



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS QUIXADÁ
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

KERLEY DE SOUSA DANTAS

**AUTOMAÇÃO DA IRRIGAÇÃO: UM SISTEMA DE IRRIGAÇÃO LOCALIZADA
BASEADO EM INTERNET DAS COISAS**

QUIXADÁ – CEARÁ

2016

KERLEY DE SOUSA DANTAS

AUTOMAÇÃO DA IRRIGAÇÃO: UM SISTEMA DE IRRIGAÇÃO LOCALIZADA
BASEADO EM INTERNET DAS COISAS

Monografia apresentada no curso de Sistemas de Informação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Sistemas de Informação. Área de concentração: Computação.

Orientador: Prof. Dr. Marcio Espíndola Freire Maia

QUIXADÁ – CEARÁ

2016

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

D213a Dantas, Kerley de Sousa.

Automação da irrigação: um sistema de irrigação localizada baseado em internet das coisas / Kerley de Sousa Dantas. – 2016.

38 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Quixadá, Curso de Sistemas de Informação, Quixadá, 2016.

Orientação: Prof. Dr. Marcio Espíndola Freire Maia.

1. Internet das coisas. 2. Irrigação localizada. 3. Automação. I. Título.

CDD 005

KERLEY DE SOUSA DANTAS

AUTOMAÇÃO DA IRRIGAÇÃO: UM SISTEMA DE IRRIGAÇÃO LOCALIZADA
BASEADO EM INTERNET DAS COISAS

Monografia apresentada no curso de Sistemas de Informação da Universidade Federal do Ceará, como requisito parcial à obtenção do título de bacharel em Sistemas de Informação.
Área de concentração: Computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Marcio Espíndola Freire Maia (Orientador)
Campus Quixadá
Universidade Federal do Ceará – UFC

Atslands Rego da Rocha
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará - UFC

Carlos Diego Andrade de Almeida
Campus Quixadá
Universidade Federal do Ceará - UFC

Aos meus pais, Conceição e Lucival, por todo o apoio que me deram durante toda a jornada até aqui. A minha irmã Aline, por todos os conselhos de fundamental importância para minha vida. A toda família Dantas Pinheiro

AGRADECIMENTOS

Ao Prof. Dr. Marcio Espíndola Freire Maia, por todos os seus conselhos e ao tempo dedicado a minha orientação.

Aos professores participantes da banca examinadora Atslands Rego da Rocha e Carlos Diego Andrade de Almeida pelo tempo, pelas valiosas colaborações e sugestões.

Aos meus amigos e colegas de turma que dividiram esse 4 anos de graduação comigo, Daniel Filho, Flávio Barros, Amanda Oliveira, Cainã Melo, Marques Amaro, Felipe Pinho, Alan Martins, Andreza Sousa, Lucas Henrique, Hinessa Caminha, Allan Vidal, João Vitor e Marciano Saraiva.

Aos meus amigos que dividiram o apartamento comigo, Paulo Rennê, Maxdyllon Rodrigues e Daniel Filho, por todas as brincadeiras geradas durante todos esses anos.

Ao meu amigo do NPI, Rayson Santos.

Aos meus amigos do PET-SI, Danrley Teixeira, Matheus Perreira, Yago Alves, Anderson Lemos e Alisson Gomes.

E a todos os meus amigos da UFC.

“A fé na vitória tem que ser inabalável”

(Dexter - Fênix)

RESUMO

Este trabalho propõe o desenvolvimento de um sistema para a irrigação localizada fornecendo comodidade aos pequenos agricultores, com o monitoramento da umidade do solo por meio de um sensor de umidade e aplicação de água por meio de uma válvula solenoide, tudo isso por um aplicativo móvel conectado a internet, utilizando dos conceitos de Internet das Coisas para o desenvolvimento da arquitetura do sistema e do próprio sistema.

Palavras-chave: Internet das Coisas. Irrigação Localizada. Automação.

ABSTRACT

This work proposes the development of a localized irrigation system, providing comfort to small farmers, monitoring soil moisture through a sensor and water application through a solenoid valve, everything for a mobile application connected to the Internet, using the concepts of Internet of Things for development of the system architecture and the system itself.

Keywords: Internet of Things. Localized Irrigation. Automation

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Arquitetura de referência | 17 |
| Figura 2 – Arquitetura do sistema | 23 |
| Figura 3 – Lista e cadastro de uma placa | 31 |
| Figura 4 – Configurações de uma placa | 32 |
| Figura 5 – Gráfico de umidade | 32 |
| Figura 6 – Gráfico de requisições | 33 |
| Figura 7 – Gráfico de umidade | 34 |

LISTA DE CÓDIGOS-FONTE

| | |
|--|----|
| Código-fonte 1 – Verificando a forma de operação da placa | 25 |
| Código-fonte 2 – Comparando os horários semanais cadastrados | 25 |
| Código-fonte 3 – Comparando o nível de umidade | 26 |
| Código-fonte 4 – Enviando mensagem ao servidor | 29 |

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------|-------------------------------------|
| CoAP | Constrained Application Protocol |
| HTTP | HyperText Transfer Protocol |
| IoT | Internet of Things |
| JSON | JavaScript Object Notation |
| MQTT | Message Queuing Telemetry Transport |
| NFC | Near Field Communication |
| NPF | Nó de Processamento Final |
| NS | Nó Sensor |
| REST | Representational State Transfer |
| WSAN | Wireless Sensor and Actor Networks |
| WSN | Wireless Sensor Network |

SUMÁRIO

| | | |
|-------|--|----|
| 1 | INTRODUÇÃO | 12 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 14 |
| 2.1 | Técnicas de Irrigação | 14 |
| 2.1.1 | <i>Técnica de irrigação por aspersão</i> | 14 |
| 2.1.2 | <i>Técnica de irrigação por superfície</i> | 14 |
| 2.1.3 | <i>Técnica de irrigação localizada</i> | 15 |
| 2.2 | Internet das Coisas | 15 |
| 2.2.1 | <i>Radio-Frequency IDentification</i> | 17 |
| 2.2.2 | <i>Near Field Communications</i> | 17 |
| 2.2.3 | <i>Wireless Sensor Networks</i> | 18 |
| 3 | OBJETIVOS | 19 |
| 3.1 | Objetivo Geral | 19 |
| 3.2 | Objetivos específicos | 19 |
| 4 | TRABALHOS RELACIONADOS | 20 |
| 5 | PROPOSTA | 22 |
| 5.1 | Arquitetura do Sistema | 22 |
| 5.1.1 | <i>Módulo Aplicação móvel</i> | 23 |
| 5.1.2 | <i>Módulo Web Service</i> | 24 |
| 5.1.3 | <i>Módulo plataforma de prototipagem</i> | 27 |
| 5.2 | Interfaces | 28 |
| 5.2.1 | <i>Protocolo e Dados</i> | 28 |
| 6 | APRESENTAÇÃO E ANÁLISE DOS RESULTADOS | 30 |
| 6.1 | Uso do sistema | 30 |
| 6.2 | Avaliação do Sistema | 33 |
| 6.2.1 | <i>Teste do envio de mensagens</i> | 33 |
| 6.2.2 | <i>Teste da umidade</i> | 34 |
| 6.2.3 | <i>Teste de horários semanais</i> | 35 |
| 7 | CONSIDERAÇÕES FINAIS | 36 |
| | REFERÊNCIAS | 37 |

1 INTRODUÇÃO

O problema da seca que afeta a região nordeste do Brasil é causado principalmente pela escassez de chuvas e as altas temperaturas que influenciam drasticamente na disponibilidade dos recursos hídricos na região (PENA, 2016). Recurso este bastante utilizado pela irrigação agrícola, a qual é responsável por aproximadamente 75% do consumo da água no Brasil (ANA, 2013).

A eficiência da irrigação é uma abordagem que trata de um manejo que aplique água apenas no momento certo e na quantidade necessária para a cultura. Mundialmente, apenas 37% da água utilizada na irrigação é aproveitada pela cultura, considerando a quantidade de água retirada da fonte pela realmente usada pela irrigação (COELHO; FILHO; OLIVEIRA, 2005). Grande parte dos agricultores determina a quantidade e quando deve-se aplicar água na cultura de forma empírica. Se a aplicação da água fosse utilizada de forma coerente, cerca de 20% da água aplicada seria economizada, evitando o uso desnecessário (LIMA; FERREIRA; CHRISTOFIDIS, 1999).

