>>. Acesso em: 28 mar. 2018.

GARTNER. **Gartner Says Worldwide Sales of Smartphones Recorded First Ever Decline During the Fourth Quarter of 2017**. fev. 2018. Disponível em: <<https://www.gartner.com/newsroom/id/3859963>>. Acesso em: 01 maio 2018.

GRAND VIEW RESEARCH. **Personal/Consumer Electronics Market Analysis By Product (Smartphones, Tablets, Desktops, Laptops/Notebooks, Digital Cameras, Hard Disk Drives, E-readers) and Segment Forecasts to 2020**. 2015. Disponível em: <<https://www.grandviewresearch.com/industry-analysis/personal-consumer-electronics.-market>>. Acesso em 29 de abr. 2018.

IBGE, INSTITUTO BRASILEIRO DE GEOGRAFIA E ESTATÍSTICA. **Levantamento Sistemático da Produção Agrícola**. Rio de Janeiro: IBGE, v. 30. n.10, p.1-83. out. 2017. Disponível em: <[ftp://ftp.ibge.gov.br/Producao_Agricola/Levantamento_Sistematico_da_Producao_Agricola_\[mensal\]/Fasciculo/lspa_201710.pdf](ftp://ftp.ibge.gov.br/Producao_Agricola/Levantamento_Sistematico_da_Producao_Agricola_[mensal]/Fasciculo/lspa_201710.pdf)>. Acesso em: 02 dez. 2017.

ITEAD STUDIO. **HC-05 – Bluetooth to Serial Port Module**. 2010. Disponível em: <<http://www.electronicaestudio.com/docs/istd016A.pdf>> Acesso em 06 de jun. 2018.

KANTAR. **Smartphone OS sales market share**. jun. 2017. Disponível em: <<https://www.kantarworldpanel.com/global/smartphone-os-market-share/>>. Acesso em: 29 out. 2017.

KIRSCHNER, S. F. **Um sistema de auxílio à coleta de dados na área de agricultura de precisão baseado em aplicações móveis**. 2012. 72 f. Dissertação (Ciência da Computação). Departamento de Ciências Exatas e Engenharias, Universidade Regional do Noroeste do Estado do Rio Grande do Sul. [Rio Grande do Sul], 2012. Disponível em: <http://bibliodigital.unijui.edu.br:8080/xmlui/bitstream/handle/123456789/1556/tcc_sabrina.p%20df?sequence=1>. Acesso em: 14 abr. 2018

LAMPARELLI, R. A. C. **Agricultura de Precisão**. Agência Embrapa de Informação Tecnológica, Brasília: Embrapa, [2012?]. Disponível em: <http://www.agencia.cnptia.embrapa.br/gestor/cana-de-acucar/arvore/CONTAG01_72_711200516719.html>. Acessado em 14 abr. 2018.

MARTINS, M. B. et al. Relação entre velocidade de deslocamento, rendimento da cana-de-açúcar e o consumo de combustível da colhedora. **Revista de Agricultura Neotropical**, Cassilândia-MS, v. 4, n. 1, p. 88–91, jan./mar. 2017.

MEIRELLES, Fernando S. **Pesquisa Anual do Uso de TI**. 28^a ed. São Paulo: FGV-EASP-GVcia, 2017. Disponível em: <<http://easp.fgvsp.br/ensinoeconhecimento/centros/cia/pesquisa>>. Acesso em: 03 dez. 2017.

MELLO, A. J. R. et al. Produtividade de híbridos de milho em função da velocidade de semeadura. **Revista Engenharia Agrícola**, v. 27, n. 02, p. 479-486, 2007.

MICROCHIP TECHNOLOGY. **PIC18F2455/2550/4455/4550 Data Sheet**. 2009. Disponível em: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39632e.pdf>>. Acesso em: 06 jun. 2018.

RECH, Willian Rodrigues da Fonseca. **Comercialização de software em plataforma mobile: um estudo de caso aplicado ao android**. 2013. 64 f. Trabalho de Conclusão de Curso (Especialização) – Especialização em Gestão de Tecnologia da Informação, Universidade Federal de Santa Maria, Frederico Westphalen, RS, 2013. Disponível em: <http://repositorio.ufsm.br/bitstream/handle/1/539/Rech_William_Rodrigues_da_Fonseca.pdf?sequence=1>. Acesso em: 04 dez. 2017.

