



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

JOÃO GUILHERME FERNANDES

**OPEN DATA SCIENCE APLICADA À AGRICULTURA INTELIGENTE
E PROJETOS IOT**

**Assis/SP
2020**



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

JOÃO GUILHERME FERNANDES

**OPEN DATA SCIENCE APLICADA À AGRICULTURA INTELIGENTE
E PROJETOS IOT**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão

Orientando: João Guilherme Fernandes

Orientador: Prof. MSc. Guilherme de Cleve Farto

Assis/SP

2020

FICHA CATALOGRÁFICA

F363o FERNANDES, João Guilherme

Open data science aplicada à agricultura inteligente e projetos
iot / João Guilherme Fernandes. – Assis, 2020.

45p.

Trabalho de conclusão do curso (Ciência da Computação). –
Fundação Educacional do Município de Assis-FEMA

Orientador: Me. Guilherme de Cleva Farto

1.Agricultura inteligente 2.Data science 3.Open data

CDD001.61

OPEN DATA SCIENCE APLICADA À AGRICULTURA INTELIGENTE E PROJETOS IOT

JOÃO GUILHERME FERNANDES

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 Cleva Farto

Avaliador: Prof. Esp. Célio Desiró

Assis/SP
2020

RESUMO

Devido ao crescimento em ritmo acelerado da população mundial, conforme estimativas apresentadas em relatório pela Organização das Nações Unidas no segundo semestre de 2019, além de ser ressaltado o aumento do consumo de alimentos, é estimado que o processo da agricultura em geral seja reformulado de modo que a perda no processo de plantio seja cada vez menor e, por conseguinte, apresente um maior grau de rentabilidade. Além disso, a grande dificuldade encontrada pelo agricultor para realizar o monitoramento de sua cultura compromete o processo de aumento de rentabilidade do plantio; isso ocorre pela falta de ferramentas tecnológicas utilizadas em campo. Tendo em vista essa questão, a proposta deste trabalho consiste em apresentar uma plataforma de monitoramento de plantio de simples utilização, consolidada a partir de um protótipo alinhado ao conceito de *Smart Agriculture*, voltado para o monitoramento de plantio; ou seja, um projeto IoT com a finalidade de ajudar agricultores a monitorarem o plantio de forma rápida e inteligente, diminuindo perdas no campo e aumentando a rentabilidade na produção de alimentos.

Palavras-chave: Internet das Coisas, Agricultura Inteligente, Data Science, Open Data.

ABSTRACT

Due to the increase in the pace of the world's population, according to estimates in a report by the United Nations in the second semester of 2019 at the latest, in addition, there is an increase in the consumption of the food, it is estimated that the process of agriculture, in general, is to be redesigned so that the loss in the process of planting, growing smaller and, therefore, to provide a greater degree of cost-effectiveness.

In addition, the main difficulty encountered by an individual farmer to carry out the monitoring of their culture that is committed to the process and to increase the profitability of the plantation; and it occurs due to a lack of the technological tools that are used in the field. In view of this question, the purpose of this work is to provide a platform for the monitoring of the planting of the easy-to-use, consolidated from the prototype in line with the concept of *Smart Agriculture* aimed to the monitoring of the planting; in other words, a project of the IoT, with the aim of helping farmers monitor growing quickly and intelligently, reducing losses in the field and increasing the yield in the production of food.

Keywords: Internet of Things, Smart Agriculture, Data Science, Open Data.

LISTA DE ILUSTRAÇÕES

Figura 1 - Gráfico Dispositivos Percapita.....	16
Figura 2 - Arduino.....	17
Figura 3 - Raspberry Pi 4.....	19
Figura 4 - Gráfico Aumento de Dados	22
Figura 5 - Logo Python	23
Figura 6 - Logo Mongo DB	24
Figura 7 - Logo PostgreSQL.....	24
Figura 8 - Arquitetura Geral	29
Figura 9 - Design da API	29
Figura 10 - Arquitetura do Sistema de Captura de Dados	35
Figura 11 - Figura de Login	36
Figura 12 - Tela Home.....	37
Figura 13 - Tela Relatório	39
Figura 14 - Tabelas do Relatório	40
Figura 15 - Gráfico de Temperatura	40
Figura 16 - Gráfico de Umidade	41
Figura 17 - Gráfico UV.....	41

SUMÁRIO

1 INTRODUÇÃO	10
1.1 OBJETIVO.....	11
1.1.1 Objetivo Geral.....	11
1.1.2 Objetivo Específico	12
1.2 JUSTIFICATIVAS	12
1.3 MOTIVAÇÃO.....	13
1.4 PERSPECTIVAS DE CONTRIBUIÇÃO	13
1.5 METODOLOGIA DE PESQUISA.....	13
1.6 RECURSOS NECESSÁRIOS	14
1.7 ESTRUTURA DO TRABALHO	14
2 INTERNET OF THINGS	16
2.1 ARDUINO	17
2.1.1 Sensores	18
2.2 Raspberry	19
2.2.1 Arquitetura	20
2.2.2 Sistemas Suportados.....	20
3 OPEN DATA SCIENCE	22
3.1 TECNOLOGIAS OPEN SOURCE	23
3.1.1 Python	23
3.1.2 Mongo DB	23
3.1.3 PostgreSQL.....	24
4 AGRICULTURA INTELIGENTE	26
4.1 APLICAÇÕES EM AGRICULTURA INTELIGENTE	27
5 PROPOSTA DE TRABALHO	28
5.1 SMART AGRICULTURE	28
5.2 RECURSOS ADOTADOS	28

5.3 DESENVOLVIMENTO DO TRABALHO	29
5.3.1 Flask API	30
5.3.2 Sistema de Captura de Dados.....	35
5.3.3 Sistema Agro Data.....	36
6 CONCLUSÃO	42
6.1 TRABALHOS FUTUROS.....	43
7 REFERÊNCIAS	44

1. INTRODUÇÃO

Com a ascensão da indústria 4.0, surgiu um novo termo denominado *IoT* ou, *Internet of Things*. De uma forma geral, entende-se por *IoT* a maneira com que objetos físicos do nosso cotidiano se interconectam na web; muitos destes equipamentos estão incorporados com inteligência ubíqua e são controlados pela Internet (COSTA, OLIVEIRA e MÓTA, 2018). Segundo a empresa norte-americana de assessoria tecnológica *Gartner* (GARTNER, 2015), em 2020 haverá mais de 50 bilhões de dispositivos *IoT* conectados à internet, sendo que uma boa parte desse número será voltada para aplicações de agricultura. Como exemplo disso, vale a pena mencionar que “[...] mais de 75 milhões de dispositivos *IoT* serão instalados (para agricultura) até o final desta década - um aumento de 150% em relação aos 30 milhões em 2015” (FAKHRUDDIN, 2017).

Além do mundo IOT, há outros dispositivos que recebem destaque como, por exemplo, o dispositivo *Raspberry*. Desenvolvido no Reino Unido pela fundação *Raspberry Pi*, tal dispositivo é um computador de baixo custo que apresenta o tamanho de um cartão de crédito. Para usá-lo, basta plugar um teclado e um mouse padrão a ele e conectá-los a um monitor ou a uma televisão (CIRIACO, 2015).

Outro dispositivo de destaque é o *Arduino*, sendo este uma plataforma *open source* de prototipagem eletrônica com hardware e software flexíveis e fáceis de usar, destinado a artistas, designers, hobbistas e qualquer pessoa interessada em criar objetos ou ambientes interativos (CIRIACO, 2015).

Com todos esses sensores, a quantidade de dados gerados é considerada exorbitante, já que “[...] o volume médio de *big data* gerado e gerenciado por *farms* individuais também mostrará um aumento impressionante entre 2017 (<0,5 milhão de pontos de dados) e 2050 (> 4,0 milhões de pontos de dados)” (FAKHRUDDIN, 2017). É importante ressaltar que a agricultura inteligente¹ tem sua eficácia assegurada a partir de sua habilidade de monitorar e unir dados meteorológicos angariados por meio do acompanhamento das condições climáticas durante toda a gestação obtida da

¹ A agricultura inteligente é considerada um sistema eficaz e de contribuição significativa para o agricultor no processo de análise das principais condições de produção.

coleta e análise de dados de estações meteorológicas e, também, de sensores presentes no meio do plantio. Tais sensores captam informações como: umidade do ar, umidade do solo, raio UV, entre outros.

Com o uso da agricultura inteligente, torna-se possível obter o controle da atuação da terra, bem como o desenvolvimento das plantações e da disponibilidade regular de água na região da vegetação, de modo que o acesso seguro a tais informações contribui significativamente para a minimização de perdas na produção (AGSOLVE, 2017).

Para realizar a análise de dados, é necessário o uso de algumas ferramentas de desenvolvimento como, por exemplo, a linguagem Python. Embora Python não tenha um conjunto de pacotes e bibliotecas tão abrangente quanto os disponíveis para a linguagem R, a combinação de Python com ferramentas como *Pandas*, *Numpy*, *Scipy*, *Scikit-learn* e *Seaborn* torna a linguagem uma das principais escolhas entre os Cientistas de Dados.

A possibilidade de se criar uma plataforma que garanta a captação de dados de uma determinada plantação e, que realize a análise de dados e de *insights* ao usuário baseado na plataforma *Arduino* e *Raspberry*, permite usufruir de todos os benefícios que o mundo *open source* oferece, possibilitando a criação de uma plataforma segura e de baixo custo.

1.1 OBJETIVOS

1.1.1 Objetivo geral

O objetivo geral do projeto consiste em pesquisar sobre conceitos *IoT* e *Data Science* voltados para a agricultura inteligente, bem como ferramentas e arquiteturas que auxiliem na construção da plataforma proposta.

A partir dos artefatos deste projeto, será possível desenvolver uma plataforma de monitoramento de plantio utilizando conceitos de *IoT*, agricultura inteligente e *Data Science*, os quais irão colaborar com o usuário no que se refere a informações acessadas em tempo real e que automatize processos de monitoramento da lavoura.

1.1.2 Objetivo Específico

Este trabalho pretende desenvolver uma plataforma de monitoramento de plantio, utilizando tecnologias e dispositivos *Open Source*. Para que a elaboração e a execução das etapas teóricas e práticas se tornem possíveis, foram instituídos os seguintes objetivos:

- Pesquisar e analisar protocolos de comunicação como *wi-fi* e *Bluetooth*;
- Pesquisar e analisar a plataforma *Raspberry*;
- Pesquisar e analisar a plataforma *Arduino*;
- Pesquisar e analisar o desenvolvimento de *WebServices*, utilizando *Python*;
- Pesquisar e analisar o desenvolvimento de páginas *Web* com *React*;
- Pesquisar e analisar técnicas de análise de dados utilizando *Python*;
- Pesquisar e analisar a plataforma *firebase*;
- Pesquisar e analisar o funcionamento de sensores voltados à agricultura.