As técnicas de irrigação são utilizadas por agricultores como solução para contornar a escassez da água e como meio de determinar a quantidade de água necessária para a cultura, assim chegando a um aumento na produtividade. A técnica de irrigação pode ser dividida em algumas categorias como por aspersão, localização ou superficial (TESTEZLAF, 2011).

As técnicas de irrigação utilizam-se de alguns componentes (aspersores, válvulas e motobombas) no ambiente, causando uma dificuldade e perda de tempo no monitoramento do solo pelo agricultor da cultura, como também o seu deslocamento até a região da cultura para o acionamento de bombas ou abertura de válvulas. Este projeto busca solucionar essa dificuldade por meio da implementação do conceito de Internet das Coisas (IoT), que é a interação entre diversos sensores e atuadores por meio de um ponto central, por exemplo um servidor Web ou uma aplicação móvel, que tome decisões sobre o que fazer na cultura. Como resultado final da interação entre os sensores e atuadores, será de uma automatização ou facilidade nos processos do nosso cotidiano, oferecendo uma ideia de um processo inteligente aos agricultores.

Para a Internet das Coisas, objetos inteligentes colaboram com objetos do mundo físico de forma transparente e autônoma, a qual fornecerá serviços e aplicações, possibilitando a interação entre usuários e máquinas. Uma tecnologia que implementa a IoT é a Radio-Frequency Identification (RFID), que permite a comunicação entre os objetos presentes em uma rede e são identificadas por meio de um sistema único de numeração (LEITE et al., 2014).

Considerando o problema de utilização de muitos componentes, o monitoramento destes e o deslocamento do agricultor até a área de cultivo, o projeto propôs desenvolver uma plataforma de prototipagem para coletar regularmente dados da umidade do solo e acionar a abertura ou o fechamento de uma válvula solenoide, as informações coletadas serão enviadas para um servidor Web utilizando uma comunicação sem fio por dados móveis. O servidor ficará encarregado de responder à plataforma de prototipagem se é pra realizar a abertura ou o fechamento da válvula.

Este projeto também desenvolveu um aplicativo móvel para o monitoramento do estado da irrigação e para alteração das configurações do processo de irrigação. O aplicativo móvel está conectado a um servidor Web, proporcionando uma melhor comodidade do agricultor. O servidor Web será responsável por guardar informações do agricultor. O trabalho buscou um melhor custo/benefício do produto para facilitar o seu acesso aos pequenos agricultores, principal público alvo deste projeto.

Este projeto também tem como foco principal automatizar o técnica de irrigação por localização, desde que ela seja utilizado em um terreno plano, pelo fato de ter uma distribuição uniforme de água na cultura, necessitando assim um único sensor de umidade do solo para determinar a porcentagem de umidade de toda a área de plantio e uma única válvula para que seja liberada a água na cultura (PRADO; NUNES; TINOS, 2014).

2 FUNDAMENTAÇÃO TEÓRICA

A seguir serão abordados conceitos necessários para fundamentar o desenvolvimento e análises deste trabalho.

2.1 Técnicas de Irrigação

A irrigação é uma técnica utilizada na agricultura para suprir as necessidade de água em uma região de plantio. Essa técnica utiliza de diversos equipamentos, acessórios e técnicas de manejo. Existem três tipos principais de irrigação, superficial, localizada e aspersão. Algumas característica definem qual tipo de irrigação é adequada para o plantio, como por exemplo o tipo do solo, o clima e entre outros (AGROSMART, 2016).

2.1.1 Técnica de irrigação por aspersão

A técnica por aspersão tenta imitar o efeito da chuva distribuindo a água em formas de gotas na área por meio de aspersores espalhados na cultura. Esta técnica é bastante utilizado por grandes áreas de plantio, por causa da grande área coberta pelos aspersores (TESTEZLAF, 2011).

A técnica por aspersão é dividida em convencionais ou mecanizados. Os convencionais utilizam de motobomba, tubulações e aspersores espalhados pela cultura, assim cada aspersor cobre uma área fixa ao mesmo tempo. Para o mecanizado é construído uma estrutura no qual as aspersores são montados. Essa estrutura percorre toda a cultura irrigando uniformemente (TESTEZLAF, 2011).

2.1.2 Técnica de irrigação por superfície

A técnica de irrigação por superfície consiste na aplicação direta da água no solo, desde que este esteja nivelado. Com ajuda da gravidade a água se movimenta uniformemente pela área de cultivo, abrangendo toda a área, e se infiltra no solo (AGROSMART, 2016). Esta técnica é bastante utilizada na produção de arroz.

A técnica de irrigação por superfície pode ser classificada em dois modos: 1. Por sulcos ou 2. Por inundação. A técnica de irrigação por sulcos consiste na aplicação da água em pequenos canais nos quais percorrem todas as linhas da cultura e escoando por sulcos. Para o

de inundação, a água é aplicada em toda a área e se acumula até se formar um “espelho d’água” (TESTEZLAF, 2011).

2.1.3 Técnica de irrigação localizada

A técnica de irrigação localizada tem como característica a utilização de pouca água e é bastante utilizada nos dias atuais, geralmente ela é utilizada em pequenas áreas planas, para não alterar na pressão de cada gotejador, e em determinados tipos de culturas. Basicamente essa técnica consiste na aplicação de água na raiz ou tronco de cada planta, formando uma pequena área irrigada ao redor da planta (TESTEZLAF, 2011).

A técnica pode ser dividida em duas categorias, a primeira por microaspersão e a segunda por gotejamento. A técnica por gotejamento resume-se em tubulações que levam a água até a raiz de cada planta. Com ajuda de gotejadores, a água é aplicada na raiz da planta constantemente ou em intervalos de tempos. A técnica por microaspersão tem como característica um microaspersor para cada planta, no qual é encarregado em distribuir a água ao redor da planta por meio de gotas de água jogadas ao ar (TESTEZLAF, 2011).

Segundo Prado, Nunes e Tinos (2014) a técnica de irrigação localizada possui uma uniformidade de distribuição de água para os tubos autocompesantes de 84,9% à 93,1%, e para os não autocompesantes de 90,8% à 96,0%, indicando que possuem uma distribuição excelente.

2.2 Internet das Coisas

A Internet das Coisas (IoT) é uma revolução tecnológica responsável por criar um ambiente inteligente, que pode ou não interagir com pessoas. Seu principal objetivo é automatizar ou facilitar processos do nosso cotidiano - como verificar o que há em uma geladeira, fazer uma lista de itens faltantes, acessar o site do supermercado e fazer as compras para uma pessoa.

Para Sundmaecker et al. (2010), Internet das Coisas pode ser definida como uma infra-estrutura de rede de auto-configuração capaz de atuar com outros dispositivos por meio de protocolos padrão para a troca de informações. No contexto de Internet das Coisas, a palavra “Coisa” pode ser entendida como um objeto físico/real ou como também uma entidade virtual. Estes objetos utilizam de uma interface para a comunicação com outros objetos que estão integrados na rede. Eles são identificados por meio de um número ou nome atribuído.

Os dispositivos ou máquinas identificados na rede, devem coletar informações do

meio físico ou virtual, trocar essas informações com outras máquinas ou dispositivos e processar tanto as informações coletadas como as obtidas por meio da rede, para poder influenciar um desencadeamento de ações do dispositivo identificado (SUNDMAEKER et al., 2010). Ações que acabam tornando o processo automatizado ou facilitando a atividade do processo. Por causa dessas características de coletar, transmitir e processar informações automaticamente, subentende-se que o ambiente é inteligente o suficiente para adaptar-se a diferentes eventos.

