



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

LETÍCIA DE BRITO VICCIOLI

**AGRICULTURA 4.0: NOVAS TECNOLOGIAS APLICADAS AO
AGRONEGÓCIO**

Assis/SP

2020

LETÍCIA DE BRITO VICCIOLI

**AGRICULTURA 4.0: NOVAS TECNOLOGIAS APLICADAS AO
AGRONEGÓCIO**

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis - IMESA e da Fundação Educacional do Município de Assis – FEMA, como requisito do Curso de Bacharelado em Ciência da Computação.

Orientanda: Letícia de Brito Viccioli

Orientador: Prof. MSc. Guilherme de Cleve Farto

Assis/SP

2020

FICHA CATALOGRÁFICA

V631a VICCIOLI, Letícia de Brito

Agricultura 4.0: Novas tecnologias aplicadas ao agronegócio /
Letícia de Brito Viccioli. – Assis, 2020.

64p.

Trabalho de Conclusão de Curso (Ciência da Computação). –
Fundação Educacional do Município de Assis – FEMA

Orientador: Me. Guilherme de Cleva Farto

1. Monitoramento agrícola 2. Agrícola 4.0 3. Arduino

CDD: 005.133

AGRICULTURA 4.0: NOVAS TECNOLOGIAS APLICADAS AO AGRONEGÓCIO

LETÍCIA DE BRITO VICCIOLI

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: Prof. MSc. Guilherme de Cleve Farto
Examinador: Prof. Esp. Célio Desiró

**Assis/SP
2020**

DEDICATÓRIA

Dedico este trabalho primeiramente à Deus, aos meus pais, por terem me dado todo apoio para que eu nunca desistisse, e principalmente para a minha avó, que me ajudou a realizar esse sonho, sei que onde estiver, estará feliz por mim.

AGRADECIMENTOS

Primeiramente agradeço a Deus, por ter me dado saúde e inteligência para ter chego até aqui.

Agradeço à minha mãe Suzimar de Brito Viccioli, meu pai Leonardo Viccioli e meu irmão Eduardo Viccioli, que sempre estão ao meu lado, em qualquer situação.

A minha avó, Maria Aparecida Guadahim, que mesmo não estando mais presente, me ajudou muito nessa etapa da minha vida.

A todos meus amigos, parentes e familiares, que me ajudaram de forma direta ou indireta, me motivando e nunca me deixando desistir.

Ao Professor MSc. Guilherme de Cleve Farto, por me orientar e colaborar para a conclusão dessa fase tão importante.

A todos os professores da instituição, que juntos formam uma excelente equipe e estão sempre dispostos a compartilhar conhecimento.

E por fim agradeço a todos que colaboraram de forma direta ou indiretamente para a execução deste trabalho.

*“O homem não teria alcançado o possível se, repetidas
vezes, não tivesse tentado o impossível.”*

Max Weber (1864 – 1920)

RESUMO

A agricultura é a principal fonte de recursos para a vida humana. Desde que o homem abandonou sua vida de nômade e fixou sua moradia em apenas um lugar, passou a produzir seu alimento por meio do cultivo da terra. A princípio, o cultivo era manual. Com o surgimento da tecnologia, o trabalhador foi agregando ao manejo de ferramentas para facilitar o dia a dia. Surge então a Agricultura 4.0, que reúne quatro aspectos principais, são eles: Gestão baseada em dados, produção a partir de novas ferramentas e técnicas, sustentabilidade e profissionalização. A proposta deste trabalho foi a de desenvolver um aplicativo utilizando a plataforma Arduino e Android, fazendo uma integração das duas tecnologias para construção de um protótipo de monitoramento agrícola, capaz de captar informações para o usuário, além de apontar possíveis problemas no local.

Palavras-chave: Monitoramento agrícola; Agricultura 4.0; Arduino.

ABSTRACT

Agriculture is the main source of resources for human life. Ever since man abandoned his nomadic life and fixed his dwelling in only one place, began to produce his food by cultivating the land. At first, cultivation was manual. With the emergence of technology, the worker was adding to the management tools to facilitate day to day. Then emerges Agriculture 4.0, which brings together four main aspects, they are: Data-based management, production from new tools and techniques, sustainability and professionalization. The proposal of this work was to develop an application using the Arduino and Android platform, making an integration of the two technologies to build a prototype of agricultural monitoring, able to capture information for the user, besides pointing out possible problems in the place.

Keywords: Agricultural monitoring; Agriculture 4.0; Arduino.

LISTA DE ILUSTRAÇÕES

Figura 1 - Agricultura 4.0 em conjunto com Internet of Things	22
Figura 2 – Comparação de número de brasileiros para dispositivos conectados	23
Figura 3 – Estatísticas dos dispositivos desejados pelo brasileiro.....	24
Figura 4 - Diagrama de Blocos	26
Figura 5 - Arduino Uno.....	27
Figura 6 - Arduino Duemilanove	28
Figura 7 - Arduino Mega 2560	28
Figura 8 - Arduino Nano.....	29
Figura 9 - Arduino UNO	30
Figura 10 - IDE do Arduino	31
Figura 11 - Divisão da IDE do Arduino.....	32
Figura 12 - Toolbar do Arduino	32
Figura 13 - Servidor Node.js x servidor tradicional	37
Figura 14 - Exemplo React Native	39
Figura 15 - Ilustração das formas de energia em um sensor.....	41
Figura 16 - Ilustração da variação de uma grandeza física de um sensor analógico.	41
Figura 17 - Ilustração da leitura da posição de um objeto por um encoder incremental. ...	42
Figura 18 - Arduino Uno.....	44
Figura 19 - Sensor de umidade do solo	46
Figura 20 - Módulo sensor de chuva	47
Figura 21 - Sensor de gás e fumaça.....	48
Figura 22 - Sensor de Temperatura.....	48
Figura 23 - Conexões Arduino e Sensores	49
Figura 24 - Código do arduino	51
Figura 25 - Instanciação da SerialPort.....	52
Figura 26 - Abrir porta do servidor	52
Figura 27 - Convertendo objetos em JSON	53
Figura 28 - Console.log.....	53
Figura 29 - Socket.id.....	54
Figura 30 - Socket.emit.....	54
Figura 31 - Endereço do servidor	54
Figura 32 - Sinal de gás detectado.	55

Figura 33 – Configurações de alerta.....	56
Figura 34 - Alerta de irrigação	57
Figura 35 – Animações da aplicação.....	57
Figura 36 - If ternário das animações	58
Figura 37 - Demonstração de chuva na aplicação.....	58
Figura 38 - Gráfico temperatura e umidade do ar.....	59
Figura 39 - Configuração das cores no gráfico.....	59

LISTA DE TABELAS

Tabela 1 - Características Arduino Uno R3	45
---	----

SUMÁRIO

1. INTRODUÇÃO	15
1.1 OBJETIVOS	16
1.1.1 Objetivo Geral	16
1.1.2 Objetivos Específicos	17
1.2 JUSTIFICATIVAS	17
1.3 MOTIVAÇÃO	18
1.4 PERSPECTIVA DE CONTRIBUIÇÃO.....	18
1.5 METODOLOGIA DA PESQUISA	18
1.6 RECURSOS NECESSÁRIOS	19
1.7 ESTRUTURAS DO TRABALHO	19
2. AGRICULTURA 4.0	21
3. INTERNET DAS COISAS (IoT)	23
4. PLATAFORMA ARDUINO	25
4.1 ESTRUTURA DO ARDUINO.....	25
4.2 PLATAFORMA ARDUINO.....	26
4.2.1 O hardware Arduino	26
4.2.2 Ambiente de desenvolvimento	30
4.3 LINGUAGEM DE PROGRAMAÇÃO DO ARDUINO	33
5. NODE.JS E REACT NATIVE	36
5.1 NODE.JS	36
5.2 REACT NATIVE	37
5.2.1 Socket.io	39
6. SENSORES	40
6.1. SENSORES ANALÓGICOS.....	41
6.2 SENSORES DIGITAIS	42

7. ESTUDO DE CASO	43
7.1 DEFINIÇÕES DO PROBLEMA	43
7.2 ARQUITETURA DA APLICAÇÃO	44
7.2.1 Arduino Uno REV3.....	44
7.2.2 Sensor de umidade do solo	45
7.2.3 Sensor de Chuva.....	46
7.2.4 Sensor de Gás Inflamável e Fumaça.....	47
7.2.5 Sensor de Temperatura LM35DZ.....	48
7.3 ESTRUTURA DA APLICAÇÃO	49
7.3.1 Configuração Arduino	49
7.3.2 Utilizando o Node.js.....	52
7.3.3 Utilizando o React Native	54
8. CONCLUSÃO	60
8.1 TRABALHOS FUTUROS	60
REFERÊNCIAS	61

1. INTRODUÇÃO

No mundo atual, é praticamente impossível imaginar uma área de atuação profissional na qual a tecnologia não esteja envolvida, fazendo parte de quase tudo ao nosso redor. Algumas áreas já estão trabalhando com a substituição da mão de obra humana, por um serviço robótico.

Nos últimos vinte anos, milhares de *startups* “explodiram”, inovando modelos de negócios dado como definitivo por séculos. No mundo do agronegócio e da agricultura essa mudança se faz cada vez mais presente (Dino, 2018).

O uso da tecnologia na área agrícola tem crescido cada vez mais, o termo utilizado atualmente para esse fenômeno no campo é “agricultura de precisão”. Essa tecnologia visa facilitar a vida dos agricultores, evitando maiores gastos e desperdícios em suas lavouras.

Segundo GOMES (2016, p. 1):

“A Agricultura de Precisão foi a principal responsável pela busca por novas tecnologias e avanços para as plantações. Este tipo de agricultura é baseado na observação, monitoramento e gestão de todos os insumos necessários e principalmente de toda a plantação. As novas tecnologias vieram para atender a todas estas demandas e alcançar resultados favoráveis e aumentar significativamente a produção.”.

O termo internet das coisas vem de um conceito tecnológico onde todos os objetos que se usam no cotidiano estariam conectados através da internet. Chamada também de IoT (*Internet of Things*), traz consigo a possibilidade de integração entre todos seus objetos tecnológicos, e até alguns que nem se consideraria nesta lista. Com isso, é possível coletar, compartilhar e analisar dados que foram obtidos no campo a uma longa distância. O maior benefício dessa tecnologia é conseguir tomar decisões que sejam estratégicas para o negócio como um todo.

Mas além de facilitar a vida cotidiana, esta tecnologia está se alastrando cada vez no mundo dos negócios. Facilitando diversas tarefas e projetos a internet das coisas veio para ser a revolução tecnológica do século XXI.

Atualmente é possível coletar, compartilhar e analisar dados que foram obtidos no campo a uma longa distância. O maior benefício dessa tecnologia é conseguir tomar decisões que sejam estratégicas para o negócio como um todo.

Ao instalar diversos dispositivos em lugares estratégicos, os agricultores conseguem controlar toda a produção durante a safra.

Os grandes agricultores não precisam mais aplicar água, fertilizantes, pesticidas e alguns insumos uniformemente em toda a plantação.

Com o uso de tecnologias é possível utilizar a quantidade mínima requerida em cada área específica, também se consegue tratar cada planta de maneira única e diferenciada.

Dentre todas as tecnologias disponíveis para a elaboração desse projeto, foi escolhido trabalhar com a plataforma do Arduino.

O Arduino é um conjunto de ferramentas de prototipagem eletrônica *open source* que visa tornar mais fácil a criação de aparelhos eletrônicos. As aplicações do Arduino são basicamente ilimitadas e dependem muito mais da criatividade e disciplina do desenvolvedor do que das limitações da plataforma ou das dificuldades de programação (OPServices, 2015).

Além disso, oferece a oportunidade de trabalhar com sensores, seja de umidade, infravermelho, clima e outros diversos.

Nesse sentido, o objetivo desse trabalho é explorar o funcionamento do Arduino na área agrícola e produzir um protótipo que possa ser utilizado nos celulares para obter informações sobre a área do agricultor.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Este trabalho tem por objetivo o desenvolvimento de um aplicativo que possa monitorar as estufas agrícolas, usando a tecnologia Arduino, trazendo ao usuário informações sobre o solo, clima e umidade. É também objetivo, proporcionar um fácil manuseio e um baixo custo para o consumidor, tornando a aplicação acessível.

1.1.2 Objetivos Específicos

Pretende-se com esse trabalho, adquirir conhecimento necessário para a implementação de um protótipo usando a plataforma Arduino, para monitoramento agrícola de estufas ou jardins. De forma a tornar possível a elaboração do protótipo, foram instituídos alguns objetivos específicos:

- Pesquisar e analisar as possíveis linguagens de programação em Arduino;
- Pesquisar e analisar a plataforma Arduino;
- Pesquisar e analisar os sensores para Arduino, tornando possível a integração e uso dessas tecnologias;
- Pesquisar e analisar as tecnologias Node.js e React Native utilizadas para integração do protótipo a API mobile;
- Especificar um estudo de caso:
 - Modelar o problema;
 - Desenvolver a arquitetura do ambiente;
 - Desenvolver a documentação do software;
 - Implementar o protótipo;
 - Testar e validar;
 - Descrever os resultados obtidos.

1.2 JUSTIFICATIVAS

Este trabalho se justifica pela criação de um protótipo utilizando as tecnologias da plataforma Arduino, visa facilitar o monitoramento das estufas pelos agricultores mesmo estando distante do local. Um outro fator importante da elaboração do presente trabalho é o fato de permitir controlar a umidade do solo, deixando nas condições adequadas para o ambiente. O aplicativo não visa somente os agricultores, por ser um sistema de baixo custo, pode ser implantado em casas com jardins e no qual o proprietário deseje um sistema similar ao protótipo desenvolvido.

Um levantamento realizado pela Comissão Brasileira de Agricultura de Precisão (CBAP) divulgou que pelo menos 67% das propriedades agrícolas no país já adotaram algum tipo de inovação tecnológica dentro ou fora de campo. (PAZETO, 2017).

