

UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ALISSON GUIMARA PEREIRA

**SISTEMA DE IRRIGAÇÃO AUTOMATIZADO UTILIZANDO O CONCEITO DE IOT
PARA TOMADA DE DECISÃO**

CRICIUMA
2020

ALISSON GUIMARA PEREIRA

**SISTEMA DE IRRIGAÇÃO AUTOMATIZADO UTILIZANDO O CONCEITO DE IOT
PARA TOMADA DE DECISÃO**

Trabalho de Conclusão de Curso, apresentado para obtenção do grau de Bacharel no curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. Me. Gustavo Bisognin

**CRICIUMA
2020**

ALISSON GUIMARA PEREIRA

**SISTEMA DE IRRIGAÇÃO AUTOMATIZADO UTILIZANDO O CONCEITO DE
IOT PARA TOMADA DE DECISÃO**

Trabalho de Conclusão de Curso aprovado
pela Banca Examinadora para obtenção do
Grau de Bacharel, no Curso de Ciência da
Computação da Universidade do Extremo
Sul Catarinense, UNESC, com Linha de
Pesquisa em Engenharia de Software.

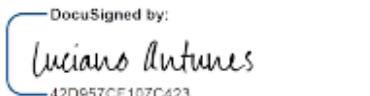
Criciúma, 06 de agosto de 2020.

BANCA EXAMINADORA

DocuSigned by:

0353997087914DS...

Prof. Gustavo Bisognin - Me - (UNESC) - Orientador

DocuSigned by:

42D957CF107C423...

Prof. Luciano Antunes - Me - (UNESC)

DocuSigned by:

0D6DD42F4F2F4BE...

Prof. Sérgio Coral - Esp - (UNESC)

RESUMO

O crescimento acentuado da população mundial está gerando cada vez mais demanda por produção de alimentos. O desafio é aumentar a produtividade na mesma área cultivada, uma das principais formas de conseguir isto é através do uso da irrigação, que pode aumentar em até 2,5 vezes a produção em relação a áreas sequeiras. O modelo proposto neste trabalho é o uso de IoT na irrigação, realizando o monitoramento da umidade do solo e da previsão do tempo para decidir o melhor momento para irrigar, gerando economia de água e de recursos. A versatilidade da Raspberry Pi, integrando sensores de umidade de solo e a API de previsão de tempo Advisor, possibilitaram a construção do modelo proposto. Para realizar o monitoramento da irrigação, foi desenvolvido um aplicativo *mobile* em Android, que pode gerar gráficos de consumo de água e acompanhamento dos dados referentes ao talhão cadastrado. O resultado foi de acordo com a lógica programada para realizar uma irrigação, porém verificou-se que o sensor usado tem a vida útil curta e deve ser trocado por outro modelo numa aplicação á campo. O R^2 , que é o coeficiente de determinação, obtido na calibração do sensor foi de 93,9%, podendo ser melhorado com mais testes de calibragem.

Palavras-chave: Irrigação. Sensores de umidade. Raspberry Pi. Aplicativo mobile.

ABSTRACT

The strong growth of the world population is generating more demand in food production. The challenge is to increase productivity in the same cultivated area, and one of the solutions is through the use of irrigation, which can be increased by up to 2.5 times the production in relation to dry areas. However, irrigation generates expenditure of water and resources such as electricity and fuel, if not used correctly. So the proposed model was to use the Raspberry Pi as a controller board, integrating soil moisture sensors and the Advisor weather forecast API, applying the concept of IoT in irrigation, so the decision to irrigate can be taken based on moisture and forecast data, generating reports of humidity and water consumption through a mobile application. The result was according to the logic programmed to carry out an irrigation, but it was found that the sensor used has a short useful life and must be replaced by another model in a field application. The R^2 , which is the determination coefficient, obtained in the sensor calibration was 93.9%, which can be improved with more calibration tests.

Keywords: Irrigation. Humidity sensors. Raspberry Pi. Mobile Application.

LISTA DE ILUSTRAÇÕES

Figura 1 - Irrigação por inundaçāo	16
Figura 2 - Irrigação Sub-superficial	17
Figura 3 - Aspersão convencional	17
Figura 4 - Irrigação localizada por gotejamento	18
Figura 5 - Posição dos instrumentos de medição.....	20
Figura 6 - Versões do Raspberry Pi	26
Figura 7 - Retorno de previsão 72 horas.....	32
Figura 8 - Retirada de amostra de solo	37
Figura 9 - Laudo da análise de solo	38
Figura 10 - Indices de CC e PMP	39
Figura 11 - Android Things Console.....	40
Figura 12 - Interface do Android Things	40
Figura 13 - Pinos GPIO	43
Figura 14 - Sensor de umidade	43
Figura 15 - Pinos do ADC0832.....	44
Figura 16 - Configuração do conversor ADB0832.....	45
Figura 17 - relação entre umidade real x valor do sensor	46
Figura 18 - gráfico da reta de regressão	46
Figura 19 - Classe Interface	48
Figura 20 - Dados salvos pela Raspberry	49
Figura 21 - conexão do módulo relé	50
Figura 22 - Tela inicial	51
Figura 23 - Cadastro de talhāo.....	51
Figura 24 - Dados salvos do talhāo.....	52
Figura 25 - Cadastro de plantas	52
Figura 26 - Dados salvos da planta.....	53
Figura 27 - Relatório do sensor	54
Figura 28 - Relatório de gasto de água	54
Figura 29 - projeto completo.....	58

LISTA DE TABELAS

Tabela 1 – comparativo entre as principais versões da Raspberry Pi 25

LISTA DE ABREVIATURAS E SIGLAS

APT	<i>Advanced Packaging Tool</i>
API	Interface de Programação de Aplicações
ATD	Água Total Disponível no solo
CC	Capacidade de Campo
COM	Comum
d	densidade do solo (em gr/cm ³)
ETc	Evapotranspiração
f	coeficiente de disponibilidade ($0 < f < 1$)
GPIO	<i>General Purpose Input-Output</i>
IoT	Internet das Coisas
JSON	<i>JavaScript Object Notation</i>
Kc	Coeficiente da cultura
LL	Lâmina líquida
NC	<i>Normally Closed</i>
NO	<i>Normally Open</i>
ONU	Organização das Nações Unidas
PMP	Ponto de murcha permanente
PoE	<i>Power over Ethernet</i>
R ²	coeficiente de determinação
REST	Transferência de Estado Representacional
SBC	<i>Single Board Computers</i>
SO	Sistema Operacional
SOA	<i>Service Oriented Architecture</i>
SOAP	<i>Simple Object Access Protocol</i>
XML	<i>eXtensible Markup Language</i>
z	profundidade do sistema radicular (em cm)

SUMÁRIO

1 INTRODUÇÃO	11
1.1 OBJETIVO GERAL.....	13
1.2 OBJETIVOS ESPECÍFICOS	13
1.3 JUSTIFICATIVA	13
1.4 ESTRUTURA DO TRABALHO.....	14
2 IRRIGAÇÃO NA AGRICULTURA	15
2.1 ESTRESSE HIDRICO	15
2.2 SISTEMAS DE IRRIGAÇÃO	15
2.2.1 Irrigação superficial	16
2.2.2 Irrigação subsuperficial.....	16
2.2.3 Irrigação por aspersão.....	17
2.2.4 Irrigação localizada	17
2.3 VARIÁVEIS DA IRRIGAÇÃO	18
2.3.1 Conceitos básicos para programar a irrigação	19
2.3.2 Frequência de irrigação fixa.....	19
2.3.3 Irrigação de acordo com instrumentos de medição de umidade.....	20
3 USO DE IOT NA IRRIGAÇÃO	22
3.1 TECNOLOGIAS PARA MONITORAMENTO E CONTROLE DE IRRIGAÇÃO ...	22
4 RASPBERRY PI	24
4.1 VERSÕES DO RASPBERRY PI	25
4.2 SISTEMAS OPERACIONAIS	26
4.2.1 Raspbian	27
4.2.2 Android Things.....	27
5 API ADVISOR	29
5.1 PADRÃO REST.....	29
5.1.1 Rest versus XML	30
5.2 DOCUMENTAÇÃO	31
5.2.1 Busca de cidades por ID ou por nome	31
5.2.2 Previsão para 72 horas	31
6 TRABALHOS CORRELATOS.....	33

6.1 ADAPTING WEATHER CONDITIONS BASED IOT ENABLED SMART IRRIGATION TECHNIQUE IN PRECISION AGRICULTURE MECHANISMS	33
6.2 AUTOMAÇÃO NO PROCESSO DE IRRIGAÇÃO NA AGRICULTURA FAMILIAR COM PLATAFORMA ARDUÍNO	33
6.3 SMART WATER MANAGEMENT PLATFORM: IOT-BASED PRECISION IRRIGATION FOR AGRICULTURE	34
6.4 SISTEMA IOT PARA MONITORAMENTO E CONTROLE DE IRRIGAÇÃO.....	34
7 SISTEMA DE IRRIGAÇÃO AUTOMATIZADO UTILIZANDO O CONCEITO DE IOT PARA TOMADA DE DECISÃO	35
7.1 DESCRIÇÃO DOS RECURSOS UTILIZADOS	35
7.1.1 Hardware	35
7.1.2 Software	36
7.2 METODOLOGIA.....	36
7.2.1 Análise de solo	37
7.2.2 Configuração da Raspberry com Android Things.....	39
7.2.3 Banco de dados Real Time Database	41
7.2.4 Desenvolvimento do módulo Raspberry	42
7.2.5 Desenvolvimento do módulo mobile	50
7.2.6 Lógica para programar a irrigação	55
7.3 RESULTADOS OBTIDOS	56
7.4 ANÁLISE DOS DADOS.....	59
8 CONCLUSÃO	60

1 INTRODUÇÃO

A água é um fator muito importante em um mundo que cada vez mais existe necessidade de se produzir alimentos, de acordo com a Organização das Nações Unidas (ONU, 2015), o aumento da população vem em uma crescente, estima-se que deve aumentar em mais de 1 bilhão de pessoas nos próximos 15 anos, chegando a 8,5 bilhões de pessoas em 2030, e aumentar ainda mais para 9,7 bilhões em 2050.

A evolução da agricultura vem acompanhando a evolução do homem, passando de nômade para viver em vilas, comunidades e cidades. De acordo com Silva e Araújo (2005), há evidências de uma civilização muito antiga, cerca de 2500 a.C, na Mesopotâmia, onde hoje se situa o Iraque, que obtinham uma excelente produção agrícola, provavelmente devido a suas terras férteis e a um sistema de irrigação.

A área irrigada corresponde a cerca de 20% de toda a área produtiva no mundo, mas é responsável pela produção de mais de 40% de todo alimento consumido. Estima-se uma produção mundial de áreas irrigadas de 2,5 vezes a mais do que em áreas sequeiras. Com estes dados, é possível ver a importância da irrigação no contexto alimentar (KÖGLER; SÖFFKER, 2017, tradução nossa).

Paulino et. al. (2011) cita que o Brasil possui cerca de 12% da disponibilidade de água no mundo, e possui um potencial de expansão para a agricultura irrigada de 30 milhões de hectares. Atualmente possui cerca de 5 milhões de hectares de terras irrigadas, sendo assim, verifica-se um grande potencial neste ramo de atuação.

Segundo Kijne (2010, tradução nossa) a história e a evolução da irrigação no mundo acompanham o crescimento populacional. Principalmente depois da segunda guerra mundial é que houve um grande investimento nesta área, ocorrendo um salto de 138 para 210 milhões de hectares de área irrigada no mundo, no período entre 1961 a 1980. Mais de 90% de todas as grandes barragens no mundo foram construídas neste período, totalizando cerca de 45.000 grandes barragens no final do século XX. Já no final do século, o banco mundial investiu menos na agricultura, influenciando diretamente no desenvolvimento de novas tecnologias para irrigação, dando ênfase assim, no melhoramento da eficiência das já existentes.

Um dos sistemas mais conhecidos de irrigação no Brasil é o AgroSmart, que é um software de gerenciamento de irrigação, que analisa dados do solo, planta

e atmosfera. Porém sistemas como este citado anteriormente, tem custos elevados para pequenas propriedades rurais, pois para o seu funcionamento é necessário a instalação de uma estação meteorológica na propriedade, tornando inviável a aquisição por grande parte dos pequenos produtores.

Sendo assim, este projeto proposto tem como base criar um protótipo de um sistema de irrigação, levando em conta variáveis como a planta, solo e clima, utilizando previsões do tempo disponíveis na internet, dispensando o gasto com a estação meteorológica e gerando economia de água.

Para isto, será utilizado uma Raspberry Pi para realizar o controle do sistema de irrigação. Para gerenciar a placa controladora, propõe-se desenvolver um software que irá extrair da internet os dados da previsão do tempo por meio de uma Interface de Programação de Aplicações, do inglês *Application Programming Interface* (API), e será alimentado com informações do solo por meio de sensores de umidades e informações da planta. O software será responsável por analisar os dados e mandar uma resposta para o Raspberry Pi, este por sua vez, estará conectado a uma bomba de água.

Os dados climáticos serão extraídos de uma API chamada *Advisor*, que é uma ferramenta de previsão de tempo criada pela ClimaTempo, com ela é possível integrar ao projeto dados meteorológicos de alta precisão e qualidade (CLIMATEMPO, 2018). Esta é uma API que aplica os conceitos de Transferência de Estado Representacional (REST), que retorna ao usuário da aplicação uma resposta no formato *JavaScript Object Notation* (JSON), que é uma forma mais leve de representação e troca de informações, obtendo uma certa vantagem sobre o *eXtensible Markup Language* (XML), que possui uma resposta mais carregada.

Com isso, pretende-se desenvolver uma tecnologia eficiente e acessível aplicando os conceitos da Internet das coisas, do inglês *Internet of Things* (IoT), que é reconhecida como uma das áreas mais importantes da tecnologia, e o seu conceito é a interconexão entre os aparelhos, ou seja, o fato de vários dispositivos se comunicarem entre si, e se integrar num sistema como um todo (LEE; LEE, 2015, tradução nossa).

1.1 OBJETIVO GERAL

Empregar o conceito de IoT em um sistema de irrigação, utilizando um Raspberry Pi como ferramenta controladora e a API Advisor para auxiliar na tomada de decisão.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste projeto consistem em:

- a) aplicar os conceitos de IoT aplicado no setor agrícola no processo de irrigação;
- b) desenvolver um software para o controle e administração da irrigação, bem como, utilizar a API Advisor para obter informações da previsão do tempo;
- c) utilizar o Raspberry Pi como controlador do sistema de irrigação, conectando-o a um sensor de umidade e numa bomba de água;
- d) simular uma mini plantação irrigada, utilizando-se das tecnologias desenvolvidas.

1.3 JUSTIFICATIVA

O estudo de economia das águas e tecnologias precisas de irrigação tem sido objeto de estudos científico em todo mundo, tendo como objetivo o aumento da eficiência no uso da água (KÖGLER; SÖFFKER, 2017, tradução nossa).

Goap et. al. (2018, tradução nossa) e Kamienski et. al. (2019, tradução nossa) afirmam que o uso de IoT surge como uma escolha natural para o gerenciamento inteligente das águas. Rajalakshmi e Mahalakshmi (2016, tradução nossa) dizem que o uso de sensores para monitoramento da umidade do solo, aliado com a irrigação localizada por gotejamento pode economizar até 53% de água em relação a irrigação por aspersão.

Para Richardson e Wallace (2013) quando se fala de Raspberry Pi, é lembrado de algumas características importantes dele, como: pequeno, barato, hackeável e educacional. Porém ele não é apenas um dispositivo *plug-and-play*, que se conecta em um monitor e se obtém uma imagem na tela. Depende do que se

pretende fazer com ele, é necessário tomar uma série de decisões sobre dispositivos periféricos e softwares, ao instalá-lo e configurá-lo. Os autores ainda dizem que o baixo preço do Raspberry Pi é uma parte importante na história, fazendo que ele fosse distribuído no mundo todo. Hoje é possível encontrar na internet a distribuição do Raspberry Pi 3, a mais atual, por cerca de 200,00 reais.

Foi em 2006 que o engenheiro britânico Eben Upton e uma equipe de *hackers* de hardware que deram vida ao Raspberry Pi, baseados no modelo Atmel ATmega, como meio de fornecer tecnologia mais acessíveis às pessoas interessadas no assunto.

De acordo com Montanheiro, Carvalho e Rodrigues (2017), o REST é uma arquitetura de web services, proposta inicialmente pelo Roy Fielding em sua tese de doutorado. Esta é uma arquitetura que utiliza o protocolo HTTP/HTTPS, e formatos de dados mais simples, do que seu antecessor o *Simple Object Access Protocol* (SOAP) que utiliza unicamente o formato XML, que é mais carregado e complexo do que o JSON, o formato mais comumente usado em API's com arquitetura REST.

Pode-se dizer então, que sendo mais simples e leve, a arquitetura REST possibilita obter uma melhor performance, tornando-se amplamente utilizada.

O uso do Raspberry Pi, aliada com os dados da previsão do tempo obtidos por meio da API Advisor que aplica o conceito REST, visando construir uma tecnologia de irrigação acessível para as pequenas propriedades, obtendo economia de dinheiro e economia de água, tornará este projeto viável e justificável.

1.4 ESTRUTURA DO TRABALHO

O trabalho proposto será divido em 8 seções apresentadas a seguir. No capítulo 2 deste trabalho, é descrita a importância da irrigação na agricultura, bem como, as variáveis e equações que incidem no sistema de irrigação. No capítulo 3 foi abordado o uso de IoT na agricultura, correlacionando com o seu uso na irrigação.

O capítulo 4 demonstra os recursos do Raspberry Pi e quais as suas funcionalidades. Na seção 5 descreve-se a API Advisor, que é uma plataforma desenvolvida pela ClimaTempo, onde é possível realizar consulta de previsões de tempo conforme a cidade. Na seção 6 é abordado os trabalhos correlatos, na 7 é descrito o trabalho desenvolvido, metodologia e resultados. Por fim no capítulo 8, é apresentado a conclusão.

2 IRRIGAÇÃO NA AGRICULTURA

De acordo com informações da Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA, 2012), a quantidade certa de água e o momento ideal da aplicação são os pontos chaves para se obter um manejo adequado, favorecendo a economia de água e energia, resultando assim, num melhor desempenho das plantas.

Porém, para que a irrigação seja eficiente é necessário que vários fatores sejam analisados, relacionados não somente ao cultivo, como também questões edafoclimáticas. Clima e solo são os fatores que mais interferem na decisão do sistema de irrigação (EMBRAPA, 2012).