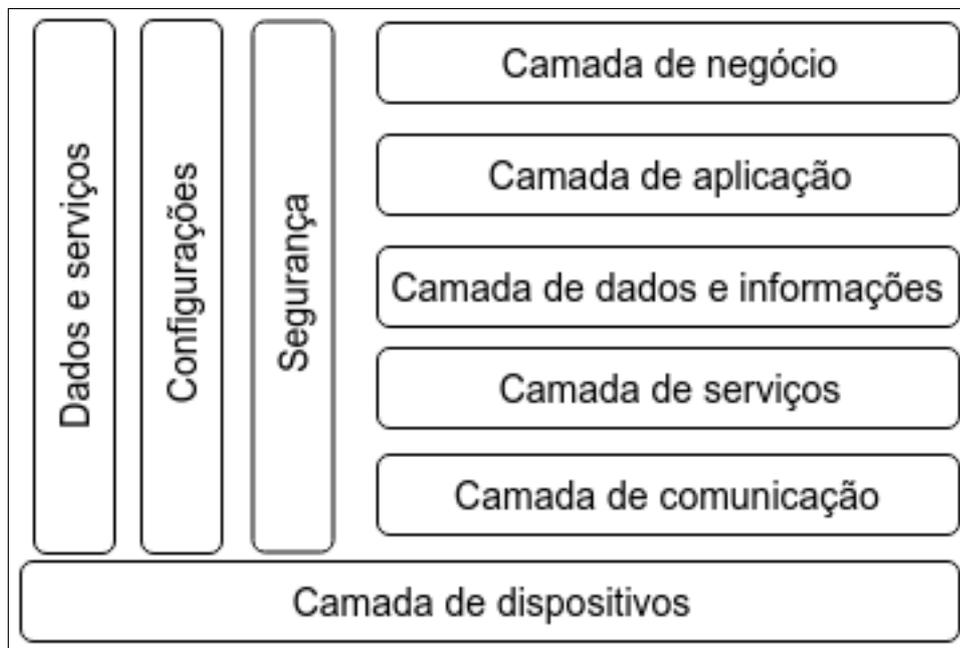
De acordo com Agrawal e Vieira (2013), as tecnologias *Radio-Frequency Identification* (RFID), *Near Field Communications* (NFC) e *Wireless Sensor Networks* (WSN), abordadas nas seções 2.2.1, 2.2.2 e 2.2.3 respectivamente, são tecnologias que implementam o conceito de IoT, sendo que RFID é a principal tecnologia que implementa IoT.

Holler et al. (2014) propuseram uma arquitetura de referência para IoT, Figura 1, em que descrevem os serviços fornecidos por um sistema, as relações entre esses elementos e os princípios das relações. Não necessariamente as camadas se comunicam com as suas adjacentes, elas podem se comunicar com qualquer outra.

A camada de dispositivos representa os microcontroladores ou as plataformas de prototipagem junto com seus componentes, que podem ser sensores ou atuadores. A camada de comunicação é responsável por interligar a camada de dispositivos com a camada de serviços, existem três principais tipos de comunicações que são o MQTT, CoAP e o HTTP. Para a realização deste trabalho foi utilizado a comunicação HTTP.

Na camada de serviços encontram-se os servidores Web que junto com a camada de dados e informações são especificados como os cliente se comunicam com o servidor. As aplicações de IoT encontram-se na camada de aplicações que junto com a camada de negócios integra os requisitos dos *stakeholders* com IoT (HOLLER et al., 2014).

Figura 1 – Arquitetura de referência



Fonte: Adaptado de Holler et al. (2014)

2.2.1 *Radio-Frequency IDentification*

Radio-Frequency IDentification (RFID) é uma tecnologia com o objetivo de etiquetar objetos ou pessoas para facilitar sua identificação por meio de dispositivos de computação (JUELS, 2006).

Para Peris-Lopez et al. (2006), o RFID é dividido em três componentes principais: um transpônder, que é um microchip de identificador único que possui uma antena de comunicação, implantado em objetos ou pessoas; um banco de dados, que possui todos os identificadores dos transpônder como suas informações, localização ou nome; além de leitores, que permitem a comunicação com os transpônderes e tem acesso ao banco de dados e geralmente responsáveis pelo processamento interno e a lógica.

2.2.2 *Near Field Communications*

Near Field Communications (NFC) é uma tecnologia de interação entre dispositivos próximos no ambiente, com uma distância de aproximadamente 10 centímetros, sem a necessidade de o usuário fazer alguma ação para estabelecer uma conexão, trocar dados ou fazer uma transação. Ao passo que os dispositivos se distanciam um do outro e ultrapassa o limite de comunicação, é terminada a conexão entre os dispositivos. (COSKUN; OZDENIZCI; OK,

2015).

O NFC possui três modos de operações para a comunicação entre os dispositivos (RAHUL et al., 2015): leitura/escrita, possibilitando que cada dispositivo envie ou receba dados, sendo o menos seguro por não necessitar de identificação; emulação de cartão, onde o dispositivo simula um cartão de crédito pessoal, esse modo é totalmente seguro; Peer to Peer (P2P), semelhante ao leitura/escrita porém necessita de uma identificação entre os dispositivos.

2.2.3 *Wireless Sensor Networks*

Para Abbasi et al. (2014), sensor é uma tecnologia capaz de colher dados do mundo físico e transforma-los em informação para um monitorador. Alguns exemplos de sensores são: pressão, temperatura, umidade, velocidade do vento e outros. (*Wireless Sensor Network* (WSN) é um conjunto de vários sensores chamados de nó sensores. Esses nós são capazes de colher dados do mundo físico e processa-los para então enviá-los para uma aplicação remota apenas o necessário, tornando-os assim dispositivos inteligentes (AKYILDIZ et al., 2002).

A arquitetura da WSN é dividido em três camadas: nó sensores, são nós que coletam ou executam alguma ação no ambiente; nó coletor, tem como função agrupar todas as informações geradas pelos nó sensores e envia-las a um nó de processamento; nó de processamento, são nós que tem a maior taxa de processamento na rede e são encarregados de fazer todo o processamento das informações enviadas pelo nó agrupador (ABBASI et al., 2014).

Yick, Mukherjee e Ghosal (2008) dizem existir 5 tipos de WSNs: terrestre, subaquática, subterrânea, móvel e multi-mídia. A WSN terrestre são sensores implantados sobre o solo em uma determinada área, diferente da subterrâneo, que são sensores implantados dentro do solo. Para os nós sensores subaquáticas, há uma dificuldade maior de implantação devido a interferência da água entre a comunicação do nó sensor e o nó de processamento. Os móveis são sensores que conseguem se locomover sobre o ambiente físico, permitindo a coleta de dados em diferentes pontos da área. Por fim a WSN multi-mídia é uma rede de sensores que captam dados através de áudio, vídeo ou imagem, o principal problema da multi-mídia e a alta demanda da largura da banda.

Existe também a *Wireless Sensor and Actuator Network* (WSAN) é uma variação da WSN, a única diferença é a presença de nós atuadores localizadas no mesma camada dos nós sensores. Com a presença dos nós atuadores, a WSAN passa a ter controle sobre o mundo físico, como o controle de válvulas, bombas e outras coisas (ABBASI et al., 2014).

3 OBJETIVOS

3.1 Objetivo Geral

O objetivo deste trabalho foi desenvolver um sistema de monitoramento e controle de uma técnica de irrigação agrícola. O controle da aplicação da água na cultura foi determinado a partir de três meios: quando a umidade do solo atingir uma determinada marca especificada pelo agricultor, configurada previamente pelo agricultor para ser aplicada em horários semanais ou quando o sistema não opera nenhum dos modos citados anteriormente. O agricultor poderá alternar entre estes meios de aplicação.

3.2 Objetivos específicos

- Desenvolver uma plataforma de prototipagem capaz de controlar a aplicação de água na cultura.
- Desenvolver uma plataforma de prototipagem capaz de coletar dados sobre a umidade do solo.
- Desenvolver uma plataforma de prototipagem capaz de enviar e receber informações de um servidor Web.
- Desenvolver um servidor Web capaz de receber e enviar informações à uma plataforma de prototipagem e à um aplicativo móvel.
- Desenvolver um aplicativo móvel capaz receber e enviar informações à um servidor Web.
- Desenvolver um aplicativo móvel que seja capaz de configurar como sistema de irrigação irá funcionar.
- Desenvolver um aplicativo móvel que seja capaz de gerar um gráfico representando umidade do solo durante o dia.