1.3 MOTIVAÇÃO

No decorrer dos anos os sistemas de automação vêm se tornando cada vez mais úteis, ampliando as possibilidades de trabalho no mercado a partir disso. Muitas empresas fornecem um software do tipo, porém, muitas vezes não atendem a real necessidade do usuário final. O protótipo tem como intuito de que o proprietário consiga acompanhar a sua estufa ou jardim mesmo estando distante do local. Pretende-se desenvolver algo de fácil utilização e baixo custo, para que seja acessível a todas as classes sociais.

Estima-se que em 2050 a população mundial ultrapasse os nove bilhões de pessoas. Para que se possa atender essa crescente demanda, é preciso aumentar a produção de alimentos em 70%. O aumento da demanda implicará em desafios que tornam o contexto ainda mais complexo, como: as mudanças climáticas, que interferem na capacidade produtiva; e restrição de recursos naturais, como a água e o solo. Contudo, o papel da inovação passa a ser essencial para garantir que as próximas gerações possam ser alimentadas com qualidade. (AGRICULTURA DIGITAL, 2018)

1.4 PERSPECTIVA DE CONTRIBUIÇÃO

A criação dessa aplicação visa facilitar a vida do agricultor, trazendo resultados em tempo real. Com o sensor de umidade do solo, a intenção é que o desperdício de água seja diminuído, emitindo avisos quando o solo atingir a umidade adequada, além de não emitir avisos necessitando de irrigação caso esteja chovendo.

1.5 METODOLOGIA DA PESQUISA

A proposta e objetivos deste trabalho acadêmico serão alcançados por meio de pesquisas teóricas, de forma a adquirir os conhecimentos necessários por meio da leitura de artigos científicos, guias práticos e técnicos, livros e fontes digitais, tornando possível a elaboração e implementação de um protótipo para monitoramento agrícola. A metodologia para a execução deste projeto de conclusão de curso é experimental, visto que seu objetivo, além de fomentar a pesquisa acadêmica, é o desenvolver um protótipo de sistema de irrigação com sensores embutidos, utilizando os métodos e conceitos estudados, assim como fazer uso da plataforma Arduino.

1.6 RECURSOS NECESSÁRIOS

Para desenvolver a pesquisa serão necessários os recursos de *hardware* e *software* citados a seguir:

- **Hardware**
 - *Notebook Dell Latitude E5470*
 - Processador *Intel Core i5 2.4 Ghz.*
 - Disco Rígido RPM SATA de 500MB.
 - Memória DDR3 8 GB.
 - Arduino Uno R3
 - Sensores Arduino

- **Software**
 - **Arduino IDE** – Usado para escrever e fazer upload de programas para a placa Arduino.
 - **Node.js** – Ambiente de desenvolvimento integrado para desenvolver para a plataforma mobile.

1.7 ESTRUTURAS DO TRABALHO

A estrutura deste trabalho se compõe das seguintes partes:

- **Capítulo 1 – Introdução:** Neste capítulo é contextualizada a área de estudo e apresentaram os objetivos, justificativas, motivação, perspectivas de contribuição e metodologia de pesquisa para o desenvolvimento deste trabalho.
- **Capítulo 2 – Agricultura 4.0** – Neste capítulo introduz-se uma apresentação sobre o que é Agricultura 4.0, ou Agricultura Digital, e algumas funções.
- **Capítulo 3 – Internet of Things** – Neste capítulo é contextualizado o conceito de Internet of Things, em português, Internet das Coisas, além das suas utilidades diárias.
- **Capítulo 4 – Plataforma Arduino:** Neste capítulo, introduz-se uma apresentação ao recurso utilizado para a elaboração do projeto e alguns de seus modelos.
- **Capítulo 5 – Node.JS e React Native:** Breve apresentação da plataforma utilizada para realização da integração do aplicativo com o Arduino.

- **Capítulo 6 – Sensores:** Aborda os tipos de sensores, descreve sua estrutura e exemplos práticos de aplicação.
- **Capítulo 7 – Estudo de Caso:** Neste capítulo, é apresentada a implementação da abordagem proposta na prática, descrevendo os componentes e tecnologias utilizadas na construção do mesmo.
- **Capítulo 8 – Conclusão:** Aborda a conclusão após o término do estudo e suas considerações finais.
- **Referências**

2. AGRICULTURA 4.0

Agricultura 4.0, termo derivado da Indústria 4.0, que remete à digitalização dos processos de produção. Esse fenômeno vai além da simples mecanização do campo. As operações e decisões passam a ser orientadas com base em dados retirados do clima, da terra, da lavoura, entre outros. (JACTO, 2018)

De acordo MASSRUHÁ (2016) os principais desafios para a humanidade daqui 50 anos serão água, energia, alimentos, pobreza e ambiente, ainda havendo uma suposição que em 2050 a população mundial será de aproximadamente nove bilhões de habitantes.

Estando os recursos naturais cada vez mais escassos, torna o desafio de produzir alimento ainda maior. Diante de tudo isso é indispensável que haja uma mudança na agricultura para que possamos proporcionar uma segurança alimentar para o mundo (MOLIN, 2015).

Segundo o presidente da Jacto, Gonçalves (2018):

"A agricultura 4.0 surge como uma importantíssima ferramenta, pois a aplicação de técnicas sustentáveis no campo deixou de ser uma tendência revolucionária e se tornou uma necessidade para a garantia do abastecimento da cadeia produtiva de alimentos para as próximas gerações. Drones, veículos autônomos, máquinas conectadas, agricultura de precisão, biotecnologia, internet das coisas, big data, entre outros, são alguns dos novos aspectos do trabalho no campo."

De acordo com Gonçalves, quando utilizadas em conjunto, essas técnicas têm como objetivos principais melhorar a produtividade e a eficiência na utilização de insumos, reduzir custos, aumentar a segurança dos trabalhadores e diminuir os impactos ambientais causados pela atividade agrícola.

Em pesquisa realizada pela Fapesp, a economista Nicole Rennó Castro destaca que, o processo de modernização tem reduzido o número de postos de agropecuário, mas, ao mesmo tempo, abre oportunidades para uma mão de obra mais qualificada, inclusive em outros segmentos do agronegócio, como nas agroindústrias e nos agros serviço.

O último Senso Agropecuário do Instituto Brasileiro de Geografia e Estatística (IBGE) mostrou que 1,5 milhão de produtores rurais acessam dados por meio de dispositivos

3. INTERNET DAS COISAS (IoT)

O conceito de Internet das Coisas, ou *Internet of Things (IoT)* é o de uma enorme rede de dispositivos conectados, mas não limitada aos suspeitos habituais. Seu computador, smartphone, tablet ou set-top box, entre outros, são gadgets que dependem da internet para funcionar apropriadamente, assim como equipamentos de grande porte como servidores de grandes empresas. (GOGONI, 2019).

Um dos principais pontos da *Internet of Things* é a ampla abertura para o uso do conceito. Isso porque não há limites nos objetos físicos que podem ser conectados, desde que ofereçam funções. Uma lâmpada, por exemplo, pode ser controlada à distância ou a partir de sensores de movimento ajustados por um aplicativo. (GIANTOMASO, 2018)

Entre o período de 2003 e 2010 o número de dispositivos conectados à rede ultrapassou o número da população mundial e no período de 2015 a 2020 esses números dobraram, o que prova o avanço tecnológico que temos nesta área, conforme pode ser observado na Figura 2.

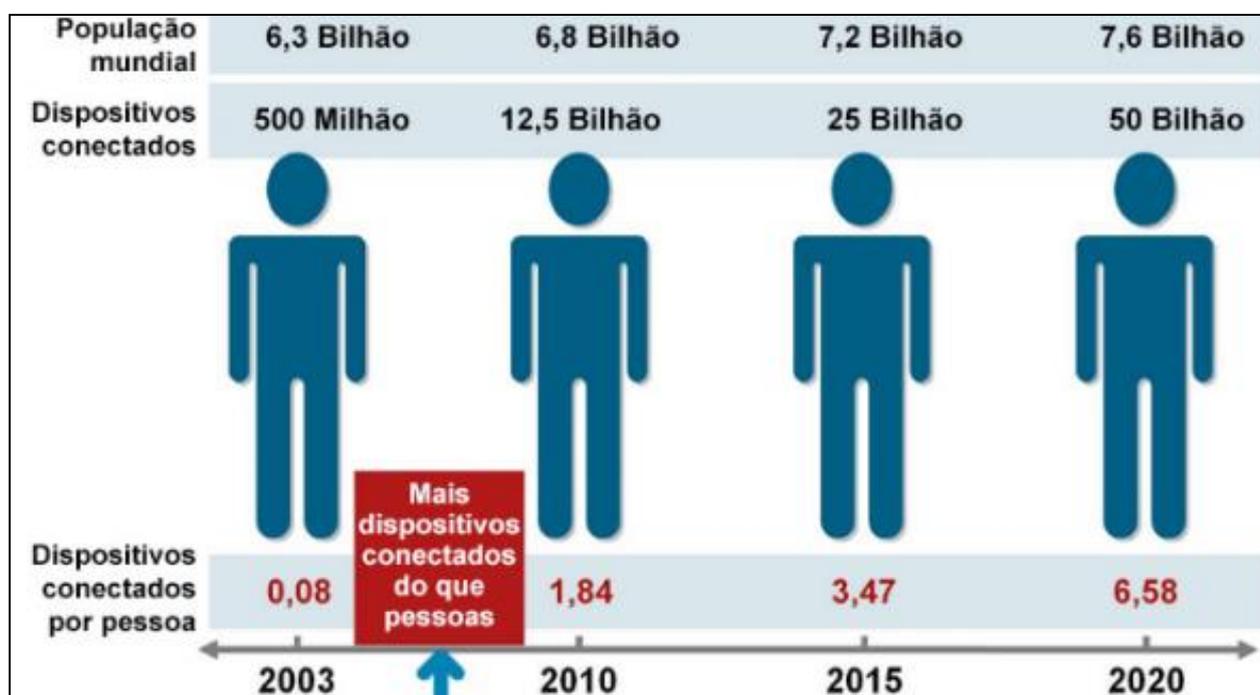


Figura 2 – Comparação de número de brasileiros para dispositivos conectados
 Fonte: EVANS, 2011, p.3

Além de implementar a Internet das Coisas por toda a casa, desde os objetos do quarto e da sala até os eletrodomésticos da cozinha, a conexão com a Internet também pode chegar às ruas e avenidas, tornando cidades inteiras inteligentes. (GIANTOMASO, 2018)

No geral, qualquer coisa pode ser conectada à internet, mas isso não significa que tudo deveria sê-lo. A ideia principal por trás da Internet das Coisas é a de facilitar a vida dos usuários e clientes, tornando o uso de certos elementos mais simples e até permitindo a automação de tarefas. (GOGONI, 2019).

Em síntese, o que descrevemos acima é a prática da Internet das Coisas: promover a conexão via internet de “coisas”. Ou seja, a união via internet de tudo que é passível de conexão.

Uma pesquisa realizada pela Avast em 2019 apresenta alguns produtos que fazem uso do conceito na lista de desejos de aquisição do brasileiro. Em primeiro lugar se encontra o famoso *Smartwatch* (relógio inteligente), que consegue captar diversas informações ao seu redor conforme a figura 3.

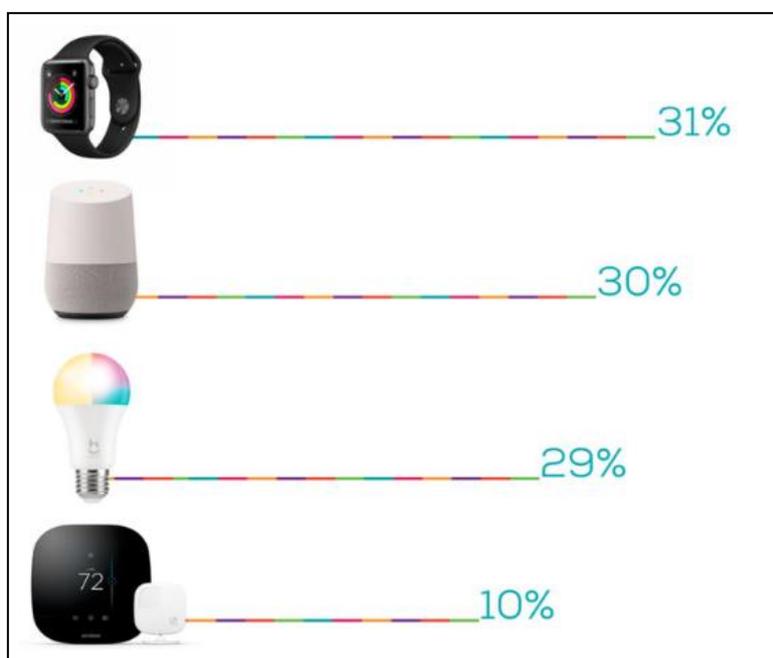


Figura 3 – Estatísticas dos dispositivos desejados pelo brasileiro
Fonte: Avast Brasil

4. PLATAFORMA ARDUINO

O Arduino é uma plataforma de hardware *open-source*, projetada sobre o microcontrolador Atmel AVR, que pode ser programado através de uma linguagem de programação similar a C/C++, permitindo a elaboração de projetos com um conhecimento mínimo ou mesmo nenhum de eletrônica. Foi criado com o objetivo de fornecer uma plataforma de fácil prototipação de projetos interativos, unindo software e hardware, características da Computação Física. (OLIVEIRA; ZANETTI, 2015)

Elaborado em 2005 na Itália por Massimo Banzì e David Cuartielles, e tem como o objetivo facilitar o desenvolvimento de projetos dos mais simples até os mais complexos.