Existem basicamente dois modos de manejo de irrigação, um que utiliza informações como clima, temperatura, umidade relativa do ar, vento e radiação solar, para calcular a quantidade necessária de água. O outro sistema utiliza somente a umidade do solo como parâmetro de análise, o que muitas vezes acaba dando um resultado errado pois não leva em consideração os fatores climáticos (EMBRAPA, 2012).

2.1 ESTRESSE HIDRICO

Com base nos estudos de Kögler e Söffker (2017, tradução nossa), os principais processos responsáveis pelo desenvolvimento das plantas são absorção e transporte de água e nutrientes, transpiração, fotossíntese e atividades metabólicas. A água tem um papel fundamental na fisiologia da planta, atuando como solvente, reagente, componente celular e regulador de temperatura.

Um estresse hídrico ocorre quando a necessidade de água de uma cultura é maior que a disponibilidade de água no solo, e resultam em deficiências morfológicas e metabólicas.

2.2 SISTEMAS DE IRRIGAÇÃO

Existem diferentes tipos de sistemas de irrigação, cada um com um diferente propósito. Todos os sistemas possuem características próprias, custos, vantagens e desvantagens. Para decidir qual sistema utilizar na propriedade, deve-se levar em consideração a disponibilidade de água, tipo de planta, relevo do solo e

custos de implantação. Estes sistemas podem ser classificados como: superficial, subsuperficial, aspersão e localizado. Estes sistemas serão descritos nas seções a seguir, com base na integra dos estudos de Marouelli e Silva (2011).

2.2.1 Irrigação superficial

Neste sistema a água é distribuída diretamente sobre a superfície do solo, compreende os sistemas de sulco, faixa e inundação. Este sistema geralmente requer um menor investimento inicial e menor uso de energia, porém demanda bastante mão-de-obra e água. Para implantar este sistema, não se recomenda solos com alta taxa de permeabilidade, a exemplo dos solos arenosos. Na figura 1 pode-se observar um sistema de irrigação por inundação.

Figura 1 - Irrigação por inundação



Fonte: Seagro (2015).

2.2.2 Irrigação subsuperficial

Neste sistema, a água é aplicada sob a superfície do solo, por meio da criação de um lençol freático. Este lençol pode ser mantido fixo, ou regulado de acordo com o sistema radicular da cultura instalada. Este sistema exige pouco investimento inicial e baixa utilização de energia, porém requer solos planos ou sistematizados e exige uma camada de solo permeável sobrepondo uma camada de solo impermeável em torno de 1,5 metros de profundidade. Na figura 2 apresenta um solo irrigado neste sistema, com regulagem do lençol freático.

Figura 2 - Irrigação Sub-superficial



Fonte: UNICAMP (2014)

2.2.3 Irrigação por aspersão

Neste sistema a água é aplicada na forma de chuva, dando destaque para o modo convencional, auto propelida e pivô-central. Este método exige menor custo com mão-de-obra, e possibilita melhor distribuição da água no solo. Pode ser usada para qualquer tipo de topografia e, permite a automatização e aplicação de fertilizantes e agrotóxicos via foliar.

Em contrapartida, exige maior demanda de energia, sofre com a interferência do vento, e em climas secos há maior perda de água pela evaporação. Além disso, a água aplicada sobre a folhagem propicia um ambiente favorável ao surgimento de doenças e prejudica a polinização. Na figura 3, é possível observar um sistema convencional.

Figura 3 - Aspersão convencional



Fonte: Portal Vida no Campo (2017)

2.2.4 Irrigação localizada

Neste sistema a água é aplicada próximo à planta, sem molhar toda a superfície do solo, em regime de alta frequência e baixo volume. É conhecido deste

sistema os métodos de gotejamento e micro-aspersão. São caracterizados por usarem um baixo volume de água e energia, podem ser usados em solos com certo grau de declividade e permitem a automação total da irrigação. As principais desvantagens são o alto investimento inicial e o risco de entupimentos dos tubos gotejadores. Na figura 4, representa o método de irrigação localizada por gotejamento.

Figura 4 - Irrigação localizada por gotejamento



Fonte: Site agronegócio interior (2016)

2.3 VARIÁVEIS DA IRRIGAÇÃO

A água na agricultura irrigada é um insumo de alto custo, tanto economicamente, como, ecologicamente. De acordo com informações de Kögler e Söffker (2017, tradução nossa), a retirada de água doce pelo setor agrícola representa em torno de 70% de toda a água consumida no mundo.

É óbvio afirmar que a aplicação de pouca água (déficit de irrigação) pode trazer prejuízos, visto que a cultura não poderia atingir todo seu potencial, mas por outro lado, podemos afirmar também que a aplicação excessiva pode ser muito mais destrutiva, pois ocorre a saturação do solo, pouca aeração, lixiviação de nutrientes e deixa o ambiente mais propício ao aparecimento de doenças (ALBUQUERQUE, 2010).

Relacionando as informações de Kögler e Söffker (2017, tradução nossa) e Albuquerque (2010), citadas acima, afirma-se que é fundamental saber quando e quanto irrigar, obtendo-se assim, melhor desempenho do sistema de irrigação e economia de água.

Nos tópicos a seguir, são apresentadas as principais variáveis de irrigação, tendo como base os estudos de Albuquerque (2010), Engenheiro agrícola e pesquisador da EMBRAPA.

2.3.1 Conceitos básicos para programar a irrigação

Existem vários pontos a serem analisados para decidir o momento ideal da irrigação e a quantidade de água, alguns métodos utilizam somente as características físicas do solo e evapotranspiração da cultura, outros métodos são baseados em sensores para monitorar o potencial ou umidade do solo.

A evapotranspiração (ETc) é a combinação de dois processos, que é a evaporação da água do solo + transpiração das plantas, resultando na água necessária para uma cultura.

O coeficiente da cultura (Kc) é influenciado pelas características da planta, estágio de desenvolvimento e condições climáticas. Cada cultura possui um Kc diferente.

Á Água Total Disponível no solo (ATD) é o total de água que pode ser absorvida pela planta, está entre a umidade da Capacidade de Campo (CC – limite superior de água disponível) e o Ponto de Murcha Permanente (PMP – limite inferior de água disponível).

2.3.2 Frequência de irrigação fixa

Também chamada de turno de rega, a frequência de irrigação fixa é o intervalo entre duas irrigações sucessivas. Neste método há uma grande facilidade no manejo, porém as taxas de erros são maiores, pois ela é baseada num valor fixo de ETc, entretanto o seu valor é variável de acordo com a fase vegetativa da cultura.

A Lâmina Líquida (LL) pode ser calculada pela equação 1:

$$\text{LL} = \frac{(CC - PMP)}{10} \cdot d \cdot f \cdot x \cdot z \quad (1)$$

Em que:

- a) LL = lâmina líquida (mm)
- b) CC = capacidade de campo

- c) PMP = Ponto de murcha permanente
- d) d = densidade do solo (em gr/cm³)
- e) f = coeficiente de disponibilidade ($0 < f < 1$)
- f) z = profundidade do sistema radicular (em cm)

O coeficiente f da água do solo assume valores diferentes de acordo com a textura do solo. O valor de $f = 0$ representa um solo com umidade na capacidade de campo, já um valor de $f = 1$ representa um solo sem água disponível, no ponto de murcha permanente. O valor de f é definido tendo em vista o maior ou menor grau de dificuldade que a planta poderá ter para retirar água do solo. Assim por exemplo, define-se o valor de f para culturas mais exigentes, como hortaliças, em torno de 0,25. Culturas como cereais e pastagens podem ter o valor de f em torno de 0,5 e para frutíferas em torno de 0,4.

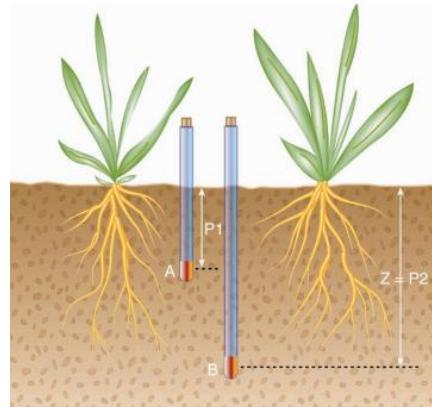
A frequência de irrigação, de acordo com a evapotranspiração da cultura será de acordo com a equação 2:

$$F = \frac{LL}{ETc} \quad (2)$$

2.3.3 Irrigação de acordo com instrumentos de medição de umidade

Instrumentos que medem a umidade do solo, são muito uteis para determinar a quantidade de água necessária para a irrigação. É importante destacar que utilizando este método, a época de irrigar fica independente do estabelecimento das frequências de irrigação vistas no método anterior. Porém, deve-se acompanhar o crescimento do sistema radicular (z), e considerar a leitura feita no ponto médio dessa profundidade, como indica na figura 5.

Figura 5 - Posição dos instrumentos de medição



Fonte: Albuquerque (2010)

Usando deste método, a lâmina líquida de irrigação (LL) é calculada usando a equação 3.

$$LL = \frac{(CC - U_i)}{10} \cdot d \cdot z \quad (3)$$

Em que:

- a) LL = lâmina líquida (mm);
- b) CC = capacidade de campo;
- c) U_i = Conteúdo de água no solo no limite para início da irrigação, capturado pelo sensor;
- d) d = densidade do solo (em gr/cm³);
- e) z = profundidade do sistema radicular (em cm).

Percebe-se que a equação 3 é semelhante a equação 1, a diferença está na variável f que não aparece na equação 3, pois este valor já está implícito ao se estabelecer um valor mínimo para o início da irrigação (U_i).

3 USO DE IOT NA IRRIGAÇÃO

O conceito de IoT é a capacidade de conectar objetos físicos (casas, carros, eletrodomésticos etc.), ou seja, tudo que possa coletar informações por meio de sensores e compartilhar com outros dispositivos (MUXITO et. al., 2018).

Para Rajalakshmi e Mahalakshmi (2016, tradução nossa), o conceito de IoT é a possibilidade de compartilhar dados e informações entre dispositivos, desde que haja conexão com a internet. O uso de IoT na agricultura é um fator muito importante para o futuro alimentar, pois pode produzir comida para quase 10 bilhões de pessoas até 2050. Rajalakshmi e Mahalakshmi (2016, tradução nossa) ainda dizem que o uso deste tipo de tecnologia ajuda a economizar fertilizantes, água, custos operacionais e aumenta o rendimento da colheita.

Como foi visto no capítulo 2 deste trabalho, a água é um recurso muito importante para a manutenção da vida na terra. A escassez de água limpa no mundo gerou uma necessidade de utilização ideal, sendo que o uso de tecnologias é indispensável para que isto ocorra. Devido à falta de sistemas de irrigação inteligentes com uma boa relação custo-benefício, os países em desenvolvimento acabam consumindo mais água do que os desenvolvidos, para alcançar um mesmo rendimento (GOAP et. al., 2018, tradução nossa).

A IoT surge como um fenômeno que envolve muitos fatores, entre eles, dispositivos baratos, acessibilidade a tecnologia, armazenamento e processamento em nuvem, estruturas para lidar com dados não estruturados de redes sociais e recursos computacionais capazes de processar uma quantidade gigantesca de dados, sendo mais conhecido como big data (KAMIENSKI et. al., 2019, tradução nossa).

O propósito da IoT não se pode resumir a uma única tecnologia. Na verdade, existe uma infinidade de fatores que contribuem e formam o seu conceito. Porém existe três componentes essenciais para que ocorra a existência de IoT: dispositivos de automação, rede de comunicação e um sistema de controle (MUXITO et. al., 2018).

3.1 TECNOLOGIAS PARA MONITORAMENTO E CONTROLE DE IRRIGAÇÃO

Davis e Dukes (2010, tradução nossa) realizaram a comparação de quatro controladores de irrigação com base na evapotranspiração (ETc) do solo, realizando

a configuração de acordo com cada fabricante. Esses controladores são programados para realizar estimativas de ET_c conforme visto na equação 1 e utilizaram sensores de chuva. Davis e Dukes (2010, tradução nossa) obtiveram resultados satisfatórios conforme as métricas usadas no seu trabalho, tendo como parâmetro a qualidade da cultura implantada mantendo uma boa aceitabilidade.

Também se destaca o projeto de Roopaei, Rad e Choo (2017, tradução nossa), que aplicam o uso de IoT usando imagens feitas por drone para realizar o monitoramento da irrigação usando a termografia das folhas das plantas. Essas imagens são processadas em nuvem por meio de um algoritmo desenvolvido, podendo-se assim saber qual a necessidade de água de uma determinada cultura.

No trabalho de Muxito et.al. (2018) foi aplicado o uso de IoT no desenvolvimento de uma plataforma que envolve Arduino e uma aplicação Web para o controle do processo de irrigação. As funcionalidades deste sistema seriam a leitura da umidade do solo, automatização da decisão de irrigar, histórico de registros de leitura, apresentação dos registros e painel para controle do tempo e agendamento do ciclo de irrigação.

Contudo, ainda existem alguns obstáculos que impedem o uso generalizado da IoT para irrigação. Um dos fatores está ligado ao desenvolvimento de aplicativos inteligentes para o uso de IoT, como na irrigação, ainda não está totalmente automatizada. Outro fator é a falta de uma plataforma que integre diferentes tecnologias, como a própria IoT, análise de big data, computação em nuvem, para implantar no uso de gerenciamento de irrigação (KAMIENSKI et. al., 2019, tradução nossa).

4 RASPBERRY PI

O Raspberry Pi tem sido muito bem-sucedido em inúmeros projetos eletrônicos, tendo em vista que seu objetivo inicial era incentivar crianças e jovens a envolverem-se com a programação e com os conceitos de ciência da computação. Para conseguir isto, era necessário que estes jovens tivessem acesso a máquinas de baixo custo, usando como substituto ao PC da família, que em muitos casos não poderia ser usado para tal objetivo (KÖLLING, 2016, tradução nossa).

Foi em fevereiro de 2012 que o primeiro dispositivo Raspberry Pi foi comercializado, desenvolvido pelo engenheiro britânico *Eben Upton* e sua equipe, com a intenção de tornar acessível para todos os jovens uma tecnologia capaz de incentivá-los a ingressar no mundo da computação (HEIN, 2013).

Assim, o propósito original do Raspberry Pi não era ser um dispositivo para fins especiais, como coletar dados de sensores ou como plataforma de controle de projetos eletrônicos, mas sim, para servir de substituto do computador pessoal, onde os jovens poderiam usar para programar e desenvolver, sem correr o risco de danificar o computador pessoal usado pela família (KÖLLING, 2016, tradução nossa).

De acordo com Johnston e Cox (2017, tradução nossa), ao se adicionar alguns periféricos que não estão inclusos (mouse, teclado, monitor), é possível obter rapidamente um computador totalmente funcional rodando Raspbian, que é um sistema operacional Linux baseado em Debian.

Muitas vezes o Raspberry Pi é dito como um computador de placa única, ou SBC (*Single Board Computers*, em inglês), o que significa que ele é capaz de executar um sistema operacional e tem periféricos suficientes para iniciar sua operação sem a adição de hardware. A fundação Raspberry Pi tornou o SBC acessível para quase todos, disponibilizando não apenas um computador de baixo custo, mas também, os pinos de conexão *General Purpose Input-Output* (GPIO). Estes pinos podem ser controlados a partir do sistema operacional e suportam uma variedade de recursos, que podem ser usados para conectar uma grande quantidade de dispositivos eletrônicos (JOHNSTON; COX, 2017, tradução nossa).

4.1 VERSÕES DO RASPBERRY PI

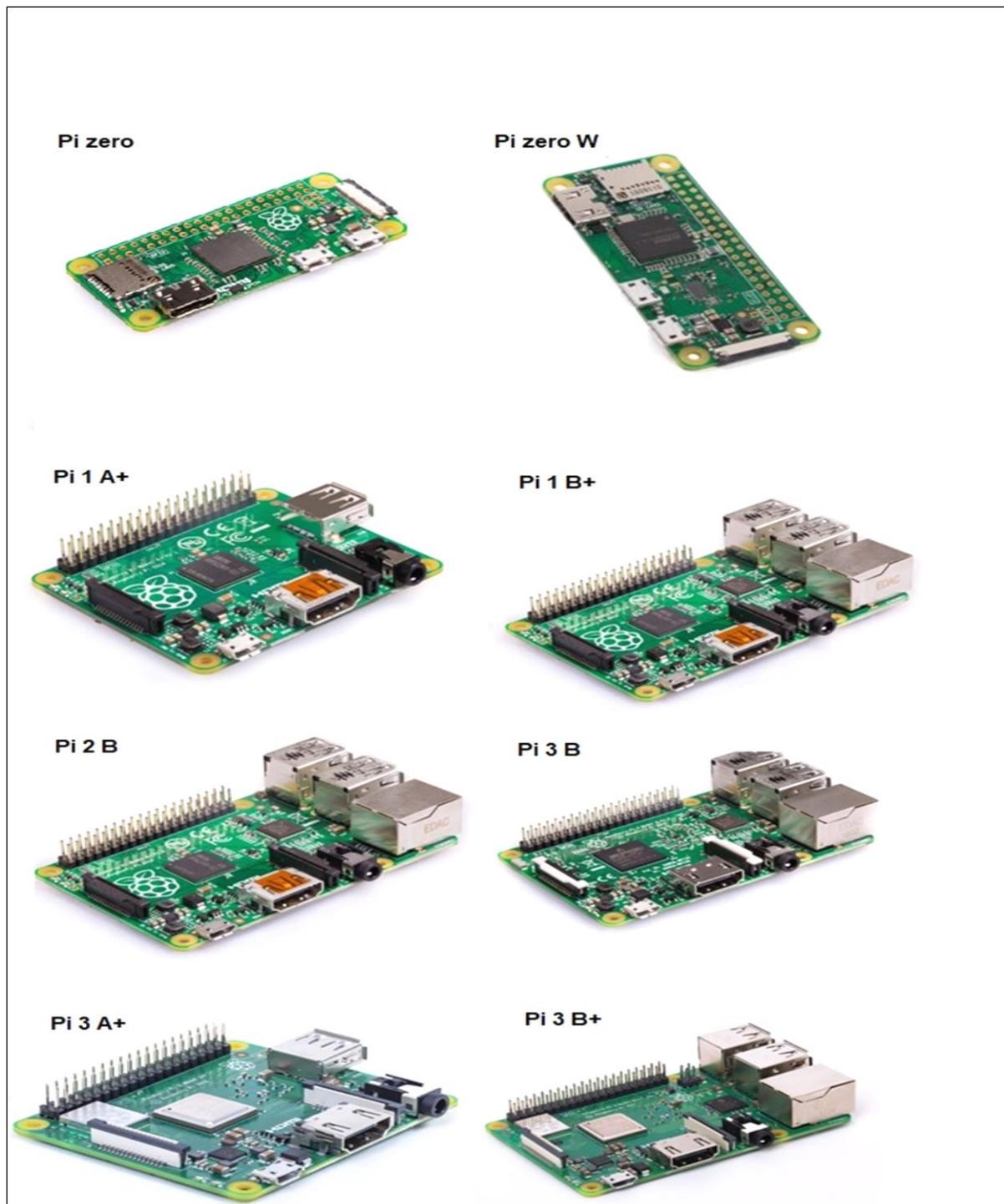
Desde 2012, ano de seu lançamento, a fundação Raspberry pi lançou algumas versões do seu SBC (figura 6), sendo a mais atual o modelo Pi 3B+. Em setembro de 2016, apenas 4 anos após o lançamento, o fundador Eben Upton anunciou a marca de 10 milhões de unidades vendidas em todo o mundo. Na tabela 1 é verificado o comparativo entre as principais versões disponíveis (RASPBERRY PI FOUNDATION, 2016, tradução nossa).