4 TRABALHOS RELACIONADOS

Em Xiao, Xiao e Luo (2010), foi proposta uma solução para uma irrigação inteligente com base na coleta de dados sobre a umidade do solo em diferentes pontos da cultura, utilizando uma conexão sem fio para a comunicação entre os sensores e a base central de dados. A base central tomava todas as decisões sobre onde seria aplicada a irrigação e a sua duração, e posteriormente a base envia instruções para o sistema de controle da irrigação que é responsável pela abertura dos registros eletrônicos. Xiao, Xiao e Luo (2010) concluem que a irrigação inteligente produziu um consumo de 62,22% de água utilizada na plantação.

O trabalho apresentado por Xiao, Xiao e Luo (2010) é deficiente no aspecto da energia disponível no campo: ele utiliza energia disponível na rede elétrica que muitas vezes não é acessível a partir da zona rural. Outro ponto inadequado é não prover a interação com os agricultores, que podem ter a necessidade de interação com o sistema para uma tomada de decisão específica, como no caso a necessidade de irrigação para um replantio. Este trabalho busca solucionar os pontos falhos apresentados anteriormente.

Já em Harun et al. (2015), foi desenvolvida uma arquitetura WSN para a troca de informações entre os dispositivos por meio do XBee. Essa arquitetura é capaz de coletar informações do estado da estufa, como o potencial hidrogeniônico, temperatura e umidade do solo e do ar. Ela as envia a um sistema central na internet para a exibição em forma gráfica dos dados para o usuário final. O sistema central é responsável pelo monitoramento e pela tomada de decisão. Quando os dados obtidos chegam a um nível mínimo, o sistema central aciona as bombas e válvulas gerando a irrigação na estufa.

O trabalho apresentado por Harun et al. (2015) utilizam de tecnologias que podem ser desnecessárias para o uso em uma cultura de campo aberto, causando um alto custo na obtenção do projeto, como o sensor de umidade do ar. Outra característica do trabalho de Harun et al. (2015) é a interação com o usuário, porém, o usuário não pode tomar decisões para o acionamento da irrigação, uma característica importante que este trabalho apresentará. Um outro ponto deficiente é a obtenção de energia pela rede elétrica, um fator que pode ser indisponível na zona rural.

Outro trabalho que vale ser citado é o de Vicente, Rocha e Neves (2010) que propõem o GolfSense: uma aplicação de monitoramento do ambiente para utilizar como dado de entrada para o manejo da irrigação em um campo de golf. O trabalho utilizou WSN para a comunicação entre os Nós Sensores (NS) e um Nó de Processamento Final (NPF). Cada NS é composto

por um sensor de umidade do solo, um sensor de bateria e um MicaZ, que é responsável por comunicar-se com o NPF, e possui um controle sobre os sprinklers. O NPF é responsável por processar os dados recebidos do Nó Sensor, se o processamento resultar na necessidade de irrigação o NPF envia uma resposta de início e tempo de duração da irrigação para o NS, caso contrário, ele responde com uma configuração padrão.

Em Vicente, Rocha e Neves (2010), a interação com o usuário é feita a partir do NPF, porém o usuário só visualiza o estado do ambiente do campo de golfe. Este trabalho visa contemplar a possível necessidade de aplicação de água pelo agricultor.

5 PROPOSTA

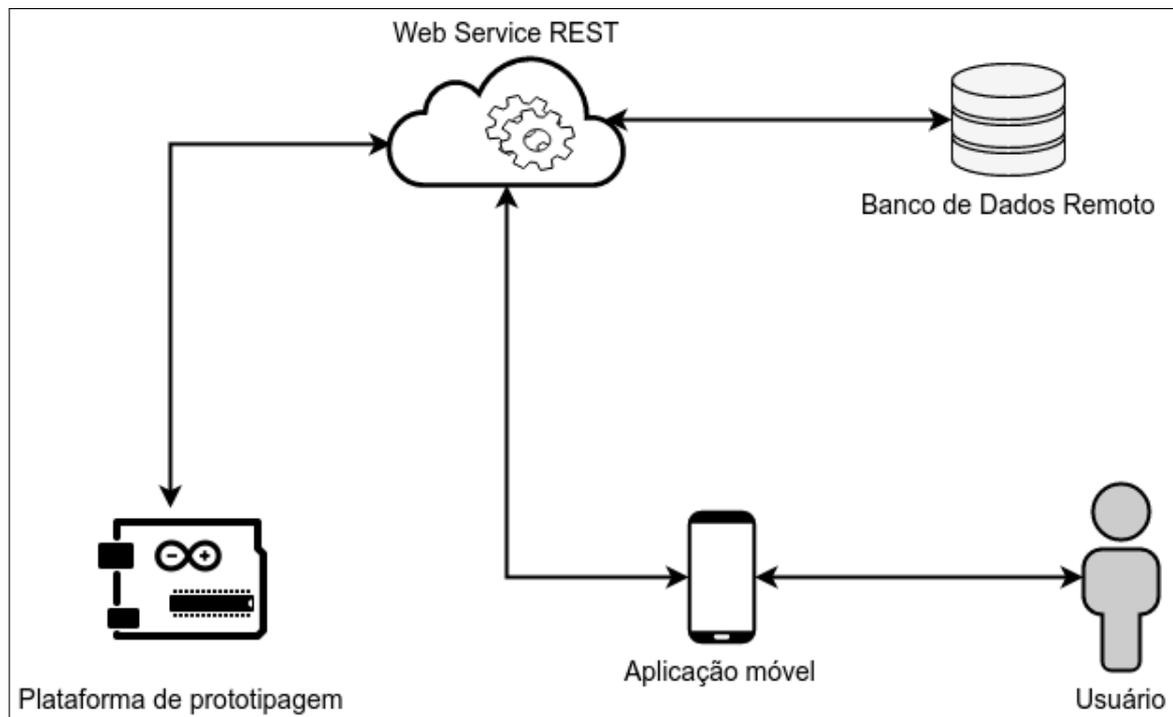
Este trabalho propõe um sistema para a automação da irrigação agrícola, no qual foi utilizado conceitos e tecnologias citados no **capítulo 4** para conseguir atingir os objetivos citados no **capítulo 3**

A técnica de irrigação utilizada neste trabalho foi a localizada, que tem como característica a distribuição uniforme da aplicação de água e a pequena área que a água é aplicada, permitindo que o sistema possa ter um único ponto de coleta de umidade para que o dado sirva para as demais plantas. Outro ponto importante é que o solo deve ser do mesmo tipo para toda a área de cultivo, para evitar que a evaporação ou infiltração sejam diferentes em locais da cultura, prejudicando assim a coleta de umidade do solo. Definido isso, o sistema poderá ser configurado para que a aplicação de água possa ser feito com base na coleta de umidade do solo.

5.1 Arquitetura do Sistema

O sistema proposto pode ser representado em três partes principais: Web Service, aplicação móvel e plataforma de prototipagem. Na Figura 2 são representados os módulos do sistema e as interações entre eles. A aplicação móvel é responsável por interagir com o usuário, no caso o agricultor, onde o usuário poderá configurar o sistema e adicionar novas placas de prototipagens. O módulo Web Service é responsável por armazenar e fornecer informações à plataforma de prototipagem e à aplicação móvel. As informações fornecidas são armazenadas em um banco de dados remoto. A plataforma de prototipagem é responsável por interagir com os objetos físicos, e tem a capacidade de obter informações sobre o solo e atuar sobre o controle da saída de água por meio de componentes.

Figura 2 – Arquitetura do sistema



Fonte: Elaborada pelo autor

A seguir serão descritos detalhadamente cada parte do sistema, como suas tecnologias utilizadas.

5.1.1 Módulo Aplicação móvel

O módulo aplicação móvel é responsável por coletar as informações fornecidas pelo agricultor e enviá-las ao servidor Web. O módulo também é responsável por exibir as informações sobre a umidade de cada placa cadastrada pelo agricultor. Cada placa cadastrada possui um gráfico em linha representando o nível de umidade, colhida por essa placa, ao longo do dia. Cada agricultor terá um cadastro para poder ter acesso ao sistema. No sistema o agricultor poderá cadastrar uma ou mais placas e configurá-la a maneira a qual o sistema de irrigação irá funcionar, se é por horário semanais, por umidade ou por nenhum dos dois.