Um ponto forte sobre sua utilização, é que todo material disponibilizado, como a IDE de desenvolvimento, bibliotecas e até o projeto eletrônico das placas, são *open-source*, permitindo a utilização e reprodução sem restrição sobre os direitos autorais dos idealizadores do projeto.

4.1 ESTRUTURA DO ARDUINO

O Arduino é uma plataforma de computação física e tem como base uma simples placa microcontrolada de Entrada/Saída para que se possa conectá-los a outros circuitos ou sensores. São sistemas digitais ligados a sensores e atuadores, que permitem construir sistemas que percebam a realidade e respondam com ações físicas. (FONSECA; BEPPU, 2010).

O Arduino é um computador como qualquer outro, possuindo:

- Microprocessador (responsável pelos cálculos e tomada de decisão)
- Memória ram (utilizada para guardar dados e instruções, volátil)
- Memória flash (utilizada para guardar o software, não volátil)
- Temporizadores (*timers*)
- Contadores
- Clock, e etc.

Através de sensores conectados aos terminais de entrada, o Arduino é capaz de receber e interpretar variáveis e converte-las em sinais elétricos, controlando uma variedade de

luzes, motores ou outra saída conectada ao seu periférico de saída, como ilustrado na figura 4.



Figura 4 - Diagrama de Blocos
Fonte: Apostila Arduino p.2

4.2 PLATAFORMA ARDUINO

De acordo com SOUZA (2013), o Arduino é composto por hardware e software, sendo o primeiro a placa sobre a qual são montados os projetos (prototipagem) e o último o programa no qual será feita a programação (sketch) que nada mais é que a sequência de comandos para coordenar as funções que o Arduino vai executar depois, a programação feita pelo projetista é enviada posteriormente à placa (upload).

Campos (2014, p. 24) caracteriza o Arduino:

O Arduino é um dispositivo formado principalmente por 2 componentes básicos: a placa Arduino, que é o elemento de hardware utilizado para construir seus objetos e a IDE (*Integrated Development Environment*) do Arduino, que é um programa executado no computador por onde escrevemos o código (chamado de sketch) e que fará o upload para a placa Arduino.

4.2.1 O hardware Arduino

A placa do Arduino é um pequeno circuito microcontrolado, onde são colocados todos os componentes necessários para que este funcione e se comunique com o computador. Existem diversas versões de placas que são mantidas sob licença open-source.

O Arduino também pode ser estendido utilizando Shields, que são placas de circuito contendo outros dispositivos, como displays de LCD, módulos de Ethernet, receptores de GPS, entre outros, que conectados ao Arduino, acrescentam funcionalidades adicionais.

Dentre os todos os modelos de Arduino, destacam-se as placas Arduino Duemilanove (2009), Arduino Mega, Arduino Nano e Arduino Uno.

A figura 5 apresenta um modelo de placa do Arduino Uno, detalhando a função de alguns de seus conectores, que podem ser encontrados em outras versões da placa:

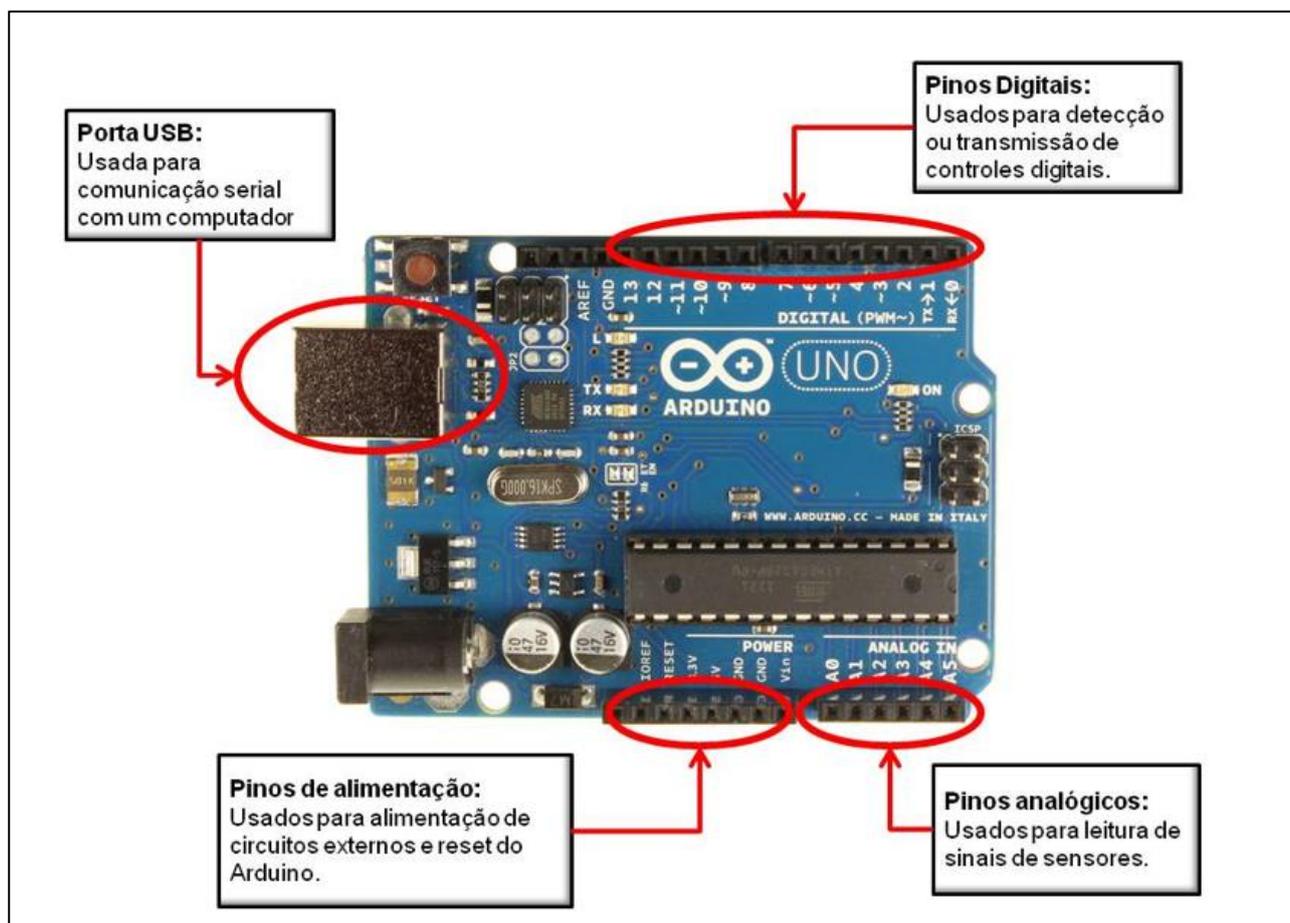


Figura 5 - Arduino Uno
Fonte: VICCIOLI, 2020

Arduino Duemilanove: (“2009” em italiano), é uma placa micro controladora baseada no ATmega168 ou ATmega328, tem 14 pinos de entrada / saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um oscilador de cristal de 16 MHz , uma conexão USB, um conector de energia, um conector ICSP e um botão de reset. Para utilizá-lo basta conectá-lo a um computador com um cabo USB ou ligá-lo a um adaptador de CA-CC ou bateria.

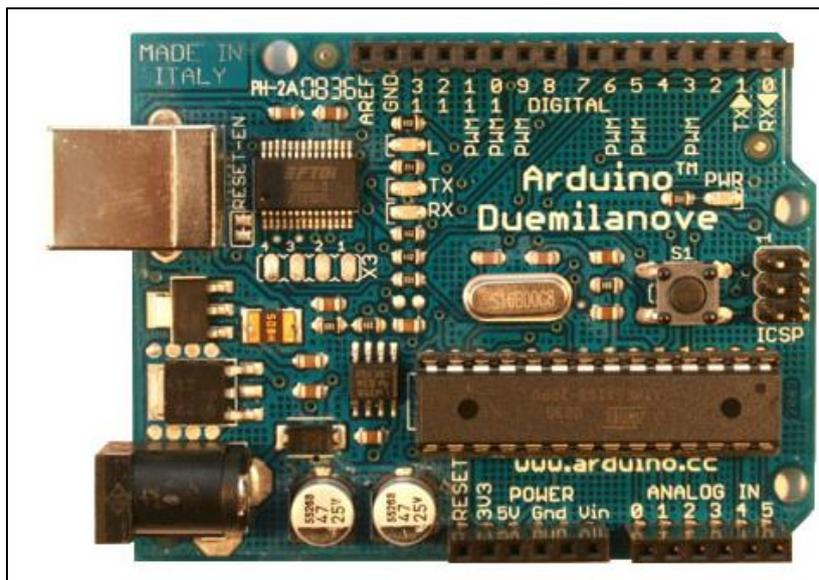


Figura 6 - Arduino Duemilanove
Fonte: Arduino, 2020

Arduino Mega 2560: foi desenvolvida para projetos mais complexos. Com 54 pinos digitais de Entrada / Saída e 16 entradas analógicas, é a placa recomendada para impressoras 3D e projetos de robótica.

O mega é compatível com a maioria dos Shields projetados para o Arduino Duemilanove ou Diecimila. (Arduino, 2020).



Figura 7 - Arduino Mega 2560
Fonte: Arduino, 2020

Arduino Nano: uma versão pequena do Arduino, completa e fácil de usar, baseada no ATmega328P. Suas funcionalidades são parecidas com a do Arduino Duemilanove, mas em um pacote diferente. Esta placa não possui um conector para fonte externa, mas é possível alimentá-la pelo pino Vin. O Arduino Nano automaticamente seleciona a maior alimentação fornecida.

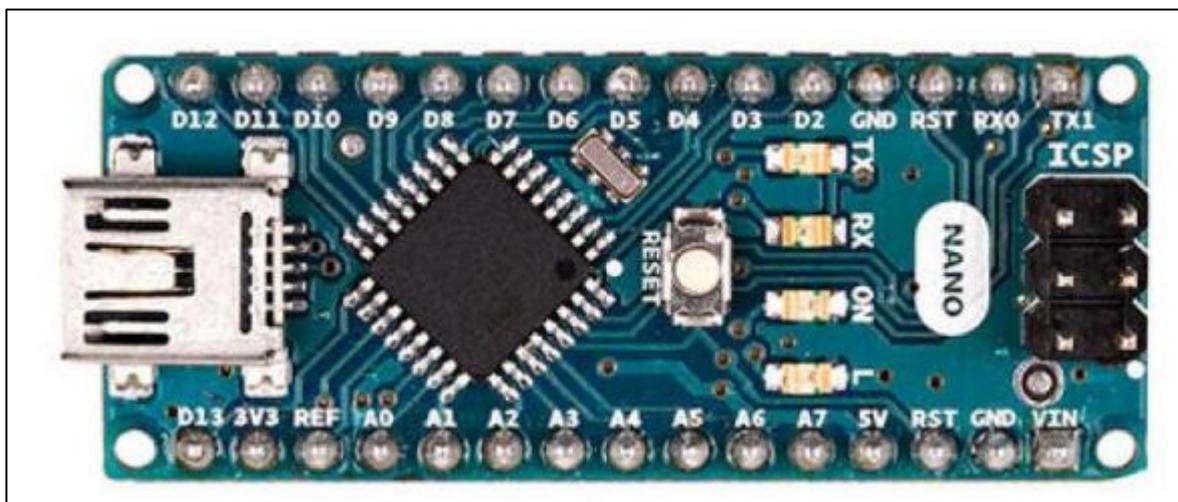


Figura 8 - Arduino Nano
Fonte: Arduino, 2020

Arduino UNO: O Arduino UNO é uma placa micro controladora de código aberto baseada no microcontrolador Microchip ATmega328P. A placa é equipada com conjuntos de pinos de entrada / saída (E / S) digitais e analógicos que podem ser conectados a várias placas de expansão e outros circuitos. A placa possui 14 pinos digitais, 6 pinos analógicos e programável com o Arduino IDE (ambiente de desenvolvimento integrado) por meio de um cabo USB tipo B. Ele pode ser alimentado por um cabo USB ou por uma bateria externa de 9 volts, embora aceite tensões entre 7 e 20 volts.



Figura 9 - Arduino UNO
Fonte: Arduino, 2020

Conforme visto a figura 9, a placa Arduino UNO possui diversos conectores que servem para interface com o mundo externo.