Tabela 1 – comparativo entre as principais versões da Raspberry Pi

	Raspberry Pi 3B	Raspberry Pi zero	Raspberry Pi 2B	Raspberry Pi B+
Lançamento	29/02/2016	25/11/2015	02/02/2015	14/07/2014
SoC	BCM2837	BCM 2835	BCM 2836	BCM 2835
CPU	Quad Cortex A53 @ 1.2 GHz	ARM 11 @ 1GHz	Quad Cortex A7 @ 900MHz	ARM 11 @ 700MHz
GPU	400 MHz Video Core IV	250 MHz Video Core IV	250 MHz Video Core IV	250 MHz Video Core IV
RAM	1 GB SDRAM	512 MB SDRAM	1 GB SDRAM	512 MB SDRAM
Armazenamento	MicroSD	MicroSD	MicroSD	MicroSD
Ethernet	10/100	-	-	-
Wireless	802.11n / Bluetooth 4.0	-	-	-
Saída de vídeo	HDMI / composto	HDMI / composto	HDMI / composto	HDMI / composto
Saída de áudio	HDMI / Headphone	HDMI	HDMI / Headphone	HDMI / Headphone
GPIO	40	40	40	40

Fonte: Do autor

Figura 6 - Versões do Raspberry Pi



Fonte: Raspberry Foundation (2016)

4.2 SISTEMAS OPERACIONAIS

Um Sistema Operacional (SO) tem a tarefa de controlar todos os recursos de um hardware qualquer. Em geral, os sistemas de computação são muito

complexos, pois consistem em um ou mais processadores, teclado, tela, interfaces de rede entre outros dispositivos de entrada e saída. Programar algo que controla todos estes recursos é uma tarefa extremamente difícil, se todo programador tivesse que se preocupar com o gerenciamento de tudo isto, provavelmente muitos softwares não poderiam ser escritos. A solução para isto, foi colocar uma camada de software sobre o hardware, e apresentar ao usuário uma interface mais fácil de entender e programar, sendo esta camada chamada de sistema operacional (TANENBAUM; WOODHULL, 2006, tradução nossa).

4.2.1 Raspbian

O *Raspbian* é o sistema operacional recomendado para o uso normal em um Raspberry pi. Ele é um sistema livre baseado no Debian e otimizado para o hardware do Raspberry pi. O *Raspbian* vem com mais de 35000 pacotes de software pré-compilado em um formato de fácil instalação (RASPBERRY PI FOUNDATION, 2019, tradução nossa).

O *Raspbian* é o principal sistema operacional para o Raspberry, apesar de existirem outros sistemas disponíveis, como, MATE que é uma versão do *Ubuntu*; o OSMC que é um sistema dedicado a mídias; o *Recalbox* pra quem deseja usar o Raspberry Pi para jogar e o *Pidora* que é uma distribuição Linux de *Red Hat* (GARRET, 2016).

Para realizar a instalação do *Raspbian* é necessário instalar a imagem em um cartão SD. Esta imagem é encontrada no site oficial do Raspberry Pi, ou em distribuições alternativas em fornecedores terceirizados. É recomendado o uso do *Etcher*, que é uma ferramenta gráfica de gravação de imagens de cartões SD (RASPBERRY PI FOUNDATION, 2019, tradução nossa).

Após a instalação do SO, o usuário pode usar o *Advanced Packaging Tool* (APT), que é uma ferramenta usada para a instalação, atualização e remoção de softwares (RASPBERRY PI FOUNDATION, 2019, tradução nossa).

4.2.2 Android Things

O Android Things é uma ferramenta da Google que foi criada no intuito de servir como um SO específico para desenvolvimento de IoT. Ele fornece a maioria das

ferramentas e APIs disponíveis no ambiente Android, facilitando o processo de desenvolvimento e aprendizagem de quem irá trabalhar com esta ferramenta (SOUZA, 2018).

Este SO permite criar aplicativos inteligentes para dispositivos conectados, melhorando a experiência do desenvolvedor com IoT. Tem o suporte para o Raspberry Pi 3B e NXP iMX7D (DEVELOPER ANDROID, 2019).

Para usar o Android Things no Raspberry é preciso atender dois requisitos: usar a Raspberry Pi 3B, a versão 1 e 2 não funcionam e ter cartão microSD de no mínimo 8GB classe 10 para gravar o SO (ZARANZA, 2016).

5 API ADVISOR

Uma Interface de Programação de Aplicativos (API), fornece recursos, dados e informações para outro aplicativo, que pode utilizar destes serviços sem precisar implementar mais nada, gerando reutilização de código. As APIs são vitais para muitas arquiteturas de software, pois fornecem abstrações de alto nível, que facilitam as tarefas de programação (MENG; STEINHARDT; SCHUBERT, 2017, tradução nossa).

Para Myers e Stylos (2016, tradução nossa), as APIs são usadas por praticamente todo código, por incluírem bibliotecas, estruturas, kits de ferramentas e desenvolvimento. Elas permitem a reutilização de código, para que os programadores possam desenvolver em cima do que outros, ou eles mesmos já fizeram, evitando em começar do zero o programa.

O aprendizado de uma nova API, está intimamente ligado à sua documentação, embora muitas vezes a dificuldade de aprendizado se deve as propriedades da API, como o domínio que ela abrange, aspectos específicos e até mesmo o design se torna relevante. Porem a forma como ela é abordada em sua documentação é o fator que mais interfere (MENG; STEINHARDT; SCHUBERT, 2017, tradução nossa).

O Advisor é uma API de previsão de tempo, que fornece dados meteorológicos com precisão, em tempo real (CLIMATEMPO, 2018). O Advisor aplica o conceito de integração de dados no padrão REST, que será abordado na seção 6.1, e sua documentação será abordada na seção 6.2.

5.1 PADRÃO REST

REST teve a origem na tese de doutorado de Roy Fielding, coautor de um dos protocolos mais utilizados no mundo, o HyperText Transfer Protocol (HTTP) (SAUDATE, 2013).

Para realizar a integração entre sistemas de informação, visando o compartilhamento de dados, podem ser utilizadas técnicas como Arquitetura Orientada a Serviços, conhecida como *Service Oriented Architecture* (SOA). Diversas tecnologias oferecem suporte na implementação do SOA, entre elas está o uso de

Web Services. Alguns padrões e tecnologias estão ligados ao desenvolvimento de Web Services, como o REST e o SOAP (GARCIA; ABILIO, 2017).

5.1.1 Rest versus XML

Saudate (2013) mostra a diferença entre o uso do XML em relação ao uso de REST. Por exemplo, para listar os clientes de um determinado sistema, é necessário enviar o seguinte XML para o servidor:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
    <soap:Body>
        <listarClientes xmlns="http://brejaonline.com.br/administracao/1.0/service"
        </soap:Body>
    </soap:Envelope>
```

Que se terá como resposta:

```
<soap:Envelope xmlns:domain="http://brejaonline.com.br/administracao/1.0/domain"
    xmlns:soap="http://schemas.xmlsoap.org/soap/envelope">
    <soap:Body>
        <listarClientesResponse
            xmlns="http://brejaonline.com.br/administracao/1.0/
<domain:clientes>
    <domain:cliente domain:id="1">
        <domain:nome>Alexandre</domain:nome>
        <domain:dataNascimento>2012-12-01</domain:dataNascimento>
    </domain:cliente>
    <domain:cliente domain:id="2">
        <domain:nome>Paulo</domain:nome>
        <domain:dataNascimento>2012-11-01</domain:dataNascimento>
    </domain:cliente>
    </domain:clientes>
</listarClientesResponse>
</soap:Body>
</soap:Envelope>
```

A complexidade do código devido a verbosidade fica evidente, principalmente quando se pensa em comunicação feitas com plataformas móveis, onde o uso de rede deve ser o mais suscinto possível. Ao se utilizar o padrão REST, para realizar a mesma listagem de clientes do sistema, pode se utilizar o navegador para abrir a URL: <http://localhost:8080/cervejaria/clientes>. Obtendo o seguinte resultado (SAUDATE, 2013):

```
<clientes>
    <cliente id="1">
        <nome>Alexandre</nome>
        <dataNascimento>2012-12-01</dataNascimento>
    </cliente>
    <cliente id="2">
        <nome>Paulo</nome>
```

```

<dataNascimento>2012-11-01</dataNascimento>
</cliente>
</clientes>
```

5.2 DOCUMENTAÇÃO

Por meio da documentação do Advisor é possível identificar os comandos de ação, como, busca de cidade por ID ou por nome, previsão para 15 dias, previsão para 72 horas, lista de estações meteorológicas e histórico de previsão. Para acessar estas funcionalidades, é necessário gerar um token, que identificará qual o projeto e qual usuário realizará as requisições

5.2.1 Busca de cidades por ID ou por nome

Busca de cidades por ID:

<http://apiadvisor.climatempo.com.br/api/v1/locale/city/3477?token=643c73fff94c5b2fe221c5ede99cbeb4>

Temos como resposta:

`{"id":3477,"name":"São Paulo","state":"SP","country":"BR"}`

Busca de cidades por nome:

<http://apiadvisor.climatempo.com.br/api/v1/locale/city?name=Crici%C3%BAma&state=SC&token=643c73fff94c5b2fe221c5ede99cbeb4>

Temos como resposta:

`[{"id":4898,"name":"Criciúma","state":"SC","country":"BR"}]`

5.2.2 Previsão para 72 horas

É possível obter a previsão para 72 horas ou para 15 dias, no exemplo a seguir será mostrado o formato para previsão de 72 horas, e na figura 7 se exibe os campos e seus retornos.

<http://apiadvisor.climatempo.com.br/api/v1/forecast/locale/4898/hours/72?token=643c73fff94c5b2fe221c5ede99cbeb4>

Temos como resposta:

```
{"date":"2019-06-10 00:00:00","date_br":"10/06/2019 00:00:00","rain": {"precipitation":0}, "wind": {"velocity":6.6,"direction":"NNW","directiondegrees":357.5,"gust":9.1}, "temperature": {"temperature":16}}
```

Figura 7 - retorno de previsão 72 horas

Success 200		
Campo	Tipo	Descrição
<code>id</code>	Number	ID da cidade.
<code>name</code>	String	Nome da cidade.
<code>state</code>	String	Estado da cidade.
<code>country</code>	String	País da cidade.
<code>data</code>	Object[]	Lista de objetos com dados de previsão por hora.
<code>date</code>	Date	Data no formato "aaaa-mm-dd" (Ex.: 2017-10-01).
<code>date_br</code>	Date	Data no formato do Brasil "dd/mm/aaaa" (Ex.: 01/10/2017).
<code>rain</code>	Object	Dados de chuva.
<code>precipitation</code>	Number	Precipitação em milímetro.
<code>wind</code>	Object	Dados de vento.
<code>velocity</code>	Number	Intensidade do vento em km/h.
<code>direction</code>	String	Direção do vento.
<code>direction_degrees</code>	Number	Direção do vento em graus.
<code>gust</code>	Number	Rajada do vento em km/h.
<code>temperature</code>	Object	Dados de temperatura.
<code>temperature</code>	Number	Temperatura em graus celsius.

Fonte: ClimaTempo (2019).

6 TRABALHOS CORRELATOS

Para o desenvolvimento deste trabalho foram usados alguns trabalhos como referência, que serão apresentados a seguir.

6.1 ADAPTING WEATHER CONDITIONS BASED IOT ENABLED SMART IRRIGATION TECHNIQUE IN PRECISION AGRICULTURE MECHANISMS

Este trabalho foi desenvolvido por Keswani et. al. (2018, tradução nossa), com o objetivo de desenvolver um ambiente de sensores sem fio utilizando a tecnologia Zigbee, criando um mapa de umidade do solo da propriedade, onde cada nó (sensor) é conectado a um nó coordenador, que é projetado com um Raspberry Pi3. O nó coordenador envia os dados coletados para a nuvem para calcular a necessidade de irrigação da propriedade.

Keswani et al. (2018, tradução nossa) também aplicou em seu trabalho um modelo para prever a umidade do solo com uma hora de antecedência, integrado ao modelo de lógica fuzzy para gerar comandos de controle de válvulas de irrigação de cada nó independentemente.

6.2 AUTOMAÇÃO NO PROCESSO DE IRRIGAÇÃO NA AGRICULTURA FAMILIAR COM PLATAFORMA ARDUÍNO

Este projeto foi desenvolvido por Cunha e Rocha (2015), na Faculdade de Tecnologia do Estado de São Paulo (FATEC), na cidade de Presidente Prudente. O objetivo foi montar um protótipo de automação de um sistema de irrigação, com base nas informações da umidade do solo, temperatura e umidade do ar. Para capturar estas informações, foram usados sensores ligados ao microcontrolador Arduino.

Neste modelo desenvolvido, a bomba é acionada quando a umidade captada do solo indica o nível de solo seco, ou também, quando a umidade do ar captada pelo sensor indicar nível crítico. A temperatura do ar não teve implicância no acionamento da bomba, porém todos os dados foram armazenados no banco de dados para análise.

O projeto de irrigação conta com uma bomba de água ligada a um relé. Quando o relé for acionado, inicia-se o processo de irrigação.

6.3 SMART WATER MANAGEMENT PLATFORM: IOT-BASED PRECISION IRRIGATION FOR AGRICULTURE

Este trabalho foi realizado por Kamienski et al. (2019, tradução nossa), e foi desenvolvido por autores do Brasil, Finlândia e Itália. Foi apresentado na *2018 Global Internet of Things Summit (GloTS)*, em português, Cúpula Global de Internet das Coisas de 2018.

No trabalho de Kamienski et al. (2019, tradução nossa) foi desenvolvido o projeto SWAMP, que é uma plataforma de gerenciamento de irrigação baseada em IoT. Esta plataforma pode ser configurada e implantada de diversas maneiras, sendo possível adequar de acordo com a realidade de cada região, formando diferentes sistemas SWAMP.

É dividido em 5 camadas, onde na camada 1 se encontra os sensores de solo e clima, drones e toda a tecnologia responsável pela coleta de dados. Na camada 2 estão os protocolos de segurança e componentes de software para a aquisição de dados como o LoRa server e o MQQT. A camada 3 é responsável pelo gerenciamento de dados, distribuída em servidores na nuvem. Na camada 4 estão os modelos de distribuição de água, como, gotejamento e aspersão. E a camada 5 é responsável pela interface do usuário.

6.4 SISTEMA IOT PARA MONITORAMENTO E CONTROLE DE IRRIGAÇÃO

Este projeto foi desenvolvido por Gonçalves et al. (2018), no estado do Ceará, no Instituto Federal de Educação, Ciência e Tecnologia do Estado do Ceará (IFCE) e na Universidade Federal do Ceará.

Foi desenvolvido um sistema baseado no conceito de IoT, que consiste em automatizar a irrigação com o uso do microcontrolador NodeMCU 12E, programado pela IDE do Arduíno. A placa capta a umidade do solo utilizando-se um sensor de umidade e é responsável pelo acionamento da bomba de água.

Para o controle de irrigação foi desenvolvido uma página Web em PHP, que recebe os dados do microcontrolador enviados via rede WiFi, encapsulados no formato JSON. Então pode ser feito o monitoramento em tempo real da umidade do solo. A bomba de água é acionada quando a umidade atingir um valor mínimo programado.

7 SISTEMA DE IRRIGAÇÃO AUTOMATIZADO UTILIZANDO O CONCEITO DE IOT PARA TOMADA DE DECISÃO

Com o embasamento científico adquirido durante a pesquisa do levantamento bibliográfico, realizou-se o desenvolvimento do protótipo capaz de realizar a irrigação do solo. Primeiramente será apresentado os recursos utilizados no projeto, logo após a metodologia e por fim os resultados obtidos.

7.1 DESCRIÇÃO DOS RECURSOS UTILIZADOS

Para descrever os recursos utilizados na construção do protótipo, será descrito a seguir separadamente entre software e hardware.

7.1.1 Hardware

Em posse do autor, foi utilizado um Notbook Acer com 8 GB de memória RAM, HD de 1000 GB e processador Intel i5-7200U 2.5GHz com sistema operacional Windows 10.

Além deste, foi necessário adquirir alguns hardwares com recursos do autor, como a Raspberry Pi 3B, que conta com um processador Broadcom BCM2837 quad-core de 1,2 GHz, memória RAM de 1GB. Juntamente com a placa, foi necessário adquirir alguns acessórios, como:

- a) um case plástico;
- b) fonte de energia 5V, 2.2 amperes;
- c) um cartão de memória micro SD classe 10 de 16 GB.
- d) fios jumpers de conexão macho-macho e macho-fêmea;
- e) protoboard;
- f) sensor de umidade do solo modelo A080;
- g) conversor de sinal analógico/digital ADC0832;
- h) bomba de água 12V, vazão de dois litros de água por minuto;
- i) módulo relé 3.3V.

7.1.2 Software

Os softwares usados no projeto consistem em ambientes de programação, bibliotecas, banco de dados em tempo real, além de recursos auxiliares para o desenvolvimento, que serão descritos a seguir.

Foi usado como ambiente de desenvolvimento o Android Studio, com linguagem de programação JAVA. O Raspberry vem de fábrica sem configuração de sistema operacional, para isto, foi utilizado o Android Things, que foi desenvolvido para rodar em hardwares simples, com foco em desenvolvimento de IoT. Foi utilizado o software Win32DiskImager para gravar a imagem do Android Things no cartão de memória.

Para realizar a comunicação com a API Advisor, foi utilizado a biblioteca Retrofit, que foi desenvolvida para consumir web services no Android. Para o banco de dados, foi usado o serviço Real Time Database, do Firebase.

Por fim, foi utilizado a ferramenta Vysor, que é um serviço de espelhamento do Android, empregado na Raspberry Pi para obter a parte visual do sistema operacional diretamente no Notebook.

7.2 METODOLOGIA

A primeira parte prática do desenvolvimento foi a retirada da amostra de solo e envio para o laboratório de análises, a fim de determinar a textura do solo.

Após isto, foi realizado a configuração da Raspberry Pi por meio da instalação do Android Things. Após este passo, foi realizado também a conexão do sensor de umidade na placa.

A terceira parte da realização do projeto foi o desenvolvimento do software, este foi dividido em dois módulos: Raspberry e Mobile. Nesta etapa também foi realizada a construção da base de dados utilizando a ferramenta Realtime Database, do Firebase. Por fim, foi realizado a conexão na API de previsão de tempo Advisor.

A seguir, foi realizado a calibragem do sensor de umidade, utilizando o protocolo de secagem da amostra em forno de micro-ondas e o cálculo da equação da reta de regressão linear. Finalmente, foi feita a conexão da bomba de água á Raspberry Pi e realizado os testes de funcionamento.

7.2.1 Análise de solo

Uma análise de solo é capaz de identificar as características químicas e físicas do solo, ou seja, pode ser utilizada para identificar os nutrientes presentes no solo ou a proporção de areia, silte e argila que compõe. No que diz respeito a irrigação do solo, o que se deve levar em consideração é a textura e a densidade, sendo que estes parâmetros podem ser determinados na análise física.

A amostra foi retirada numa propriedade rural no município de Praia Grande, Santa Catarina, na profundidade de 0 a 20 centímetros (figura 8).

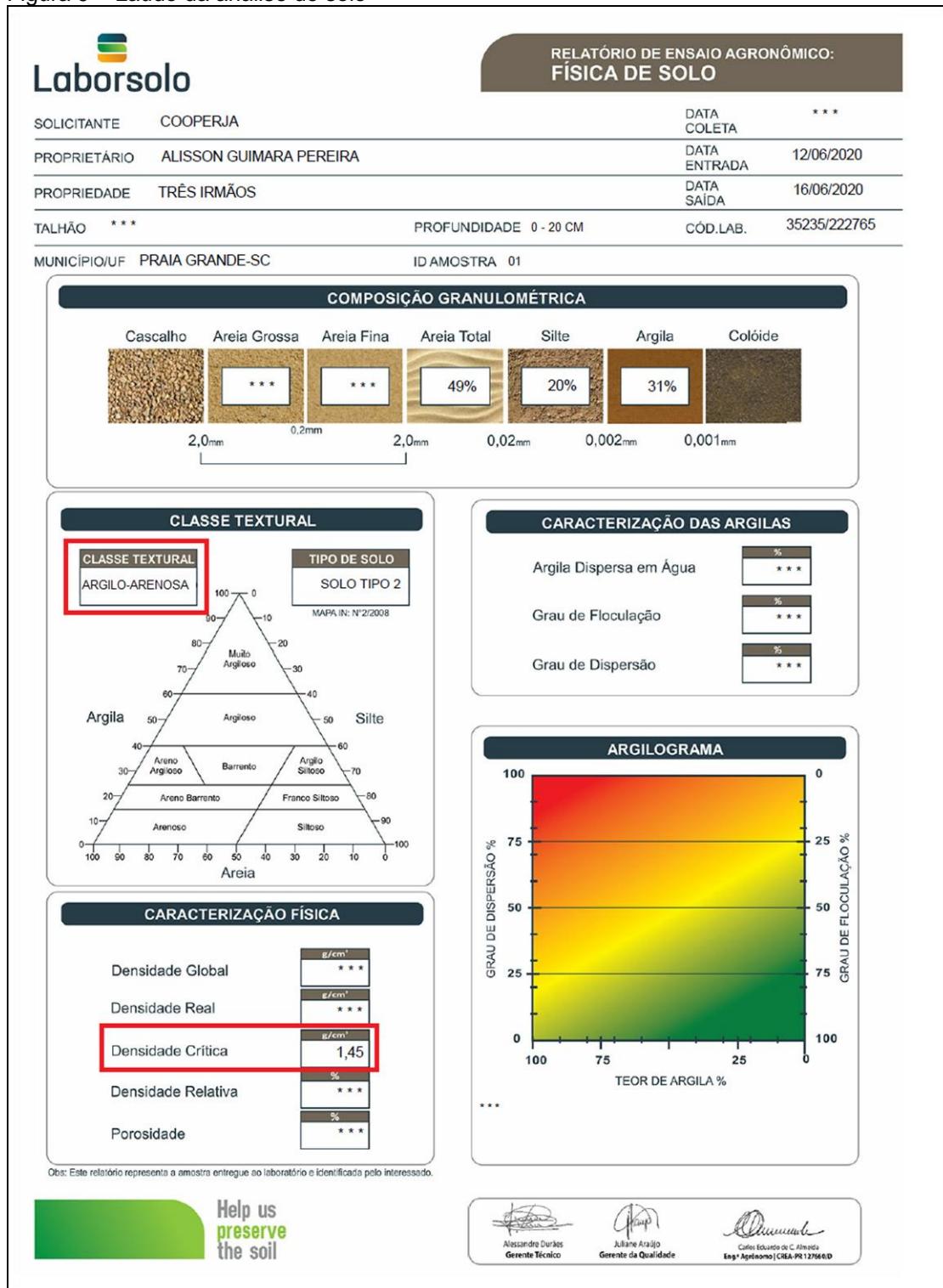
Figura 8 – retirada de amostra de solo



Fonte: Do autor.

Esta amostra foi secada na sombra por três dias, para retirar o excesso de umidade e após isto, foi empacotada e encaminhada para o laboratório de análises de solo Laborsolo. Após o retorno do laudo, foi possível determinar a textura do solo e a sua densidade (figura 9).

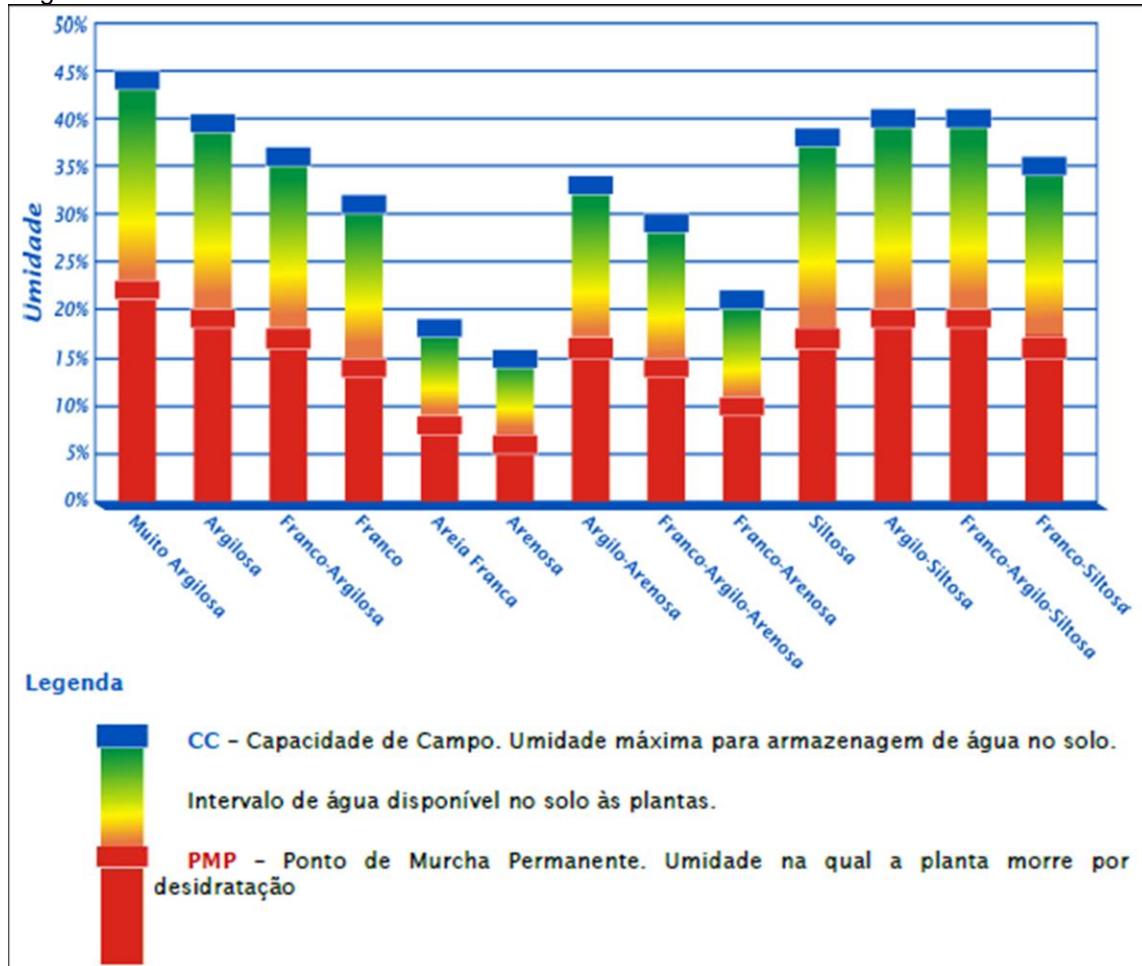
Figura 9 – Laudo da análise de solo



Fonte: Laborsolo (2020).

No laudo foi obtido o valor da densidade de 1.45 gr/cm³, e por meio da textura argilo-arenosa é possível descobrir o valor do PPM e da CC, que é de 16% e 33% respectivamente (figura 10).

Figura 10 - Índices de CC e PMP



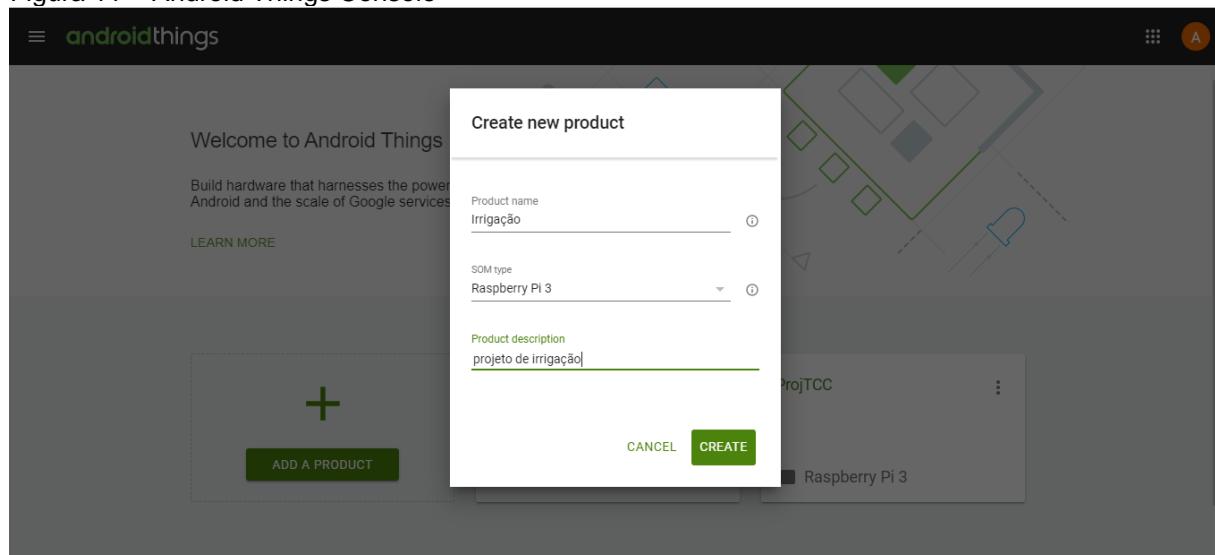
Fonte: Falker (2010).

7.2.2 Configuração da Raspberry com Android Things

A configuração da Raspberry consistiu em instalar o sistema operacional Android Things. Para isto, foi necessário criar uma conta no Android Things Console e criar um projeto (figura 11), onde define-se que a plataforma a ser usada seria a Raspberry Pi. Com o projeto finalizado, o Android Things Console disponibiliza para download uma imagem do sistema operacional com as configurações necessárias para a instalação na placa.

Feito isso, necessitou-se do uso do Win32DiskImager para gravar a imagem do SO no cartão microSD, destaca-se a importância de o cartão estar formatado em FAT32, com mínimo de 8GB de memória e classe 10, configurações diferentes destas citadas podem não funcionar como esperado.

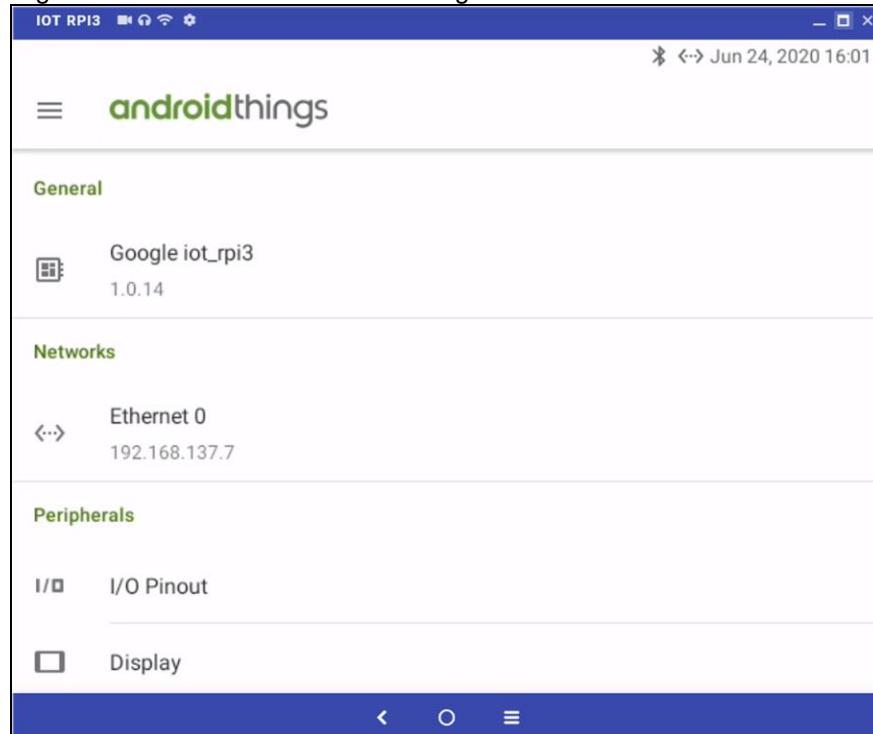
Figura 11 – Android Things Console



Fonte: O autor (2020).

Foi utilizado a ferramenta Vysor, que permitiu a visualização da interface do sistema operacional utilizando-se do IP da Raspberry Pi diretamente no Notebook, dispensando o uso de um monitor (figura 12). Esta interface serviu para facilitar a configuração da data e da hora da placa, que são utilizadas na gravação dos dados no Firebase, por este motivo elas devem estar corretas.

Figura 12 – Interface do Android Things



Fonte: Do autor.

7.2.3 Banco de dados Real Time Database

O Real Time Database do Firebase é um banco de dados NoSQL hospedado na nuvem, sincronizado com todos os clientes em tempo real. Estes dados são armazenados no padrão JSON. A configuração deste serviço ao projeto no Android Studio consistiu em criar um arquivo inicial na plataforma Firebase e registrar o pacote do aplicativo neste projeto, onde foi necessário informar o certificado de assinatura de depuração “SHA-1”, que é encontrado no *Gradle* do projeto criado no Android Studio.

Após esta etapa, o Firebase cria um arquivo de configuração, que deve ser adicionado dentro do diretório do modo raiz do projeto. Feito isto, o aplicativo já está pronto para utilizar todos os recursos disponíveis do Firebase. Para utilizar o Real Time Database, deve ser adicionado no arquivo *build.gradle* de cada módulo, dentro do *dependencies* o comando “*implementation 'com.google.firebaseio:firebase-database:18.0.0'*”;

7.2.3.1 Salvar dados

Para salvar os dados, pode-se obter uma referência à instância do banco do tipo *DatabaseReference*, como mostra-se no exemplo a seguir:

```
DatabaseReference firebaseReferencia = FirebaseDatabase.getInstance().getReference();
```

Por meio desta referência, obtém-se acesso a todos os nós criados no banco ou então é possível criar um nó *child*, ou seja, um nó filho onde pode-se armazenar os dados, como no exemplo abaixo, onde é criado um nó “planta” e dentro deste nó é criado outro nó “nome” e inserido o valor “arroz”.