RESENDE, Á. V. et al. **Agricultura de Precisão no Brasil: Avanços, Dificuldades...** In: XVIII REUNIÃO BRASILEIRA DE MANEJO E CONSERVAÇÃO DO SOLO E DA ÁGUA, 2010, Teresina. **Anais...** Teresina: Embrapa, 2010. Disponível em: <<https://www.alice.cnptia.embrapa.br/alice/bitstream/doc/870646/1/Agriculturaprecisao.pdf>>. Acesso em: 14 abr. 2018

SANTOS, A. J. M. et al. Análise espacial da distribuição longitudinal de sementes de milho em uma semeadora-adubadora de precisão. **Bioscience Journal**, Uberlândia, v. 27, n. 01, p. 16-23, 2011. Disponível em:

<<http://www.seer.ufu.br/index.php/biosciencejournal/article/view/7355>>. Acesso em: 05 abr. 2018.

SENAR, Serviço Nacional de Aprendizagem Rural. **Máquinas Agrícolas: Tecnologias de Precisão**. Brasília: SENAR, v. 159, p. 76, 2012. Disponível em: <http://www.senar.org.br/sites/default/files/158_-_maquinas_ap_a5.pdf>. Acesso em: 14 abr. 2018.

STARTAGRO. **10 coisas que você não sabia sobre o produtor rural e a tecnologia**. set. 2016. Disponível em: <<http://www.startagro.agr.br/o-produtor-rural-e-a-tecnologia/>>. Acesso em: 04 dez. 2017.

UNIVERSOAGRO. **Produtor rural usa cada vez mais internet para se informar**. jun. 2017. Disponível em: <<http://www.uagro.com.br/editorias/agricultura/2017/06/01/produtor-rural-usa-cada-vez-mais-internet-para-se-informar.html>>. Acesso em: 04 dez. 2017.

AGROCOMPANY, 2018 Disponível em: <<http://www.agrocompany.agr.br/agricultura-de-precisao.html>> Acesso em: 14 abr. 2018.

Google Play

Disponível em: <https://play.google.com/store/apps/details?id=lt.noframe.farmisfieldnavigator.free&hl=pt_BR> Acesso em: 20 abr. 2018.

Efarmer Disponível em: <<https://efarmer.mobi>>. Acesso em: 20 abr. 2018.

Machineyeri Guide Disponível em: <<http://machineryguideapp.com/en>>. Acesso em: 20 abr. 2018

APÊNDICE A - Documento de Levantamento de Requisitos

DOCUMENTO DE LEVANTAMENTO DE REQUISITOS

1 Introdução

1.1 Propósito

Este documento especifica os requisitos a serem desenvolvidos no aplicativo móvel, fornecendo ao desenvolvedor as informações necessárias para o projeto e implementação.

2 Visão Geral do Produto

O aplicativo é uma ferramenta que auxilia o condutor do implemento agrícola a manter o alinhamento nas atividades de semeadura, colheita ou pulverização. Além de permitir que ele saiba a qual a área que ele trabalhou, a distância, o tempo.

3 Requisitos Funcionais

São descritos a seguir os requisitos funcionais a serem implementados.

3.1 <RF001> Medição da Área

O aplicativo deve mediar a área de trabalho em hectares.

3.2 <RF002> Efetuar trabalhos em Retas AB

O aplicativo deve permitir marcar pontos A e B e assim traçar retas paralelas para o alinhamento e paralelismo do implemento agrícola no campo.

3.3 <RF003> Tempo de Atividade

O aplicativo deve contar o tempo total em que o usuário esteve em atividade.

3.4 <RF004> Controle de Velocidade

O aplicativo deve informar para o usuário a velocidade e se ele está dentro do limite estabelecido.

3.5 <RF005> Distância Percorrida

O aplicativo deve calcular a distância percorrida pelo veículo.

3.6 <RF006> Gerar relatório

O aplicativo ao final do trabalho deve gerar um relatório com as informações relativas ao serviço.

3.7 <RF007> Escolher o serviço GPS

O aplicativo deve permitir escolher entre o GPS do dispositivo ou um GPS externo através do Bluetooth

3.8 <RF008> Largura do implemento

O aplicativo deve permitir o usuário informar qual a largura do implemento agrícola.

4 Requisitos Não Funcionais

São descritos a seguir os requisitos não funcionais a serem implementados.