1.2 JUSTIFICATIVA

Considerando o crescimento acelerado da população mundial e, conseqüentemente, a pressão em torno da necessidade de se produzir mais alimentos com altos padrões de qualidade, verifica-se que tal crescimento exige a presença da tecnologia no mercado agrícola global. De acordo com uma matéria publicada no site (BOAS PRATICAS AGRONÔMICAS, 2019), aplicativos via *smartphone*, computadores ou *tablets* têm sido utilizados com o propósito de auxiliar no monitoramento de plantações remotas, o que confirma a contribuição significativa da tecnologia no cenário agrícola.

Vale a pena ressaltar que esse mecanismo tecnológico é acionado devido ao auxílio promovido pelos sensores instalados na propriedade monitorada, os quais são conectados à internet, como os utilizados nas plataformas do *Raspberry* e *Arduino*.

1.3 MOTIVAÇÃO

O desenvolvimento deste projeto de pesquisa é oportuno, visto que a agricultura inteligente e *IoT* são temas explorados com frequência na comunidade científica devido às contribuições que ambas têm proporcionado ao cenário agrícola e, conseqüentemente, ao cenário econômico do país. Por meio deste projeto de pesquisa, também se espera contribuir para a automatização agrícola e para o aumento da qualidade de alimentos.

A agricultura inteligente auxilia o produtor rural no monitoramento da lavoura de forma prática e eficaz, valendo-se de métodos de análise de dados e sensoriamento, reduzindo riscos de perdas por eventos climáticos, eventuais problemas com pragas, otimização do uso de insumos, redução de impactos negativos ao meio ambiente e aumento na produtividade da lavoura.

Outra motivação é a chance de demonstrar a força que as tecnologias e ferramentas *open source* têm sobre o desenvolvimento de soluções para a área da agricultura e *IoT*, de modo a inspirar futuras pesquisas e trabalhos acadêmicos que possam ampliar e amadurecer essa área de estudo.

1.4 PERSPECTIVAS DE CONTRIBUIÇÃO

Ao término deste projeto, pretende-se publicá-lo no formato de artigo científico e disponibilizá-lo em instituições de ensino e/ou para pessoas com interesse na área, tendo como objetivo compartilhar os conhecimentos e resultados alcançados. O desenvolvimento da plataforma de monitoramento de plantio possibilitará uma comparação com métodos e aplicações utilizados em agricultura inteligente, com a finalidade de contribuir para a realização de projetos e pesquisas futuras relacionados a essa área.

1.5 METODOLOGIA DE PESQUISA

Os objetivos deste trabalho acadêmico foram alcançados por meio de pesquisas teóricas, de forma a adquirir os conhecimentos necessários a partir da leitura de artigos científicos, livros, monografias, dissertações, teses, guias práticos e técnicos, livros e fontes digitais confiáveis. Desse modo, foi possível elaborar e implementar uma plataforma de monitoramento de lavoura funcional.

Primeiramente, foram realizados estudos das plataformas *Arduino* e *Raspberry*, as quais possibilitaram a utilização de módulos e sensores neles acoplados. Em seguida, foram realizados estudos de sensores e de módulos aplicados na captação de informações do campo. Após esse processo, um *webService* foi desenvolvido com o objetivo de processar e analisar os dados obtidos na plantação. Por fim, foi desenvolvida uma plataforma *Web* cuja função consiste em monitorar o plantio em tempo real para o usuário.

1.6 RECURSOS NECESSÁRIOS

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

- **Hardware**
 - Notebook HP NP-RC420
 - Processador *Intel Core* i3 2.33 Ghz.
 - Disco Rígido *Hitachi* 5400 RPM SATA de 500 GB.
 - Memória de 1333 Mhz DDR3 4 GB.
 - *Arduino*
 - *Raspberry Pi3*
 - Sensores de Ambiente
- **Software**
 - Banco de Dados Mongo DB
 - *Python*
 - C
 - *React*
 - *Visual Studio Code*

1.7 ESTRUTURA DO TRABALHO

Este trabalho está estruturado da seguinte forma:

- **Capítulo 1 – Introdução:** Neste capítulo, é contextualizada a área de estudo, bem como os objetivos, a justificativa e a motivação para o desenvolvimento desta pesquisa.

- **Capítulo 2 – Internet of Things (IoT):** Compete ao segundo capítulo abordar conceitos de *IoT* e tecnologias *open source* como *Raspberry* e *Arduino*, tendo em vista o propósito central de exemplificar seu funcionamento.
- **Capítulo 3 - Open Data Science:** Este capítulo comenta as principais tecnologias *open source* direcionadas ao *Data Science*, as quais foram utilizadas ao longo do projeto.
- **Capítulo 4 – Agricultura Inteligente:** O quarto capítulo trata dos principais conceitos sobre agricultura inteligente e suas aplicações envolvendo IoT.
- **Capítulo 5 – Proposta de Trabalho:** Apresenta a plataforma utilizada no projeto, bem como os fundamentos de agricultura inteligente a serem desenvolvidos.
- **Capítulo 6 – Conclusão:** Consiste na apresentação e discussão acerca das vantagens da agricultura inteligente bem como a utilização de IoT no campo.
- **Referências**

2. INTERNET OF THINGS (IOT)

O termo Internet of Things (IoT), ou internet das coisas, foi empregado no final da década de 1990 pelo pesquisador Kevin Ashton, quando o mesmo fez uma proposta para a gigante multinacional de bens de consumo P&G, com o propósito de melhorar o fluxo dos produtos e informações sem ter interferências diretas do homem nos processos, utilizando a tecnologia de *RFID*. (COSTA, OLIVEIRA e MÓTA, 2018).

Desde então, o termo *Internet of Things* costuma ser relacionado com a indústria 4.0, pois o setor industrial constatou que com a utilização de IoT seria possível acarretar a diminuição de interferência humana em tarefas de operacionalização tecnológica e, também, a diminuição de custos industriais. Além disso, com a *Internet of things* a eficiência é maior e, por isso mesmo, mais evidente.

Para que os benefícios do IoT possam ser compreendido, é necessária a utilização de algumas tecnologias presentes no cotidiano como: *RFID*, *Bluetooth*, sensores, arquiteturas de rede, protocolos de comunicação e conexões sem fio. (COSTA, OLIVEIRA e MÓTA, 2018).

Na primeira metade da década de 2010, o número de dispositivos IoT conectados e utilizados na indústria, em geral, chegou à quantidade de 5 bilhões. A estimativa é que para o ano de 2020 essa quantidade seja 10 vezes maior, ou seja, 50 bilhões de dispositivos conectados à internet (CHASE, 2013), conforme demonstrado na Figura abaixo:

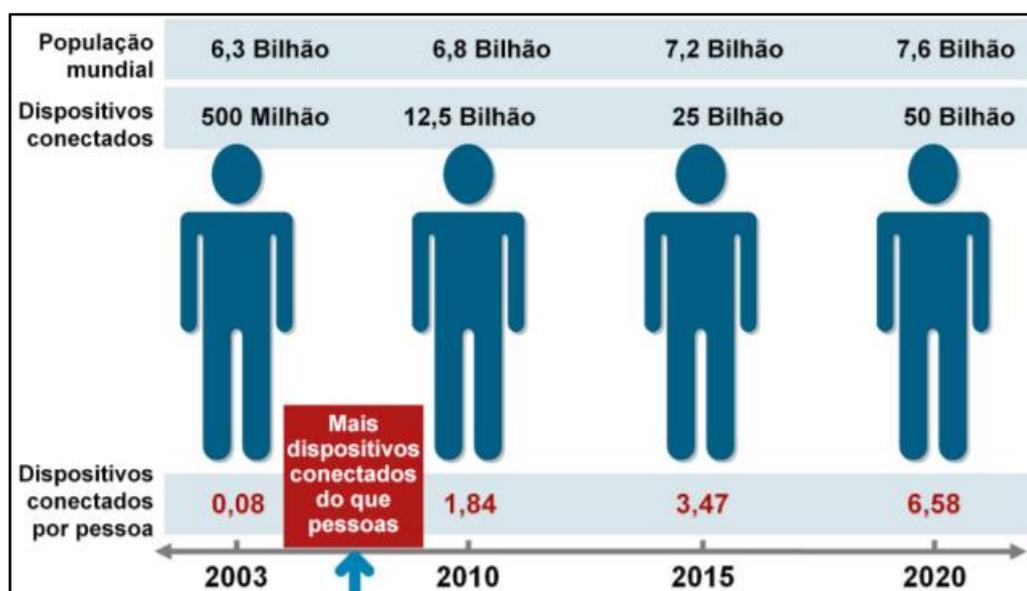


Figura 1: Gráfico dispositivos per capita

Fonte: Cisco IBSG, abril de 2011

De acordo com (CAVALCANTI, 2019), apesar do IoT apresentar resultados satisfatórios em projetos desenvolvidos a partir de suas ferramentas, há algumas dificuldades que retardam a evolução dessa tecnologia, bem como a sua manutenção em diversas áreas da atividade humana. Em uma aplicação de IoT em um ambiente agrícola, por exemplo, a maior dificuldade de implementação do projeto justifica-se a partir da falta de sensores autossustentáveis que possam se manter com pouco consumo de energia. Em projetos assim teriam de ser utilizadas placas solares, o que ocasionaria um alto custo financeiro.

2.1 ARDUINO

De acordo com (WHEAT, 2011), *Arduino* foi desenvolvido originalmente na cidade de Ivrea, na Itália, pelo professor Massimo Banzi, no ano de 2005.

A placa do *Arduino* é baseada no processador Atmel AVR. Ela é constituída de uma porta serial, um circuito de alimentação, conectores e uma grande expansão para diversos tipos de componentes suportados (WHEAT, 2011).



Figura 2: *Arduino*

Fonte: *FlipFlop*, setembro de 2014.

Segundo (MCROBERTS, 2011, p. 22) , o *Arduino* é titulado como uma plataforma embarcada, pois ele pode interagir com seu ambiente por meio de *hardware* (sensores e *Shields*) e, também, por meio de *software* (Programas que utilizam da capacidade computacional e *hardware* do *Arduino*).

De maneira geral, o *Arduino* se assemelha aos computadores comuns, visto que possui um *hardware* com as seguintes características: memória RAM, processador, barramentos, entre outros dispositivos que compõem sua arquitetura.

De acordo com (EVANS, NOBLE e HOCHENBAUM, 2013, p. 25-26), a plataforma *Arduino* teve um grande aumento em sua popularidade logo após os usuários perceberem suas principais características, as quais, em sua maioria, apresenta benefícios relevantes e visíveis, tais como: fácil utilização, baixo custo e um enorme leque de possibilidades de projetos que se pode fazer com essa placa como, por exemplo, projetos envolvendo agricultura de precisão e agricultura inteligente.