```
firebaseReferencia.child("planta").child("nome").setValue("arroz");
```

Se acaso o nó “child” já existir no banco, a informação contida no “*setValue*” será apenas sobreescrita caso for diferente da que está salva.

7.2.3.2 Ler dados

Para ler os dados de um caminho e detectar as alterações em tempo real, é possível usar o método *addValueEventListener()*, para adicionar um evento de

leitura ao *DatabaseReference*. Desta forma é possível ler e detectar as alterações em todo conteúdo de um caminho.

Para usar o *addValueEventListener()*, é necessário definir uma referência do caminho desejado, como mostra o exemplo a seguir, onde se tem referência do nó “talhao”:

```
DatabaseReference plantaReferencia = firebaseReferencia.child("talhao");
```

Usando o método *onDataChange()*, é possível ler um *snapshot* de conteúdo, esse método será acionado uma vez quando o *listener* for anexado e sempre que os dados forem alterados, incluindo os filhos. O retorno desta chamada receberá um *snapshot* com todos os dados no local, se não houver dados o *snapshot* retornará *false* e quando chamar o *getValue()* nele, obterá o valor *null*. O exemplo a seguir mostra como foi realizado a leitura de dados, captando o valor da densidade do solo:

```
densidade.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        //ação
    }

    @Override
    public void onCancelled(@NonNull DatabaseError databaseError) {
        }
});
```

7.2.4 Desenvolvimento do módulo Raspberry

Nesta etapa do projeto, foi desenvolvido todo o módulo da placa, que envolve a leitura da umidade do sensor e calibração, leitura da previsão do tempo para 3 dias, gravação dos dados no Real Time Database, conexão com a bomba de água e a aplicação que monitora e define a irrigação.

Como dito anteriormente, a linguagem de programação usada foi o JAVA, utilizando o ambiente de programação Android Studio.

O Raspberry Pi 3B conta com 40 pinos GPIO e 4 entradas USB, na figura 13 é possível visualizar a posição, a função e o nome de cada pino. É importante observar que o nome do pino não é igual a sua posição, sendo que para definir e configurá-lo no Android Things, deve-se levar em consideração o seu nome.

Figura 13 – Pinos GPIO

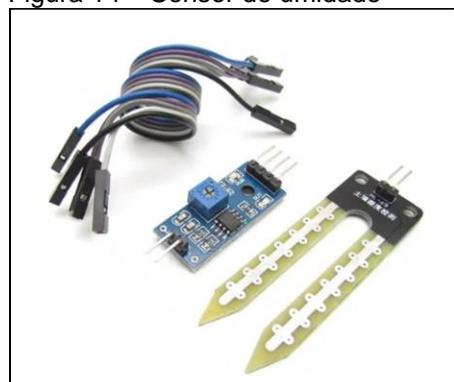
Raspberry Pi 3 GPIO Header			
Pin#	NAME		NAME
01	3.3v DC Power		DC Power 5v
03	GPIO02 (SDA1 , I ² C)		DC Power 5v
05	GPIO03 (SCL1 , I ² C)		Ground
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14
09	Ground		(RXD0) GPIO15
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18
13	GPIO27 (GPIO_GEN2)		Ground
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23
17	3.3v DC Power		(GPIO_GEN5) GPIO24
19	GPIO10 (SPI_MOSI)		Ground
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08
25	Ground		(SPI_CE1_N) GPIO07
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC
29	GPIO05		Ground
31	GPIO06		GPIO12
33	GPIO13		Ground
35	GPIO19		GPIO16
37	GPIO26		GPIO20
39	Ground		GPIO21

Fonte: FilipeFlop (2019).

7.2.4.1 Instalação do sensor de umidade

Foi necessário utilizar um sensor resistivo de umidade do modelo A080 (figura 14), que é capaz de medir a umidade do solo por meio da resistência elétrica presente, ou seja, quanto mais água presente no solo, menor será o valor lido da resistência elétrica. Este sensor tem saída de valor digital e analógica, na leitura digital é possível obter o valor lógico 0 e 1, sendo 0 para solo com umidade boa e 1 para solo seco, tendo sua regulagem no potenciômetro disponível no sensor. Na leitura analógica, é possível obter o valor da resistência elétrica da água do solo, sendo este utilizado para o projeto.

Figura 14 – Sensor de umidade



Fonte: Mercado Livre (2020)

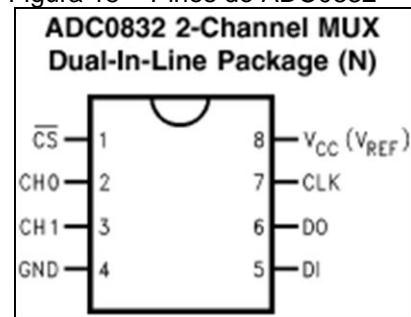
O sensor possui alimentação de 3.3V ou 5V, pinos de saída analógica, digital, GND e fase. Como este sensor opera na mesma tensão dos pinos GPIO da Raspberry, pode-se ligar diretamente, não sendo necessário utilizar nenhum tipo de resistor.

O pino fase do sensor foi conectado na coluna positiva da protoboard, esta por sua vez, está conectada no pino 01 de 3.3V da placa, o pino GND do sensor foi conectado na coluna GND da protoboard, sendo que esta foi ligada no pino 39 da placa. O pino utilizado para leitura foi o analógico, que foi conectado ao pino do canal 0 do conversor de sinal ADC0832, que transforma o sinal em digital.

O conversor ADC0832 possui 8 pinos (figura 15), dois canais de entrada que variam de 0V a 5V com uma resolução de 8 bits, o ADC0832 transforma o sinal analógico em sinal digital, assim pode-se realizar a leitura do valor pela placa. Nos demais pinos do conversor foi realizado a seguinte conexão:

- a) pino 4 do conversor na coluna GND da protoboard;
- b) pino 8 do conversor no pino 2 da Raspberry, que disponibiliza 5V;
- c) pino CS do conversor (pino 1), ligado no pino 12 (GPIO18) da placa;
- d) pino CLK do conversor (pino 7), ligado no pino 18 (GPIO 24) da placa;
- e) pino DI do conversor (pino 5), ligado no pino 16 (GPIO 23) da placa;
- f) pino D0 do conversor (pino 6), ligado no pino 38 (GPIO 20) da placa.

Figura 15 – Pinos do ADC0832



Fonte: Futurlec.com (2020).

Para realizar a leitura do valor da umidade utilizando-se do conversor de sinal, foi adaptado um driver do Gialluca (2017), sendo que o valor obtido na leitura é um primitivo do tipo “int” que contém o valor da resistividade da água presente no solo. Na figura 16 apresenta-se o modo que é configurado o conversor na programação, com base no driver citado anteriormente.

Figura 16 – Configuração do conversor ADB0832

```

120    public int getADCChannelValue(int channel) throws IOException {
121        if (channel != 0 & channel != 1) return Integer.MIN_VALUE;
122
123        int ad = 0;
124        // CS LOW.
125        gpioCS.setValue(true);
126        gpioCS.setValue(false);
127
128        // Start clock
129        gpioCLK.setValue(false);
130
131        // Input MUX address
132        for (int i = 0; i < 3; i++) {
133            if (i == 0 || i == 1 || channel == 1)
134                | gpioD1.setValue(true);
135            else gpioD1.setValue(false);
136
137            gpioCLK.setValue(true);
138            gpioCLK.setValue(false);
139        }
140
141        // Read 8 bits from ADC
142        for (int i = 0; i < 8; i++) [
143            gpioCLK.setValue(true);
144            gpioCLK.setValue(false);
145            ad = ad << 1;
146            boolean value = gpioD0.getValue();
147            if (value)
148                | ad |= 0x01;
149        ]
150
151        // Reset
152        gpioCS.setValue(true);
153
154        return ad;
155    }

```

Fonte: O autor.

7.2.4.2 Calibração do sensor de umidade

Como visto no capítulo anterior deste trabalho, o valor obtido do sensor não corresponde ao valor real da umidade do solo e sim ao valor da resistência da água presente. Para resolver este problema, foi necessário utilizar um método de calibração, que consistiu em aferir a umidade do solo utilizando-se do protocolo da EMBRAPA de secagem da terra em micro-ondas e aplicar uma equação de regressão linear.

Para aferir o valor real da umidade, realizou-se a secagem de 180 gramas de terra por 10 minutos em potência máxima no forno micro-ondas, de acordo com o protocolo da EMBRAPA (2002), neste processo a terra perde toda a umidade retida.

Após esta etapa, foi separado uma amostra de 130 gramas de solo seco e realizado a leitura pelo sensor de umidade. Seguindo o processo, foi adicionado porções de água no solo seco e aferido o valor pelo sensor. Primeiramente foi adicionado 13 ml de água obtendo valor de umidade de 10%, por segundo foi adicionado o total de 26 ml de água obtendo o valor de umidade de 20%, por terceiro

foi adicionado o total de 39 ml de água obtendo o valor de umidade de 30% e por último adicionado o total de 52 ml de água obtendo o valor de umidade em 40%.

Cada vez que era adicionado água, foi realizado o aferimento do sensor, assim foi possível obter pares de valores “umidade real x valor do sensor” (figura 17), sendo possível aplicar a equação de regressão linear.

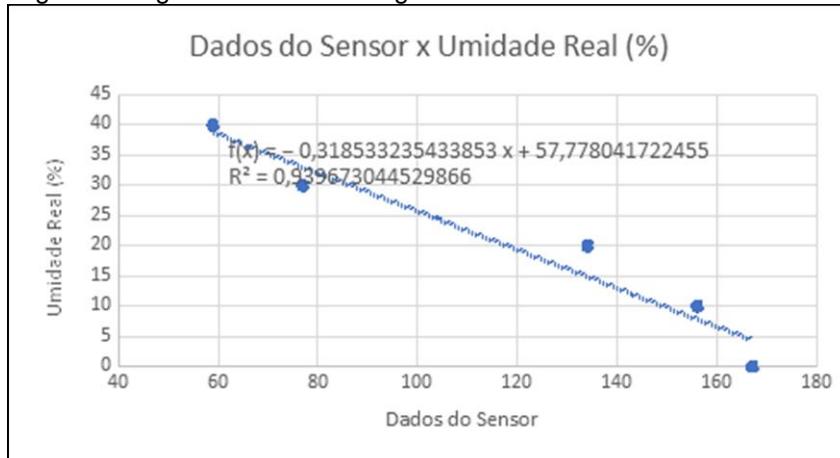
Figura 17 – relação entre umidade real x valor do sensor

água adicionada	umidade real (%)	valor do sensor
0	0	167
13 ml	10	156
26 ml	20	134
39 ml	30	77
52 ml	40	59

Fonte: O autor.

Aplicando estes dados no gráfico, foi obtido o resultado como mostra a figura 18. Foi alcançado o valor do coeficiente de determinação R^2 de 93,9%, isso mostra o quanto os dados estão próximos da linha de regressão ajustada.

Figura 18 – gráfico da reta de regressão



Fonte: O autor.

Por fim, a equação que transforma o dado do sensor em umidade real ficou como mostra a equação 4:

$$\text{Umidade} = (-0,3185 * (\text{valor do sensor})) + 57,778 \quad (4)$$

7.2.4.3 Comunicação com API Advisor

Nesta etapa foi realizado a comunicação da Raspberry com a API Advisor. Esta API utiliza como forma de comunicação o padrão REST com as mensagens no formato JSON. Foi utilizado a versão grátilis, que é capaz de realizar até 100

requisições por dia, ter uma chave de acesso e consultar a previsão para até 72 horas. A API também fornece versões pagas, sendo possível obter mais recursos, porém não foi necessário ao projeto.

Primeiramente foi criado uma conta de usuário na API, e gerado um token que serve para identificar o usuário que realiza as requisições. Para realizar a leitura da previsão do tempo, é necessário informar o ID da cidade que se deseja, então foi utilizado o recurso de descobrir o ID pelo nome da cidade, executando o seguinte comando:

```
http://apiadvisor.climatempo.com.br/api/v1/locale/city?name=Praia%20Grande&state=sc&token=643c73fff94c5b2fe221c5ede99cbeb4
```

Como retorno obteve-se:

```
[{"id":4782,"name":"Praia Grande","state":"SC","country":"BR"}]
```

Com o ID da cidade o token do projeto é possível acessar a previsão para 72 horas, executando o comando:

```
http://apiadvisor.climatempo.com.br/api/v1/forecast/locale/4782/hours/72?token=643c73fff94c5b2fe221c5ede99cbeb4
```

Para melhor visualização, será mostrado o retorno do JSON da primeira e da segunda hora do dia:

```
{"id":4782,"name":"Praia Grande","state":"SC","country":"BR","data":[{"date":"2020-06-27 00:00:00","date_br":"27/06/2020 00:00:00","rain":{"precipitation":0.6}, "wind":{"velocity":10.2, "direction":"SSW", "directiondegrees":223.1, "gust":26.7}, "temperature":{"temperature":12}}, {"date":"2020-06-27 01:00:00", "date_br":"27/06/2020 01:00:00", "rain":{"precipitation":0.6}, "wind":{"velocity":10, "direction":"SSW", "directiondegrees":206.9, "gust":24.5}, "temperature":{"temperature":12}}...]
```

A informação que foi relevante na previsão do tempo para este projeto, foi a precipitação de chuva obtida no campo “*precipitation*” do JSON. Este dado serviu para posteriormente montar a lógica de programação capaz de identificar o melhor momento para irrigação.

Para consumir esta API no JAVA, foi utilizado a biblioteca Retrofit. Com ela foi possível ganhar simplicidade no desenvolvimento, sendo necessário implementar uma simples interface e escrever alguns métodos.

Foi preciso adicionar ao arquivo *build.gradle* do projeto as seguintes dependências:

```
compile 'com.squareup.retrofit2:retrofit:2.1.0'
compile 'com.squareup.retrofit2:converter-gson:2.0.0-beta4'
```

A primeira dependência é a própria biblioteca do Retrofit, a segunda é o conversor Gson, que é utilizado para converter objetos JAVA na representação JSON.

Para estabelecer a conexão foi criado uma classe Interface, sendo informado uma String com a URL contendo a rota estabelecida para obter a previsão para 72 horas, como mostra a figura 19:

Figura 19 – Classe Interface

```
import retrofit2.Call;
import retrofit2.http.GET;

public interface ApiInterface {

    String BASE_URL = "http://apiadvisor.climatempo.com.br/api/v1/forecast/locale/4782/hours/";
    //List<RespostaServidor> list = new ArrayList<>();
    @GET("72?token=643c73fff94c5b2fe221c5ede99cbeb4")
    Call<classeData> lista();