4.2.2 Ambiente de desenvolvimento

Assim como em qualquer dispositivo programável, a plataforma Arduino necessita que os programas sejam desenvolvidos em uma linguagem de programação, compilados e, posteriormente, transferidos para o Arduino. (LIVRO ARDUINO)

Para se criar sketches para a placa arduino é necessário um programa executando em um computador e esse programa é chamado de IDE (*Integrated Development Environment*). É nesse software que se escreve o programa para carregar o arduino e consequentemente, o que ele realizará. A programação do arduino é feita em ciclos e pode ser basicamente dividida nas etapas a seguir (Banzi, 2010):

- Conectar a placa a uma porta USB do computador;
- Escrever o código na IDE para dar vida à placa;
- Fazer o upload (o carregamento) desse código (sketch) para o arduino através da conexão USB, aguardando alguns instantes a sua reinicialização;

- E por fim, o arduino irá executar o *sketch* escrito;

Após a instalação será possível abrir a IDE do Arduino, que tem a seguinte aparência:

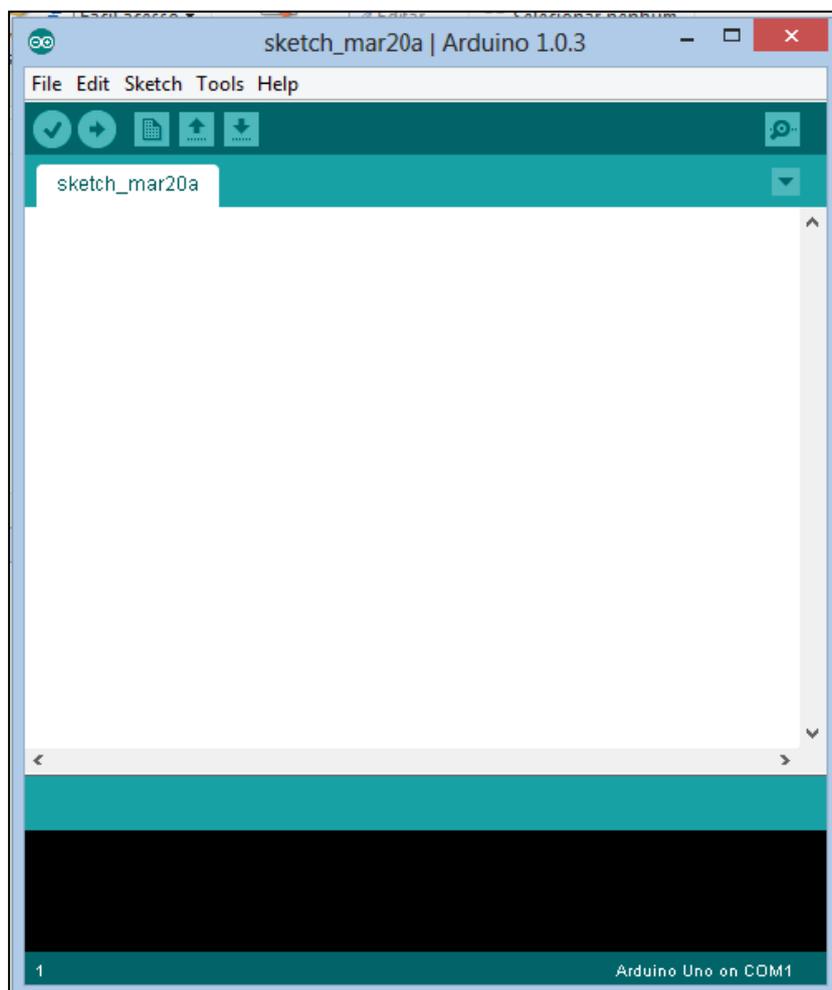


Figura 10 - IDE do Arduino
Fonte: Arduino, 2020

O IDE do Arduino é dividido em três partes: A Toolbar, o código ou a Sketch Window, e a janela de mensagens na base, conforme exibido na figura 11.

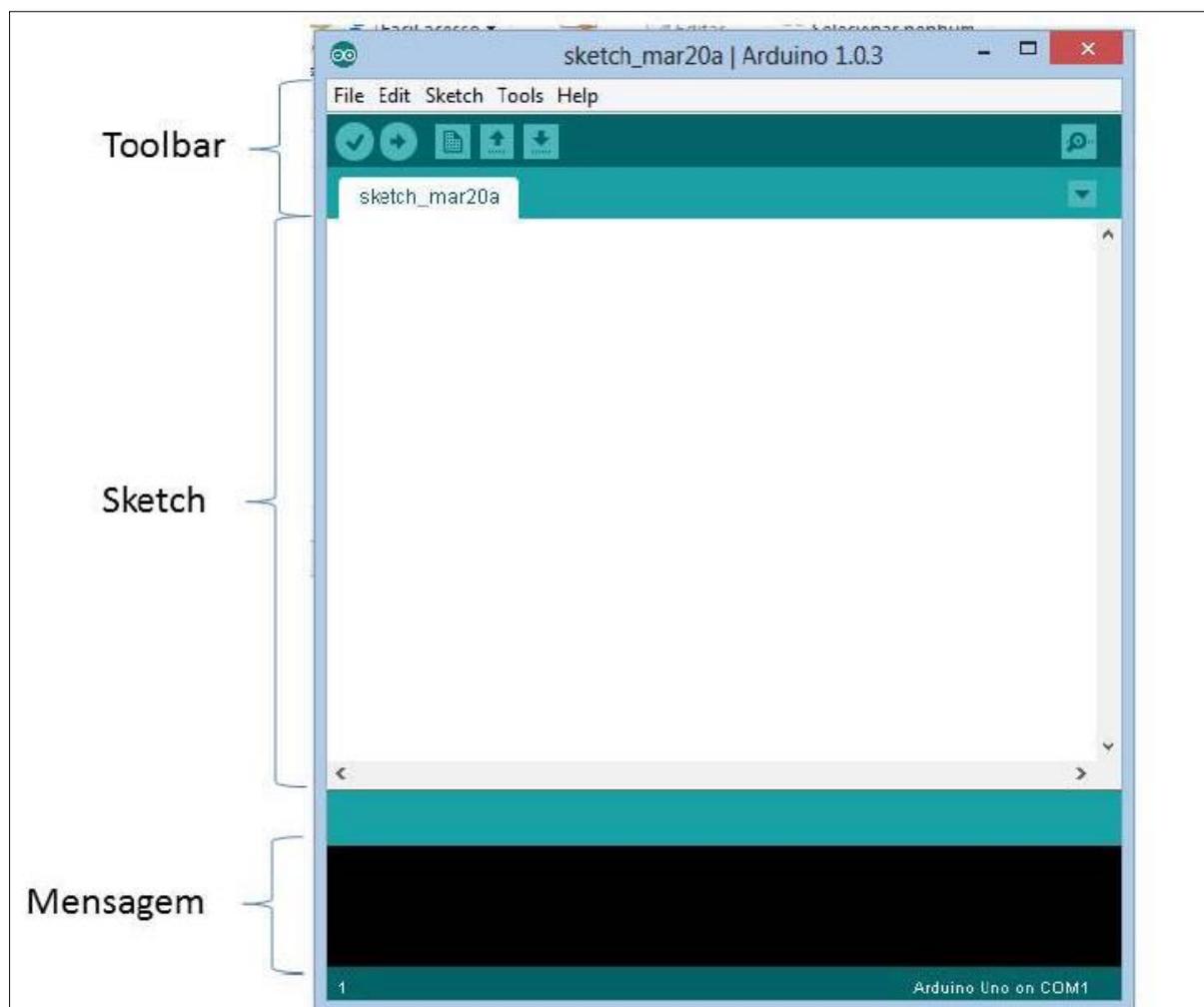


Figura 11 - Divisão da IDE do Arduino
Fonte: Arduino, 2020

Há um conjunto de guias na Toolbar, com o nome do arquivo do sketch. Também há um botão posicionado no lado direito que habilita o serial monitor. Ao longo do topo há uma barra de menus, com os itens File, Edit, Sketch, Tools e Help. Os botões na Toolbar fornecem acesso conveniente às funções mais utilizadas dentro desses menus.

A Toolbar, na figura a seguir, é composta por botões que fornecem acesso direto às funções mais utilizadas pelo programador, conforme a figura 12.



Figura 12 - Toolbar do Arduino
Fonte: Arduino, 2020

Abaixo são identificados os ícones de atalho da IDE em sua devida sequência:

- **Verify / Verificar** – A função desse botão é verificar se o Sketch (código) está livre de erros antes de ser feito o seu *upload* para a placa do Arduino.
- **Upload / Carregar** – O botão carregar tem a função de compilar o código e realizar o upload para a placa do Arduino.
- **New / Novo** – Cria um novo Sketch em branco.
- **Open / Abrir** - Apresenta uma lista de sketches armazenados em seu sketch book. O sketch book é uma lista com exemplos de sketch já prontos, que tem como objetivo dar uma visão de como serão os projetos aos usuários iniciantes.
- **Save / Salvar** – Salva o Sketch ativo.
- **Monitor Serial** - O monitor serial é uma ferramenta muito útil, especialmente para depuração de código. O monitor exibe no console os dados seriais enviados do Arduino.

4.3 LINGUAGEM DE PROGRAMAÇÃO DO ARDUINO

O Arduino utiliza uma linguagem de programação simples, adaptada da linguagem *Processing*, que é chamada de *Wiring*. O que torna essa linguagem simples é o fato de quando pressionado o botão de *upload*, o código escrito é traduzido para a linguagem C. (Banzi, 2010).

A estrutura em si para programação em Arduino é bem básica, podendo ser formada por dois blocos de funções que carregam outros blocos de funções escritas em linguagem C/C++. O primeiro bloco de funções forma a função **setup()**; o segundo, a função **loop()**.

- **setup()**: Essa é a primeira função a ser chamada quando o programa inicia. Esta é uma função de preparação: ele dita o comportamento dos pinos do Arduino e inicializa a porta serial.
- **loop ()**: A função loop() é chamada logo a seguir e todas as funções embarcadas nela são repetidamente executadas. Ela fica lendo os pinos de entrada do Arduino e comandando os pinos de saída e porta serial.

Os símbolos usados na construção de funções são os seguintes:

- **{ }** - Dentro das chaves vão os procedimentos (*statements*) que a função deve executar;

- ; - O ponto-e-vírgula é usado para marcar o final de um procedimento.
- // - Comentário de uma linha: qualquer caractere depois das duas barras é ignorado pelo programa;
- /*...*/ - Comentário em várias linhas: qualquer texto colocado entre esses símbolos também é ignorado pelo programa.

Os programas para o Arduino são implementados tendo como referência a linguagem C++, conservando sua sintaxe clássica na declaração de variáveis, nos operadores, nos ponteiros, nos vetores, nas estruturas e em muitas outras características da linguagem. (FONSECA; BEPPU, 2010).

As estruturas de referências são representadas por:

- Estruturas de controle if, else, break, ...
- Sintaxe básica define, include, , ...
- Operadores aritméticos e de comparação +, -, =, ==, !=, ...
- Operadores booleanos, ||, !.
- Acesso a ponteiros *
- Operadores compostos ++, -, +=, ...
- Operadores de bits |, ^, ...

Os valores de referências são representados por:

- Tipos de dados byte, array, int, char, ...
- Conversões char(), byte(), int(), ...
- Variável de escopo e de qualificação variable scope, static, volatile, ...
- Utilitários sizeof(), diz o tamanho da variável em bytes.

Algumas das outras funções que podem ser utilizadas no Arduino:

- **pinMode():** Configura o pino do Arduino especificado para que se comporte ou como entrada (INPUT) ou saída (OUTPUT).
- **digitalWrite():** Essa função aciona um valor HIGH ou LOW em um pino digital. Se o pino for configurado como saída (OUTPUT) com a função pinMode(), sua tensão será acionada para o valor correspondente: 5V (ou 3.3V em placas alimentadas com 3.3V como o DUE) para o valor HIGH, 0V (ou ground) para LOW.

- **digitalRead():** Lê o valor de um pino digital especificado e retorna um valor HIGH ou LOW.
- **Serial.begin():** Ajusta a taxa de transferência em bits por segundo para uma transmissão de dados pelo padrão serial.
- **delay():** Suspende a execução do programa pelo tempo (em milissegundos) especificado.

5. NODE.JS E REACT NATIVE

As plataformas Node.js e React Native quando trabalhadas em conjunto, oferecem inúmeras possibilidades de aplicação. São ferramentas poderosas, neste capítulo será abordado o básico sobre ambas.

5.1 NODE.JS

O Node.js se caracteriza como um ambiente de execução JavaScript. O usuário pode criar aplicações sem depender do browser para isso. Com alta capacitação de escalabilidade, boa flexibilidade, arquitetura e custo baixo, acaba se tornando uma excelente opção para programação. (PROJECT.TI, 2020).

A utilização do Node.js tem seus benefícios, sendo um deles a reutilização de códigos, ajudando na produtividade. Além de ser uma plataforma versátil, que pode ser utilizada em vários cenários. O gerenciador de pacotes se classifica como o maior repositório de software disponível.

Os recursos computacionais são menos exigidos ao usar a linguagem Node.js por conta do seu formato single thread.

O uso do Node.js é mais comum nos casos de aplicações em tempo real. Recomendado para ambiente escaláveis pelo seu potencial de suportar um número maior de conexões simultâneas comparadas com servidor tradicional.

Enquanto um servidor tradicional faz uso do sistema multi-thread para tratar as requisições concorrentes, o Node.js realiza o mesmo efeito através das chamadas de entrada e saída (E/S) não-bloqueantes. Isso significa que as entradas e saídas das operações são assíncronas e não bloqueiam a thread. No caso dos servidores tradicionais, a thread não espera que a operação seja concluída para prosseguir com a continuação.