2.1.1 Sensores

Segundo (ANTONIO A.F. LOUREIRO, 2014), nas últimas décadas houve um grande aumento do avanço tecnológico nas áreas dos sensores e dos circuitos integrados que, por sua vez, podem ser usados em diversos meios, como no monitoramento e controle de florestas e até mesmo de oceanos.

Ainda de acordo com o autor, esse avanço tecnológico é mais abrangente na área dos micro-controladores e, também, na comunicação sem fio, que tem estimulado o grande desenvolvimento e o uso dos sensores inteligentes, os quais podem ser utilizados em diversas áreas ligadas a processos físicos. A tendência para as próximas décadas consiste em realizar a produção desses sensores em larga escala. Com isso, verifica-se o baixo custo na aquisição dos novos recursos tecnológicos e o desenvolvimento aprofundado desses dispositivos, os quais são responsáveis por trazer melhoras significativas no trabalho de monitoramento.

A utilização dos sensores tem angariado grandes melhorias obtidas a partir do IoT em diversas áreas de estudos e, também, na elaboração de novos produtos. As áreas que mais se beneficiam são:

- Indústria 4.0: A indústria sem dúvida é o setor que mais se beneficia com a utilização de sensores. Com a utilização desses dispositivos houve uma revolução na indústria, visto que processos humanos foram automatizados, melhorando assim a performance da fábrica, monitoramento da produção e a diminuição de custos.
- Agricultura: A Agricultura usufrui dos benefícios promovidos tanto pelos sensores, quanto pelo IoT. É importante lembrar que a utilização de sensores na agricultura é recente, no entanto, já são perceptíveis determinadas melhorias significativas como: monitoramento e controle de mudanças de

ambiente, automatização de veículos agrícolas e aumento de produtividade com a diminuição de impactos ambientais.

- Aeroespacial: Os sensores também são muito utilizados por instituições de monitoramento e pesquisas espaciais. Esses dispositivos são utilizados nos processos de sensoriamento remoto e também no monitoramento de florestas.

Com tudo, a perspectiva é que nos próximos anos as tecnologias que envolvem os sensores evoluam ao ponto de se tornarem autônomos no que se diz respeito ao consumo de energia, de modo a facilitar e a viabilizar muitos projetos, cujo maior problema na atualidade consiste no consumo de energia.

2.2 RASPBERRY

De acordo com (EBERMAM, PRESENTE, *et al.*, 2017), o projeto *Raspberry* surgiu no início do ano de 2006, quando os pesquisadores Eben Upton, Rob Mullins, Jack Lang e Alan Mycroft resolveram desenvolver um computador de pequeno porte e, de preço acessível para crianças, no laboratório da Universidade de Cambridge, Inglaterra. O principal propósito para que o *Raspberry* fosse criado se deu pela observação de Eben, o qual constatou que a maioria dos estudantes do Departamento de Ciência de Computação não estavam apresentando habilidades e domínio das máquinas.

O projeto despertou interesse no público pois, além de ser simples e de baixo custo, seu código é totalmente aberto, fazendo com que um grande leque de possibilidades de desenvolvimento seja acessado com a contribuição de seus usuários.



Figura 3: *Raspberry Pi 4*
Fonte: dobitaobyte, janeiro de 2016.

2.2.1 Arquitetura

Atualmente, a família *Raspberry* possui cerca de oito dispositivos principais: *Raspberry Pi A+*, *Raspberry Pi model B*, *Raspberry pi 2*, *Raspberry Pi Zero*, *Raspberry Pi Zero W*, *Raspberry Pi 3*, *Raspberry Pi 3 model B+* e o *Raspberry Pi 4 Model B*.

Apesar das diferenças entre as versões do *Raspberry*, no quesito aparência e composição tais versões são muito semelhantes entre si.

De acordo com (Cavalcanti, 2019), a maioria das placas *Raspberry* possui um circuito integrado chamado *system-on-chip* (SoC), que tem como função fornecer fontes de processamento, renderização gráfica e recursos de I/O ao dispositivo. Entre as variações dos modelos, esse circuito integrado pode ser o *Broadcom2837*, *BCM2836 quad-core* e, também, o *BCM2837* de 64 *bits*.

A maioria das versões das placas *Raspberry* tem seus componentes internos muito semelhantes ao de computadores tradicionais, com exceção de sua capacidade de processamento. O *Raspberry Pi 3*, por exemplo, possui 4 portas USB utilizadas para conectar dispositivos, tais como: teclado e mouse. Além disso, possui uma saída HDMI (*High-Definition Multimedia Interface*) utilizada para realizar a transmissão de áudio e vídeo para um monitor. Ademais, possui porta de vídeo composto que também é utilizado para a transmissão de áudio e vídeo, mas somente para monitores e televisores mais antigos; apresenta também uma porta *Ethernet* 10/ 100 RJ 45, memória *Ram*, placa integrada de *wi-fi* e *Bluetooth*.

2.2.2 Sistemas suportados

O *Raspberry* oferece grande gama de sistemas operacionais disponíveis para a instalação. Essa grande variedade de sistemas disponíveis faz com que o *Raspberry* tenha bastante variedade de projetos podendo, inclusive, ser uma central multimídia e, até mesmo, um protótipo de carro autômato. Entre os sistemas disponíveis, estão:

- *Raspbian*: Este é o principal sistema operacional e também oficial do *Raspberry*. Esse sistema é baseado no Debian e, quando instalado, traz uma grande variedade de softwares que possibilita ao usuário utilizá-lo como se fosse um computador convencional.

- *Windows 10 IoT Core*: Esse sistema foi desenvolvido pela *Microsoft*, sendo este uma versão otimizada do *Windows 10*, voltado especificamente para a utilização em projetos de Internet das Coisas.
- *Ubuntu Server*: Desenvolvida pela Canonical, essa é uma versão otimizada do *Ubuntu Server* x86 e x64 para processadores ARM, possibilitando a construção de servidores e *clusters* com dispositivos *Raspberry*.

3. OPEN DATA SCIENCE

Com a indústria 4.0, a quantidade de dados gerados é cada vez maior. De acordo com uma pesquisa desenvolvida pela EMC [1], o volume de dados saltará de 130 *exabytes* para 40.000, ou 40 trilhões de *gigabytes* até 2020, conforme demonstrado na Figura 4. Após esta data, a tendência é que o volume de dados dobre a cada dois anos. Esse amplo aglomerado de dados recebeu o nome de *Big Data*.

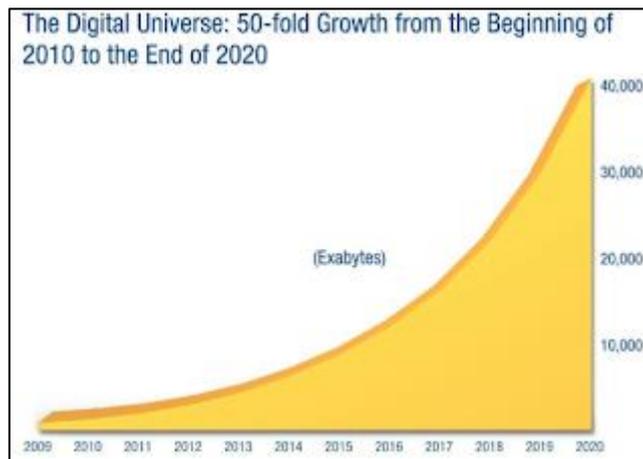


Figura 4: Gráfico Aumento de Dados

Fonte: IDC's *Digital Universe Study*, 2012.

Dessa forma, o termo *Data Science* veio à tona. Vale assinalar que Ciência de Dados ou *Data Science* envolve o uso de métodos para analisar grandes quantidades de dados (*Big Data*) e extrair o conhecimento que eles contêm (CIELEN, MEYSMAN e ALI, 2016). Muitas empresas ao redor do mundo perceberam esse potencial e, por isso, começaram a realizar a análise de dados, de modo que poderiam tomar decisões com mais segurança usando gráficos e relatórios.

Data Science nada mais é do que um campo multidisciplinar que se apropria dos campos da matemática, estatística e da ciência da computação, a partir dos quais é possível realizar mineração de dados, projetos de *Machine Learning*, análise preditiva e *Business Intelligence* (BI) por meio da extração, estudo e análise dos dados, os quais possibilitam a extração de informações de grande importância por meio dos dados. Contudo, esse campo de pesquisa tem como objetivo diagnosticar possíveis ocorrências, prever consequências futuras e gerar *insights*, auxiliando assim o usuário no que pode ser feito.

3.1 TECNOLOGIAS OPEN SOURCE

Nesta seção, serão apresentadas as tecnologias *open-source* ligadas à *Data Science*, as quais serão utilizadas no decorrer do desenvolvimento do projeto.

3.1.1 Python

Python é uma linguagem de programação orientada a objetos de altíssimo nível (VHLL – *Very High Level Language*), criada no final da década de 1990 pelo pesquisador Guido Van Rossum. Essa é uma linguagem gratuita e de código aberto que se encaixa na categoria de linguagem de propósito geral. Desse modo, é possível construir aplicações tanto em nível comercial quanto para programas de nível científico (MANZANO, 2014).

Em geral, *Python* é uma das linguagens mais bem aceitas no meio científico, principalmente por cientistas de áreas que não estão necessariamente ligadas à Tecnologia da Informação, mas que apresentam a necessidade de aplicar a computação em algum momento de suas pesquisas (CRUZ, 2018). Isso acontece pelo fato da linguagem *Python* apresentar um grande número de bibliotecas para cunho científico e por contar com uma didática e sintaxe muito simples, facilitando o aprendizado.



Figura 5: Logo Python
Fonte: *Python Org*, abril 2018

3.1.2 Mongo DB

Mongo DB é um banco de dados não SQL ou NoSql. Segundo (SOLAGNA e LAZZARETTI, 2016), é um termo genérico para uma classe de banco de dados que

rompe com o paradigma dos bancos de dados embasados nos modelos relacionais existentes.



Figura 6: Logo *Mongo DB*
Fonte: Luiz Tools, setembro de 2017.

Esse banco de dados foi desenvolvido na linguagem C, e tem como propósito principal ser uma aplicação de alto desempenho em que se utiliza uma maneira de armazenar dados que se distancia das estruturas rígidas como tabela, por exemplo. Em vez disso, ele os armazena em documentos vagamente definidos. (SOLAGNA e LAZZARETTI, 2016).

Por apresentar uma linguagem de consulta de alto desempenho, o *mongo DB* acaba sendo uma excelente opção como banco de dados em projetos que se envolvem diretamente com IoT e análise de dados.

3.1.3 PostgreSQL

Segundo (SOLAGNA e LAZZARETTI, 2016), PostgreSQL é um sistema gerenciador de banco de dados que foi desenvolvido no final da década de 1990 por um programador chamado Michael Stonebraker. Esse projeto foi desenvolvido no departamento de Ciência da Computação da Universidade da Califórnia. A princípio, a ideia era criar um servidor de banco de dados relacionais, denominado Postgres.