}
```

Fonte: O autor.

Dentro da classe *Call* da Interface foi informado qual o retorno esperado, que no caso é a “*classeData*”. Dentro desta classe existe uma lista de datas, que corresponde as datas recebidas pelo JSON. Esta lista de datas representa outra classe, que contém o objeto “*precipitation*”, desta forma, obtém-se o retorno esperado da precipitação. O método implementado está contido no apêndice A deste trabalho.

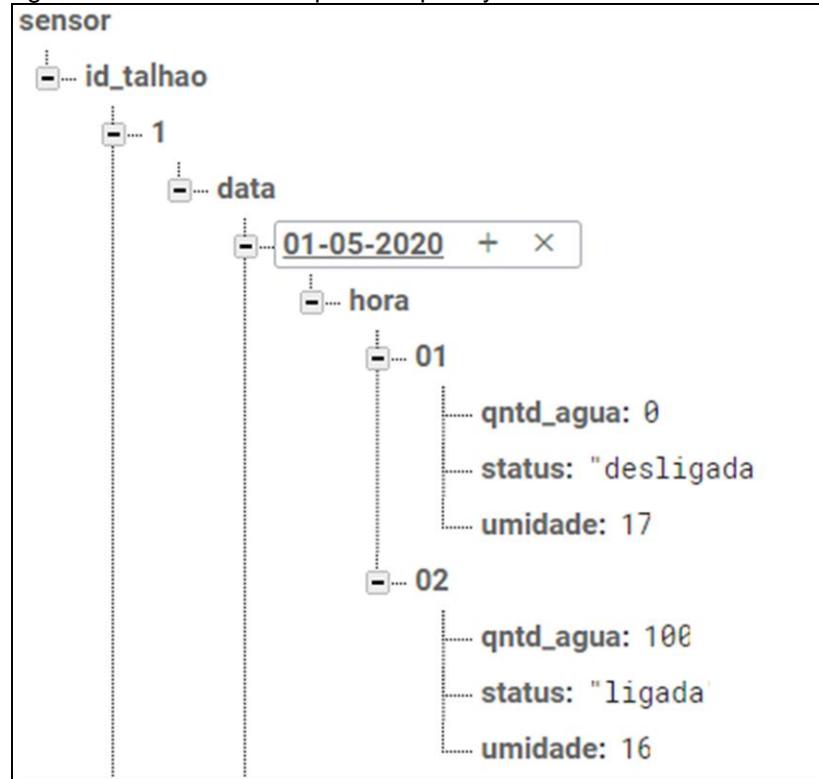
7.2.4.4 Salvando os dados no RealTime Database

Os dados salvos foram divididos por data e hora, para posteriormente serem gerados relatórios de consumo de água e dados do talhão conforme a data que o usuário desejar.

É também possível gravar os dados de acordo com o talhão cadastrado, ou seja, pode-se ter mais de um sensor de umidade de solo lendo os dados de talhões diferentes. Deste modo, garante-se que cada sensor possa corresponder a um talhão diferente, com plantas diferentes e com necessidade de água de acordo com a característica identificada. Os campos armazenados foram a umidade do solo, o

status da bomba de água (ligada ou desligada) e a quantidade de água gastada naquele momento (figura 20).

Figura 20 – Dados salvos pela Raspberry



Fonte: O autor.

7.2.4.5 Conexão com a bomba de água

Para realizar a conexão com a bomba de água 12V, foi necessário utilizar um módulo relé, pois a tensão da placa Raspberry é de 3.3V, logo se fosse realizado a conexão diretamente a placa poderia sofrer algum tipo de dano irreversível.

Este módulo relé possui um canal 3.3V que permite controlar cargas de até 30V de corrente contínua ou até 250V de corrente alternada. Este é um módulo *Low Level*, ou seja, quando se é inserido um valor lógico zero ele é acionado. Ele possui 3 pinos de conexão com a Raspberry, sendo um pino para o GND, um pino para o VCC e um pino para o sinal. Possui também três portas de saída, sendo uma normalmente aberta, do inglês *normally open* (NO), uma normalmente fechada, do inglês *normally closed* (NC) e uma porta comum (COM).

O pino GND do módulo foi conectado na linha GND da protoboard, o pino VCC do módulo foi conectado na protoboard na linha de 3.3V correspondente ao pino 1 da placa. E o pino do sinal do módulo foi conectado ao pino 35 da placa.

Na conexão com a bomba, foi utilizado a porta COM e a NC. Na porta COM foi conectado o fio positivo da fonte e na porta NC foi conectada ao pino positivo da bomba. Desta forma, quando o módulo relé recebe um valor *Low* da placa Raspberry, ele fecha a conexão da porta COM na porta NC e liga a bomba de água. A figura 21 mostra como foi realizada a conexão.

Figura 21 – conexão do módulo relé



Fonte: Do autor

7.2.5 Desenvolvimento do módulo mobile

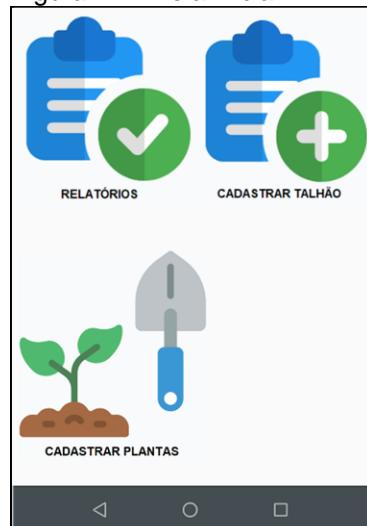
Neste módulo foi desenvolvido toda a aplicação do lado do usuário, implementado as telas de cadastro de planta, cadastro de talhão, relatórios de dados do talhão e consumo de água e tela inicial.

Cada tela construída corresponde a uma Activity diferente, isto serve para separar e organizar as funcionalidades diferentes de cada tela. Para criar esta classe, foi preciso estender a classe “Activity”, para que ela herde todas as características das Activities. Após isto, foi necessário sobrescrever o método “onCreate”, que é usado para configurar a interface do usuário.

7.2.5.1 Tela inicial

A tela inicial é bem simples, foi projetada para possibilitar ao usuário escolher entre três opções: Cadastro de talhão, cadastro de planta e relatórios (figura 22). Cada imagem corresponde ao um evento de clique, que direciona o usuário para as outras Activities.

Figura 22 – Tela inicial



Fonte: Do autor.

7.2.5.2 Cadastro de talhão

Esta *activity* permite ao usuário cadastrar dados referentes ao talhão (figura 23). Com estes dados salvos, é possível recuperá-los e usá-los como parâmetros na equação do cálculo de água.

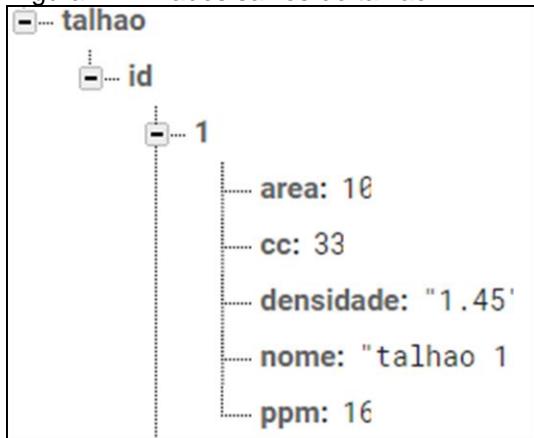
Figura 23 – Cadastro de talhão

Cadastro De Talhão	
Id:	<input type="text"/>
Nome:	<input type="text"/>
Area(HA):	<input type="text"/>
PPM:	<input type="text"/>
CC:	<input type="text"/>
Densidade:	<input type="text"/>
VOLTAR	CADASTRAR

Fonte: Do autor.

Estes dados são salvos no *RealTime Database* usando o Id do talhão como nó chave e o restante dos dados como nós filhos (figura 24), desta forma é possível recuperar os dados utilizando o Id do talhão como forma de busca.

Figura 24 – Dados salvos do talhao

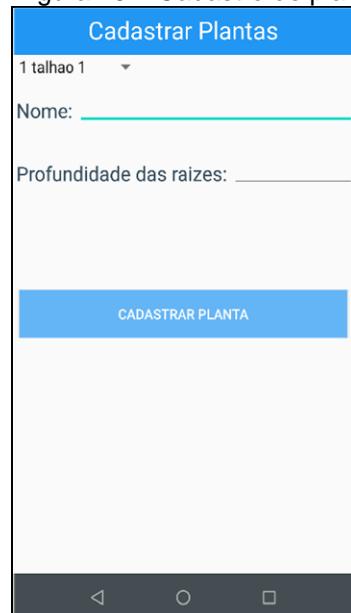


Fonte: Do autor.

7.2.5.3 Cadastro de planta

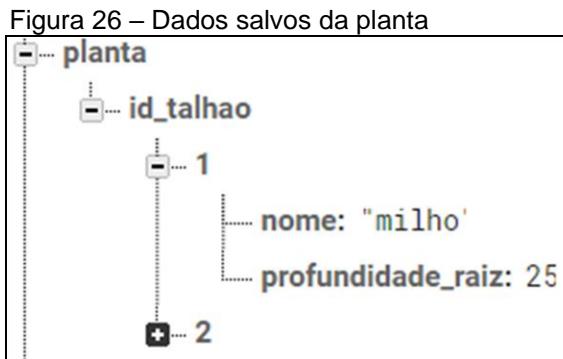
No cadastro de planta é mostrado ao usuário dentro de um *spinner* o Id e o nome do talhão, sendo que o nome da planta e a profundidade das raízes serão atribuídas aquele talhão selecionado. A figura 25 mostra como é o layout da tela de cadastro de plantas.

Figura 25 – Cadastro de plantas



Fonte: Do autor.

Os dados foram salvos dentro de um nó chave, referente ao Id do talhão, como mostra a figura 26.



Fonte: Do autor.

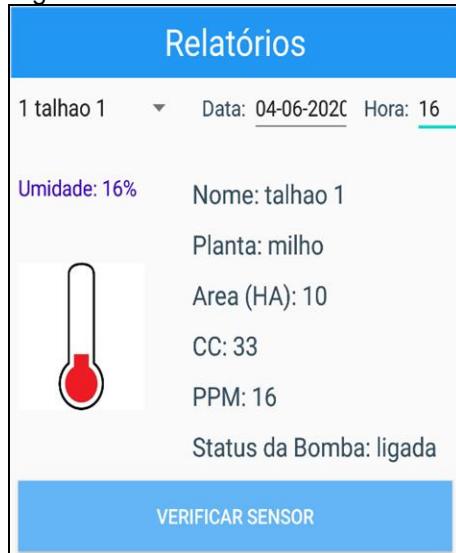
7.2.5.4 Relatórios

Esta *activity* mostra ao usuário, os dados referentes ao talhão selecionado no *spinner*. Ela é dividida em duas partes, sendo que a primeira exibe os dados do talhão, planta e umidade do solo captada pelo sensor e a segunda parte exibe os dados de gasto de água de acordo com o mês selecionado.

7.2.5.4.1 Relatório dos dados do talhão

Como mostra a figura 27, este relatório permite ao usuário selecionar a data e a hora que deseja obter as informações, ele vem por padrão selecionado a data e hora atual. É exibido um termômetro, que muda de acordo com a umidade do solo, quando a umidade está próxima ao PPM ele fica vermelho, quando a umidade está ideal ele fica verde e quando a umidade está próxima ou acima da CC ele fica azul.

Figura 27 – Relatório do sensor



Fonte: Do autor.

As informações que ficam ao lado do termômetro são obtidas por meio da leitura de dados do *RealTime Database*, sendo utilizado o Id do talhão captado na *spinner* como parâmetro. Elas são carregadas na tela quando o usuário clica no botão “**“VERIFICAR SENSOR”**”, sendo criado um evento de botão, que recupera os dados e insere eles na tela.

7.2.5.4.2 Relatório de gasto de água

Este relatório exibe um *GraphView*, que é um gráfico em forma de barras (figura 28), foi utilizado desta forma para melhor visualização dos dados, porém um *GraphView* também pode-se tornar um gráfico de linha

Figura 28 – Relatório de gasto de água



Fonte: Do autor.

Para alimentar o gráfico, primeiramente o usuário deve selecionar o mês no *spinner* e clicar no botão “**VERIFICAR GASTO DE ÁGUA**”. Este clique gera um evento de botão, que chama a classe que lê os dados.

Para realizar a leitura, é usado o mês capturado no *spinner* dentro de uma estrutura de repetição *for*, que vai de 0 a 31, que representa os dias do mês. A cada repetição, é concatenado o dia com o mês, usando isto como chave para realizar a leitura no banco de dados, somando a quantidade de água gasta em cada dia em um *arrayList*. Se o mês não ter sido completo ainda ou não ter 31 dias, a leitura será finalizada no último dia lido.

Para exibir os dados na tela, é lido o *arrayList* e adicionado no *GraphView*, sendo que cada barra corresponde a um dia do mês.

7.2.6 Lógica para programar a irrigação

A última etapa do projeto consistiu na construção da lógica para realizar a irrigação automatizada. Esta parte envolve tudo o que foi desenvolvido até aqui, pois ela utiliza a umidade do solo, previsão do tempo, dados da análise de solo, leitura e gravação no banco de dados e a conexão com a bomba de água.

A lógica de programação desenvolvida utilizou a equação 3 para calcular a necessidade de água, de acordo com a bibliografia. Primeiramente foi testada as condições da umidade e previsão, para decidir ligar a bomba, como mostra-se a seguir:

a) a primeira condição foi testar se a umidade do solo está boa, comparando se ela está maior que 3% em relação ao PPM, que ficou definido em 16% de acordo com a análise de solo. Se isto for verdadeiro, a bomba fica desligada;

b) se a primeira condição for falsa, então é testado se a umidade do solo está menor ou igual que o PPM, ou seja 16%. Se isto for verdadeiro, então é acionado a bomba de água, independente da previsão do tempo;

c) se a condição acima for falsa, significa que a umidade do solo está entre 17% á 19%. Neste ponto é realizado a leitura da previsão do tempo para 3 dias, se a precipitação neste período for igual a zero, então é iniciado a irrigação, pois presume-se que não irá chover, consequentemente a umidade do solo atingirá o PPM;

d) se a condição acima for falsa, significa que a umidade do solo ainda está entre 17% á 19%, porém a previsão para os próximos 3 dias diz que pode ocorrer incidênciа de chuva. Se a precipitação for maior ou igual que 25% da necessidade de água, então a bomba fica desligada;

e) se a condição acima também for falsa, significa que a previsão de precipitação de chuva é maior que zero, porém será menor que 25% da necessidade de água. Então decide-se ligar a bomba, pelo fato de que a chuva pode não ser suficiente para realizar a irrigação do solo.

Quando alguma das condições forem verdadeiras, é realizado a chamada do método que salva as informações no banco, sendo salvo os dados da umidade do solo, quantidade de água gasta e status da bomba de água. A placa foi programada de modo que faça a leitura da umidade e salvar os dados a cada hora.

7.3 RESULTADOS OBTIDOS

De acordo com o levantamento bibliográfico realizado, uma irrigação por sensoriamento deve ou não ser iniciada de acordo com a umidade do solo, sendo que em nenhuma hipótese o solo deve atingir a umidade menor que a do PPM, caso isto ocorra, a planta não consegue retirar água do solo, acarretando em danos econômicos para a cultura.

O modo como foi desenvolvido a lógica para programar a irrigação do solo, permite que a umidade nunca fique menor que a definida como PPM, considerando a previsão do tempo para tomar a decisão de irrigar.

Para realizar o cálculo da necessidade de água utilizou-se a equação 3 e os dados da análise de solo e umidade. Por exemplo, supondo que a umidade do solo esteja em 17% e a previsão de precipitação seja igual a zero, temos:

$$\boxed{LL = \frac{(CC - U_i)}{10} \cdot d \cdot z} \quad (3)$$

$$LL = \frac{(33 - 17)}{10} \cdot (1.45) \cdot 20$$

$$LL = 46,4 \text{ mm de água por metro quadrado}$$

Sendo que:

- a) CC é igual a 33% definido na análise de solo;
- b) Ui é igual a 17%, que é a umidade captada pelo sensor;
- c) d é igual a 1.45, que é a densidade do solo definida na análise;
- d) z é igual a 20cm, usado como exemplo definido pelo usuário.

De acordo com estes resultados obtidos, é tomado a decisão de ligar a bomba de água. Foi desenvolvido então um método que calcula qual o tempo que a bomba deve ficar ligada por meio da equação 4, levando em consideração a quantidade de água, tamanho da área e vazão da bomba.

$$\boxed{tempo(\text{seg}) = \frac{\text{quantidade de água(litros/m}^2) \cdot \text{área(m}^2)}{\text{vazão da bomba (litros por minuto)}} \cdot 60} \quad (4)$$

Considerando que a quantidade de água é de 46,4 litros por metro quadrado, uma área de 1 metro quadrado e a vazão da bomba de 2,0 litros por minuto, temos:

$$tempo(\text{seg}) = \frac{46,4 \cdot 1}{2} \cdot 60$$

$$tempo(\text{seg}) = 1392$$

Este resultado é usado para determinar o tempo em que a bomba ficará ligada para que o solo atinja a CC.

Os testes realizados para verificação do sistema de irrigação não foram no campo, pois a fragilidade dos componentes da placa não permite que ela fique exposta diretamente ao clima, sendo necessário criar uma estrutura de proteção. Porém foi considerado todas as condições possíveis de umidade de solo. A figura 29 mostra o projeto final, com todas as conexões realizadas, sendo utilizado uma porção de solo para os testes.

Figura 29 – projeto completo



Fonte: Do autor.

Primeiramente foi testado em condições de umidade ideal, obtendo o resultado da bomba desligada, assim como o esperado. O teste com a umidade do solo abaixo do PPM também foi realizado, a bomba foi ligada assim que o sistema realizou a verificação do sensor.

Além destes dois testes citados, também foi verificado os casos em que o solo está com a umidade próxima ao PPM, devendo ser consultado a previsão do tempo para a tomada de decisão. Neste cenário, o solo estava com umidade de 19%, então o sistema realizou uma consulta de precipitação de chuva e foi verificado volume zero para os próximos 3 dias, então foi iniciado a irrigação, assim como esperado. No segundo caso, foi testado no solo com 18% de umidade, o sistema verificou que a previsão de precipitação seria suficiente para realizar a irrigação, então a bomba de água não foi acionada.