O módulo foi desenvolvido utilizando o *framework* Ionic¹, que é um *framework* de desenvolvimento de aplicações móveis híbridas, ou seja, o mesmo código-fonte utilizado para o desenvolvimento servirá tanto para sistema operacional Android quanto para o iOS. A linguagem utilizada pelo Ionic para o desenvolvimento é JavaScript para *back-end* e HTML e

¹ <https://ionicframework.com/>

CSS para o *front-end*. O Ionic foi escolhido principalmente por ser um *framework* híbrido, não sendo necessário desenvolver duas aplicações para dois sistemas operacionais. Outro fator para a escolha deste *framework* é que seus objetos são no formato JSON, que são os mesmos recebidos pelo servidor Web, sendo assim não há a necessidade de converter os objetos, facilitando ainda mais o desenvolvimento da aplicação. Na seção 6.1 é possível visualizar as telas do aplicativo móvel.

5.1.2 Módulo Web Service

O módulo Web Service é responsável pelo gerenciamento das informações e também o armazenamento desta informações. Na arquitetura de referência, Figura 1, o módulo se encontra na camada de serviços. Para o desenvolvimento do Web Service foi utilizado a linguagem JavaScript, por meio do *framework* NodeJS². O NodeJS utiliza da arquitetura REST (*REpresentational State Transfer*) junto com protocolo de comunicação HTTP para a transferência de informações. O protocolo HTTP possui os métodos *Post*, *Get*, *Put* e *Delete*, para especificar os tipos de requisições, respectivamente os métodos são utilizados para inserir, visualizar, atualizar e excluir. Os motivos pelo qual o NodeJS foi escolhido por causa da facilidade e rapidez no desenvolvimento de um servidor, que tem como base uma arquitetura REST que não está atrelado a um tipo de aplicação e sim a uma comunicação. A arquitetura REST facilita no desenvolvimento de novas aplicações, já que para a comunicação entre esta não será necessário mudar ou criar novos serviços no NodeJS. O servidor Web foi hospedado na plataforma de serviços Heroku³, por causa da necessidade que o servidor Web esteja na Internet para que o módulo de prototipagem possa se comunicar com o servidor.

Para o armazenamento das informações foi utilizado um banco de dados remoto, o MongoDB⁴. O MongoDB é baseado em um banco de dados orientado a documentos, similares ao JSON(JavaScript Object Notation). Ele foi escolhido por causa de suas características, como esquema dinâmico, respostas no formato JSON e por ser bastante compatível com servidores desenvolvidos pelo NodeJS que utilizam a arquitetura REST.

O Web Service ficou encarregado de tratar as mensagens recebidas pela plataforma de prototipagem e retornar uma resposta contendo a indicação se deve ou não abrir a válvula. Para que o servidor responda a plataforma, ele precisará guardar as informações recebidas e

² <http://nodeframework.com/>

³ <https://www.heroku.com/home>

⁴ <https://www.mongodb.com/>

acessar as informações da placa para obter o tipo de operação que ela está funcionando. Com as informações obtidas, o servidor verificará qual tipo de operação será executada para dar a resposta a placa de prototipagem. O trecho de código a seguir é referente ao processo descrito acima.

```

1 Board.getBoardById(board._id, function(err, board){
2   if(err){
3     throw err;
4   }
5
6   var response;
7   if (board.operation == 1){
8     response = weekly(board);
9   }else if (board.operation == 2){
10    response = moisture(board, collect);
11  }else{
12    response = "0"
13  }
14 });

```

Código-fonte 1 – Verificando a forma de operação da placa

Para cada tipo de operação foi atribuído um valor numérico para representá-lo. O valor 0 representa que o servidor não fará nenhuma verificação e sempre responderá a plataforma com o valor 0. O valor 1 significa que a placa foi configurada para operar dentro de horários semanais, ou seja, o agricultor deseja estabelecer intervalo fixo de horário em um ou mais dias da semana. O valor 2 define que a placa está funcionando no estilo de um valor mínimo de umidade do solo, sendo assim, o servidor verificará a coleta realizada naquele instante e fará a comparação com o valor mínimo estabelecido pelo agricultor. Os códigos-fontes a seguir são referentes as chamadas de função das linhas 8 e 10 do Código-fonte 1 respectivamente.

```

1 function weekly(board){
2   var dayNow = new Date().getDay()
3   var day
4   for (var i = board.days.length - 1; i >= 0; i--) {
5     var temp = board.days[i]
6     if (dayNow == temp.day) {
7       day = temp

```

```

8         break
9     };
10
11 };
12
13 if(!day){
14     return "0";
15 }
16 console.log(day)
17 var hours = day.hours;
18
19 var temp = new Date()
20 var now = new Date(0,0,0,temp.getHours(), temp.getMinutes(),0)
21 for (var i = hours.length - 1; i >= 0; i--) {
22     var hour = hours[i]
23
24     var arrStart = hour.start.split(":")
25     var arrEnd = hour.end.split(":")
26
27     var start = new Date(0,0,0,arrStart[0], arrStart[1],0)
28     var end = new Date(0,0,0,arrEnd[0], arrEnd[1],0)
29
30     var diffStart = start.getTime() - now.getTime()
31     var diffEnd = end.getTime() - now.getTime()
32
33     if (diffStart < 0 && diffEnd > 0) {
34         return "1"
35     }
36 };
37 }
38 return "0"

```

Código-fonte 2 – Comparando os horários semanais cadastrados

```

1 function moisture(board, collect){
2     if (board.minimum > collect.soilMoisture){
3         return "1"
4     }else{

```

```

5     return "0"
6     }
7 }

```

Código-fonte 3 – Comparando o nível de umidade

O Código-fonte 2 utiliza do dia da semana (0 para Domingo, 1 para Segunda, 2 para Terça, 3 para Quarta, 4 para Quinta, 5 para Sexta e 6 para Sábado) atual para comparar com os dias cadastrados para a placa, caso não seja encontrado nenhum dia cadastrado igual ao dia atual, o valor de retorno será 0, caso seja encontrado, a hora atual será comparada com as horas cadastradas para aquele dia. Se a hora atual estiver entre o horário de início e de término especificados pelo agricultor, o valor de resposta será 1, caso contrário sera 0.

O Código-fonte 3 analisa o valor de umidade coletado naquele momento com o mínimo estabelecido pelo agricultor. Para o caso da umidade atual está inferior ao mínimo estabelecido, o valor de retorno será 1, do contrário será 0.

5.1.3 Módulo plataforma de prototipagem

A plataforma de prototipagem escolhida para o desenvolvimento do módulo foi o Arduino, por causa do seu baixo custo e que suas funções atenderiam ao que o projeto é proposto. Outro fator importante para a escolha do Arduino é que a comunidade é bastante popular, fazendo com que tenha mais suporte.

O módulo plataforma de prototipagem é encarregado de coletar a umidade do solo por meio um sensor de umidade e enviá-la a um servidor Web. Com a resposta do envio, a plataforma processará a informação e verificará se é preciso abrir a válvula solenoide, caso ela esteja fechada, continuar aberta, fechar a válvula solenoide, caso esteja aberta, ou continuar fechada.

Um *shield* GSM/GPRS sim900 é utilizado pelo Arduino para poder ter acesso a rede de dados móveis. Isso permite que o Arduino possa estar em um local isolado, desde que tenha acesso a rede de dados móveis, portanto o Arduino conseguirá se comunicar com o servidor mesmo que a área de cultivo esteja distante da cidade. O *shield* foi conectado ao Arduino através das portas 2 e 3.

5.2 Interfaces

Como abordado na seção 5.1.2, a comunicação entre o servidor Web e o Arduino é feita por meio de requisições HTTP. É necessário que ao o término de um cadastro de uma placa, o ID gerado automaticamente para referenciar a placa pelo MondoDB e exibido na aplicação móvel, após o cadastro, seja inserido em uma constante presente no código do Arduino, para que o servidor Web possa identificar qual placa está fazendo requisições à ele.

5.2.1 Protocolo e Dados

Foi estabelecido o protocolo de mensagens HTTP como a comunicação entre a aplicação móvel e o servidor Web. O formato dos dados utilizado na comunicação, foi o JSON por causa da sua cobertura oferecida pela linguagem JavaScript, que foi utilizada na aplicação móvel como também no servidor Web.