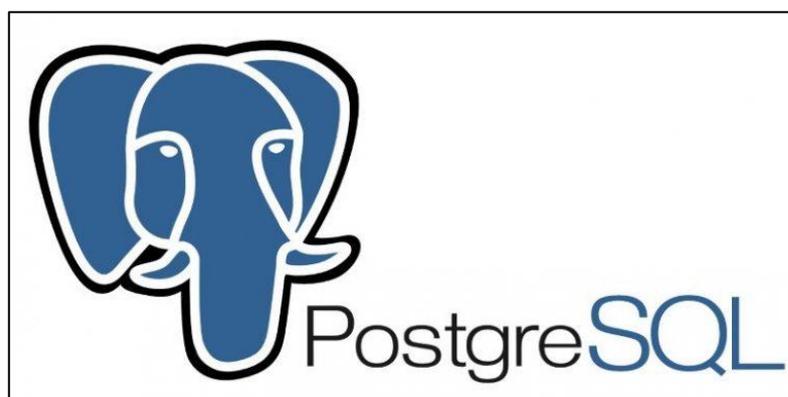


Figura 7: Logo *PostgreSQL*
Fonte: Save Point, novembro de 2019.

A linguagem utilizada neste SGBD é o SQL, ou seja, trata-se de uma linguagem semelhante a outros tipos de SGBD também existentes nos dias de hoje. Porém, o PostgreSQL apresenta um diferencial em relação aos outros sistemas de banco de dados, que é a descendência do código de fonte aberta (*open source*), do então projeto desenvolvido no Departamento de Computação da Universidade da Califórnia. Com o passar do tempo, PostgreSQL ganhou espaço no mercado de desenvolvimento de *software*, o que estimulou os desenvolvedores a produzirem mais modelos, fazendo com que se sentissem mais confiantes, sobretudo por causa da maneira em que o *software* foi projetado. Em resumo, o PostgreSQL demonstrou ser uma ferramenta bem robusta e confiável, sendo utilizada em grandes *softwares*.

4. AGRICULTURA INTELIGENTE

Atualmente, a população mundial é em torno de 7 bilhões de habitantes. Estudos apontam que no ano de 2050 a população global será por volta de 9 bilhões de habitantes.

Um relatório apresentado pela FAO (Organização das Nações Unidas para Agricultura e Alimentação), constatou que até o ano de 2050 será necessário aumentar a produção alimentícia global em até 70% para suprir o grande aumento populacional. Com isso, entende-se que é necessário reforçar e aumentar, de forma eficiente, o tamanho da produção agrícola como um todo.

Existem dois grandes aliados para ajudar a suprir esse grande aumento de produtividade agrícola, a agricultura de dados e a agricultura de precisão, que juntas originaram um novo termo: agricultura inteligente. A Agricultura Inteligente se baseia na utilização de equipamentos em campo, sensores e maquinários, a partir dos quais é possível realizar o monitoramento de produtividade e rendimentos do campo de forma eficaz.

Entretanto, também são necessárias ferramentas de análise de dados, as quais possibilitam a realização de estudos sobre os dados coletados em campo, auxiliando agricultores a decidirem com mais precisão e confiabilidade.

Um dos grandes integrantes e responsáveis pelo surgimento do termo “agricultura inteligente” é o IoT (*Internet of Things*). De acordo com (MEOLA, 2019), até o ano de 2020, 75 milhões de dispositivos IoT serão utilizados diretamente na agricultura como um todo.

Ainda de acordo com o pesquisador, no ano de 2017 o número de pontos de dados era de <0,5 milhões>. A previsão para o ano de 2050 é que o número de pontos de dados na agricultura seja de 4 milhões. Isso indica que não só grandes fazendas utilizam a agricultura inteligente como uma ferramenta para o aumento de produtividade, mas que também pequenos e médios produtores individuais comecem a utilizar essa ferramenta como auxílio.

4.1 APLICAÇÕES EM AGRICULTURA INTELIGENTE

Aplicações para agricultura inteligente estão crescendo cada vez mais no Brasil. Porém são poucos *softwares* que realizam um monitoramento completo de áreas e safras em tempo real, e que ofereça *insights* com precisão ao agricultor.

O SIMA é um *software* de monitoramento de plantio e culturas que utiliza os conceitos de IoT para realizar a coleta de dados e visualizar informações sobre o plantio de determinada cultura. Tal *software* apresenta uma arquitetura bem interessante, sendo basicamente dividido em três partes:

- **Sensores:** Esse *software* utiliza sensores divididos em lotes, nos quais é realizada a coleta de dados, de maneira que esses dados são enviados a um servidor que os submete ao processamento.
- **WebService:** Aqui os dados coletados no campo passam por uma etapa de processamento, no qual são utilizados algoritmos de análise de dados para gerar informações úteis e simplificadas para o agricultor.
- **App:** Depois de realizados todos os devidos processamentos das informações, é utilizado um aplicativo *android* que consome as informações já processadas no *WebService*, de maneira que o agricultor pode consultar os dados. Nesse *App*, é possível prever se em determinado lote há chances de ter pragas e até mesmo monitorar o nível de umidade em determinados lotes.

Porém, esse *software* possui algumas limitações para o público brasileiro nos seguintes quesitos: realizar consultas somente no *smartphone* e ser de origem argentina, o que inviabiliza a possibilidade de oferecer suporte a toda região brasileira.

5. PROPOSTA DE TRABALHO

Este trabalho tem como objetivo desenvolver uma aplicação de agricultura inteligente com base no conceito de IoT, incluindo *Arduino* com sensores de temperatura de ambiente, umidade do ar, umidade de solo e de raio ultra violeta, juntamente com o *Raspberry*, comunicando ao *Arduino*. É importante esclarecer que neste dispositivo ficará hospedado a API do *backend Python* e, também, o *frontend* construído em *react*. Essa aplicação será desenvolvida a fim de auxiliar agricultores em suas plantações, de forma que esses profissionais tenham em mãos uma ferramenta de monitoramento de plantio completa, com uma interface capaz de mapear as condições climáticas e de realizar decisões com base na visualização de dados.

O projeto também visa criar uma infraestrutura com *Arduino* e seus sensores, integrado à placa *RaspBerry* que, por sua vez, será responsável por executar o *backend*, garantindo que a API disponibilize os dados necessários para que o *frontend* possa exibi-los.

5.1 SMART AGRICULTURE

Smart Agriculture tem como propósito oferecer apoio e facilidade ao agricultor na hora de realizar o monitoramento de plantios por meio da análise dos dados coletados em campo. Tendo em vista esse propósito, este trabalho se dedicou a desenvolver uma solução *open source* de baixo custo que supra as necessidades do agricultor.

A princípio, os sensores ligados no *Arduino* coletaram dados como: umidade da terra e do ar, temperatura do solo e do ambiente e, também, a quantidade de raios UV. Essas informações foram passadas ao *Webservice*, que processou e disponibilizou os dados para a aplicação web. Dessa forma, o agricultor poderá monitorar sua plantação, além de tirar *insights* e relatórios.

5.2 RECURSOS ADOTADOS

Em um primeiro momento, foi realizada a programação de um *software* para o *Arduino*, possibilitando a comunicação com os sensores e também com o *webservice*. Depois de finalizada a programação, foi iniciado o processo de criação da API, a qual foi construída com *framework* Flask, presente na linguagem *Python*. Essa API foi executada no *RaspBerry* e teve como propósito processar todos os dados captados pelos sensores, além de criar a regra de funcionamento de toda a plataforma.

Em seguida, foi realizada a construção do *frontend* com as ferramentas *React*, uma plataforma de aplicações web baseada na linguagem *TypeScript*, e também do *Chart.js*, responsável pela geração de gráficos.

5.3 DESENVOLVIMENTO DO TRABALHO

Neste trabalho, foi realizado o desenvolvimento de uma plataforma de monitoramento agrícola com base no conjunto de três aplicações: Sistema de captura de dados, *Flask rest api* (responsável pelo recebimento e processamento dos dados e, também, pela disponibilização dos mesmos para a aplicação *front-end*), e a própria aplicação *react*, responsável por exibir os dados processados. A comunicação entre as aplicações é realizada via internet, a partir da utilização de requisições e respostas HTTP para o envio e recebimento de dados, conforme ilustrado na figura 8.

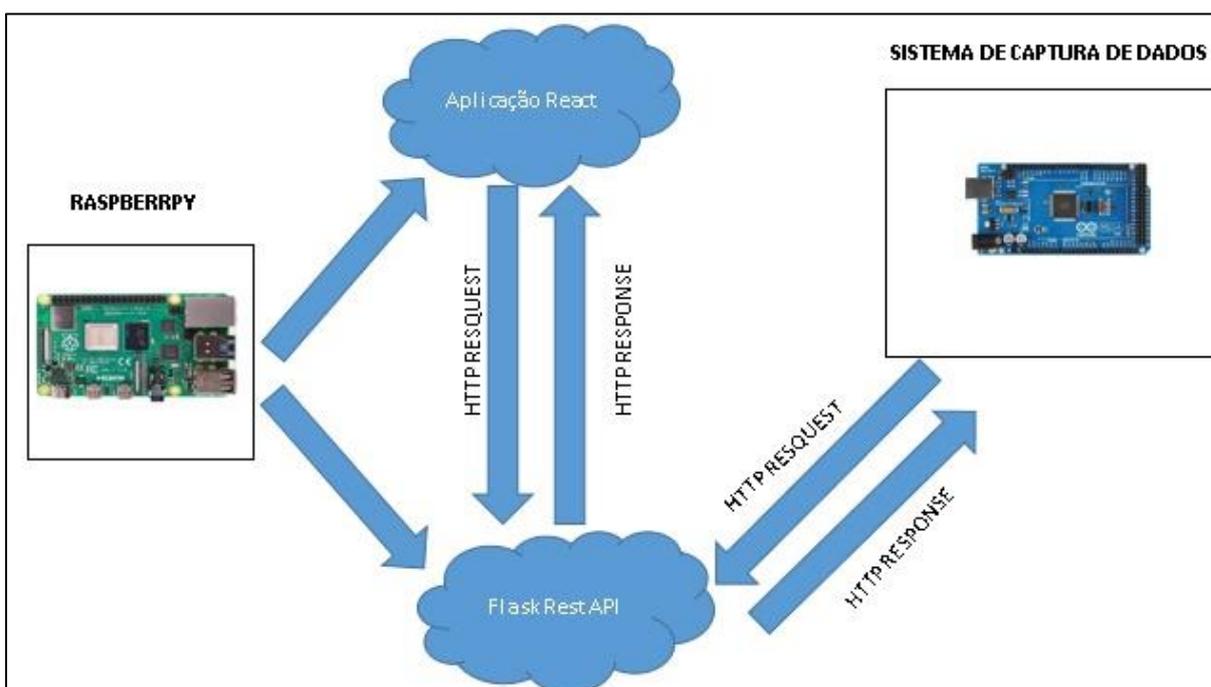


Figura 8: Arquitetura Geral.