Como toda previsão de tempo não é 100% exata, há chances de o sistema verificar precipitação e ela não ocorrer, por isso é decidido na lógica de programação que realize a irrigação quando o solo atingir o nível de PPM, independente da previsão de chuva, garantindo assim que não ocorra danos econômicos na cultura implantada.

No início do desenvolvimento deste projeto foi considerado que a bomba de água seria desligada quando a umidade captada pelo sensor fosse igual a

determinada na CC. Porém o problema encontrado foi em relação a estrutura física do solo, pois em solos muito argilosos por exemplo, a densidade faz com que a água demore mais a infiltrar do que em solos arenosos. Isto pode gerar um gasto de água maior do que o necessário. Este problema foi resolvido utilizando-se da equação 3 e 4, conforme foi mostrado anteriormente.

Outro problema identificado foi em relação a vida útil do sensor de umidade. Em pesquisas levantadas na internet, foi observado que este sensor não é o ideal para trabalho de campo, porém para protótipos ele pode ser usado, isto se deve ao fato de suas partes metálicas oxidarem rapidamente. O sensor utilizado no projeto necessitou ser trocado após 50 dias de uso, pois não apresentava as leituras corretamente.

7.4 ANÁLISE DOS DADOS

De acordo com os trabalhos correlatos levantados no decorrer deste projeto, foi desenvolvido o sistema de irrigação automatizado baseado em IoT. Levando em consideração os resultados obtidos por Kamienski et al. (2019, tradução nossa), no desenvolvimento da plataforma de gerenciamento de irrigação SWAMP, obteve-se resultados semelhantes em relação a divisões das camadas.

Os resultados obtidos nos trabalhos de Cunha e Rocha (2015) e Gonçalves et al. (2018) também levam em conta a umidade mínima definida para iniciar a irrigação, porém não consideram a previsão do tempo. Baseado neste fato, pode-se dizer que o sistema desenvolvido neste projeto tem a capacidade de gerar maior economia de água, visto que a previsão do tempo está integrada na tomada de decisão.

Por fim os resultados do trabalho de Keswani et. al. (2018, tradução nossa) serão considerados para o desenvolvimento de um projeto futuro, visando a construção de um mapa de sensores na propriedade, conectando vários sensores.

8 CONCLUSÃO

O proposito inicial deste trabalho foi baseado em um grave problema de abastecimento de alimentos relacionado com o alto crescimento populacional do mundo. Que tem como estratégia de solução aumentar a produtividade nas áreas já cultivadas, tendo a irrigação como um importante aliado.

Diante dos levantamentos bibliográficos, foi realizado a implementação de um protótipo de irrigação automatizada, que utiliza dados do clima e sensoriamento do solo para decidir o melhor momento para iniciar uma irrigação.

Alguns pontos críticos foram observados durante o desenvolvimento deste projeto. O primeiro deles foi em relação a calibração do sensor de umidade utilizando uma equação de regressão linear, obtendo um R^2 de 93,9%, isto significa que na conversão do valor da resistividade do sensor para a umidade real do solo pode haver uma variação de até 6,1% nos dados. Uma forma de se melhorar este fator é realizar mais testes de calibração, até ajustar este valor acima de 95% como foi encontrado na bibliografia.

Outro ponto crítico é sobre a vida útil do sensor de umidade, como foi citado anteriormente nos resultados da pesquisa. Uma possível solução para este problema seria a troca dos sensores resistivos por sensores capacitivos.

Também foi observado a dificuldade de realizar os testes no campo, devido a fragilidade dos componentes físicos da placa. Uma possível solução para este problema seria criar uma estrutura de proteção, que permita que a placa fique exposta a condições climáticas por um grande tempo, podendo assim captar uma quantidade grande de dados.

Nos testes realizados in loco, foi observado que o projeto se comportou como o esperado. Realizando as irrigações quando foram necessárias, salvando os dados no RealTime Database. Assim, foi possível gerar relatórios de gasto de água e informações pertinentes aos talhões cadastrados, sendo possível acompanhar os dados conforme a data e hora que o usuário desejar.

Ao finalizar o projeto foi possível atingir os objetivos estabelecidos, criando uma tecnologia baseada em IoT, que pode ser aplicada em larga escala utilizando-se da mesma lógica que foi aplicada no desenvolvimento deste projeto.

Como trabalho futuro, sugere-se o desenvolvimento de uma rede de sensores de umidade de solo, que se conectam com a placa controladora

remotamente. Deste modo seria possível controlar vários talhões simultaneamente, realizando a leitura dos dados e tomando a decisão de irrigar cada talhão separadamente. Assim seria possível que a placa controladora, sendo ela a Raspberry Pi ou não, ficasse fixada em um único lugar dentro da propriedade.

REFERÊNCIAS

ALBUQUERQUE, Paulo Emílio Pereira de. Estratégias de Manejo de Irrigação: Exemplos de Cálculo. **Circular Técnica**, Sete Lagoas, v. 136, n. 1, p.1-25, Set. 2010.

CUNHA, K. C. B. da; ROCHA, R. V. Automação no processo de irrigação na agricultura familiar com plataforma Arduíno. **RECoDAF – Revista Eletrônica Competências Digitais para Agricultura Familiar**, Tupã, v. 1, n. 2, p. 62-74, jul./dec. 2015. ISSN: 2448-0452

DAVIS, S.I.; DUKES, M.d. Irrigation scheduling performance by evapotranspiration-based controllers. **Agricultural Water Management**, [s.I.], v. 98, n. 1, p.19-28, dez. 2010. Elsevier BV. <http://dx.doi.org/10.1016/j.agwat.2010.07.006>.

DEVELOPER ANDROID. **Android things**: raspberry pi. 2019. Disponível em: <<https://developer.android.com/things/hardware/raspberrypi>>. Acesso em: 17 jun. 2019.

EMBRAPA (Brasil). Determinação de matéria seca e umidade em solos e plantas com forno de microondas doméstico. **CIRCULAR TECNICA**, SÃO CARLOS, ed. 33, 2002. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/33322/1/CrcularTecnica33.pdf>. Acesso em: 27 maio 2020.

GARCIA, Cristiano M.; ABILIO, Ramon. Integração entre Sistemas utilizando Web Services REST e SOAP: Um Relato Prático. **Sistemas de Informação**, Macaé, v. 19, n. 1, p.34-41, jun. 2017.

GARRET, Filipe. **Cinco sistemas operacionais para usar no Raspberry Pi**. 2016. Disponível em: <<https://www.techtudo.com.br/listas/noticia/2016/09/cinco-sistemas-operacionais-para-usar-no-raspberry-pi.html>>. Acesso em: 25 maio 2019

FALKER (Porto Alegre). Aplicações do HidroFarm. **Nota de Aplicação – Hfm 2010**, Porto Alegre, v. 1, n. 1, p.1-5, dez. 2010.

GIALLUCA, Giovanni di. **Android Things Drivers**. 2017. Disponível em: <https://github.com/giovannidg/androidthingsdrivers/tree/master/adc0832>. Acesso em: 20 maio 2020.

GOAP, Amarendra et al. An IoT based smart irrigation management system using Machine learning and open source technologies. **Computers And Electronics In Agriculture**, [s.I.], v. 155, p.41-49, dez. 2018. Elsevier BV. <http://dx.doi.org/10.1016/j.compag.2018.09.040>.

GONCALVES, Diego et al. Sistema IoT para monitoramento e controle de irrigação. In: ESCOLA REGIONAL DE INFORMÁTICA DO PIAUÍ (ERIPI), 4., 2018, Teresina. **Anais da IV Escola Regional de Informática do Piauí**. Porto Alegre: Sociedade Brasileira de Computação, oct. 2018 . p. 310 - 315.

HEIN, Werner. **Raspberry Pi aplicado a projetos do mundo real**. 2013. Disponível em:
[<http://blogs.linuxmagazine.com.br/images/uploads/pdf_aberto/LM_100_60_65_06_tut_raspberry_pi.pdf>](http://blogs.linuxmagazine.com.br/images/uploads/pdf_aberto/LM_100_60_65_06_tut_raspberry_pi.pdf). Acesso em: 24 fev. 2019.

JOHNSTON, Steven; COX, Simon. The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams. **Electronics**, Southampton, v. 6, n. 3, p.51-58, 12 jul. 2017. MDPI AG. <http://dx.doi.org/10.3390/electronics6030051>.

KAMIENSKI, Carlos et al. Smart Water Management Platform: IoT-Based Precision Irrigation for Agriculture. **Sensors**, [s.l.], v. 19, n. 2, p.276-284, 11 jan. 2019. MDPI AG. <http://dx.doi.org/10.3390/s19020276>.

KESWANI, Bright et al. Adapting weather conditions based IoT enabled smart irrigation technique in precision agriculture mechanisms. **Neural Computing And Applications**, [s.l.], v. 31, n. 1, p.277-292, 26 set. 2018. Springer Nature. <http://dx.doi.org/10.1007/s00521-018-3737-1>.

KIJNE, Jacob W.. Teaching irrigation science and water management: accepting professional diversity. Editorial. **Irrigation Science**, [s.l.], v. 29, n. 1, p.1-10, 27 out. 2010. Springer Nature. <http://dx.doi.org/10.1007/s00271-010-0239-z>.

KÖGLER, F.; SÖFFKER, D.. Water (stress) models and deficit irrigation: System-theoretical description and causality mapping. **Ecological Modelling**, [s.l.], v. 361, p.135-156, out. 2017. Elsevier BV. <http://dx.doi.org/10.1016/j.ecolmodel.2017.07.031>.

KÖLLING, Michael. Educational Programming on the Raspberry Pi. **Electronics**, Canterbury, v. 5, n. 4, p.33-50, 24 jun. 2016. MDPI AG. <http://dx.doi.org/10.3390/electronics5030033>.

LEE, In; LEE, Kyoochun. The Internet of Things (IoT): Applications, investments, and challenges for enterprises. **Business Horizons**, [s.l.], v. 58, n. 4, p.431-440, jul. 2015. Elsevier BV. <http://dx.doi.org/10.1016/j.bushor.2015.03.008>

MARQUELLI, Waldir A.; SILVA, Washington L.c.. **SELEÇÃO DE SISTEMA DE IRRIGAÇÃO PARA HORTALIÇAS**. 2011. Disponível em:
[<https://ainfo.cnptia.embrapa.br/digital/bitstream/item/75698/1/ct-98.pdf>](https://ainfo.cnptia.embrapa.br/digital/bitstream/item/75698/1/ct-98.pdf). Acesso em: 22 abr. 2019.

MENG, Michael; STEINHARDT, Stephanie; SCHUBERT, Andreas. Application Programming Interface Documentation: What Do Software Developers Want?. **Journal Of Technical Writing And Communication**, Merseburg, v. 48, n. 3, p.295-330, 26 jul. 2017. SAGE Publications. <http://dx.doi.org/10.1177/0047281617721853>.

MYERS, Brad A.; STYLOS, Jeffrey. Improving API usability. **Communications Of The Acm**, [s.l.], v. 59, n. 6, p.62-69, 23 maio 2016. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/2896587>.

ONU (Estados Unidos). **POPULATION**. 2015. Disponível em: <<https://www.un.org/en/sections/issues-depth/population/index.html>>. Acesso em: 18 abr. 2019.

PAULINO, Janaina et al. **SITUAÇÃO DA AGRICULTURA IRRIGADA NO BRASIL DE ACORDO COM O CENSO AGROPECUÁRIO 2006**. 2011. Disponível em: <<https://repositorio.unesp.br/bitstream/handle/11449/72671/2-s2.0-80052502561.pdf?sequence=1&isAllowed=y>>. Acesso em: 22 abr. 2019.

RAJALAKSHMI, P; MAHALAKSHMI, S. Devi. IOT based crop-field monitoring and irrigation automation. **2016 10th International Conference On Intelligent Systems And Control (isco)**, Coimbatore, p.1-6, jan. 2016. IEEE. <http://dx.doi.org/10.1109/isco.2016.7726900>.

RASPBERRY PI FUNDATION (Reino Unido). **Ten millionth Raspberry Pi, and a new kit**. 2016. Disponível em: <<https://www.raspberrypi.org/blog/ten-millionth-raspberry-pi-new-kit/>>. Acesso em: 13 abr. 2019.

SOUZA, Ramon. **Android Things**: o SO da Google para IoT, é finalmente lançado em edição estável. [S. l.], 2018. Disponível em: <<https://canaltech.com.br/internet-das-coisas/android-things-o-so-da-google-para-iot-e-finalmente-lancado-em-edicao-estavel-113308/>>. Acesso em: 16 abr. 2019.

MUXITO, Ezequiel Manuel et al. IoT na Agricultura: Automação de Pivôs e Canais de Irrigação com Arduino e Webservice. In: III CONGRESSO INTERNACIONAL ADVENTISTA DE TECNOLOGIA (CIAT), 3., 2018, Hortolândia. **Congresso**. Hortolândia: Researchgate, 2018. p. 1 - 10.

ROOPAEI, Mehdi; RAD, Paul; CHOO, Kim-kwang Raymond. Cloud of Things in Smart Agriculture: Intelligent Irrigation Monitoring by Thermal Imaging. **Ieee Cloud Computing**, [s.l.], v. 4, n. 1, p.10-15, jan. 2017. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/mcc.2017.5>.

SAUDATE, Alexandre. **Rest**: construa AP'Is inteligentes de maneira simples. São Paulo: Casa do Código, 2013. 315 p.

TANENBAUM, Andrew s; WOODHULL, Albert s. **OPERATING SYSTEMS DESIGN AND IMPLEMENTATION**. 3. ed. Massachusetts: Prentice Hall, 2006. 1080 p.

ZARANZA, bruno. **Hello World com Raspberry Pi 3 usando Android Things**. [S. l.], 19 dez. 2016. Disponível em: <https://medium.com/android-dev-br/hello-world-com-raspberry-pi-3-usando-android-things-2b10fa6dc21d>. Acesso em: 17 jun. 2019.

APÊNDICE(S)

APÊNDICE A – classe implementada para a leitura da precipitação