A lógica principal para a comunicação entre o Arduino e o Servidor Web, é que o Arduino faça uma requisição ao servidor Web com o método GET, fornecido pelo protocolo HTTP, passando como parâmetros pela URL o ID da placa e a umidade do solo respectivamente. Foi utilizada a biblioteca InetGMS para tratar todo o processo de comunicação entre o Arduino, o *shield* sim900 e o chip telefônico, nela é possível utilizar alguns métodos do protocolo HTTP, no caso foi utilizado o método GET que recebe como parâmetros o Domínio do servidor Web, a porta, o caminho, um vetor de caracteres para a resposta e o tamanho do vetor de caracteres. O servidor, por sua vez recebe os dados da requisição e persiste a coleta na base de dados remota. Com a coleta persistida, o MongoDB lhe atribui um ID que é inserido em um vetor de ID's de coletas da placa.

Como resposta a requisição, o servidor processará a informação da placa para saber que tipo de operação a placa está funcionando, como foi explicado na seção 5.1.2. Após o termino do processamento as possíveis resposta do servidor serão 0 para o caso de fechamento da válvula ou 1 para o de abertura da válvula.

O método GET, utilizado pelo Arduino, retorna o tamanho da resposta do servidor, incluso cabeçalho e dados, e preenche o vetor de caracteres passado como parâmetro com a resposta. Como foi definido anteriormente que o servidor retornará apenas 1 caractere indicando a abertura ou o fechamento da válvula, o Arduino poderá acessar diretamente a resposta no vetor, que está na ultima posição, por que as primeiras posições são referente ao cabeçalho da resposta

e as ultimas são referentes aos dados. O Código-fonte a seguir representa o processo descrito anteriormente.

```
1 char envia_GSM(String texto)
2 {
3
4     if (inet.attachGPRS("timbrasil.br", "tim", "tim")){
5         Serial.println(F("status=Conectado..."));
6     }
7     else{
8         Serial.println(F("status=Nao conectado !!"));
9         started = false;
10        return "0"
11    }
12    delay(100);
13    char msg[250];
14    char temp_string[55];
15    int numdata = -1;
16
17    String valor = "/collect/"+String(id_board)+"&"+texto;
18    valor.toCharArray(temp_string, 55);
19
20    numdata = inet.httpGET("autirri.herokuapp.com", 80, temp_string
21        , msg, 250);
22    return msg[numdata - 1];
23 }
```

Código-fonte 4 – Enviando mensagem ao servidor

6 APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Nesta seção é apresentado os resultados do desenvolvimento do trabalho proposto para automatizar a irrigação localizada, além de problemas encontrados durante todo o processo de implementação do trabalho.

Apesar da irrigação localizada e o Arduino existirem a muito tempo, foram encontradas algumas dificuldades durante o processo de desenvolvimento. Como novas tecnologias vem surgindo para o conceito de IoT, foi proposto implantar algumas delas neste trabalho, para simplificar ou melhorar o desempenho do mesmo, como é o caso do protocolo mensagens MQTT. Porém, foram encontradas dificuldades em utilizar a biblioteca do *shield* sim900 junto com a biblioteca do MQTT, por causa da incompatibilidade entre o recursos requisitados e oferecidos pelas mesmas. O biblioteca do MQTT requisita de um cliente a qual não é oferecido pela a biblioteca do *shield* sim900, inviabilizando o uso do MQTT no módulo plataforma de prototipagem. Para contornar este problema foi utilizado a biblioteca InetGSM, que oferece recursos de comunicação por meio de requisições HTTP.

Outra dificuldade encontrada durante a implementação da comunicação do Arduino com o servidor Web foi na utilização de requisições POST pelo arduino. Como a requisição POST oferecido pela biblioteca envia os dados no formato Form-Data, o servidor Web não conseguia identificar os dados por que ele estava configurado para receber requisições com os dados no formato JSON. Para solucionar o problema citado poderíamos configurar o servidor para receber dados no formato Form-Data, porém isso iria fazer com que as informações passadas pela aplicação móvel não fosse recebidas pelo servidor. Então foi decidido que o Arduino utilizasse do método GET para passar as informações e que essa requisição fosse tratado no serviço do servidor, já que era mais custoso mudar a configuração do servidor por causa da maior requisição serviços entre a aplicação móvel e o servidor. Essa alteração de configuração iria acarretar a transformação de cada mensagem, tanto no lado do servidor como no lado da aplicação, para que possam serem enviadas e recebidas as mensagens.

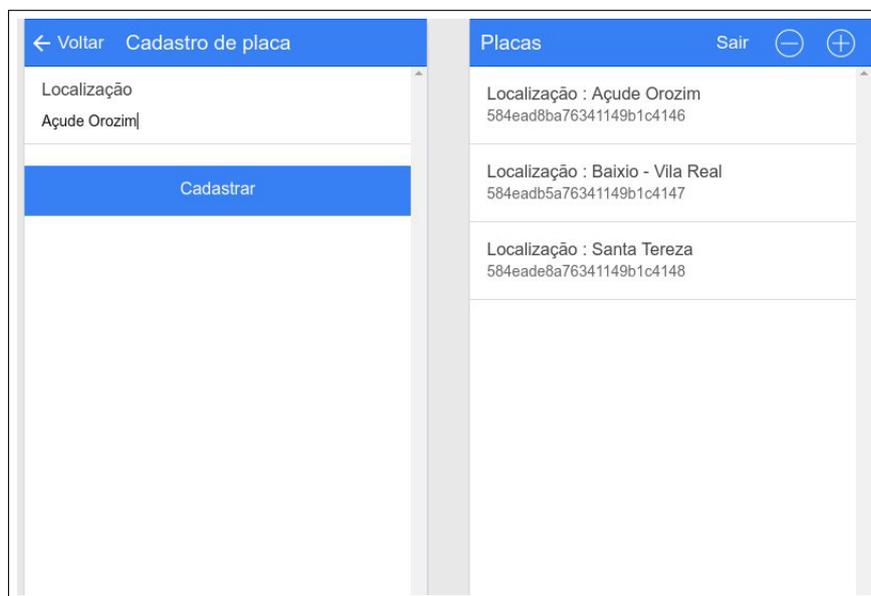
6.1 Uso do sistema

As dificuldades apresentadas anteriormente fizeram com que o projeto evoluísse ao longo do tempo em busca de soluções que resolvessem os problemas, como também proporcionou um conhecimento de novas tecnologias, como foi o caso do *framework* Ionic para

o desenvolvimento da aplicação móvel. Apesar de todas as dificuldades encontradas, o projeto conseguiu atingir seus objetivos específicos abordados no capítulo 3.

Todos os módulos do projeto atenderam as suas respectivas responsabilidades, A figura 3 representam a tela de cadastro de uma placa e visualização de todas as placas por um agricultor. É possível notar que na visualização das placas é mostrado o local da placa cadastrado pelo agricultor e logo abaixo o ID referente a placa. Com esse ID, faz-se necessário colocar no código do Arduino para que possa ser identificado pelo servidor Web. Para o caso do Arduino ser transferido de local, é apenas necessário alteração da localização dele na aplicação móvel, sem a necessidade de atribuir novamente um ID.

Figura 3 – Lista e cadastro de uma placa



Fonte: Elaborada pelo autor

A figura 4 representa as configurações de uma placa. Na segunda tela, da figura, é possível observar de qual modo a placa está operando e escolher outros modos de operação. Caso a operação escolhida seja por umidade, será requisitado qual o valor mínimo em porcentagem para que seja necessário a aplicação da água, já para o semanal é possível cadastrar os horários semanais para a operação. É importante ressaltar que a coleta de umidade pela plataforma e o envio ao servidor independe do modo de operação, com isso sempre está sendo feito a coleta e armazenado no servidor Web.