A figura abaixo apresenta a diferença no funcionamento de um servidor web tradicional e um Noje.js:

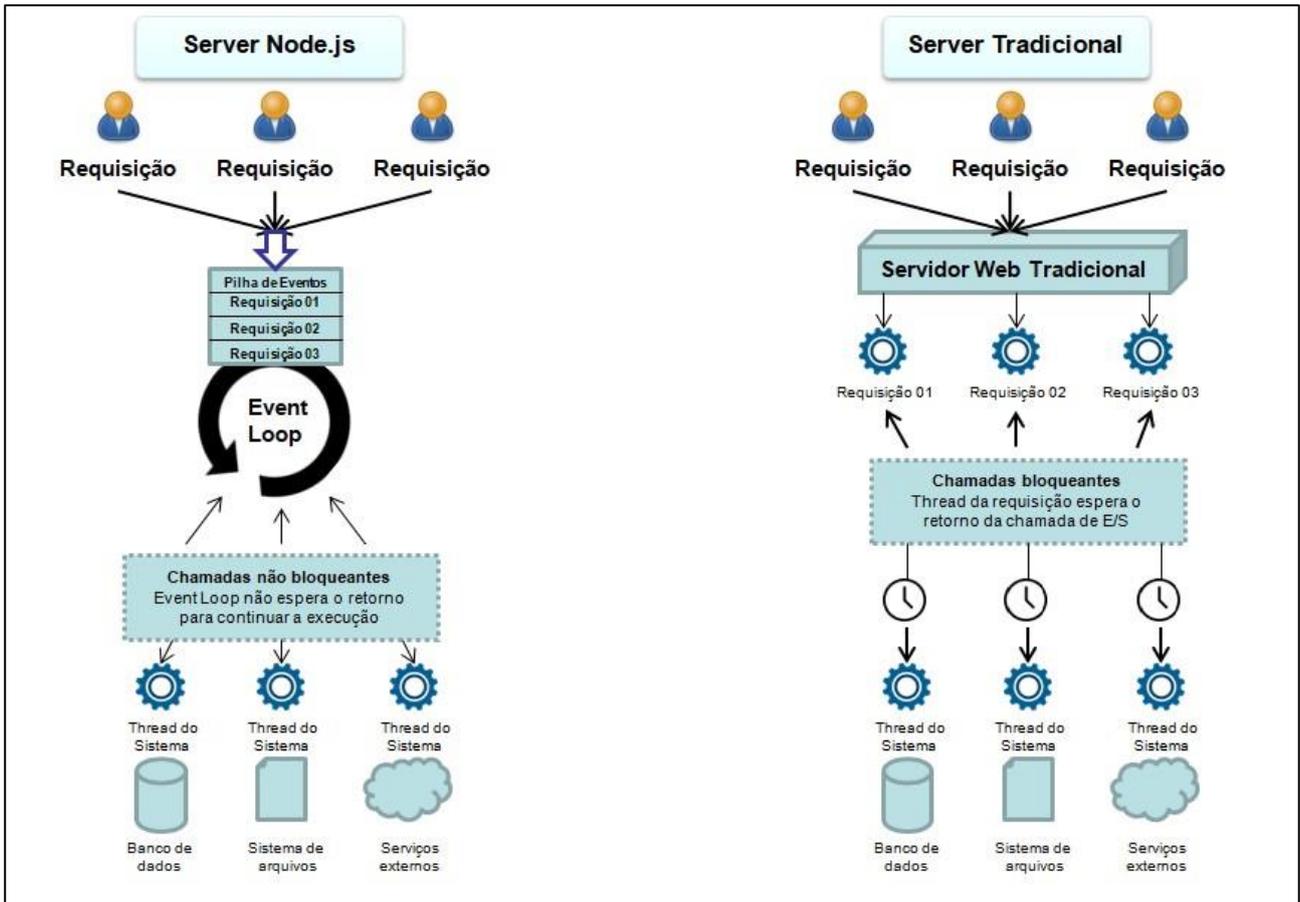


Figura 13 - Servidor Node.js x servidor tradicional

Fonte: <https://www.opus-software.com.br/node-js/>

5.2 REACT NATIVE

O React Native consiste em um conjunto de ferramentas que propiciam a criação de aplicativos móveis nativos para as plataformas iOS e Android, utilizando o conceitos de front-end (ROMERO; SAAD; BASTOS, 2018).

Os princípios de funcionamento do React Native são virtualmente idênticos ao ReactJs, exceto que o React Native não manipula o DOM através do Virtual DOM.

O DOM é uma API que permite acessar e manipular documentos HTML e XML válidos. O Virtual DOM, por sua vez, é uma representação do DOM mantida em memória, realizando uma busca ou alteração mais rápida que por meio de sincronização em um chamado “Reconciliação”, gera uma análise permitindo que haja um menor número de acessos possíveis ao DOM.

Ele é executado em um processo em segundo plano (que interpreta o JavaScript escrito pelos desenvolvedores) diretamente no dispositivo final e se comunica com a plataforma nativa por meio de uma serialização, uma ponte assíncrona e em lote.

Esse framework possui diversas características fundamentais e marcantes para sua ampla adoção:

- Por ser multiplataforma, podemos desenvolver aplicações com React Native utilizando qualquer sistema operacional (Windows, macOS ou Linux).
- Todo código desenvolvido é convertido para a linguagem nativa do sistema operacional;
- O código produzido é semelhante ao React para Web;
- Com o React Native consegue-se desenvolver aplicações para Android e iOS utilizando um código único;
- Possui a base de conhecimento compartilhada entre o desenvolvimento mobile e front-end;
- Acessar a interface e os recursos nativos do Android e iOS utilizando JavaScript;

O diferencial no desenvolvimento com o framework React Native é a não necessidade de uma camada de renderização baseado em Web, utilizando componentes internos para a tarefa (LEITE et al., 2018).

São diversas funcionalidades muito interessantes que aumentam a praticidade e a produtividade do desenvolvimento, mas vale enfatizar uma delas. O Hot Reloading, que faz com que o programa fique rodando em desenvolvimento, e a cada atualização no código uma versão nova do arquivo modificado é injetado na aplicação, levando menos de 1 segundo para atualizar. (BECKER, 2020)

Apesar de ser baseado no React, o React Native não utiliza tags HTML para a criação de seus elementos. Ao invés disso, os elementos são criados utilizando tags nativas do React Native, como podemos ver no trecho de código apresentado na Figura 14.

```
import React from 'react';
import { Text } from 'react-native';

export default function Ola() {
  return (
    <Text>Olá, mundo!</Text>
  );
}
```

Figura 14 - Exemplo React Native
Fonte: TreinaWeb

5.2.1 Socket.io

Socket.io é uma biblioteca que permite a comunicação em tempo real, bidirecional e baseada em eventos entre o navegador e o servidor.

A biblioteca Socket.io convencional é escrita em JavaScript tanto para front end, quanto para back-end, por isso foi projetada para rodar em um servidor Node.js. (ZIM, 2013).

Segundo o site da Socket.io, a função dessa biblioteca é tornar possível que os aplicativos sejam executados em tempo real, em navegadores e dispositivos móveis, minimizando a diferença entre o mecanismo de transportes diferentes.

No caso do desenvolvimento desse protótipo, o socket.io está sendo usado como uma biblioteca para troca de informações em tempo real do Node.js para o React Native.

6. SENSORES

Um Sensor em Eletrônica é um dispositivo que capta e converte um fenômeno físico, como temperatura, umidade ou luminosidade, em um sinal elétrico. Desta forma, os sensores fazem parte da interface entre o mundo físico e o mundo dos dispositivos eletrônicos, como os computadores e redes de dados. (REIS, 2018).

Um sensor nem sempre tem as características elétricas necessárias para ser utilizado em um sistema de controle. Normalmente o sinal de saída deve ser manipulado antes da sua leitura no sistema de controle. Isso geralmente é realizado com um circuito de interface para produção de um sinal que possa ser lido pelo controlador. (THOMAZINI, ALBUQUERQUE, 2009).

Depois que o sensor recebe o estímulo, a sua função é emitir um sinal que seja capaz de ser convertido e interpretado pelos outros dispositivos.

Os elementos sensoriais são definidos como transdutores por converter uma grandeza de entrada para uma grandeza elétrica, como corrente, tensão ou resistência elétrica.

Os sensores são um dos principais equipamentos necessários para um sistema domótico, pois é com ele que o sistema vai captar os diversos estímulos que ocorrem dentro de uma residência e converter estes estímulos em sinais elétricos e manda-los para o controlador, para que este mande um sinal para os atuadores executarem alguma ação de acordo com o que foi programado. Cada tipo de sensor é adequado para um determinado tipo de percepção e quanto mais sensores instalados melhor (HUIDOBRO et al., 2007).

Supondo que a saída de um sensor, ao ser sensibilizado por uma energia externa, é dada por um nível de tensão muito baixo, torna-se necessária a sua amplificação. Essa interface seria então um amplificador capaz de elevar o nível do sinal para sua efetiva utilização. (THOMAZINI, ALBUQUERQUE, 2009).

A figura 15 demonstra uma ilustração adaptada das formas de energia em um sensor.

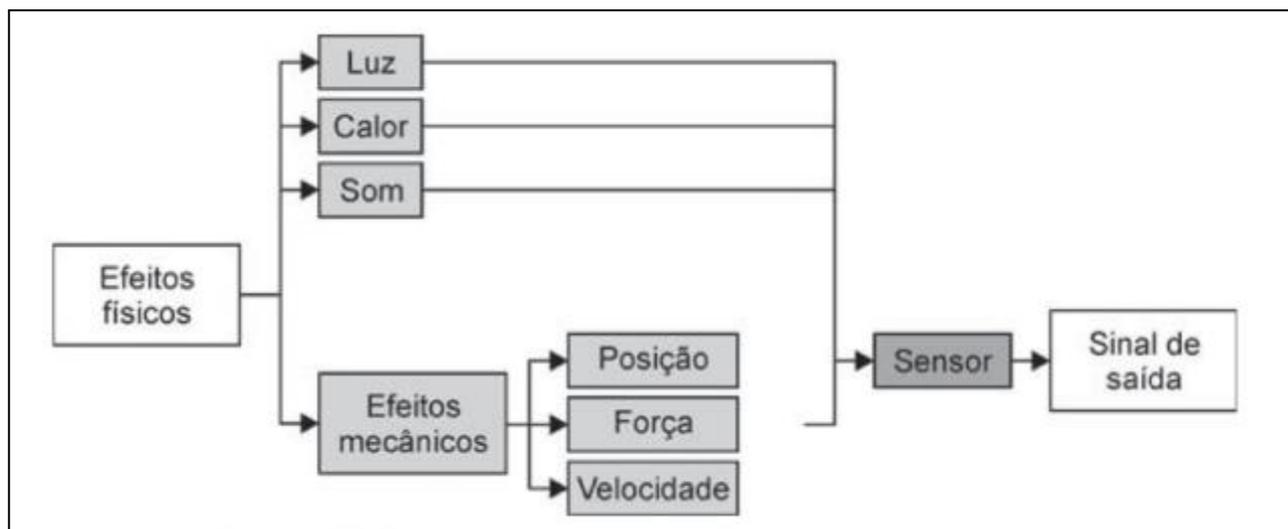


Figura 15 - Ilustração das formas de energia em um sensor.
Fonte: Adaptado de (THOMAZINI; ALBUQUERQUE, 2005)

6.1. SENSORES ANALÓGICOS

Algumas das grandezas físicas que podem assumir qualquer valor ao longo do tempo são: pressão, temperatura, velocidade, umidade, vazão força, ângulo, distância, torque, luminosidade. Essas variáveis são mensuradas por elementos sensíveis com circuitos eletrônicos não digitais. (THOMAZINI, ALBUQUERQUE, 2009).

A figura a seguir, ilustra a variação de uma grandeza física (temperatura) de forma analógica.

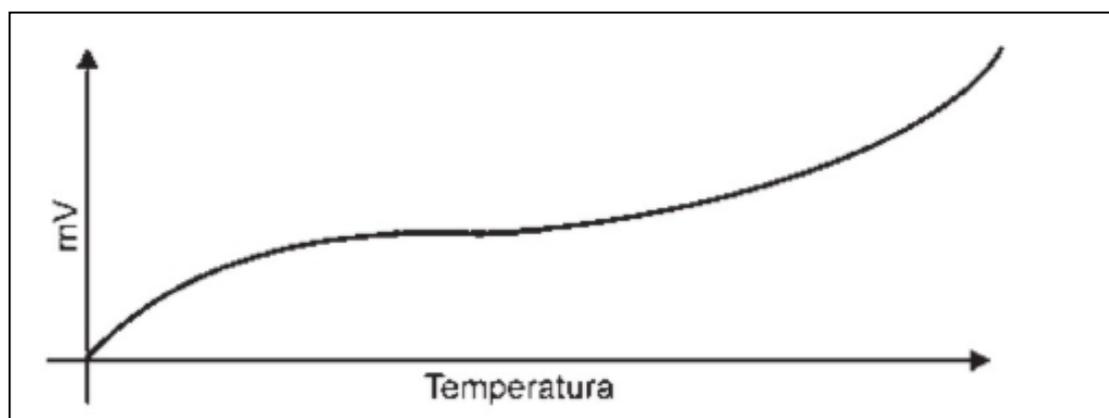


Figura 16 - Ilustração da variação de uma grandeza física de um sensor analógico.
Fonte: Adaptado de Thomazini, 2005

Os sensores podem ser classificados de acordo a saída do sinal, sendo eles, sensores discretos que possuem um sinal quantificado que indica a existência ou não de um evento podendo assim assumir os valores zero ou um ou uma combinação destes, e sensores analógicos que tem como saída um sinal cuja informação pode assumir qualquer valor dentro de uma determinada faixa. (FUENTES, 2005).

6.2 SENSORES DIGITAIS

Esse tipo de sensor pode assumir apenas dois valores no seu sinal de saída ao longo do tempo, que podem ser interpretados como zero ou um. Não existem naturalmente grandezas físicas que assumam esses valores, mas eles são assim mostrados ao sistema de controle após serem convertidos pelo circuito eletrônico do transdutor. (THOMAZINI, ALBUQUERQUE, 2009).

Geralmente são utilizados em detecção de passagem de objetos, encoders na determinação de distância ou velocidade, etc.

A Figura 17 ilustra a variação da posição de um objeto lida por um encoder incremental.

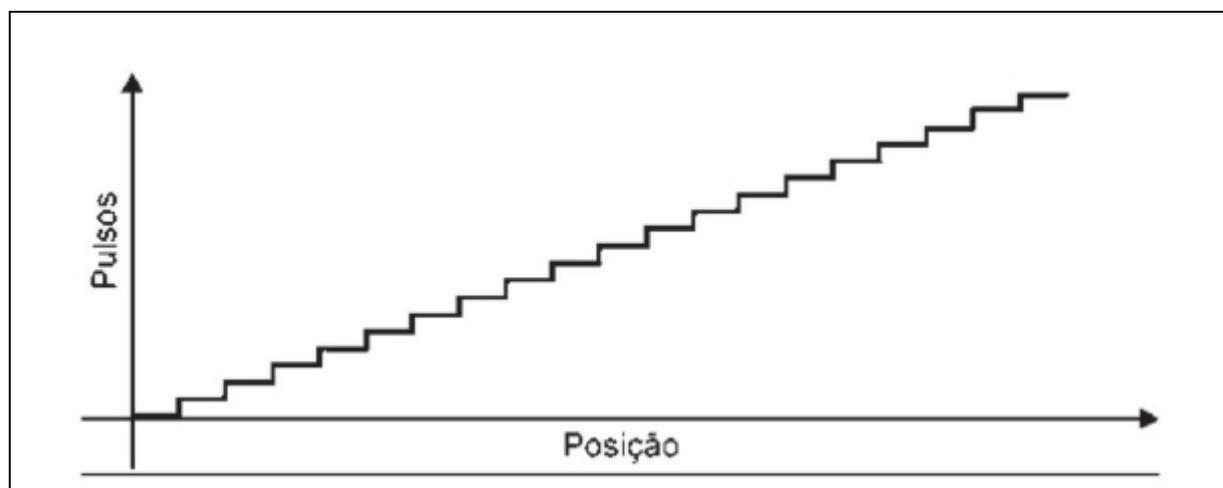


Figura 17 - Ilustração da leitura da posição de um objeto por um encoder incremental.