O desenvolvimento dessa plataforma foi realizado em três etapas sequenciais:

- Construção da aplicação *Flask rest API*;
- Construção do sistema de captura e envio de dados com *arduino*;
- Construção da aplicação *React* para a exibição dos dados capturados.

Nas próximas subseções será demonstrado, de forma mais detalhada, como foram utilizadas as tecnologias para desenvolver cada etapa até chegar na plataforma final.

5.3.1 Flask API

A primeira aplicação a ser desenvolvida, entre as 3 principais, é a *Rest API*. Compreende-se que a API é a etapa central de todo o projeto, ou seja: a etapa responsável por receber os dados capturados a partir do sistema de captura (*arduino* + sensores) que, por sua vez, são processados, armazenados e disponibilizados para uma aplicação web *react.y*.

A construção da API foi possível graças ao *framework web* para *python*, denominado *Flask*. Este foi escolhido pois se trata de um framework limpo e totalmente modularizado, visto que o desenvolvedor escolhe quais pacotes e bibliotecas serão utilizadas no projeto, eliminando a possibilidade de importações de pacotes que não serão utilizados, o que tornaria a aplicação mais lenta. Certamente, a utilização desse *framework* faz com que as aplicações construídas a partir dele sejam mais rápidas e eficientes.

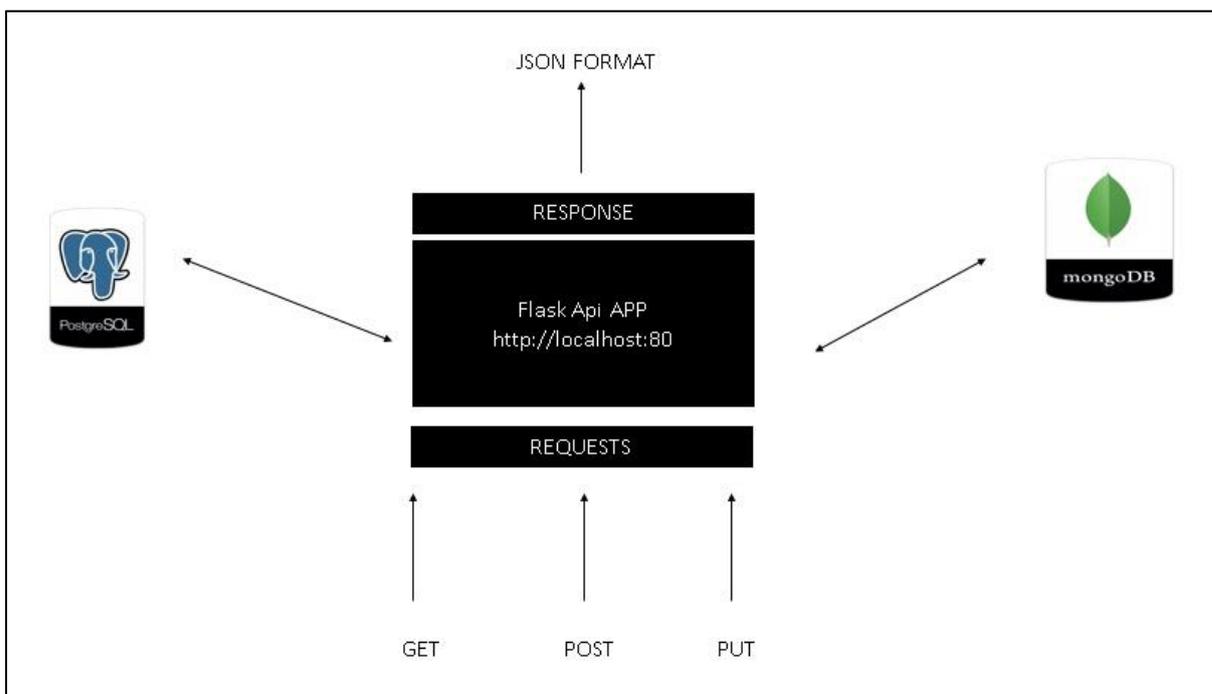


Figura 9: design da api

Conforme ilustrado na figura 9, a API foi desenvolvida a partir de dois bancos de dados diferentes: *mongodb* e *PostgreSql*. Vale ressaltar que o banco de dados *PostgreSql* é utilizado para armazenar todos os dados captados pelo sistema de captura de dados, enquanto o banco *mongodb* é utilizado, exclusivamente, para retornar dados para o sistema *web react*. Esse banco armazena apenas os últimos dados enviados à API, o que torna a pesquisa mais rápida e eficiente.

No total, a aplicação possui 5 rotas (listadas abaixo) protegidas, as quais utilizam o método de autenticação *jwt*, que possibilita que outras aplicações, pertencentes ao projeto, enviem ou recebam dados de forma segura.

- GET <http://localhost:80/area/{area}> -> retorna as últimas informações captadas pelos sensores (umidade da terra, temperatura ambiente e UV), de acordo com a área informada por parâmetro.
- PUT <http://localhost:80/area/{area}> -> Além de ser necessário informar a área por parâmetro, também é necessário enviar os parâmetros *temp_ambiente*, *humi_solo*, *raio_uv* e *data*. Essa rota é de acesso exclusivo do *arduino* e, tem como objetivo, inserir as informações passadas para o banco *PostgreSql* e atualizar as informações presentes no banco *mongodb*.
- GET http://localhost:80/process?data_inicio=value&data_final=value&area=value -> apresenta um relatório em json sobre o plantio de acordo com o range de data, informando de maneira precisa a área analisada. No relatório informado, constam-se informações diárias relacionadas à média de temperatura, média de raios UV, média da umidade, média da temperatura do período estacional, média de umidade do período, além de apresentar um parecer sobre os índices de favorabilidade climática de cada dia. A condição climática do dia é obtida por meio de um algoritmo de classificação, com base nos conceitos *machine learning*.
- POST <http://localhost:80/login> - > é necessário informar, via parâmetro json, o usuário e senha. Após isso, um *token* de acesso será emitido na tela, de modo que a partir dele o usuário terá a possibilidade de acessar as demais rotas e, também, a aplicação *web react*.
- POST <http://localhost:80/logout> - > é necessário informar pelo *Header* o parâmetro *authorization* contendo o *token* de acesso pois, quando acessado, o token de acesso informado é colocado em uma *blacklist*, de modo que o usuário perde o acesso às rotas e à aplicação *web react*.

No flask as rotas são tratadas como *resources*, de modo cada *resource* pode conter até quatro funções. Tais funções são responsáveis, respectivamente, por cada tipo de requisição http como: *get*, *post*, *put* e *delete*. A seguir, tem-se um exemplo utilizado no projeto com o *Resource login*:

```

from flask_restful import Resource, reqparse
from models.users import UserModel
from werkzeug.security import safe_str_cmp
from flask_jwt_extended import create_access_token, jwt_required, get_raw_jwt
from blacklist import BLACKLIST

atributos = reqparse.RequestParser()
atributos.add_argument('user_login', type=str, required=True, help="The field
'login' cannot be left blank")
atributos.add_argument('user_senha', type=str, required=True, help="The Field
'senha' cannot be left blank")

class UserLogin(Resource):

    @classmethod
    def post(cls):
        #1º captura os parâmetros passados
        dados = atributos.parse_args()

        #2º realiza a busca no banco postgres utilizado o USER_LOGIN como para
metro
        user = UserModel.find_by_login(dados['user_login'])

        #3º Realiza a validação dos dados informados
        if user and safe_str_cmp(user.user_senha, dados['user_senha']):
            #4 caso seja encontrado é criado um token do tipo JWT para o usar
io
            token_de_acesso = create_access_token(identity=user.user_id)
            #5 é retornado o token de acesso
            return {'access_token': token_de_acesso}, 200
        return {'Message': 'The username or password is incorrect'}, 401

```

Neste exemplo foi criada a classe *UserLogin*, inspirada a partir do modelo *Resource*, na qual também está presente o método *post*. Para definir qual o *path* que irá acessar o método *post* da classe *UserLogin*, é necessário registrar essa informação no arquivo *app.py* do projeto, conforme consta no código abaixo:

```

#End Points Configurados
api.add_resource(Area, '/area/<string:name_area>')
api.add_resource(ProcessInformation, '/process')
api.add_resource(UserLogin, '/login')
api.add_resource(UserLogout, '/logout')

```

No geral, todas as configurações iniciais como: definir banco de dados, definir rotas e funções principais são realizadas dentro do arquivo app.py, uma vez que ele é executado para iniciar a API. O código abaixo é o código contido no arquivo app.py da aplicação API criada.

```

from flask import Flask, jsonify
from flask_restful import Api
from resources.area import Area, ProcessInformation
from resources.users import UserLogin, UserLogout
from flask_cors import CORS
from flask_jwt_extended import JWTManager
from blacklist import BLACKLIST

app = Flask(__name__)
#Configurações iniciais do banco de dados MONGODB
app.config['MONGODB_SETTINGS'] = {
    'db' : 'plant'
}

#Configurações iniciais do banco de dados Postgress
app.config['SQLALCHEMY_DATABASE_URI'] = 'postgres+psycopg2://postgres:1234@localhost/plant'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
#configurações iniciais do JWT
app.config['JWT_SECRET_KEY'] = 'DontTellAnyone'
app.config['JWT_BLACKLIST_ENABLED'] = True
#iniciando o cors
CORS(app)
#iniciando a api
api = Api(app)
#iniciando o JWT
jwt = JWTManager(app)

@jwt.token_in_blacklist_loader
def verifica_blacklist(token):
    return token['jti'] in BLACKLIST

@jwt.revoked_token_loader
def token_de_acesso_invalidado():
    return jsonify({'Message' : 'You have benn Logged out.'}), 401

#End Points Configurados
api.add_resource(Area, '/area/<string:name_area>')
api.add_resource(ProcessInformation, '/process')
api.add_resource(UserLogin, '/login')
api.add_resource(UserLogout, '/logout')

```

```

from flask import Flask, jsonify
from flask_restful import Api
from resources.area import Area, ProcessInformation
from resources.users import UserLogin, UserLogout
from flask_cors import CORS
from flask_jwt_extended import JWTManager
from blacklist import BLACKLIST

app = Flask(__name__)
#Configurações iniciais do banco de dados MONGODB
app.config['MONGODB_SETTINGS'] = {
    'db' : 'plant'
}

#Configurações iniciais do banco de dados Postgress
app.config['SQLALCHEMY_DATABASE_URI'] = 'postgres+psycopg2://postgres:1234@localhost/plant'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
#configurações iniciais do JWT
app.config['JWT_SECRET_KEY'] = 'DontTellAnyone'
app.config['JWT_BLACKLIST_ENABLED'] = True
#iniciando o cors
CORS(app)
#iniciando a api
api = Api(app)
#iniciando o JWT
jwt = JWTManager(app)

@jwt.token_in_blacklist_loader
def verifica_blacklist(token):
    return token['jti'] in BLACKLIST

@jwt.revoked_token_loader
def token_de_acesso_invalidado():
    return jsonify({'Message' : 'You have benn Logged out.'}), 401

#End Points Configurados
api.add_resource(Area, '/area/<string:name_area>')
api.add_resource(ProcessInformation, '/process')
api.add_resource(UserLogin, '/login')
api.add_resource(UserLogout, '/logout')

if __name__ == '__main__':
    from mon_engine import db_mon
    from sql_alchemy import banco
    banco.init_app(app)
    db_mon.init_app(app)
    app.run(debug=True, host="192.168.1.103", port=80)

```

Ao executar o arquivo `app.py`, o servidor da API é iniciado, o que demonstra que o dispositivo está pronto para receber requisições e enviar respostas em formato *json*.