Figura 4 – Configurações de uma placa

The image shows three sequential mobile application screens for configuring a plate. The first screen, titled 'Configurações', has a blue header with a back arrow and the text 'Configurações'. It contains several settings: 'Estilo de Funcionamento' with a dropdown menu, 'Por Umidade' with a checked checkbox, 'Por Dias' with a dropdown menu, 'Nenhum' with a dropdown menu, 'Umidade atual:' with a dropdown menu, 'Umidade (%)' with a text input field containing '70', and a blue 'Cadastrar' button at the bottom. The second screen, titled 'Cadastro de dia', has a blue header with a back arrow and the text 'Cadastro de dia'. It shows a 'Dia' dropdown menu set to 'Terça' and a blue 'Cadastrar' button. The third screen, titled 'Cadastro de Hora', has a blue header with a back arrow and the text 'Cadastro de Hora'. It shows 'Hora de Início' with a text input field containing '14:00', 'Hora de Término' with a text input field containing '15:00', and a blue 'Cadastrar' button.

Fonte: Elaborada pelo autor

O agricultor pode acompanhar a mudança de umidade do solo ao decorrer do dia, como é mostrado na figura 5, através de um gráfico em linha. Também é listado todas as coletas realizadas ao longo do dia.

Figura 5 – Gráfico de umidade



Fonte: Elaborada pelo autor

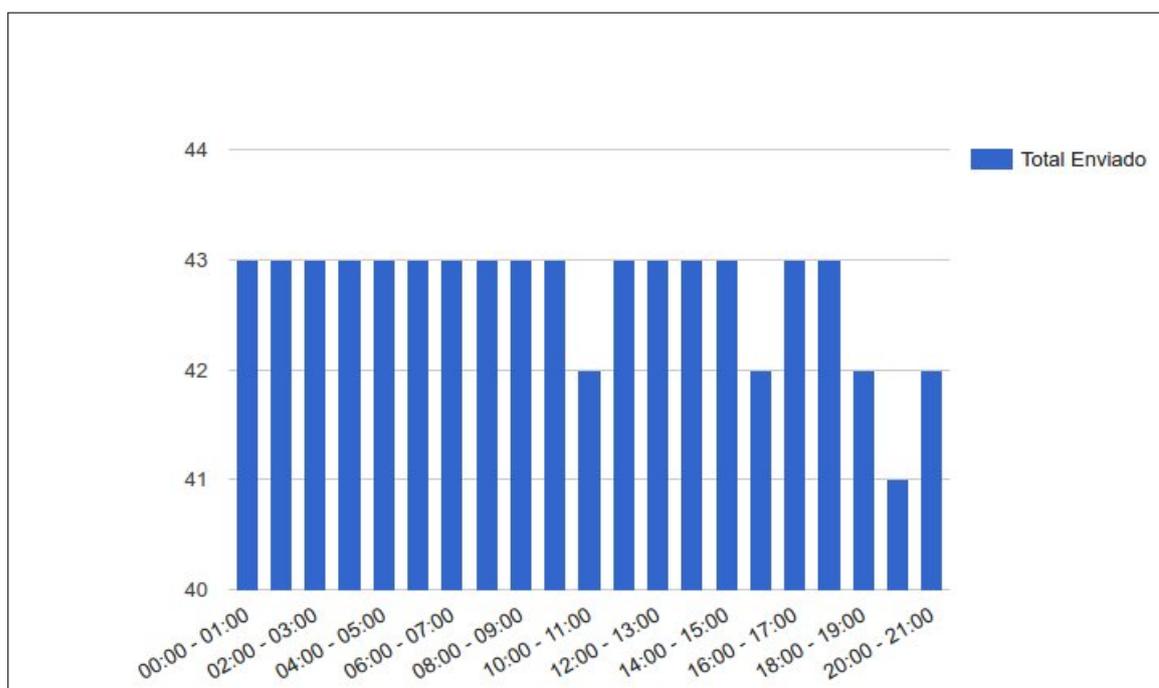
6.2 Avaliação do Sistema

Para avaliar o sistema foram utilizados três testes. O primeiro teste realizado, foi de envio de mensagens entre o Arduino e o Servidor Web ao longo do dia. O segundo teste realizado, foi de manter a umidade em um nível mínimo estabelecido pelo agricultor. E por fim, o terceiro teste realizado foi a análise sobre a abertura da válvula em horários semanais.

6.2.1 Teste do envio de mensagens

Durante um período de 21 horas, o Arduino ficou realizando requisições ao servidor no intervalo de um minuto. Antes do envio, a biblioteca utilizada executa alguns testes para poder verificar se o Arduino ainda está conectado com a rede de dados móveis, estes testes demoram entorno de quarenta segundos, que somado ao tempo estabelecido anteriormente, dá um total de uma requisição a cada um minuto e quarenta segundos. Ao todo foram realizadas 897 requisições do dia 10/12/2016 à 11/12/2016. O gráfico a seguir representa a distribuição de requisições em horas.

Figura 6 – Gráfico de requisições



Fonte: Elaborada pelo autor

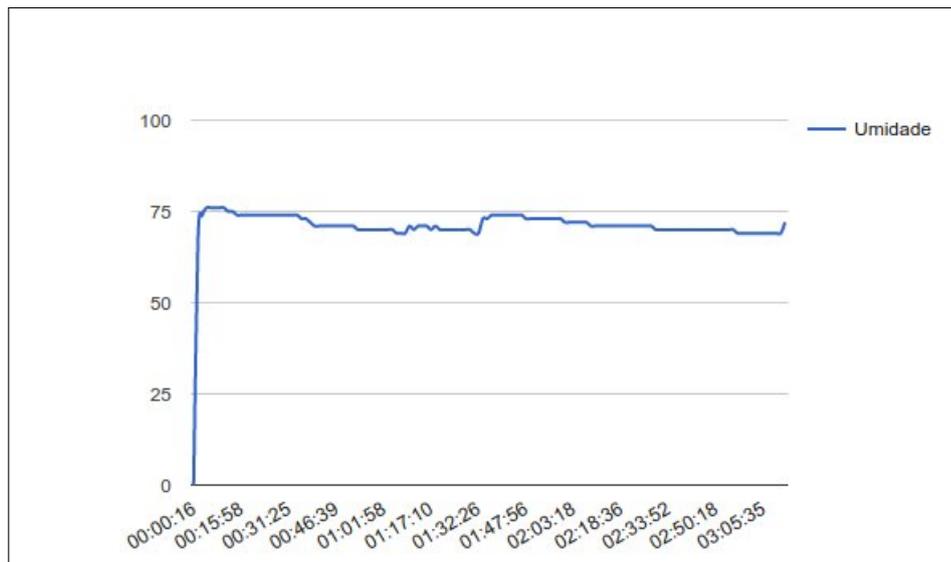
Dividindo 60 minutos, referente a uma hora, por 1 minuto e 40 segundos, referente a requisição, temos 43 requisições por hora. Logo, podemos observar no gráfico 6 que entre

às 10-11, 15-16, 18-19 e 20-21 umas das requisições não foi enviada e entre às 19-20 duas requisições não foram enviadas, fazendo com que 6 de 897 requisições não fossem enviadas no intervalo de 21 horas. Portanto, o módulo Arduino consegue manter uma comunicação satisfatória com o servidor Web.

6.2.2 *Teste da umidade*

O teste de umidade consistiu no funcionamento da placa em modo de umidade mínima. Quando a umidade atual for menor que a umidade mínima especificada, espera-se que a placa abra a válvula para que possa ser feita a aplicação da água. O valor mínimo de umidade utilizado foi de 70%. O gráfico 7 mostra o acompanhamento da umidade durante o período de três horas.

Figura 7 – Gráfico de umidade



Fonte: Elaborada pelo autor

Como podemos observar, durante as 3 horas a umidade ficou entre 69% e 76%. Como o valor de umidade ficava abaixo do mínimo estipulado, a placa respondeu com a abertura da válvula e fechando quando a próxima requisição fosse 0. Podemos concluir que a placa agiu corretamente na aplicação da água no solo, fazendo com que a umidade ficasse acima da estipulada.

Outro ponto que podemos destacar é o número de coletas repetidas que podem acabar atrapalhando a análise da umidade ao longo do dia. Isso poderia ser solucionado como o aumento do tempo de requisição, porém existe outro problema que será abordado na seção 6.2.3.