4.1 <NF001> Usabilidade

A interface com o usuário é de vital importância para o sucesso da aplicação, e deve ser de fácil manipulação.

4.2 <NF002> Plataforma

A aplicação é para o sistema operacional Android, seu código é desenvolvido na linguagem de programação Java e API 15 ou superior.

APÊNDICE B – Código XML do Layout da Figura 13 (A)

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:fab="http://schemas.android.com/apk/res-auto"
    android:background="@color/colorGreen"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"

    tools:context="com.studio.luanfssilva.tractornavigator.NavigatorActivity"
>
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"
        android:layout_margin="@dimen/margin"
        android:layout_weight="2">
        <LinearLayout
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_marginRight="@dimen/margin"
            android:orientation="vertical"
            android:layout_weight="1">
            <TextView
                android:id="@+id/textView3"
                style="@style/ThemeTextViewFixed"
                android:text="@string/text_area" />
            <TextView
                android:id="@+id/textAreaID"
                style="@style/ThemeTextView"
                android:text="@string/no_data"
                android:gravity="center" />
            <TextView
                android:id="@+id/textUnitAreaID"
                style="@style/ThemeTextViewFixed"
                android:text="@string/unit_area" />
            <TextView
                android:layout_marginTop="@dimen/margin"
                android:id="@+id/textDistance"
                style="@style/ThemeTextViewFixed"
                android:text="@string/text_distance" />
            <TextView
                android:id="@+id/textDistanceID"
                style="@style/ThemeTextView"
                android:layout_height="match_parent"
                android:text="@string/no_data" />
            <TextView
                android:id="@+id/textUnitDistanceID"
                style="@style/ThemeTextViewFixed"
                android:text="@string/unit_distance" />
        </LinearLayout>
    </LinearLayout>

```



```

        android:layout_width="wrap_content"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_weight="1">
        <TextView
            android:id="@+id/textSat"
            style="@style/ThemeTextViewFixed"
            android:text="@string/text_sat" />
        <TextView
            android:id="@+id/textSatID"
            style="@style/ThemeTextView"
            android:text="@string/no_data" />
        <TextView
            android:id="@+id/textAccuracy"
            android:layout_marginTop="@dimen/margin"
            style="@style/ThemeTextViewFixed"
            android:text="@string/text_hdop" />
        <TextView
            android:id="@+id/textAccuracyID"
            style="@style/ThemeTextView"
            android:layout_marginBottom="@dimen/margin"
            android:text="@string/no_data" />
        <TextView
            android:id="@+id/textTimeID"
            style="@style/ThemeTextView"
            android:background="@color/colorWhite"
            android:layout_weight="1"
            android:text="00:00:00" />
    </LinearLayout>
</LinearLayout>
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical"
        android:layout_marginLeft="@dimen/margin"
        android:layout_marginRight="@dimen/margin"
        android:layout_marginBottom="@dimen/margin">
        <TextView
            android:id="@+id/textSpeedID"
            style="@style/ThemeSpeed"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="@string/no_data" />
        <TextView
            android:id="@+id/textUnitSpeedID"
            android:background="@color/colorWhite"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:gravity="center_horizontal"
            android:layout_weight="1.7"

```

```

        android:text="@string/unit_speed" />
    <LinearLayout
        android:background="@color/colorWhite"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_marginTop="@dimen/margin"
        android:orientation="horizontal"
        android:layout_weight="1"
        android:paddingTop="@dimen/fab_margin">
        <ProgressBar
            android:id="@+id/progressBarE"
            android:layout_weight="1"
            style="@style/MyProgressBarTwo"
            android:rotation="180"
            android:layout_marginStart="8dp"
            android:layout_marginLeft="8dp"
            android:layout_marginEnd="4dp"
            android:layout_marginRight="4dp" />
        <ProgressBar
            android:id="@+id/progressBarD"
            style="@style/MyProgressBarTwo"
            android:layout_weight="1"
            android:layout_marginStart="4dp"
            android:layout_marginLeft="4dp"
            android:layout_marginEnd="8dp"
            android:layout_marginRight="8dp" />
    </LinearLayout>
</LinearLayout>
<com.github.clans.fab.FloatingActionMenu
    android:id="@+id/fab_menu"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="end|bottom"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:paddingBottom="@dimen/fab_margin"
    android:paddingRight="@dimen/fab_margin"
    fab:menu_backgroundColor="@android:color/transparent"
    fab:menu_fab_label="@string/fab_menu"
    fab:fab_colorPressed="#E75043"
    fab:fab_colorRipple="#99FFFFFF"
    fab:fab_showShadow="true"
    fab:menu_labels_colorNormal="#333333"
    fab:menu_labels_colorPressed="#444444"
    fab:menu_labels_colorRipple="#66FFFFFF"
    fab:menu_labels_showShadow="true"
    fab:menu_labels_maxLines="-1"
    fab:menu_labels_position="left"
    fab:menu_openDirection="up"
    fab:fab_shadowColor="#66000000"
    fab:menu_labels_ellipsize="end"
    fab:menu_labels_singleLine="true"
    tools:layout_editor_absoluteY="8dp"
    tools:layout_editor_absoluteX="8dp">
    <com.github.clans.fab.FloatingActionButton

```