5.3.2 Sistema de captura de dados

O sistema de captura de dados foi desenvolvido a partir da plataforma *arduino*, com os sensores de temperatura *dht11*, sensor higrômetro de umidade e sensor de raios UV *uvma-30a*.

Com base no conjunto de sensores, o sistema de captura de dados é responsável por realizar a captação de informações no campo, além de alimentar o banco de dados por intermédio da API. Para que o envio dos dados captados seja possível, a placa de rede *enc28j60* é utilizada, pois através dela os dados são enviados via rede, dispondo dos protocolos *http*.

Na figura 10, é possível visualizar como foi realizada a construção da arquitetura do sistema de captura de dados, utilizando os componentes necessários para o seu funcionamento. Nota-se que é utilizado uma *protoboard* para emular um circuito eletrônico, o que facilita o manuseio e a manutenção do circuito proposto, caso seja necessário.

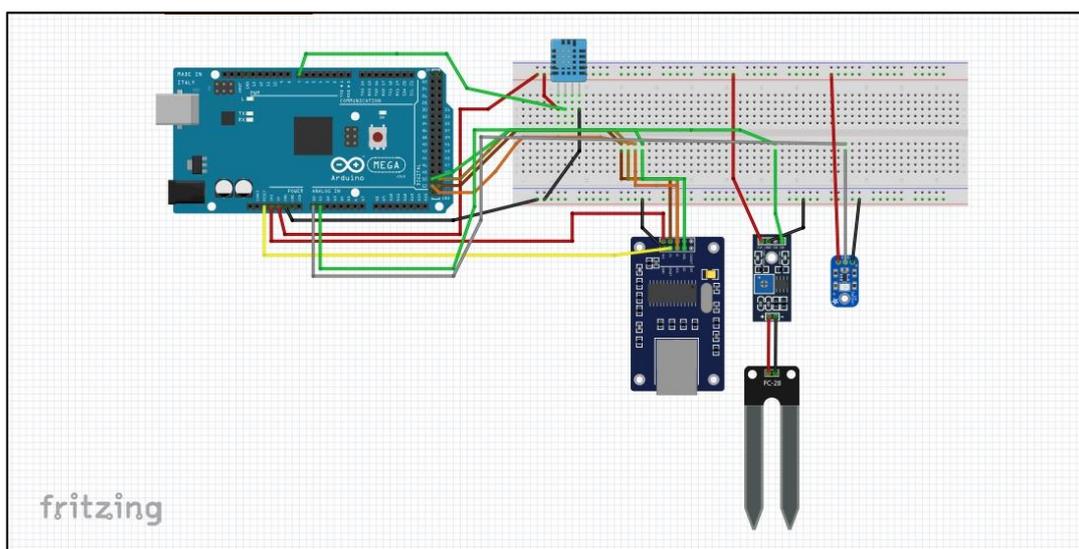


Figura 10: Arquitetura do Sistema de Captura de dados.

Com o propósito de assegurar um padrão de estrutura e de quantidade de dados suficientes para futuras análises no banco de dados, 288 envios são emitidos em um

período de 24 horas, de forma que a cada 5 minutos é realizado um envio do conjunto de dados para a API.

5.3.3 Sistema Agro Data

O sistema agro data é uma aplicação *web frontend* responsável por renderizar ao usuário todas as informações captadas pelo sistema de captura de dados e, seguidamente, processadas pela *rest api*.

Toda aplicação foi construída por meio do *framework react*, o qual proporciona uma navegação mais fluente e agradável ao usuário do sistema.

O sistema possui três páginas totalmente estilizadas, proporcionando um conforto maior ao usuário. Embora haja uma quantidade pequena de páginas, vale assinalar que elas foram desenvolvidas para proporcionar ao usuário uma visualização de dados de forma mais simples e objetiva.

Para ter acesso ao sistema, primeiro é necessário realizar a autenticação. Isso é possível por meio da tela de *login*, conforme demonstrado na Figura 11:

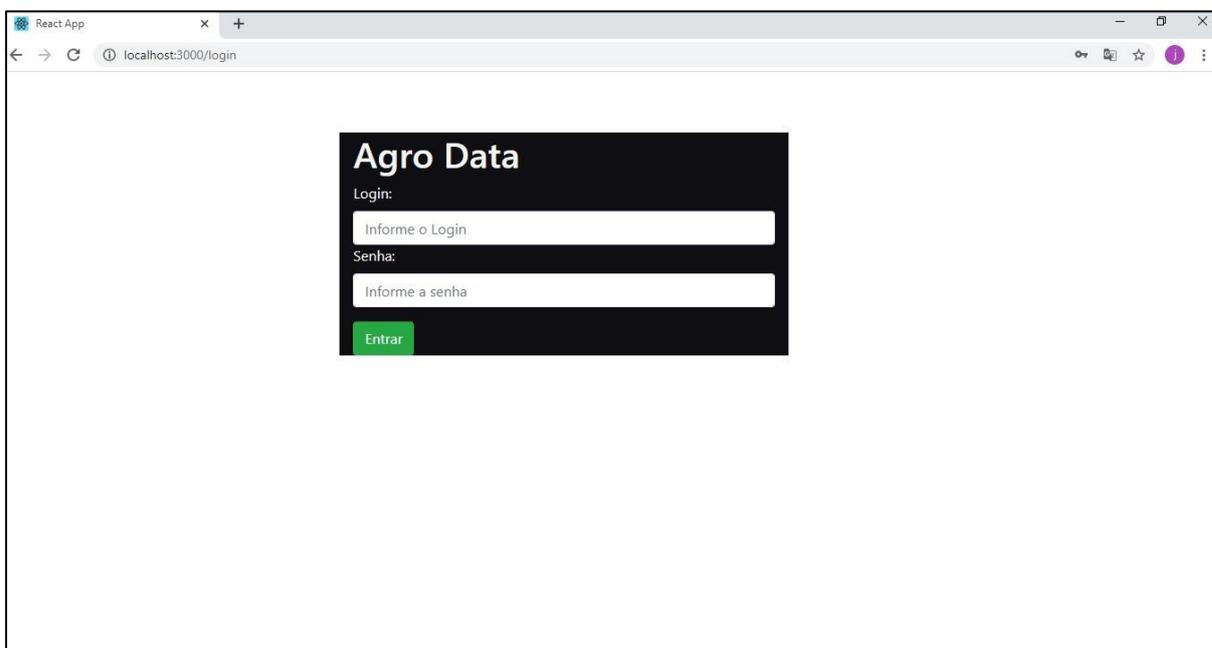


Figura 11: Tela de *Login*

Logo após realizar o *login*, o usuário irá receber um *token* de acesso, conforme consta no trecho do código abaixo:

```
function onSend(){  
    var login = document.getElementById('login');
```

```

    var senha = document.getElementById('senha');
    const response = api.post('/login',{user_login: login.value, user_senha:senha.value});
    if(response != null){
      response.then(response => {
        localStorage.setItem('token', JSON.stringify(response.data.access_token));
        api.defaults.headers.Authorization = `Bearer ${response.data.access_token}`;
        setAuthenticated(true);
        history.push('/home');
      });
    }else{
      history.push('/Login');
    }
  }
}

```

O *token* de acesso encaminhado à requisição garante que todas as requisições dirigidas à API sejam atendidas, além de assegurar o acesso às demais páginas pertencentes à aplicação.

Caso as credenciais enviadas sejam válidas, o usuário será redirecionado para a tela *home*, conforme pode ser verificado na Figura 12.

Area	Temperatura	Humidade	Uv	Data	Condição Atual
area1	29.5°C	1089	100	2020-08-05 14:16:00	ruim

Figura 12: Tela *Home*

Nesta tela, são renderizados os dados do último envio do sistema de captação de dados com o adicional da condição atual e, também, dos dados meteorológicos da

região do plantio. Como demonstrado anteriormente, após a realização de uma requisição dos últimos dados captados pelos sensores, é gerada uma classificação dos dados solicitados a partir de técnicas referentes ao *machine learning*, sendo que por meio deste é informada qual a condição atual do clima para o plantio.

O código abaixo é o responsável pela impressão dos dados na tela:

```
render(){
  const { information } = this.state;
  return(
    <div class="sd">
      <Container>
        <h1>Parametros Atuais:</h1>
        <Table hover danger bordered>
          <thead>
            <th>Area</th>
            <th>Temperatura</th>
            <th>Humidade</th>
            <th>Uv</th>
            <th>Data</th>
            <th>Condição Atual</th>
          </thead>
          <tbody>
            <tr>
              <td>
                {information.name}
              </td>
              <td>
                {information.temp_ambiente + '°C'}
              </td>
              <td>
                {information.humi_solo}
              </td>
              <td>
                {information.raio_uv}
              </td>
              <td>
                {information.data}
              </td>
              <td>
                {information.condicao}
              </td>
            </tr>
          </tbody>
        </Table>
      </Container>
    </div>
  )
}
```

```
    );  
  }  
}
```

Além da página *home*, também é possível acessar a página de relatório. Nesta página é possível realizar uma pesquisa sobre a área de plantio de acordo com o período solicitado. Em um primeiro momento, são renderizados na tela somente dois campos, de modo que cabe ao primeiro inserir a data inicial, enquanto corresponde ao segundo campo inserir a data final. Também é renderizado um botão para que seja possível enviar a requisição à API. A seguir, a Figura 13 ilustra o formato da tela:

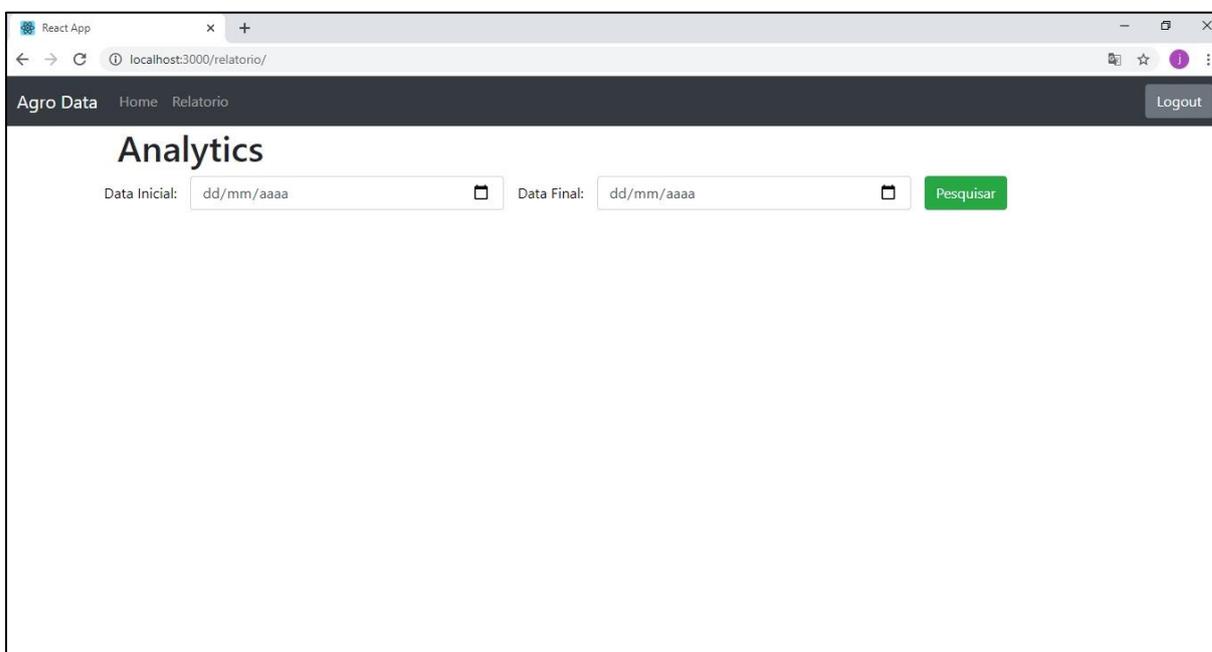


Figura 13: Tela Relatório

Ao inserir a data inicial, a data final e clicar no botão pesquisar, é gerado na tela um relatório panorâmico sobre todas as informações da área do plantio. Algumas das informações apresentadas são: uma tabela que contém média de temperatura, média de umidade, média UV e uma tabela que informa a condição climática geral por dia. Vale considerar que todas essas tabelas são geradas a partir do período informado, conforme consta na Figura 14:

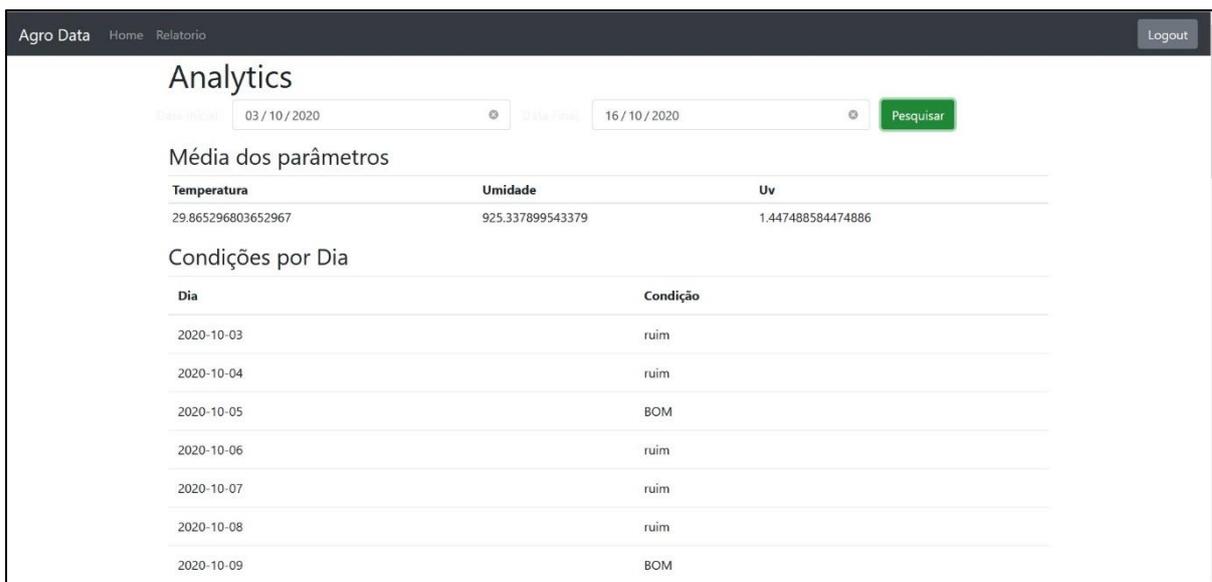


Figura 14: Tabelas do Relatório

Também são gerados três gráficos, sendo eles gráficos atualizados diariamente, tais como: gráfico de média de temperatura, gráfico de umidade e gráfico de raios UV, conforme as figuras 15, 16 e 17:

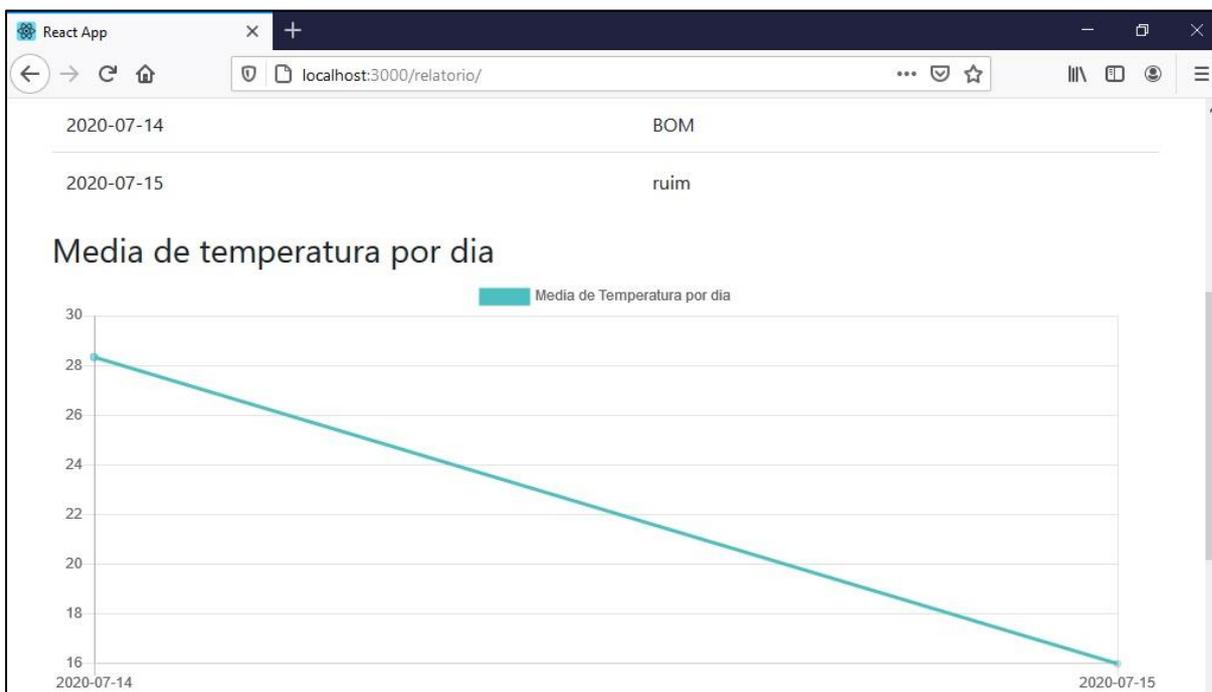


Figura 15: Gráfico de Temperatura

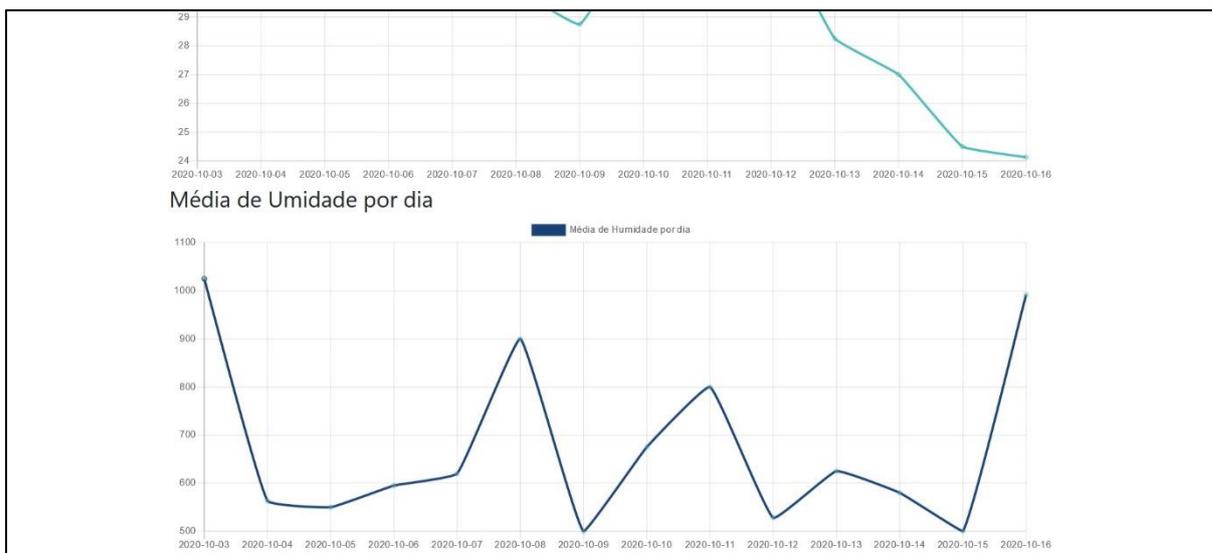


Figura 16: Gráfico de Umidade

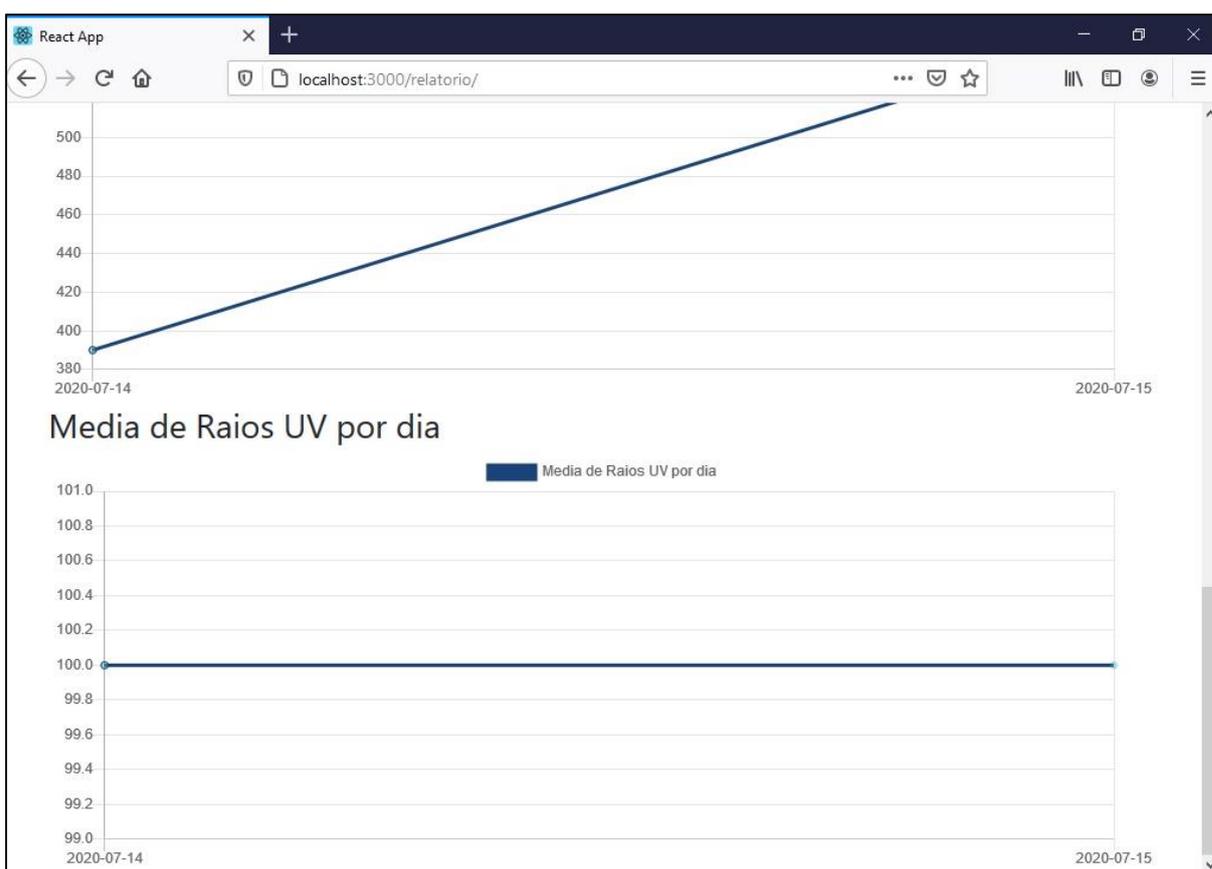


Figura 17: Gráfico UV

Todos os componentes são gerados de forma que possam apresentar fácil visualização e compreensão, sobretudo por parte do usuário, para que o mesmo possa tirar *insights* sobre os dados do plantio.

6. CONCLUSÃO

Nesta pesquisa foi abordada a grande importância do IoT na agricultura, o qual tem se mostrado capaz de revolucionar o setor agrícola ao possibilitar maior produtividade e rentabilidade.

Para realizar protótipos como os deste trabalho, a plataforma do *Arduino* foi avaliada e eleita como um recurso promissor, visto que se trata de uma tecnologia *open source*, de baixo custo, e com grande quantidade de matérias para auxiliar o desenvolvimento de qualquer projeto. Essa potente placa apresenta uma surpreendente gama de interações com os mais diversos sensores, assim como os deste trabalho, a partir dos quais foi construído um pequeno projeto utilizando sensores de monitoramento dos seguintes subsídios: umidade e temperatura do ar e umidade e temperatura da terra. Ademais, também foi utilizado um sensor capaz de captar raios UV.

Em conjunto com o *Arduino*, foi utilizado o *RaspBerry*, sendo este responsável por garantir a execução da aplicação *backend* e pelo armazenamento dos dados coletados por meio dos sensores. Por fim, foi construída uma plataforma *web* para o cliente final. A partir desta plataforma será possível que o agricultor monitore sua plantação de maneira clara e coesa, além de poder realizar a emissão de relatórios com informações relevantes sobre o índice de favorabilidade climática.

Considerando os resultados obtidos a partir da finalização deste trabalho, pode-se afirmar que ele apresenta contribuições pertinentes para a realização de estudos futuros sobre IoT, agricultura inteligente e *Data Science*, com base em tecnologias *open source*, as quais têm se mostrado cada vez mais promissoras para o desenvolvimento de projetos com o objetivo de promover não só a maior rentabilidade e produtividade dos alimentos no cenário mundial, mas também a consciência sustentável a partir dos recursos tecnológicos.

Segundo os autores² do artigo “Agricultura, Sustentabilidade e Tecnologias”, a agricultura inteligente no Brasil é indispensável para

[...] alavancar os potenciais econômicos e de sustentabilidade da nova bioindústria, tanto para intensificar a produção de alimentos, fibras e energia limpa, como para desenvolver uma nova e pujante indústria de químicos renováveis, sem competição com a produção de alimentos. As nossas instituições de pesquisa científica e tecnológica e as empresas do

² Maurício Antunes Lopes (Diretor Executivo de Pesquisa e Desenvolvimento da Embrapa no ano de 2012. Atualmente, integra o Grupo Assessor ao Direito Geral da FAO, em Roma); Elísio Contini (Pesquisador da Secretaria de Inteligência e Relações Estratégicas Embrapa).

agronegócio precisam olhar adiante, vencer obstáculos, ultrapassar barreiras. Não há tempo a perder! (2012, p. 34).

Como se pode verificar, a agricultura inteligente trata-se de uma área promissora em crescimento no Brasil, de modo que a realização de novas pesquisas e o desenvolvimento de projetos cada vez mais amadurecidos e sofisticados se fazem indispensáveis para expandir as bases de produção alimentícia, bem como as práticas sustentáveis no agronegócio, sendo que a partir dessas práticas será possível usufruir dos recursos naturais com menor impacto ambiental.

6.1 TRABALHOS FUTUROS

Tecnologias relacionadas a IOT, agricultura inteligente e análise de dados têm constantemente apresentado melhorias significativas, o que pode ser evidenciado a partir do lançamento de sensores diversificados, de técnicas agrícolas inteligentes embasadas na tecnologia de informação e de métodos classificatórios cada vez mais eficientes e precisos.

Entretanto, tendo em vista os aspectos que ainda precisam ser aprimorados, sugere-se a importância de adotar uma nova função, isto é, atribuir ao projeto do *arduino* um sistema de rega inteligente que possa utilizar como parâmetro os dados obtidos pelo sensor de umidade.

Outra medida igualmente necessária é a adoção de novos sensores, preferencialmente de sensores aptos a monitoramento agrícola, como exemplo, sensor de medição de ph da terra. Com tais sensores, seria possível promover um novo parâmetro para as análises, além de contribuírem para o maior rigor referente à precisão e confiabilidade nos dados apresentados ao usuário.

7. REFERÊNCIAS

AGSOLVE. Agricultura Inteligente: Como funciona e qual sua importância? **AgSolve**, 2017. Disponível em: <<https://www.agsolve.com.br/dicas-e-solucoes/10375/agricultura-inteligente-como-funciona-e-qual-sua-importancia>>. Acesso em: 12 Outubro 2019.

ANTONIO A.F. LOUREIRO, J. M. S. N. E. A. Rede de Sensores sem Fio, Belo Horizonte, 2014.

BOAS PRATICAS AGRONÔMICAS. Tecnologia na agricultura: maior sustentabilidade na produção agrícola. **Boas Praticas Agronômicas**, 2019. Disponível em: <<https://boaspraticasagronomicas.com.br/noticias/tecnologia-na-agricultura/>>. Acesso em: 12 Outubro 2019.

CAVALCANTI, M. PROTÓTIPO DE SMART PARKING PARA CIDADES INTELIGENTES E MOBILIDADE URBANA, Assis, 2019.

CHADE, J. Estadão. **economia Estadão**, 17 Setembro 2018. Disponível em: <<https://economia.estadao.com.br/noticias/geral,brasil-passa-a-ser-3-maior-exportador-agricola-mas-clima-ameaca-futuro,70002506105>>.

CHASE, J. The Evolution of the Internet of Things. **TI**, 2013. Disponível em: <<http://www.ti.com/lit/ml/swrb028/swrb028.pdf>>. Acesso em: 2020.

CIELEN, D.; MEYSMAN, A.; ALI, M. **Introducing Data Science: Big Data, Machine Learning, and more, using Python tools**. 1. ed. New York: Manning Publications, v. 1, 2016.

CIRIACO, D. O que é Raspberry Pi? **canaltech**, 2015. Disponível em: <<https://canaltech.com.br/hardware/o-que-e-raspberry-pi/>>. Acesso em: 12 outubro 2019.

CONTINI, Elísio; LOPES, Maurício Antunes. Agricultura, Sustentabilidade e Tecnologia. In: Agroanalysis/Especial Embrapa, fevereiro de 2012.

COSTA, C. L.; OLIVEIRA, L.; MÓTA, L. M. S. Internet das coisas (IOT): um estudo exploratório em agronegócios. **VI Simpósio da Ciência de Agronegócio**, Porto Alegre, 25 Outubro 2018.

CRUZ, L. C. D. DATA SCIENCE:DESENVOLVIMENTO DE APLICAÇÃO PARA ANÁLISE DE DADOS, Assis, 2018.

EBERMAM, E. et al. **Programação para leigos com Raspberry Pi**. 1. ed. João Pessoa: IFPB, v. 1, 2017.

EVANS, M.; NOBLE, ; HOCHENBAUM, J. **Arduino em Ação**. 1. ed. São Paulo: Novatec, v. 1, 2013. 25-26 p.

FAKHURDIN, H. Smart Agriculture: 13 Trends To Watch Out For. **Teks Mobile**, 2017. Disponível em: <<https://teks.co.in/site/blog/smart-agriculture-13-trends-to-watch-out-for/2/>>. Acesso em: 12 Outubro 2019.

GARTNER. Gartner Says Competition Is Increasing to Be the IoT Gateway to the Connected Home. **Gartner**, 2015. Disponível em: <<https://www.gartner.com/en/newsroom/press-releases/2015-08-06-gartner-says-competition-is-increasing-to-be-the-iot-gateway-to-the-connected-home>>. Acesso em: 12 Outubro 2019.

MANZANO, J. A. **Programação de computadores com C/C++**. 1. ed. São José dos Campos: Érica, v. 1, 2014.

MCROBERTS, M. **Arduino Básico**. 1. ed. São Paulo: NovaTec, v. 1, 2011.

MEOLA, A. Smart Farming in 2020: How IoT sensors are creating a more efficient precision agriculture industry. **businessinside**, 2019. Disponível em: <<https://www.businessinsider.com/smart-farming-iot-agriculture>>. Acesso em: 23 Fevereiro 2020.

SOLAGNA, E. A.; LAZZARETTI, A. T. UM ESTUDO COMPARATIVO ENTRE O MONGODB E O POSTGRESQL, 2016.

WHEAT, D. **Arduino Internals**. 1. ed. New York: Technology in Action, v. 1, 2011.