

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

JOÃO PAULO RODRIGUES BUENO

PLATAFORMA EM NUVEM PARA GESTÃO DE DISPOSITIVOS DE ΙΟΤ

Assis/SP 2018



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

JOÃO PAULO RODRIGUES BUENO

PLATAFORMA EM NUVEM PARA GESTÃO DE DISPOSITIVOS DE IOT

Projeto de pesquisa apresentado ao curso de Análise e Desenvolvimento de Sistemas 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 Paulo Rodrigues Bueno Orientador: Dr. Almir Rogério Camolesi

Assis/SP 2018

FICHA CATALOGRÁFICA

BUENO, João Paulo Rodrigues. **PLATAFORMA EM NUVEM PARA GESTÃO DE DISPOSITIVOS DE IOT** / João Paulo Rodrigues Bueno. Fundação Educacional do Município de Assis –FEMA – Assis, ano 2018.

1. IoT na Agricultura.

CDD: 600 Biblioteca da FEMA

PLATAFORMA EM NUVEM PARA GESTÃO DE DISPOSITIVOS DE IOT

JOÃO PAULO RODRIGUES BUENO

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: ____

Dr. Almir Rogério Camolesi

Examinador:

Esp. Domingos de Carvalho Villela Junior

Assis/SP 2018

DEDICATÓRIA

Dedico este trabalho especialmente aos meus pais Marcos e Mariana, aos quais sempre me deram amor e carinho, e me orientaram a encontrar uma formação profissional que eu identifica-se para atuar com virtude e humildade, agradeço por terem acreditado em mim. Meus pais são minha inspiração por serem pessoas maravilhosas e tão determinadas em seus objetivos.

Dedico esse trabalho tambem ao meu irmão Marcos Paulo ao qual eu tenho um enorme amor e carinho, por sempre estar presente em minha vida, me encorajando, dando conselhos e torcendo pelo meu futuro.

AGRADECIMENTOS

Agradeço primeiro a Deus e ao universo por conspirar ao meu favor.

Agradeço a Minha namorada Ana beatriz pelo amor, paciencia e compreensão durante esses meses, por estar me dedicando para que este trabalho fosse possivel.

Agradeço meu Sogro Marcelo por me incentivar sendo um empreendedor que busca sempre superar os desafios da vida se inovando, e as varias conversas sobre tecnologia enquanto tomavamos cafe. Agradeço aos amigos Leticia, Beto, Thais, Bruno, Mauro, Osmar e Gabriel aos quais fico feliz por ter conhecido na faculdade, as varias risadas de assuntos diversos e coca-cola no intervalo. Agradeço ao Jonas, Matheus e Sabrina otimos amigos que me ajudarão muitas vezes quando precisava durante a faculdade.

Agradeço ao Harrison e o Rafael por me darem a oportunidade de participar da equipe de Direção Tecnica de Informatica da Faculdade UNESP sendo estagiario, me ajudou quando mais precisei nesses 2 ultimos anos da faculdade, e aos jogos de RPG e amizades que fiz sempre serão lembrados.

Agradeço ao Amiliton pela amizade e pelo presente da placa Raspberry para que eu possa desenvolver outros projetos, vou usar-la com sabedoria.

Agradeço meu Vizinho e amigo Vilmar pelas mudas de alface fornecidas para os testes nesse trabalho.

Agradeço ao meu Professor e Orientador Almir nos conselhos, nos ensinamentos ao longo da faculdade e na orientação para que esse trabalho fosse concluido. Obrigado pelos ensinamentos Mestre "Jedi".

"Inteligência é a capacidade de se adaptar á mudança."

Stephen Hawking

"A mente que abre uma nova janela, jamais volta ao seu tamanho original."

Albert Einstein

RESUMO

Com a Chegada da 4° Revolução Industrial a tecnologia tem transformado todos os aspectos do processo das atividades dos seres humanos contribuindo para melhorar nosso modo de vida. A Agricultura tem sofrido processos de transformações por essa nova Onda, os processos na produção em todos seus pontos tem sido aperfeiçoados mudando as rotinas braçais por maquinas programadas que com a ajuda da internet sendo integradas ao campo muitas possibilidades tem se tornado realidade.

O Presente trabalho demonstrará a integração das Tecnologias usando os conceitos de Internet das Coisas e Desenvolvimento Hibrida de Aplicações para maximizar as atividades no Campo, podendo gerir Variedades o seu desenvolvimento, monitorar e controlar o consumo de insumos alcançando melhores resultando consumindo as informações do banco de dados em Nuvem. O Processamento do Big Data gerada no campo pelos dispositivos mostrará as respostas para melhorar a Agricultura do Futuro.