```
public class LerPrevisao {  
    private Double precipitacao = null;  
  
    // método que le a previsão do tempo para 3 dias  
    public Double previsao() {  
        final Retrofit[] retrofit = {new Retrofit.Builder().baseUrl(ApiInterface.BASE_URL).addConverterFactory(GsonConverterFactory.  
            create()).build();}  
  
        ApiInterface service = retrofit[0].create(ApiInterface.class);  
        Call<classeData> requestResposta = service.lista();  
  
        requestResposta.enqueue(new Callback<classeData>() {  
            @Override  
            public void onResponse(Call<classeData> call, Response<classeData> response) {  
                Double soma = 0.0;  
  
                if (!response.isSuccessful()) {  
                    Log.e("ERRO", "erro" + response.code());  
                } else {  
                    classeData resp = response.body();  
  
                    for (Datas d : resp.data) {  
                        soma += Double.parseDouble(d.getRain().getPrecipitation().toString());  
                    }  
                    precipitacao = soma;  
                    Log.e("CERTO", "soma1" + "-----" + precipitacao);  
                }  
            }  
            @Override  
            public void onFailure(Call<classeData> call, Throwable t) {  
                Log.e("ERRO", "erro" + t.getMessage() + " aaaaa " + t.getStackTrace());  
            }  
        });  
        return precipitacao;  
    }  
}
```

APÊNDICE B - ARTIGO

Sistema de Irrigação Automatizado Utilizando o Conceito de IoT Para Tomada de Decisão

Alisson G. Pereira¹

¹Universidade do Extremo Sul Catarinense – UNESC

Críiuma – SC – Brazil

alissonguimarap@unesc.net

Abstract. *The strong growth of the world population is generating more demand in food production. The challenge is to increase productivity in the same cultivated area, and one of the solutions is through the use of irrigation, which can be increased by up to 2.5 times the production in relation to dry areas. However, irrigation generates expenditure of water and resources such as electricity and fuel, if not used correctly. So the proposed model was to use the Raspberry Pi as a controller board, integrating soil moisture sensors and the Advisor weather forecast API, applying the concept of IoT in irrigation, so the decision to irrigate can be taken based on moisture and forecast data, generating reports of humidity and water consumption through a mobile application.*

Resumo. *O crescimento acentuado da população mundial está gerando cada vez mais demanda por produção de alimentos. O desafio é aumentar a produtividade na mesma área cultivada, sendo que uma das soluções é através do uso da irrigação, que pode aumentar em até 2,5 vezes a produção em relação a áreas sequeiras. Porém a irrigação gera gasto de água e de recursos como energia elétrica e combustível, se não for usada de forma correta. Então o modelo proposto foi usar a Raspberry Pi como placa controladora, integrando sensores de umidade de solo e a API de previsão de tempo Advisor, aplicando o conceito de IoT na irrigação, para que a decisão de irrigar possa ser tomada com base nos dados de umidade e previsão, sendo então gerados relatórios de umidade e consumo de água por meio de um aplicativo mobile.*

1. Introdução

Com o aumento da população mundial, é necessário produzir mais alimentos para evitar o desabastecimento (ONU, 2015). A maior parte das terras cultiváveis no mundo estão ocupadas, sendo então o desafio produzir mais com a mesma quantidade de terra. A forma mais eficiente de atingir este objetivo é usando a irrigação (KÖGLER; SÖFFKER, 2017, tradução nossa).

Porém com isto surge outro problema, que é o uso racional da água, que demanda cada vez mais tecnologia para se irrigar eficientemente uma lavoura. Ainda existem poucas opções tecnológicas eficientes pensando nas pequenas propriedades rurais quando se fala em irrigação, sendo inviável para um pequeno produtor adquirir um sistema completo de monitoramento de solo e das condições climáticas.

Visto isso, surge o problema computacional de unir o monitoramento das condições do solo, via sensor de umidade e das condições do clima, via API de previsão de tempo, com baixo custo para pequenos produtores.

1.1 Objetivo geral

Empregar o conceito de IoT em um sistema de irrigação, utilizando um Raspberry Pi como ferramenta controladora e a API Advisor para auxiliar na tomada de decisão.

1.2 Objetivos específicos

Os objetivos específicos deste projeto consistem em:

- a) aplicar os conceitos de IoT aplicado no setor agrícola no processo de irrigação;
- b) desenvolver um software para o controle e administração da irrigação, bem como, utilizar a API Advisor para obter informações da previsão do tempo;
- c) utilizar o Raspberry Pi como controlador do sistema de irrigação, conectando-o a um sensor de umidade e numa bomba de água;
- d) simular uma mini plantação irrigada, utilizando-se das tecnologias desenvolvidas.

1.3 Justificativa

O estudo de economia das águas e tecnologias precisas de irrigação tem sido objeto de estudos científico em todo mundo, tendo como objetivo o aumento da eficiência no uso da água (KÖGLER; SÖFFKER, 2017, tradução nossa).

Goap et. al. (2018, tradução nossa) e Kamienski et. al. (2019, tradução nossa) afirmam que o uso de IoT surge como uma escolha natural para o gerenciamento inteligente das águas. Rajalakshmi e Mahalakshmi (2016, tradução nossa) dizem que o uso de sensores para monitoramento da umidade do solo, aliado com a irrigação localizada por gotejamento pode economizar até 53% de água em relação a irrigação por aspersão.

O uso do Raspberry Pi, aliada com os dados da previsão do tempo obtidos através da API Advisor que aplica o conceito REST, visando construir uma tecnologia de irrigação acessível para as pequenas propriedades, obtendo economia de dinheiro e economia de água, tornará este projeto viável e justificável.

2. Irrigação

De acordo com informações da Empresa Brasileira de Pesquisa Agropecuária (EMBRAPA, 2012), a quantidade certa de água e o momento ideal da aplicação são os pontos chaves para se obter um manejo adequado, favorecendo a economia de água e energia, resultando assim, num melhor desempenho das plantas.

Porém, para que a irrigação seja eficiente é necessário que vários fatores sejam analisados, relacionados não somente ao cultivo, como também questões edafoclimáticas. Clima e solo são os fatores que mais interferem na decisão do sistema de irrigação (EMBRAPA, 2012).

2.1 Conceitos básicos para programar a irrigação

Existem vários pontos a serem analisados para decidir o momento ideal da irrigação e a quantidade de água, alguns métodos utilizam somente as características físicas do solo e evapotranspiração da cultura, outros métodos são baseados em sensores para monitorar o potencial ou umidade do solo.

Á Água Total Disponível no solo (ATD) é o total de água que pode ser absorvida pela planta, está entre a umidade da Capacidade de Campo (CC – limite superior de água disponível)

e o Ponto de Murcha Permanente (PMP – limite inferior de água disponível) (ALBUQUERQUE, 2010).

2.1.1 Irrigação de acordo com instrumentos de medição de umidade

Instrumentos que medem a umidade do solo, são muito uteis para determinar a quantidade de água necessária para a irrigação. É importante destacar que utilizando este método, a época de irrigar fica independente do estabelecimento das frequências de irrigação vistas no método anterior. Porém, deve-se acompanhar o crescimento do sistema radicular (z), e considerar a leitura feita no ponto médio dessa profundidade.

Usando deste método, a lâmina líquida de irrigação (LL) é calculada através da equação 1.

$$LL = ((CC - U_i) / 10) \cdot d \cdot z \quad (1)$$

Em que:

- a) LL = lâmina líquida (mm)
- b) CC = capacidade de campo
- c) U_i = Conteúdo de água no solo no limite para início da irrigação, capturado pelo sensor
- d) d = densidade do solo (em gr/cm³)
- e) z = profundidade do sistema radicular (em cm)

3. Desenvolvimento

Foi realizado uma pesquisa aplicada e descritiva, na área de circuitos digitais e desenvolvimento de software. A forma de coleta de dados para desenvolver o projeto foi baseada em pesquisa bibliográfica

3.1 Recursos necessários

Para descrever os recursos utilizados na construção do protótipo, será descrito a seguir separadamente entre software e hardware.

3.1.1 Hardware

Em posse do autor, foi utilizado um Notbook Acer com 8 GB de memória RAM, HD de 1000 GB e processador Intel i5-7200U 2.5GHz com sistema operacional Windows 10.

Além deste, foi necessário adquirir alguns hardwares com recursos do autor, como a Raspberry Pi 3B, que conta com um processador Broadcom BCM2837 quad-core de 1,2 GHz, memória RAM de 1GB. Juntamente com a placa, foi necessário adquirir alguns acessórios, como:

- a) um case plástico;
- b) fonte de energia 5V, 2.2 amperes;
- c) um cartão de memória micro SD classe 10 de 16 GB.
- d) fios jumpers de conexão macho-macho e macho-fêmea;
- e) protoboard;
- f) sensor de umidade do solo modelo A080;

- g) conversor de sinal analógico/digital ADC0832;
- h) bomba de água 12V, vazão de dois litros de água por minuto;
- i) módulo relé 3.3V.

3.1.2 Software

Foi usado como ambiente de desenvolvimento o Android Studio, com linguagem de programação JAVA. O Raspberry vem de fábrica sem configuração de sistema operacional, para isto, foi utilizado o Android Things, que foi desenvolvido para rodar em hardwares simples, com foco em desenvolvimento de IoT. Foi utilizado o software Win32DiskImager para gravar a imagem do Android Things no cartão de memória.

Para realizar a comunicação com a API Advisor, foi utilizado a biblioteca Retrofit, que foi desenvolvida para consumir web services no Android. Para o banco de dados, foi usado o serviço Real Time Database, do Firebase.

Por fim, foi utilizado a ferramenta Vysor, que é um serviço de espelhamento do Android, empregado na Raspberry Pi para obter a parte visual do sistema operacional diretamente no Notbook.

3.2 Análise de solo

Uma análise de solo é capaz de identificar as características químicas e físicas do solo, ou seja, pode ser utilizada para identificar os nutrientes presentes no solo ou a proporção de areia, silte e argila que compõe. No que diz respeito a irrigação do solo, o que se deve levar em consideração é a textura e a densidade, sendo que estes parâmetros podem ser determinados na análise física.

A amostra foi retirada numa propriedade rural no município de Praia Grande, Santa Catarina, na profundidade de 0 a 20 centímetros. Esta amostra foi secada na sombra por três dias, para retirar o excesso de umidade e após isto, foi empacotada e encaminhada para o laboratório de análises de solo Laborsolo. Após o retorno do laudo, foi possível determinar a textura do solo em argilo-arenosa e a sua densidade de 1.45 gr/cm³. De acordo com Falker (2010), um solo argilo-arenoso tem os valores de CC e PMP de 33% e 16% respectivamente.

3.3 Configuração da Raspberry Pi

A configuração da Raspberry consistiu em instalar o sistema operacional Android Things. Para isto, foi necessário criar uma conta no Android Things Console e criar um projeto, onde define-se que a plataforma a ser usada seria a Raspberry Pi. Com o projeto finalizado, o Android Things Console disponibiliza para download uma imagem do sistema operacional com as configurações necessárias para a instalação na placa.

Feito isso, necessitou-se do uso do Win32DiskImager para gravar a imagem do SO no cartão microSD, destaca-se a importância de o cartão estar formatado em FAT32, com mínimo de 8GB de memória e classe 10, configurações diferentes destas citadas podem não funcionar como esperado.

3.4 Banco de dados RealTime Database

A configuração deste serviço ao projeto no Android Studio consistiu em criar um arquivo inicial na plataforma Firebase e registrar o pacote do aplicativo neste projeto, onde foi necessário informar o certificado de assinatura de depuração “SHA-1”, que é encontrado no *Gradle* do projeto criado no Android Studio.

Após esta etapa, o Firebase cria um arquivo de configuração, que deve ser adicionado dentro do diretório do modo raiz do projeto. Feito isto, o aplicativo já está pronto para utilizar todos os recursos disponíveis do Firebase. Para utilizar o Real Time Database, deve ser adicionado no arquivo *build.gradle* de cada módulo, dentro do *dependencies* o comando “*implementation 'com.google.firebaseio:firebase-database:18.0.0'*”. Finalmente, a aplicação já está preparada para ler e salvar os dados no RealTime Database.

3.5 Desenvolvimento do Módulo Raspberry

Nesta etapa do projeto, foi desenvolvido todo o módulo da placa, que envolve a leitura da umidade do sensor e calibração, leitura da previsão do tempo para 3 dias, gravação dos dados no Real Time Database, conexão com a bomba de água e a aplicação que monitora e define a irrigação.

3.5.1 Instalação do sensor de umidade

Foi necessário utilizar um sensor resistivo de umidade do modelo A080, que é capaz de medir a umidade do solo por meio da resistência elétrica presente, ou seja, quanto mais água presente no solo, menor será o valor lido da resistência elétrica. Este sensor tem saída de valor digital e analógica, porém foi utilizado somente a leitura do sinal analógico, onde é possível obter o valor da resistência da água no solo.

O sensor possui alimentação de 3.3V ou 5V, pinos de saída analógica, digital, GND e fase. Como este sensor opera na mesma tensão dos pinos GPIO da Raspberry, pode-se ligar diretamente, não sendo necessário utilizar nenhum tipo de resistor.

Para realizar a leitura do valor da umidade utilizando-se do conversor de sinal, foi adaptado um driver do Gialluca (2017), sendo que o valor obtido na leitura é um primitivo do tipo “*int*” que contém o valor da resistividade da água presente no solo.

3.5.2 Calibração do sensor de umidade

Como visto no capítulo anterior deste trabalho, o valor obtido do sensor não corresponde ao valor real da umidade do solo e sim ao valor da resistência da água presente. Para resolver este problema, foi necessário utilizar um método de calibração, que consistiu em aferir a umidade do solo utilizando-se do protocolo da EMBRAPA de secagem da terra em micro-ondas e aplicar uma equação de regressão linear.

Para aferir o valor real da umidade, realizou-se a secagem de 180 gramas de terra por 10 minutos em potência máxima no forno micro-ondas, de acordo com o protocolo da EMBRAPA (2002), neste processo a terra perde toda a umidade retida.

Após esta etapa, foi separado uma amostra de 130 gramas de solo seco e realizado a leitura pelo sensor de umidade. Segundo o processo, foi adicionado porções de água no solo seco e aferido o valor pelo sensor, cada porção adicionada correspondia ao aumento de 10% na umidade real. Em cada adição de água, foi realizado o aferimento do sensor, assim foi possível obter pares de valores “umidade real x valor do sensor” (figura 01), sendo possível aplicar a equação de regressão linear.

água adicionada	umidade real (%)	valor do sensor
0	0	167
13 ml	10	156
26 ml	20	134
39 ml	30	77
52 ml	40	59

Figura 01. Relação entre umidade real e valor do sensor

Aplicando estes dados no gráfico, foi obtido o resultado como mostra a figura 02. Foi alcançado o valor do coeficiente de determinação R^2 de 93,9%, isso mostra o quanto os dados estão próximos da linha de regressão ajustada.

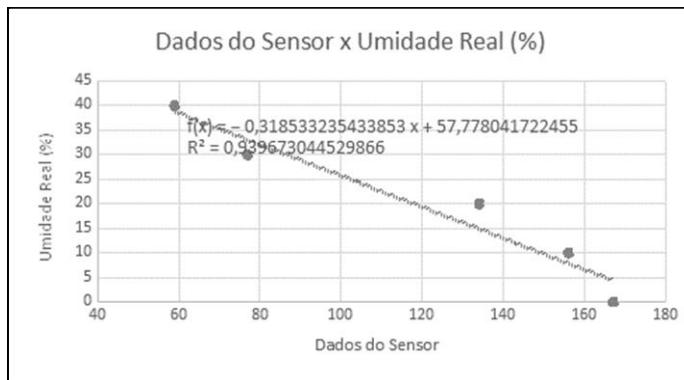


Figura 02. Gráfico da reta de regressão

Por fim, a equação que transforma o dado do sensor em umidade real ficou como mostra a equação 2:

$$\text{Umidade} = (-0,3185 * (\text{valor do sensor})) + 57,778 \quad (2)$$

3.5.3 Comunicação com a API Advisor

Primeiramente foi criado uma conta de usuário na API, e gerado um token que serve para identificar o usuário que realiza as requisições. Para realizar a leitura da previsão do tempo, é necessário informar o ID da cidade que se deseja, então foi utilizado o recurso de descobrir o ID pelo nome da cidade, executando o seguinte comando:

<http://apiadvisor.climatempo.com.br/api/v1/locale/city?name=Praia%20Grande&state=sc&token=643c73fff94c5b2fe221c5ede99cbeb4>

Como retorno obteve-se:

[{"id":4782,"name":"Praia Grande","state":"SC","country":"BR"}]

Com o ID da cidade o token do projeto é possível acessar a previsão para 72 horas, executando o comando:

<http://apiadvisor.climatempo.com.br/api/v1/forecast/locale/4782/hours/72?token=643c73fff94c5b2fe221c5ede99cbeb4>

Para consumir esta API no JAVA, foi utilizado a biblioteca Retrofit. Com ela foi possível ganhar simplicidade no desenvolvimento, sendo necessário implementar uma simples interface e escrever alguns métodos.

3.5.4 Conexão com a bomba de água

Para realizar a conexão com a bomba de água 12V, foi necessário utilizar um módulo relé, pois a tensão da placa Raspberry é de 3.3V, logo se fosse realizado a conexão diretamente a placa poderia sofrer algum tipo de dano irreversível.

Este módulo relé possui um canal 3.3V que permite controlar cargas de até 30V de corrente contínua ou até 250V de corrente alternada. Este é um módulo *Low Level*, ou seja, quando se é inserido um valor lógico zero ele é acionado. Ele possui 3 pinos de conexão com a Raspberry, sendo um pino para o GND, um pino para o VCC e um pino para o sinal. Possui também três portas de saída, sendo uma normalmente aberta, do inglês *normally open* (NO), uma normalmente fechada, do inglês *normally closed* (NC) e uma porta comum (COM).

3.6 Desenvolvimento do módulo móvel

Neste módulo foi desenvolvido toda a aplicação do lado do usuário, implementado as telas de cadastro de planta, cadastro de talhão, relatórios de dados do talhão e consumo de água e tela inicial (figura 3).

Cada tela construída corresponde a uma Activity diferente, isto serve para separar e organizar as funcionalidades diferentes de cada tela. Para criar esta classe, foi preciso estender a classe “Activity”, para que ela herde todas as características das Activities. Após isto, foi necessário sobrescrever o método “onCreate”, que é usado para configurar a interface do usuário.

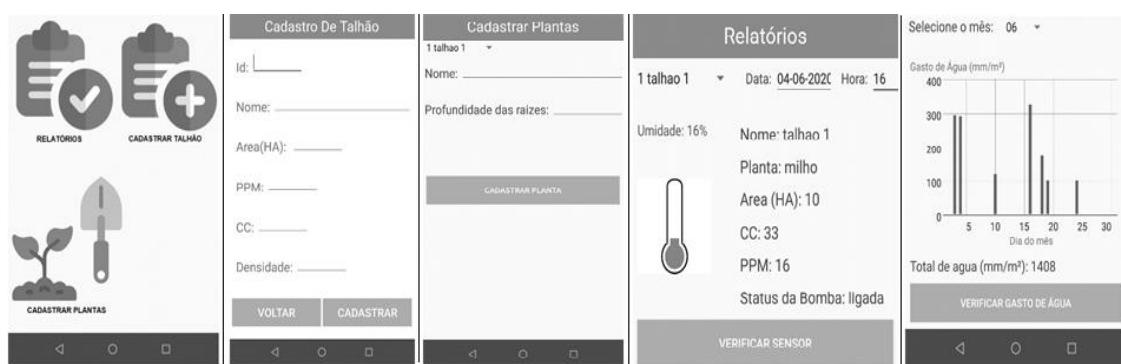


Figura 03. Telas do aplicativo móvel

3.7 Lógica para programar a irrigação

A última etapa do projeto consistiu na construção da lógica para realizar a irrigação automatizada. Esta parte envolve tudo o que foi desenvolvido até aqui, pois ela utiliza a umidade do solo, previsão do tempo, dados da análise de solo, leitura e gravação no banco de dados e a conexão com a bomba de água.

A lógica de programação desenvolvida utilizou a equação 1 para calcular a necessidade de água, de acordo com a bibliografia. Primeiramente foi testada as condições da umidade e previsão, para decidir ligar a bomba, como mostra-se a seguir:

a) a primeira condição foi testar se a umidade do solo está boa, comparando se ela está maior que 3% em relação ao PPM, que ficou definido em 16% de acordo com a análise de solo. Se isto for verdadeiro, a bomba fica desligada;

b) se a primeira condição for falsa, então é testado se a umidade do solo está menor ou igual que o PPM, ou seja 16%. Se isto for verdadeiro, então é acionado a bomba de água, independente da previsão do tempo;

c) se a condição acima for falsa, significa que a umidade do solo está entre 17% á 19%. Neste ponto é realizado a leitura da previsão do tempo para 3 dias, se a precipitação neste período for menor que 25% da necessidade de água, então é iniciado a irrigação, pois presume-se que se acontecer incidência de chuva, a mesma não será suficiente para irrigar o solo, consequentemente a umidade do solo atingirá o PPM;

d) se a condição acima for falsa, significa que a umidade do solo ainda está entre 17% á 19%, porém a previsão para os próximos 3 dias diz que pode ocorrer incidência de chuva. Se a precipitação for maior ou igual que 25% da necessidade de água, então a bomba fica desligada;

Quando alguma das condições forem verdadeiras, é realizado a chamada do método que salva as informações no banco, sendo salvo os dados da umidade do solo, quantidade de água gasta e status da bomba de água. A placa foi programada de modo que faça a leitura da umidade e salvar os dados a cada hora.

4. Resultados e discussão

De acordo com o levantamento bibliográfico realizado, uma irrigação por sensoriamento deve ou não ser iniciada de acordo com a umidade do solo, sendo que em nenhuma hipótese o solo deve atingir a umidade menor que a do PMP, caso isto ocorra, a planta não consegue retirar água do solo, acarretando em danos econômicos para a cultura.

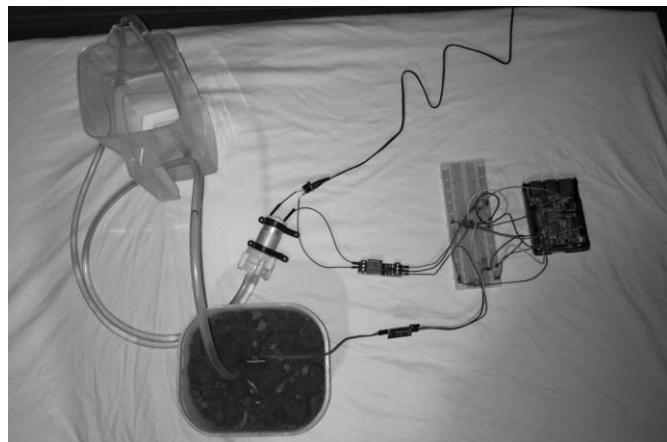
O modo como foi desenvolvido a lógica para programar a irrigação do solo, permite que a umidade nunca fique menor que a definida como PMP, considerando a previsão do tempo para tomar a decisão de irrigar.

Para realizar o cálculo da necessidade de água utilizou-se a equação 1 e os dados da análise de solo e umidade.

De acordo com estes resultados obtidos, é tomado a decisão de ligar a bomba de água. Foi desenvolvido então um método que calcula qual o tempo que a bomba deve ficar ligada por meio da equação 3, levando em consideração a quantidade de água, tamanho da área e vazão da bomba.

$$tempo(\text{seg}) = \frac{\text{quantidade de água(litros/m}^2\text{)} \cdot \text{área(m}^2\text{)}}{\text{vazão da bomba (litros por minuto)}} \cdot 60 \quad (3)$$

Os testes realizados para verificação do sistema de irrigação não foram no campo, pois a fragilidade dos componentes da placa não permite que ela fique exposta diretamente ao clima, sendo necessário criar uma estrutura de proteção. Porém foi considerado todas as condições possíveis de umidade de solo. A figura 4 mostra o projeto final, com todas as conexões realizadas, sendo utilizado uma porção de solo para os testes.



2 Figura 04. Projeto final

Como toda previsão de tempo não é 100% exata, há chances de o sistema verificar precipitação e ela não ocorrer, por isso é decidido na lógica de programação que realize a irrigação quando o solo atingir o nível de PPM, independente da previsão de chuva, garantindo assim que não ocorra danos econômicos na cultura implantada.

5. Conclusão

Diante dos levantamentos bibliográficos, foi realizado a implementação de um protótipo de irrigação automatizada, que utiliza dados do clima e sensoriamento do solo para decidir o melhor momento para iniciar uma irrigação.

Alguns pontos críticos foram observados durante o desenvolvimento deste projeto. O primeiro deles foi em relação a calibração do sensor de umidade utilizando uma equação de regressão linear, obtendo um R^2 de 93,9%, isto significa que na conversão do valor da resistividade do sensor para a umidade real do solo pode haver uma variação de até 6,1% nos dados. Uma forma de se melhorar este fator é realizar mais testes de calibração, até ajustar este valor acima de 95% como foi encontrado na bibliografia.

Outro ponto crítico é sobre a vida útil do sensor de umidade. Uma possível solução para este problema seria a troca dos sensores resistivos por sensores capacitivos.

Também foi observado a dificuldade de realizar os testes no campo, devido a fragilidade dos componentes físicos da placa. Uma possível solução para este problema seria criar uma estrutura de proteção, que permita que a placa fique exposta a condições climáticas por um grande tempo, podendo assim captar uma quantidade grande de dados.

Nos testes realizados in loco, foi observado que o projeto se comportou como o esperado. Realizando as irrigações quando foram necessárias, salvando os dados no RealTime Database. Assim, foi possível gerar relatórios de gasto de água e informações pertinentes aos talhões cadastrados, sendo possível acompanhar os dados conforme a data e hora que o usuário desejar.

Ao finalizar o projeto foi possível atingir os objetivos estabelecidos, criando uma tecnologia baseada em IoT, que pode ser aplicada em larga escala utilizando-se da mesma lógica que foi aplicada no desenvolvimento deste projeto.

6. Referências

ALBUQUERQUE, Paulo Emílio Pereira de. Estratégias de Manejo de Irrigação: Exemplos de Cálculo. **Circular Técnica**, Sete Lagoas, v. 136, n. 1, p.1-25, Set. 2010

EMBRAPA (Brasil). Determinação de matéria seca e umidade em solos e plantas com forno de microondas doméstico. **CIRCULAR TECNICA**, SÃO CARLOS, ed. 33, 2002. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/33322/1/CircularTecnica33.pdf>. Acesso em: 27 maio 2020

FALKER (Porto Alegre). Aplicações do HidroFarm. **Nota de Aplicação – Hfm 2010**, Porto Alegre, v. 1, n. 1, p.1-5, dez. 2010.

GIALLUCA, Giovanni di. **Android Things Drivers**. 2017. Disponível em: <https://github.com/giovannidg/androidthingsdrivers/tree/master/adc0832>. Acesso em: 20 maio 2020.

GOAP, Amarendra et al. An IoT based smart irrigation management system using Machine learning and open source technologies. **Computers And Electronics In Agriculture**, [s.l.], v. 155, p.41-49, dez. 2018. Elsevier BV. <http://dx.doi.org/10.1016/j.compag.2018.09.040>.

KAMIENSKI, Carlos et al. Smart Water Management Platform: IoT-Based Precision Irrigation for Agriculture. **Sensors**, [s.l.], v. 19, n. 2, p.276-284, 11 jan. 2019. MDPI AG. <http://dx.doi.org/10.3390/s19020276>.

KÖGLER, F.; SÖFFKER, D.. Water (stress) models and deficit irrigation: System-theoretical description and causality mapping. **Ecological Modelling**, [s.l.], v. 361, p.135-156, out. 2017. Elsevier BV. <http://dx.doi.org/10.1016/j.ecolmodel.2017.07.031>.

ONU (Estados Unidos). **POPULATION**. 2015. Disponível em: <<https://www.un.org/en/sections/issues-depth/population/index.html>>. Acesso em: 18 abr. 2019.

RAJALAKSHMI, P; MAHALAKSHMI, S. Devi. IOT based crop-field monitoring and irrigation automation. **2016 10th International Conference On Intelligent Systems And Control (isco)**, Coimbatore, p.1-6, jan. 2016. IEEE. <http://dx.doi.org/10.1109/isco.2016.7726900>.