```

        android:id="@+id/fabStop"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="gone"
        android:src="@drawable/ic_stop_24dp"
        fab:fab_label="@string/fab_stop"
        fab:fab_size="mini" />
    <com.github.clans.fab.FloatingActionButton
        android:id="@+id/fabPointB"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:visibility="gone"
        android:src="@drawable/ic_b_point"
        fab:fab_label="@string/fab_pointB"
        fab:fab_size="mini" />
    <com.github.clans.fab.FloatingActionButton
        android:id="@+id/fabPointA"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_a_point"
        fab:fab_label="@string/fab_pointA"
        fab:fab_size="mini" />
    <com.github.clans.fab.FloatingActionButton
        android:id="@+id/fabStart"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_play_arrow_24dp"
        fab:fab_label="@string/fab_start"
        fab:fab_size="mini" />
    <com.github.clans.fab.FloatingActionButton
        android:id="@+id/fabSettings"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_settings_24dp"
        fab:fab_label="@string/btn_settings"
        fab:fab_size="mini" />
    </com.github.clans.fab.FloatingActionButton>
</FrameLayout>
</LinearLayout>

```

APÊNDICE C – Código Fonte Sistema Embarcado

```

'*****
'* Name      : Code_GPS_Bluetooth.BAS
'* Author    : [Luan Felipe dos S. da Silva]
'* Notice    : Copyright (c) 2018 [Luan Felipe dos S. da Silva]
'* Date      : 21/04/2018
'* Version   : 1.4
'* Notes     : Programa para receber dados do GPS e enviar via
'*           : bluetooth para o smartphone
'*****

Device = 18F4550
Xtal 48

' Atribuições dos pinos LCD
Declare LCD_DataUs = 75
Declare LCD_DTPin = PORTB.4
Declare LCD_RSPin = PORTB.2
Declare LCD_ENPin = PORTB.3
Declare LCD_Interface = 4
Declare LCD_Lines = 2
Declare LCD_Type = 0

TRISB = 0
TRISC = %01000001
TRISD = 0
All_Digital TRUE

Symbol LED = PORTD.7
Symbol BT_TX = PORTC.6
Symbol BT_RX = PORTC.7 'Receptor Bluetooth'
Symbol GPS_TX = PORTC.0
Symbol GPS_RX = PORTC.1 'Receptor GPS

Symbol TRUE = 1
Symbol FALSE = 0

'-----Constantes de controle-----'
'Dim NOTCONNECTED As String * 15
'Dim CONNECTED As String * 15
'CONNECTED = "--Connected"
'NOTCONNECTED = "--NotConnected"
Dim isConnection As String * 15

'-----Variaveis GPS-----'
Dim nmea As String * 84
Dim iniCampo[13] As Byte 'Marca o inicio dos campos de dados
Dim sentence As Byte 'Informa qual sentença é RMC(1) ou GGA(2)
Dim statusDados As Byte

'---- Dados RMC ----
Dim hora As String * 2
Dim minutos As String * 2
Dim segundos As String * 2
Dim verificaDados As String * 1
Dim lat As String * 9

```

```

Dim long As String * 10
Dim hemisferioNS As String * 1
Dim hemisferioEW As String * 1
Dim vel As String * 5
Dim curso As String * 5
Dim dia As String * 2
Dim mes As String * 2
Dim ano As String * 2

'---- Dados GGA -----
Dim alt As String * 5
Dim sat As String * 2
Dim hdop As String * 3
Dim checksum As String * 2