Fonte: Adaptado de Thomazini, 2005

7. ESTUDO DE CASO

O capítulo a seguir se retrata dos procedimentos para realização do protótipo no qual foi proposto para o trabalho de conclusão.

7.1 DEFINIÇÕES DO PROBLEMA

Quando se ouve falar em agricultura, se vem em mente uma visão estereotipada do campo. A verdade é que o campo está se modernizando, e com o avanço da tecnologia está cada vez mais conectado digitalmente.

Anteriormente, esse movimento chamado pela indústria como Agricultura Digital, entra agora uma nova fase, conhecida por Agricultura 4.0. Hoje as plantações de milhões de hectares podem ser monitoradas por meio de smartphones, computadores e tablets, dando início a uma nova geração de agricultores, ávidos por tecnologias capazes de rentabilizar o seu negócio.

A nova realidade, é que a Agricultura 4.0 provoca a necessidade do agricultor buscar qualificações adequadas para esses novos tipos de equipamento. Afinal, este é um caminho sem volta, que veio para facilitar o replanejamento das empresas agroindustriais e que impacta tanto o pequeno produtor quanto os grandes grupos agrícolas.

Dessa forma esse estudo visa a implementação de um sistema de monitoramento agrícola, seja para uma estufa ou até mesmo um jardim, utilizando uma tecnologia precisa e barata. Através da tecnologia Arduino juntamente com o uso de sensores analógicos é possível construir um sistema de monitoramento agrícola que realiza as mesmas tarefas de sistemas de grandes empresas. O grande diferencial é que devido à utilização de tecnologias baratas, o sistema fica acessível a qualquer tipo de pessoa que tenha interesse.

O sensor realiza uma leitura das variáveis físicas do ambiente e as envia ao Arduino, por sua vez, recebendo as variáveis as transmite para o dispositivo mobile, informando ao usuário as condições atuais do local onde está implementado o dispositivo Arduino.

7.2 ARQUITETURA DA APLICAÇÃO

Para o desenvolvimento do protótipo de monitoramento são necessários os seguintes itens:

- Uma placa Arduino Uno
- Um sensor de umidade do solo
- Um módulo Sensor de Chuva
- Um Sensor de Gás MQ-2 - Gás Inflamável e Fumaça
- Um Sensor de Temperatura LM35DZ

7.2.1 Arduino Uno REV3

O Arduino Uno é uma placa micro controladora baseada no ATmega328P. Possui 14 pinos de entrada / saída digital (dos quais 6 podem ser usados como saídas PWM), 6 entradas analógicas, um cristal de quartzo de 16 MHz, uma conexão USB, um conector de energia, um conector ICSP e um botão de reset. Ele contém tudo o que é necessário para suportar o microcontrolador. (ARDUINO, 2020).

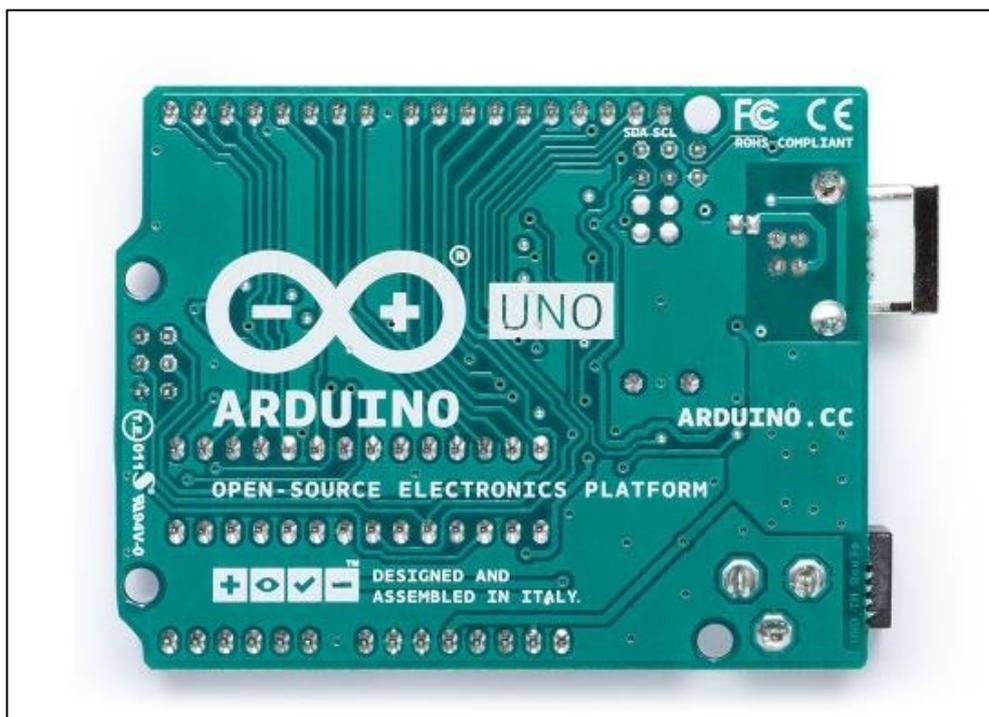


Figura 18 - Arduino Uno
Fonte: Arduino, 2020

Na tabela a seguir podemos observar algumas das características do Arduino Uno Rev3:

Micro controlador	ATmega328
Tensão de operação	5V
Tensão de alimentação (recomendada)	7-12V
Tensão de alimentação (limite)	6-20V
Entradas e saídas digitais	14 das quais 6 podem ser PWM
Entradas analógicas	6
Corrente contínua por pino de I/O	40 mA
Corrente contínua para o pino 3.3V	50 mA
Memória Flash	32 KB (ATmega328) dos quais 0.5 KB são usados pelo bootloader
Memória SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Velocidade do Clock	16 MHz
Dimensões	68,58mm x 53,34mm
Peso	150g

Tabela 1 - Características Arduino Uno R3
Fonte: Arduino, 2020

7.2.2 Sensor de umidade do solo

Este sensor foi feito para detectar as variações de umidade no solo, sendo que quando o solo está seco a saída do sensor fica em estado alto, e quando úmido em estado baixo.

O limite entre seco e úmido pode ser ajustado através do potenciômetro presente no sensor que regulará a saída digital D0. Contudo, para ter uma resolução melhor, é possível utilizar a saída analógica A0 e conectar a um conversor AD, como o presente no Arduino, por exemplo.

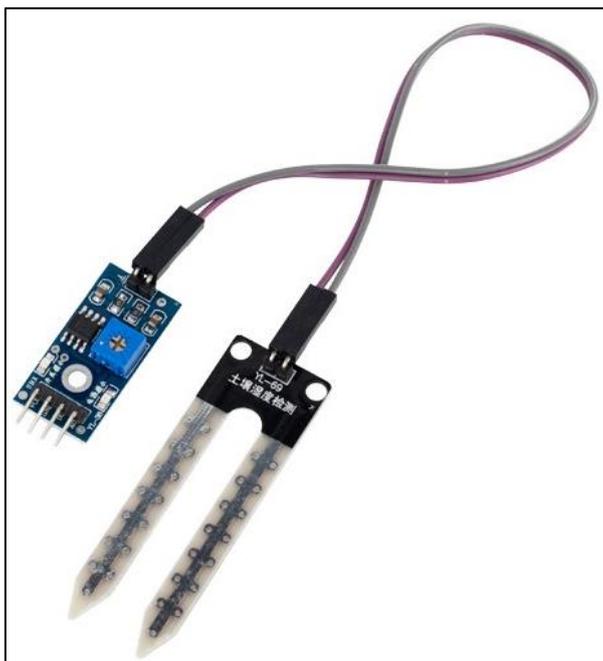


Figura 19 - Sensor de umidade do solo
Fonte: FILIPEFLOP, 2020

7.2.3 Sensor de Chuva

Este sensor pode ser usado para monitorar uma variedade de condições climáticas como gotas de chuva ou neve. Quando o clima está seco a saída do sensor fica em estado alto e quando há uma gota de chuva em estado baixo. O limite entre tempo seco e chuva pode ser ajustado através do potenciômetro presente no sensor que regulará a saída digital D0. Contudo para ter uma resolução melhor é possível utilizar a saída analógica A0 e conectar a um conversor AD, como a presente no Arduino por exemplo. A placa do Sensor de Chuva é revestida em ambos os lados com um tratamento de níquel contra oxidação, melhorando assim a condutividade, desempenho e duração.



Figura 20 - Módulo sensor de chuva
Fonte: FILIPEFLOP, 2020

7.2.4 Sensor de Gás Inflamável e Fumaça

O Detector de Gás / Sensor de Gás Inflamável e Fumaça MQ-2 é capaz de detectar gases de diferentes tipos, entre eles, gás de petróleo liquefeito, butano, propano, metano, hidrogênio, álcool, gás natural, entre outros, ou até mesmo fumaça. Quando o Sensor de Gás Inflamável e Fumaça MQ-2 detectar a presença do gás ou da fumaça, liberará um sinal de aviso ao Arduino, se for o caso, que estará programado para realizar diferentes tarefas, como por exemplo, liberar trancas de portas, emitir sinais sonoros e luminosos, etc.

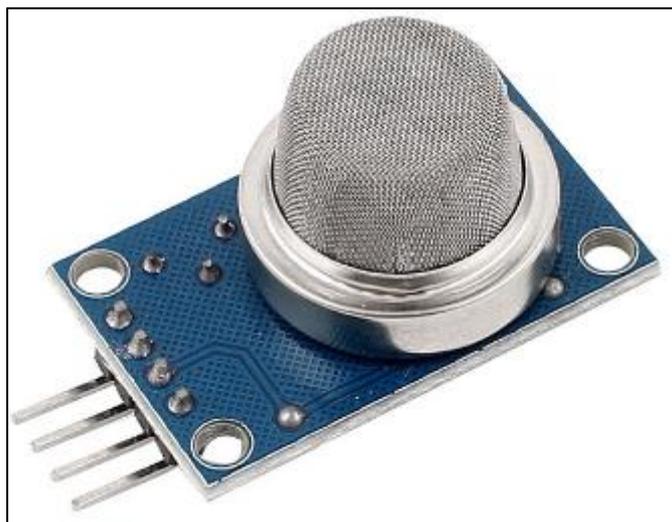


Figura 21 - Sensor de gás e fumaça
Fonte: FILIPEFLOP, 2020

7.2.5 Sensor de Temperatura LM35DZ

O Sensor de Temperatura LM35DZ é preciso, além de ter uma fácil comunicação com microcontroladores como Arduino, PIC, ARM e Raspberry_Pi. Muito usado para projetos de automação residencial ou até mesmo industrial.

A sua comunicação com o microcontrolador utiliza apenas um pino. Saída de tensão proporcional à temperatura. Possui faixa de leitura de 0°C a 100°C e precisão de 0,5°C.

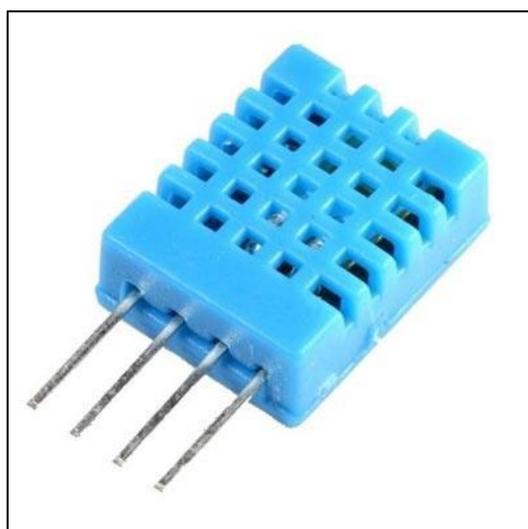


Figura 22 - Sensor de Temperatura
Fonte: FILIPEFLOP, 2020

7.3 ESTRUTURA DA APLICAÇÃO

Neste tópico será apresentado o processo realizado para finalizar o protótipo Arduino, juntamente com a aplicação mobile utilizando-se do Node.js e React Native.

7.3.1 Configuração Arduino

Para que todos os sensores pudessem ser utilizados na placa do arduino, o uso de uma protoboard foi indispensável por conta da quantidade de portas limitadas que a placa fornece.

A figura 23 representa um esboço das conexões entre o Arduino e os sensores para que os mesmos trabalhem em conjunto.