6.2.3 *Teste de horários semanais*

Para o teste, foram cadastrados na aplicação móvel um horário para o domingo de 19:30 às 19:45 e um horário para a segunda de 06:20 às 06:35. No horário estabelecido foi observado o comportamento do Arduino em relação ao horário atual para a abertura ou fechamento da válvula. Para o fechamento na segunda e a abertura no domingo da válvula, foi ocorrido no momento especificado, porém, para o fechamento no domingo e a abertura na segunda da válvula, ocorreu um atraso de pouco mais de 1 minuto.

O atraso ocorrido na abertura da válvula pode ser explicado pelo fato de as requisições estavam sendo feitas a cada 1 minutos mais o tempo dos testes da biblioteca, portanto, se o Arduino requisitasse o servidor no momento do horário de fechamento ou abertura da válvula, ele esperaria 1 minuto e 40 segundos até obter a resposta do servidor, causando o atraso na abertura ou fechamento da válvula.

O problema abordado anteriormente poderá ser solucionado diminuindo o tempo de requisição, fazendo com que o Arduino esteja em constante comunicação com servidor Web. Porém a solução deste problema entrará em conflito com a solução do problema abordado na seção 6.2.2, que é o aumento do tempo de requisição para uma coleta.

7 CONSIDERAÇÕES FINAIS

Durante o desenvolvimento do projeto foram encontrados alguns problemas que impossibilitaram de utilizar tecnologias bastantes utilizada pela IoT, porém estes problemas foram contornados para que o trabalho fosse um sucesso. Durante a avaliação do projeto foi encontrados alguns problemas que não afeta ao objetivo proposto do trabalho, a solução para este problema foi colocado para trabalhos futuros.

Concluimos que este trabalho apresenta os objetivos específicos, como: uma plataforma de prototipagem capaz de comunicar e receber dados de um servidor; um servidor Web capaz de enviar e receber dados de uma plataforma de prototipagem e uma aplicação móvel; uma aplicação móvel capaz de enviar e receber dados de um servidor Web; uma aplicação móvel capaz de configurar uma plataforma de prototipagem e capaz de gerar um gráfico da umidade ao longo do dia; uma plataforma de prototipagem capaz de controlar a aplicação de água na cultura como também a coleta da umidade do solo.

Foram identificadas algumas melhorias para o projeto durante sua avaliação. Segue algumas melhorias que poderão ser adicionadas ao projeto em trabalhos futuros:

- Avisos ao agricultor quando uma placa parar de enviar requisições ao servidor.
- Adicionar um novo componente ao Arduino para que ele possa também ligar uma bomba D'Água e não ficar restringido somente a uma válvula.
- Solucionar o problema de diminuir a requisição para que possa abrir a válvula no momento certo e aumentar a requisição para que não haja muitas coletas repetidas.
- Colocar o protótipo em um ambiente real para possam ser coletar dados do funcionamento do projeto e que sejam analisados com o objetivo de validar o protótipo.

Todos os códigos desenvolvidos neste projeto estão disponível em: <<https://github.com/kerleysol>>

REFERÊNCIAS

- ABBASI, A. Z.; ISLAM, N.; SHAIKH, Z. A. et al. A review of wireless sensors and networks' applications in agriculture. **Computer Standards & Interfaces**, Elsevier, v. 36, n. 2, p. 263–270, 2014.
- AGRAWAL, S.; VIEIRA, D. A survey on internet of things-[doi 10.5752/p. 2316-9451.2013v1n2p78](https://doi.org/10.5752/p.2316-9451.2013v1n2p78). **Abakós**, v. 1, n. 2, p. 78–95, 2013.
- AGROSMART. **Vantagens e desvantagens dos principais tipos de irrigação**. 2016. Disponível em: <<https://www.agrosmart.com.br/blog/vantagens-tipos-de-irrigacao/>>. Acesso em: 21 de nov de 2016.
- AKYILDIZ, I. F.; SU, W.; SANKARASUBRAMANIAM, Y.; CAYIRCI, E. Wireless sensor networks: a survey. **Computer networks**, Elsevier, v. 38, n. 4, p. 393–422, 2002.
- ANA. **Relatório de Conjuntura dos Recursos Hídricos no Brasil – Informe 2013**. 2013. Disponível em: <http://arquivos.ana.gov.br/institucional/spr/conjuntura/webSite_relatorioConjuntura/projeto/index.html>. Acesso em: 10 de abr de 2016.
- COELHO, E. F.; FILHO, M. A. C.; OLIVEIRA, S. d. Agricultura irrigada: eficiência de irrigação e de uso de água. **Bahia Agrícola**, v. 7, n. 1, p. 57–60, 2005.
- COSKUN, V.; OZDENIZCI, B.; OK, K. The survey on near field communication. **Sensors**, Multidisciplinary Digital Publishing Institute, v. 15, n. 6, p. 13348–13405, 2015.
- HARUN, A. N.; KASSIM, M. R. M.; MAT, I.; RAMLI, S. S. Precision irrigation using wireless sensor network. In: IEEE. **Smart Sensors and Application (ICSSA), 2015 International Conference on**. Kuala Lumpur, 2015. p. 71–75.
- HOLLER, J.; TSIATSIS, V.; MULLIGAN, C.; AVESAND, S.; KARNOUSKOS, S.; BOYLE, D. **From Machine-to-machine to the Internet of Things: Introduction to a New Age of Intelligence**. Waltham: Academic Press, 2014.
- JUELS, A. Rfid security and privacy: A research survey. **Selected Areas in Communications, IEEE Journal on**, IEEE, v. 24, n. 2, p. 381–394, 2006.
- LEITE, M. A. de A.; MASSRUHÁ, S. M. F. S.; EVANGELISTA, S. R. M.; SOUZA, K. X. S. de. **Tecnologias emergentes-futuro e evolução tecnológica das AgroTIC**. Brasília: Embrapa Informática Agropecuária, 2014. 331–349 p.
- LIMA, J.; FERREIRA, R. S. A.; CHRISTOFIDIS, D. O uso da irrigação no brasil. **O estado das águas no Brasil. Agência Nacional de Energia Elétrica.**, 1999.
- PENA, R. F. A. **Por que no nordeste há seca?** 2016. Disponível em: <<http://brasilecola.uol.com.br/brasil/por-que-nordeste-seco.htm>>. Acesso em: 10 de abr de 2016.
- PERIS-LOPEZ, P.; HERNANDEZ-CASTRO, J. C.; ESTEVEZ-TAPIADOR, J. M.; RIBAGORDA, A. Rfid systems: A survey on security threats and proposed solutions. In: SPRINGER. **Personal wireless communications**. New York, 2006. p. 159–170.

PRADO, G. do; NUNES, L. H.; TINOS, A. C. Avaliação técnica de dois tipos de emissores empregados na irrigação localizada. **Revista Brasileira de Agricultura Irrigada**, Instituto de Pesquisa e Inovação na Agricultura Irrigada-INOVAGRI, v. 8, n. 1, p. 12, 2014.

RAHUL, A.; KRISHNAN, G. G.; KRISHNAN, U. H.; RAO, S. The survey on near field communication. **International Journal on Cybernetics & Informatics (IJCI)**, Amrita Center for Wireless Networks & Applications, v. 4, n. 2, 2015.

SUNDMAEKER, H.; GUILLEMIN, P.; FRIESS, P.; WOELFFLÉ, S. **Vision and challenges for realising the Internet of Things**. Brussels: EUR-OP, 2010. v. 20.

TESTEZLAF, R. Irrigação: Métodos, sistemas e aplicações. Faculdade de Engenharia Agrícola Unicamp-FEAGRI, 2011.

VICENTE, J.; ROCHA, R.; NEVES, R. Golsense: A golf course wsn monitoring application. In: IEEE. **Computer Engineering and Systems (ICCES), 2010 International Conference on**. Cairo, 2010. p. 152–157.

XIAO, K.; XIAO, D.; LUO, X. Smart water-saving irrigation system in precision agriculture based on wireless sensor network. **Transactions of the Chinese Society of Agricultural Engineering**, Editorial Office of Transactions of the Chinese Society of Agricultural Engineering, v. 26, n. 11, p. 170–175, 2010.

YICK, J.; MUKHERJEE, B.; GHOSAL, D. Wireless sensor network survey. **Computer networks**, Elsevier, v. 52, n. 12, p. 2292–2330, 2008.