Clear

Print At 1,1, "Iniciando GPS..."
Print At 2,1, "Versao 0.1"
DelayMS 4000
Cls

main:

    Print At 1,1, "Aguardando..."
    Print At 2,1, "Parear Bluetooth"

    SerIn BT_RX,84,[isConnection] 'Aguarda mensagem de controle
    'SerOut BT_TX,84,[isConnection,13,10] 'Confirma com o Dispositivo a
    conexão

    If isConnection == "--Connected" Then
        Cls
        Print At 1,1, "Bluetooth"
        Print At 2,1, "Conectado!!"
        DelayMS 3500
        GoTo dadosGPS
    Else If isConnection == "--NotConnected" Then
        Cls
        Print At 1,1, "Nao Conectado..."
        Print At 2,1, "Tente Novamente"
        DelayMS 2000
    Else
        Print At 1,1, "Aguardando..."
        Print At 2,1, "Parear Bluetooth"
        DelayMS 2000
        Cls
    EndIf

GoTo main

'----- Responsável por ficar enviando os dados via bluetooth para o GPS---
'--'
dadosGPS:

    SerIn GPS_RX, 84,[Wait("$GPRMC"), nmea]
    sentence = 1
    'SerOut BT_TX, 84,["$GPRMC",nmea,13,10]
    GoSub dividi_dados

```

```

GoSub valida_dados

SerIn GPS_RX, 84, [Wait("$GPGGA"), nmea]
sentence = 2
'SerOut BT_TX, 84,["$GPGGA",nmea,13,10]
GoSub dividi_dados
GoSub trata_dados

If statusDados == TRUE Then
    SerOut
    84,["$GPS"," ",hora,minutos,segundos," ",verificaDados," ",lat," ",hemisf
erioNS," ",long," ",hemisferioEW," ",vel," ",curso," ",sat," ",hdop," ",a
lt," ",dia,mes,ano,13,10]
    Print At 1,1, "Enviando..."
    Print At 2,1, "Dados GPS!!"
Else
    Cls
    Print At 1,1, "Aguardando..."
    Print At 2,1, "Dados do GPS"
EndIf

GoTo dadosGPS

'-----Dividi Dados-----'
'Separa os campos da string NMEA
dividi_dados:
    Dim i As Byte
    Dim j As Byte
    Dim k As Byte
    i = 0
    j = 0

    Repeat
        If nmea[j]=="," And nmea[j+1] != "*" Then
            iniCampo[i] = j+2
            Inc i 'Incrementa a variavel
        EndIf
        Inc j
    Until nmea[j] == "*", j==83, nmea[j] == "\0"

Return

'-----Valida Dados -----'
' Verifica se os dados NMEA estão validos
valida_dados:
    verificaDados = Mid$ (nmea,iniCampo[1],1)
    'Se os dados estiverem validos já chama a rotina de tratamento
    If verificaDados == "A" Then
        statusDados = TRUE
        GoSub trata_dados
    Else
        'Print 1,1, "Dados Invalidos"
        statusDados = FALSE
    EndIf
Return

'-----Tratamento dos Dados -----'
'Extrair os dados da string NMEA
trata_dados:
    'RMC

```

```

If sentence == 1 Then
  ' Print 1,1, "Entrou RMC"
  hora = Mid$ (nmea,iniCampo[0],2)
  minutos = Mid$ (nmea,iniCampo[0]+2,2)
  segundos = Mid$ (nmea,iniCampo[0]+4,2)
  hemisferioNS = Mid$ (nmea,iniCampo[3],1)
  hemisferioEW = Mid$ (nmea,iniCampo[5],1)
  lat = Mid$ (nmea,iniCampo[2],9)
  long = Mid$ (nmea,iniCampo[4],10)
  dia = Mid$ (nmea,iniCampo[8],2)
  mes = Mid$ (nmea,iniCampo[8]+2,2)
  ano = Mid$ (nmea,iniCampo[8]+4,2)
  vel = Mid$ (nmea,iniCampo[6],5)
  curso = Mid$ (nmea,iniCampo[7],5)

  'GGA
Else
  'Print 2,1, "Entrou GGA"
  sat = Mid$ (nmea,iniCampo[6],2)
  hdop = Mid$ (nmea,iniCampo[7],3)
  alt = Mid$ (nmea,iniCampo[8],5)
EndIf
Return

```