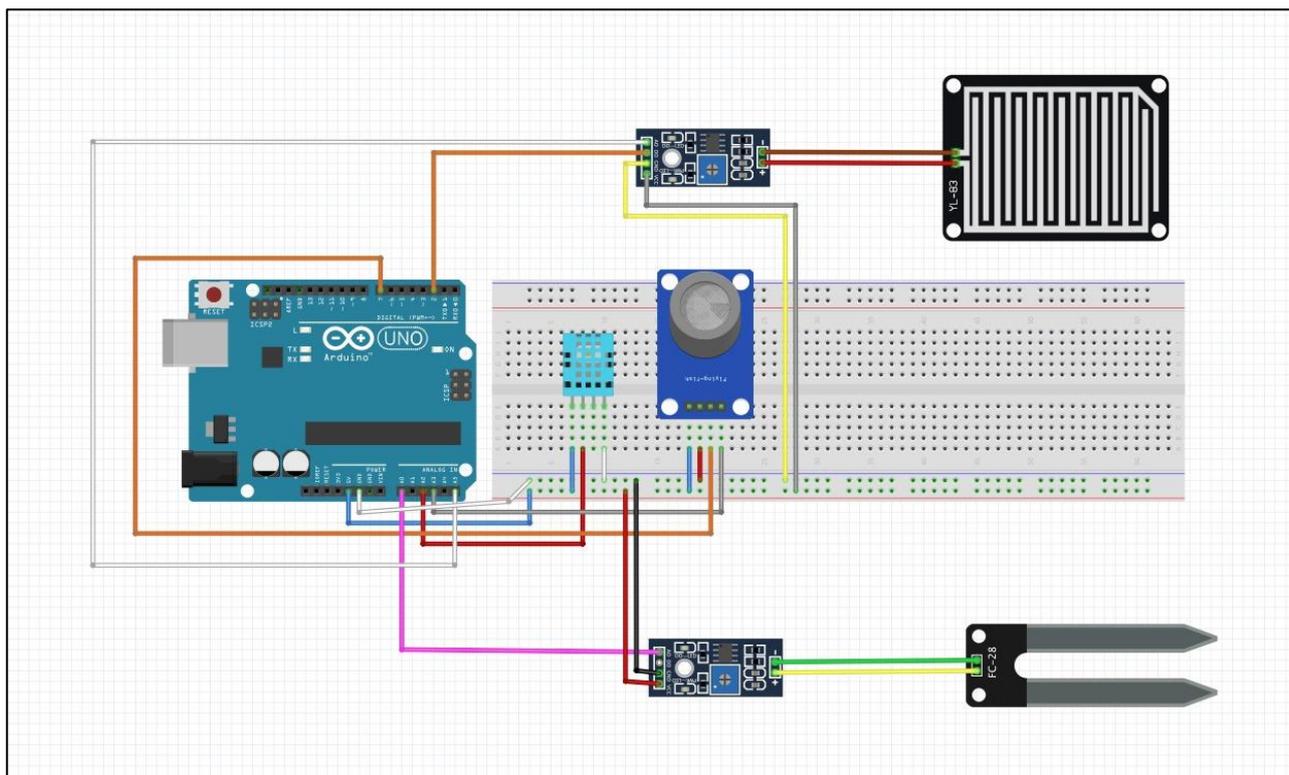


Figura 23 - Conexões Arduino e Sensores
Fonte: VICCIOLI, 2020

Realizado as conexões na placa Arduino, necessita-se carregar o código na memória, o código desenvolvido para captar o sinal dos sensores.

Para realizar a leitura das portas no qual os pinos sensoriais estão conectados, é necessário informar os mesmos na IDE do Arduino, em seguida, o valor recebido é exibido em forma de um print no monitor serial da aplicação.

O código que foi carregado na memória do Arduino para realização do protótipo é apresentado na figura 24, juntamente com os comentários em cada trecho.

```

1  #include <dht.h>
2  int pin_d0 = 7; //Sensor gás
3  int pin_a0 = A3; //Sensor temperatura
4  int nivel_sensor = 400; //Sensor gás
5  int pino_d = 2; //Sensor de Chuva
6  int pino_a = A5; //Sensor de Chuva
7  int val_a = 0; //Armazena o valor lido do pino analogico
8  const int pinoDHT11 = A2; //PINO ANALÓGICO UTILIZADO PELO DHT11
9
10 dht DHT; //VARIÁVEL DO TIPO DHT
11
12 void setup() {
13   Serial.begin(9600); //INICIALIZA A SERIAL
14   // Define os pinos de leitura do sensor de gas como entrada
15   pinMode(pin_d0, INPUT);
16   pinMode(pin_a0, INPUT);
17   delay(2000); //INTERVALO DE 2 SEGUNDO ANTES DE INICIAR
18 }
19
20 void loop() {
21   //Sensor de Gás
22   int valor_digital = digitalRead(pin_d0);
23   int valor_analogico = analogRead(pin_a0);
24   Serial.print("Valor sensor : ");
25   Serial.println(valor_analogico);
26
27   if (valor_analogico > nivel_sensor) {
28     Serial.println("Sinal de gas "); }
29   else {
30     Serial.println("Sem sinal de gas "); }
31
32   int sensorValue = analogRead(A0);
33
34   //sensor chuva
35   val_a = analogRead(pino_a);
36   // Compara valores do sensor de chuva
37   if (val_a >0 && val_a <149) {
38     Serial.println("Chovendo "); }
39   if (val_a > 150 && val_a <400) {
40     Serial.println("Sem chuva"); }
41
42   //Sensor umidade do solo
43   if (sensorValue > 600) {
44     Serial.println("Solo está seco!!!"); }
45   else{
46     Serial.println("Solo está molhado!"); }
47
48   //Sensor Umidade do Ar e Temperatura
49   DHT.read11(pinoDHT11);
50   Serial.print("Umidade: ");
51   Serial.print(DHT.humidity);
52   Serial.print("%");
53   Serial.print(" / Temperatura: ");
54   Serial.print(DHT.temperature, 0);
55   Serial.println("°C");
56   Serial.println("-----");
57   delay(5000);
58 }

```

Figura 24 - Código do arduino
Fonte: VICCIOLI, 2020

7.3.2 Utilizando o Node.js

Para realizar o envio das informações captadas pelos sensores através da plataforma Arduino, foi necessário passar as informações por um servidor, nesse momento o Node.js assume a responsabilidade, esse servidor Node.js é startado em uma máquina local.

Para que o Arduino consiga se comunicar com o Node.js, utiliza-se de uma biblioteca chamada SerialPort. Essa biblioteca é responsável por disponibilizar a conexão e transmissão de dados através da porta serial (USB) que o Arduino está conectado juntamente com os sensores que estão captando os sinais.

Inicialmente necessita-se instanciar um novo objeto SerialPort informando a entrada USB no qual a placa Arduino está conectada, e fornecer a taxa de transmissão configurada no Arduino, conforme a figura 25.

```
const port = new SerialPort("COM3", {  
  |   baudRate: 9600  
});
```

Figura 25 - Instanciação da SerialPort
Fonte: VICCIOLI, 2020

Realizado a instanciação, precisa-se abrir a porta para que o servidor comece a receber as informações que são captadas pelos sensores do Arduino. A figura 26 representa o trecho no qual essa função é realizada.

```
port.on("open", function() {  
  |   console.log("Porta aberta");  
});
```

Figura 26 - Abrir porta do servidor
Fonte: VICCIOLI, 2020

Para que seja possível receber estas informações, utiliza-se de um atributo que capta cada linha recebida e delimita quando encontra um /n, são estes que indicam o fim de um objeto recebido contendo as informações dos sensores, necessitando informar o nome que será utilizado, denominado então como "data".

As informações são recebidas com características de um objeto, no entanto, precisam ser convertidas para o formato JSON, conforme mostra a figura a seguir, para que os atributos contidos dentro delas possam ser manipulados.

```
const parser = port.pipe(new Readline({ delimiter: "\n" }));
parser.on("data", function(data) {
  let dados = JSON.parse(data);
  console.log(dados);

  socket.emit("arduino", data);
});
```

Figura 27 - Convertendo objetos em JSON
Fonte: VICCIOLI, 2020

Ao chegar no comando “console.log”, o terminal passa a imprimir as informações dos sensores conectados à placa Arduino, conforme a Figura 28.

```
{ gas: false, chuva: false, solo: 1007, umidade: 54, temperatura: 28 }
{ gas: false, chuva: false, solo: 1007, umidade: 54, temperatura: 28 }
{ gas: false, chuva: false, solo: 1007, umidade: 54, temperatura: 28 }
{ gas: false, chuva: false, solo: 1007, umidade: 54, temperatura: 28 }
{ gas: false, chuva: false, solo: 1007, umidade: 54, temperatura: 28 }
{ gas: false, chuva: false, solo: 1007, umidade: 54, temperatura: 28 }
{ gas: false, chuva: false, solo: 1007, umidade: 54, temperatura: 28 }
{ gas: false, chuva: false, solo: 1007, umidade: 54, temperatura: 28 }
```

Figura 28 - Console.log
Fonte: VICCIOLI, 2020

Estas informações são enviadas para o servidor a cada cinco segundos, conforme configuradas anteriormente.

Recebido essas informações no Node.js, as mesmas são transmitidas para uma biblioteca chamada socket.io. Essa biblioteca é importada tanto no servidor Node.js quanto no React Native, ela permite o envio e recebimento de informações em tempo real, sem a necessidade da passagem de informações por um banco de dados.

Essas informações são enviadas através da variável Dados, quem contém o objeto com o valores dos sensores para a biblioteca socket.io já convertidos no formato JSON, para

isso, utiliza-se do comando `io.on`, que recebe um atributo `connection` captando as informações do usuário que está conectado na outra ponta do socket (React Native / Cliente), sendo assim, possível visualizar o id do usuário conectado, utilizando o `console.log` e passando pela variável `socket.id`, conforme a figura abaixo.

```
io.on("connection", socket => {  
  console.log("Usuario conectado", socket.id);  
});
```

Figura 29 - Socket.id
Fonte: VICCIOLI, 2020

Realizadas as conexões, utiliza-se do comando `emit` para enviar a variável `data` no qual contém as informações, para quem está aguardando os dados que contenham o nome `arduino` no React Native.

```
socket.emit("arduino", data);  
});
```

Figura 30 - Socket.emit
Fonte: VICCIOLI, 2020

7.3.3 Utilizando o React Native

Para receber as informações enviadas pelo servidor Node.js e exibi-las no aplicativo, precisa-se instalar a biblioteca do socket.io modo cliente, configurando então uma constante com a biblioteca e passando o endereço do servidor, conforme a figura abaixo.

```
const socket = socketio("http:192.168.0.106//:3333");
```

Figura 31 - Endereço do servidor
Fonte: VICCIOLI, 2020

Para que começar a receber as informações do servidor, precisa-se utilizar a constante que esta com o endereço do servidor e coloca-la para ouvir utilizando o `.io`, feito isso, informa-se o nome do socket que se deseja receber, colocando o nome de arduino, o

mesmo utilizado no `socket.emit` do servidor, conforme a figura 30, e em seguida, utiliza-se a variável `data` para receber as informações, através do IP informado na constante `socket`, é possível que o React Native consiga buscar os dados apresentados pelo servidor para tratá-los.

Recebido as informações, é necessário convertê-las para o formato JSON, para que seja possível manipular cada objeto nela contida. Cada atributo recebido pelo aplicativo será mostrado em tela, por isto, aguarda-se cada informação em seu respectivo hook. Os hooks são funções que permitem usar o estado e métodos do ciclo de vida de um componente de classe em um componente funcional.

O aplicativo foi configurado para exibir um alerta na tela, conforme a figura 32, informando que o sensor de gás captou alguma alteração no sinal.

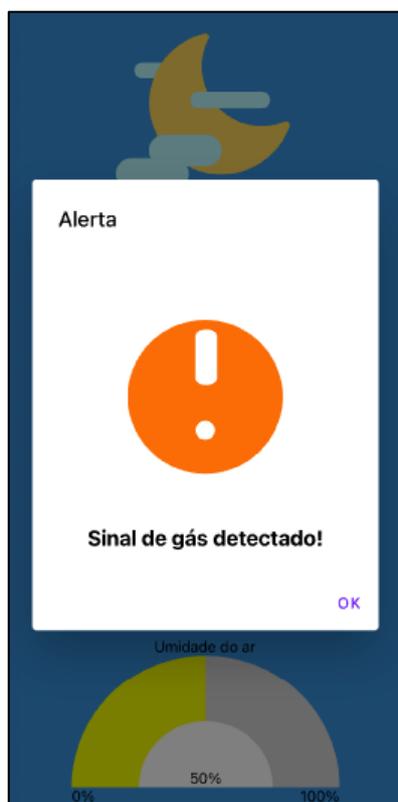


Figura 32 - Sinal de gás detectado.
Fonte: VICCIOLI, 2020

A partir de um `if` verificador o sensor de gás retorna se está captando algo, sendo `true`, um comando é enviado, para que o alerta seja mostrado em tela, essa configuração é exibida na imagem abaixo entre as linhas 40 e 42 da figura 33.

```
33 socket.on("arduino", data => {
34   setinfo2(data);
35   let dados = JSON.parse(data);
36   setchuva(dados.chuva);
37   setsolo(dados.solo);
38   setumidade(dados.umidade);
39   settemperatura(dados.temperatura);
40   if (dados.gas === true) {
41     | showDialog();
42   }
43   if (dados.chuva === false && dados.solo > 900) {
44     | setatentsolo("flex");
45   } else {
46     | setatentsolo("none");
47   }
```

Figura 33 – Configurações de alerta

Fonte: VICCIOLI, 2020

Quando o solo está seco e não existem indícios de chuva também é exibido uma mensagem na tela da aplicação, para isso, se realiza uma verificação utilizando as informações dos sensores de chuva e umidade de solo, verificado que os fatores são verdadeiros, então a variável Hook é setada como flex, pois utiliza um componente em css chamado “display” que recebendo flex é renderizado em tela, se o mesmo receber a propriedade “none”, então não é renderizado em tela, isso ocorre quando ele executa o “else” entre as linhas 45 a 47 da figura 33.



Figura 34 - Alerta de irrigação
Fonte: VICCIOLI, 2020

Utiliza-se da biblioteca Lottie para criar animações em JSON, as mesmas são baixadas do próprio site da lottie, havendo três variações na aplicação. Sol, sendo ela durante o dia até as 19h. Lua, quando o horário ultrapassa as 19h ou chuva, quando o sensor capta sinal de água.



Figura 35 – Animações da aplicação
Fonte: VICCIOLI, 2020

Este recurso faz uso de um IF ternário, que verifica a condição da variável e renderiza em tela, conforme a imagem 36.

```

<View style={styles.clima}>
  {chuva === true ? (
    <LottieView resizeMode="contain" source={chuvaa} autoPlay />
  ) : null}
  {chuva === false && horario < "19" ? (
    <LottieView resizeMode="contain" source={sol} autoPlay />
  ) : null}
  {chuva === false && horario >= "19" ? (
    <LottieView resizeMode="contain" source={lua} autoPlay />
  ) : null}
</View>

```

Figura 36 - If ternário das animações
Fonte: VICCIOLI, 2020

Quando o sensor de chuva capta água na placa, a informação de que está chovendo passa a ser true, realizando a alteração da animação que antes estava uma lua, mostrando que no momento está chovendo, como ilustra a figura 37.

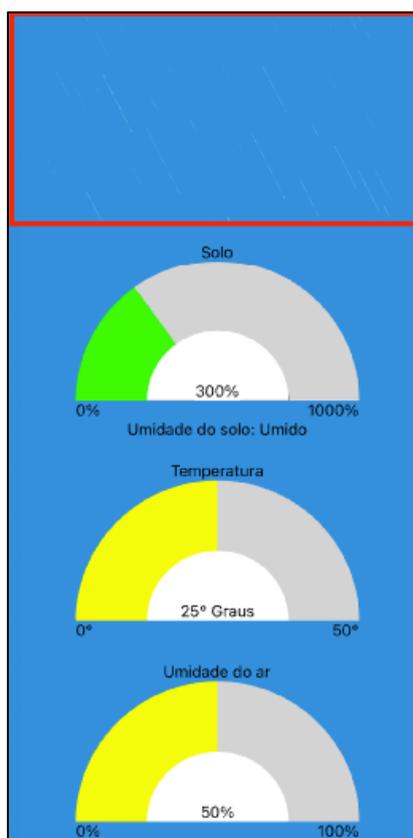


Figura 37 - Demonstração de chuva na aplicação
Fonte: VICCIOLI, 2020

A captação de sinais de temperatura e de umidade do ar são juntas, pois ambos fazem parte de apenas um sensor, o DHT11, mas neste caso, os resultados obtidos por ele são

divididos em dois objetos distintos, o `data.temperatura` e o `data.umidade`, estas informações são apresentadas em um gráfico de meia rosca, conforme exemplificação da figura abaixo.

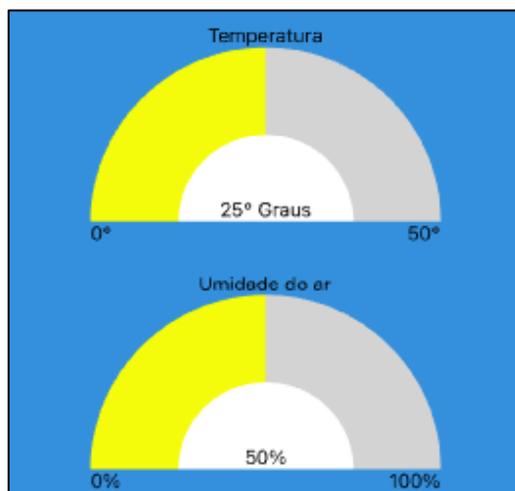


Figura 38 - Gráfico temperatura e umidade do ar
Fonte: VICCIOLI, 2020

As cores dos gráficos realizam variação de acordo com a informação apresentada, essas cores representam a intensidade de que cada sensor está captando informações, como por exemplo, a temperatura, se for de 0 a 20 graus, o gráfico se exibirá na cor verde, de 21 a 25 graus, a cor se altera para amarelo, sendo acima disso, a cor apresentada será o vermelho, como apresenta a figura 39.

```
temperatura <= 20
? "#13fc03"
: temperatura <= 25
? "#f4fc03"
: "#ff0000"
```

Figura 39 - Configuração das cores no gráfico.
Fonte: VICCIOLI, 2020

Realizando então estes procedimentos, a placa do Arduino irá captar os sinais dos sensores, enviar para o servidor Noje.js que por sua vez transmitirá as informações ao React Native apresentando os dados captados dos sensores na aplicação mobile.

8. CONCLUSÃO

A tecnologia na área agrícola tende a crescer cada vez mais, por conta dos seus dados mais precisos colaborando para que sejam evitados desperdícios e prejuízos. O desenvolvimento do protótipo proposto ocorreu da forma esperada. Conclui-se que o Arduino é uma plataforma de baixo custo, porém com estrutura ideal para estudos e criação de protótipos em nível inicial, agregado com sensores, é capaz de trazer infinitas possibilidades tecnológicas, não só na área agrícola.

Fica explícito que o Arduino é uma plataforma de fácil implementação, mesmo para quem não tem um conhecimento avançado em programação e eletrônica. Atualmente os materiais encontrados são bem explicativos e intuitivos se tratando deste assunto.

Os resultados obtidos ao captar os sinais dos sensores e em seguida transmitir a API por conta das tecnologias Node.js e React Native foram satisfatórios. A necessidade de realizar uma pesquisa e estudar sobre os componentes utilizados, no qual ainda não havia sido utilizado até a realização desta conclusão agregou muito como conhecimento.

Desta forma, pode-se afirmar que este trabalho atende aos objetivos estabelecidos propostos ainda no início do mesmo, estudar o Node.js, React Native e a plataforma Arduino, realizando assim uma interação entre os três, captando os sinais dos sensores e transmitindo a aplicação mobile em tempo real.

8.1 TRABALHOS FUTUROS

Utilizando ainda a plataforma Arduino, pretende-se dar continuidade no protótipo agregando com outros sensores para obter um monitoramento mais avançado.

Com a realização de uma coleta de dados mais ampla, pretende-se integrar uma moto bomba, para que o solo passe a ser irrigado automaticamente de acordo com os sinais captados do sensor de umidade de solo e chuva.

E para que haja uma melhor comunicação, implantar uma base Shield WiFi para permitir o monitoramento a distância.

REFERÊNCIAS

AGRÍCOLA. Jacto. Agricultura 4.0 modifica e otimiza as etapas do ciclo produtivo. Disponível em < <https://g1.globo.com/sp/bauru-marilia/especial-publicitario/jacto-agricola/noticia/2018/11/23/agricultura-40-modifica-e-otimiza-as-etapas-do-ciclo-produtivo.ghtml>>. Acesso em: 16 mar. 2019

ANDRADE. Joelson. Como funciona a internet das coisas e como pode influenciar seu negócio. Disponível em < <http://www.getcard.com.br/novo/como-funciona-a-internet-das-coisas-e-como-pode-influenciar-seu-negocio/>>. Acesso em: 12 nov. 2018

ARDUINO. Arduino. Disponível em <<https://store.arduino.cc/>>. Acesso em: 29 out. 2018.

Banzi, M. (2010). *Primeiros passos com o arduino* (Primeira ed.). São Paulo, SP, Brasil: Novatec Editora Ltda.

BARBOSA, José Wilian. Sistema de Irrigação Automatizado utilizando a plataforma Arduino. 2013. 57p. Dissertação (Bacharel) - Fundação Educacional do Município de Assis – FEMA - Assis, São Paulo, 2013.

BECKER. Lauro. O que é React Native? Disponível em <<https://www.organicadigital.com/blog/o-que-e-react-native/>>. Acesso em: 27 ago. 2020.

CAMPOS, R. A. F.; Automação residencial utilizando Arduino e aplicação web. Disponível em: <https://repositorio.uniceub.br/jspui/bitstream/235/5461/1/Monografia_Roberto.pdf>. Acesso em: 06 mar. 2019.

DEVELOPERS. Conheça o Android Studio. Disponível em: <<https://developer.android.com/studio/intro/?hl=pt-br>>. Acesso em: 08 mar. 2019.

DIGITAL, Agricultura. Para alimentar o mundo, é preciso trazer inovação para a agricultura. Disponível em <<https://agrosmart.com.br/blog/agricultura-digital/alimentar-o-mundo-trazer-inovacao-para-agricultura/>>. Acesso em: 18 nov. 2018.

DINO. Os impactos da tecnologia no campo, com a agricultura 4.0. Disponível em <<https://exame.abril.com.br/negocios/dino/os-impactos-da-tecnologia-no-campo-com-a-agricultura-40/>>. Acesso em: 8 out. 2018.

FAPESP. Agricultura 4.0. Disponível em: <<https://revistapesquisa.fapesp.br/2020/01/02/agricultura-4-0/>>. Acesso em: 7 mar. 2020.

FONSECA, Erika Guimarães Pereira da; BEPPU, Mathyan Motta. Apostila Arduino. Niterói-RJ: Universidade Federal Fluminense Centro Tecnológico, 2010. 23 p.

FUENTES, Prof. Rodrigo C. Apostila de Automação Industrial. 2005. 31 f. - Curso de Eletrotécnica, Universidade Federal De Santa Maria Colégio Técnico Industrial De Santa Maria, Santa Maria - RS, 2005.

GIANTOMASO, Isabela. O que é internet das coisas? Dez fatos que você precisa saber sobre IoT. Disponível em <<https://www.techtudo.com.br/listas/2018/08/o-que-e-internet-das-coisas-dez-coisas-que-voce-precisa-saber-sobre-iot.ghtml>>. Acesso em 15 mar. 2019.

GOGONI, Ronaldo. O que é internet das coisas?. Disponível em <<https://tecnoblog.net/263907/o-que-e-internet-das-coisas/>>. Acesso em 15 mar. 2019.

GOMES, Renato. Tecnologia Agrícola: A Importância e principais inovações. Disponível em <<https://pixforce.com.br/tecnologia-agricola/>>. Acesso em: 7 out. 2018.

GOVENDER, Sashen. Introdução ao Android Studio. Disponível em: <<https://code.tutsplus.com/pt/tutorials/getting-started-with-android-studio--mobile-22958>>. Acesso em: 08 mar. 2019.

HACHOUCHE. Anwar. Apostila Arduino Básico V1.0. Disponível em <http://apostilas.eletrogate.com/Apostila_Arduino_Basico-V1.0-Eletrogate.pdf>. Acesso em: 15 fev. 2019.

HANASHIRO, Akira. O que é DOM, Virtual DOM e Shadow DOM?. Disponível em <<https://www.treinaweb.com.br/blog/o-que-e-dom-virtual-dom-e-shadow-dom/>>. Acesso em: 16 out. 2020.

HUIDOBRO, José Manuel; NOVEL, Beatriz; CALAFAT, Crithian; SULLER, Eduardo; ESCUDERO, Adrian Nogales; TOLEDANO, José Carlos; SANTAMARÍA, Asunción; LASTRES, Carmen. La Domótica Como Solución De Futuro. Madri: Fenercom, 2007.

JACTO. Agricultura 4.0: tudo o que você precisa saber. Disponível em <<https://blog.jacto.com.br/agricultura-4-0-tudo-o-que-voce-precisa-saber/>>. Acesso em 16 mar. 2019

LEITE, F. H. et al. Petvitae: Ferramenta de auxílio ao acompanhamento e gestão de grupos pet. 2018. Citado na página 27.

MARKETING. Tecnologia na Agricultura: Conheça as 6 principais inovações. Disponível em <<http://www.superbac.com.br/tecnologia-na-agricultura-conheca-as-6-principais-inovacoes/>>. Acesso em: 7 out. 2018.

MARQUES, Jheimis Fernandes. Protótipo de projeto de Internet of Things para a coleta de dados em Agricultura Digital 2018. 44p. Dissertação (Bacharel) - Fundação Educacional do Município de Assis – FEMA – Assis, São Paulo, 2018.

MASSRUHÁ, Silvia Maria Fonseca Silveira. O papel na agricultura. AgroANALYSIS, v. 35, n. 9, p. 29-31, 2015.

MOLIN, José Paulo; DO AMARAL, Lucas Rios; COLAÇO, André. Agricultura de precisão. Oficina de Textos, 2015.

PAZETO, Marcos. Lavoura *high tech*: A importância do uso da tecnologia no agronegócio. Disponível em: <<https://computerworld.com.br/2017/12/08/lavoura-high-tech-importancia-do-uso-da-tecnologia-no-agronegocio/>>. Acesso em: 17 nov. 2018.

PROJECT.TI. Node.js: O que é, características e vantagens. Disponível em <<http://projectti.com.br/node-js-o-que-e-caracteristicas-e-vantagens/>>. Acesso em: 09 out. 2020.

SOUZA, Fábio. Arduino – Primeiros Passos. Disponível em: <<https://www.embarcados.com.br/arduino/>>. Acesso em: 20 fev. 2019.

THOMAZINI, Daniel; ALBUQUERQUE, Pedro Urbano Braga de. Sensores Industriais: Fundamentos e Aplicações. Érica.2009. 221 p.

OLIVEIRA, Cláudio Luís Vieira; ZANETTI, Augusto Piovesana. Arduino Descomplicado - Como elaborar projetos de eletrônica. Érica. 2015. 288 p.

OPSERVICES. Entenda o que é Arduino e como funciona a sua aplicação! Disponível em: <<https://www.opservices.com.br/o-que-e-o-arduino/>>. Acesso em: 8 out. 2018.

REIS, Fábio dos. O que são sensores. Disponível em: <<http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/o-que-sao-sensores>>. Acesso em: 05 nov. 2020.

ROMERO, B. A.; SAAD, E. F.; BASTOS, G. B. React-native, avaliação de performance comparando com código java nativo com listview. JORNAL DE ENGENHARIA, TECNOLOGIA E MEIO AMBIENTE-JETMA, v. 2, n. 2, p. 23, 2018. Citado na página 27.

ZIM, Joe. Conectando no Socket.IO – o básico. Disponível em <<https://imasters.com.br/devsecops/conectando-no-socket-io-o-basico>>. Acesso em: 10 out. 2020.