Palavras-chave: Internet das Coisas, Arduino, Tecnologia no campo.

ABSTRACT

With the arrival of the 4th Industrial Revolution technology has transformed all aspects of the human activities process, contributing to improve our way of life. Agriculture has undergone transformation processes for this new wave, processes in production at all points have been perfected changing as manual routines for makeup programmed with the help of internet being integrated into the field many possibilities have become a reality.

The present work will demonstrate the integration of the Technologies using the Internet concepts of Things and Hybrid Application Development to maximize how activities in the Field, being able to manage Varieties in their development, monitoring and control of consuming inputs reaching better resulting consuming like information of the bank of data in the Cloud. The Big Data Processing generated in the field by the devices shown as responses to improve the Agriculture of the Future.

Keywords: Internet of Things, Arduino, Field Technology.

LISTA DE ILUSTRAÇÕES

Figura 1: Astecas nas Chiampas.	17
Figura 2: Semeadeira Mecânica de Jethro Tull	
Figura 3: Maquina Reaper de 1887 de McCormick	
Figura 4: Painel de Maquina Agrícola.	23
Figura 4: Maquina pulverizando área de plantio	24
Figura 6: Estufa climatizada	25
Figura 7: Arquitetura SOFEA com aplicação web utilizando serviço Rest API	
Figura 8: Aplicação Mobile utilizando serviço Restful integrado ao banco de Dados	(back-
end-as-a-Service).	28
Figura 9: Dados em formato JSON	29
Figura 10: Android SDK Manager	
Figura 11: Criando Projeto Ionic.	32
Figura 12: Processo do comando Ionic serve	32
Figura 13: Projeto Ionic em execução	33
Figura 14: Estrutura de projeto Ionic	34
Figura 15: Arquivo SCSS	35
Figura 16: Arquivo HTML parte 1	
Figura 17: Arquivo HTML parte 2	
Figura 18: Arquivo HTML parte 3	
Figura 19: Arquivo HTML parte 4	40
Figura 20: Arquivo HTML parte 5	40
Figura 21: Tela Dispositivo parte 1	41
Figura 22: Tela Dispositivo parte 2	41
Figura 23: Tela Dispositivo parte 3	42
Figura 24: Tela Dispositivo parte 4	42

LISTA DE ILUSTRAÇÕES

Figura 25: Arquivo Dispositivo model Ts	43
Figura 26: Arquivo Dispositivo Pagina Ts parte 1	44
Figura 27: Arquivo Dispositivo Pagina Ts parte 2	46
Figura 28: Arquivo Dispositivo Pagina Ts parte 3	47
Figura 29: Arquivo Dispositivo Pagina Ts parte 4	48
Figura 30: Arquivo Dispositivo Pagina Ts parte 5	49
Figura 31: Arquivo Dispositivo Service Ts	50
Figura 32: Interface Firebase parte 1	52
Figura 33: Interface Firebase parte 2	52
Figura 34: Interface Firebase parte 3	53
Figura 35: Interface Firebase parte 4	53
Figura 36: Arquivo App module Ts	54
Figura 37: Processo de Compilação utilizando Cordova	55
Figura 38: Tecnologias que integram o IOT	56
Figura 39: Microcontrolador Arduino	59
Figura 40: Componentes adicionais.	59
Figura 41: Placa Esp8266 12e Especificações	60
Figura 42: Diagrama de conexão dos componentes eletronicos	62
Figura 43: Passo 1 instalar biblioteca ESP8266	62
Figura 44: Passo 1.1 instalar biblioteca ESP8266	63
Figura 45: Passo 2 instalar biblioteca ESP8266	63
Figura 46: Passo 2.1 instalar biblioteca ESP8266	64
Figura 47: Passo 3 instalar biblioteca ESP8266	64
Figura 48: Passo 4 instalar biblioteca ESP8266	65

Figura 49: Codigo Olá Mundo	65
Figura 50: Codigo Salvar int na EEPROM	66
Figura 51: Codigo Leitura simples na EEPROM	67
Figura 52: Codigo Salvando variavel String na EEPROM	68
Figura 53: Codigo Leitura de variavel String na EEPROM	69
Figura 54: Sensor umidade solo	70
Figura 55: Codigo Sensor Solo	71
Figura 56: Modulo Data RTC DS3231	72
Figura 57: Codigo Salvando Rtc ds3231	73
Figura 58: Codigo Recuperando dados do Rtc ds3231	74
Figura 59: Codigo Conectando ESP8266 12E com Wifi	76
Figura 60: Estrutura de Bases de dados e coleções no Firebase	77
Figura 61: Codigo Conectando ESP8266 12E ao Firebase	78
Figura 62: Codigo Salvar e Recuper dados ESP8266 12E com Firebase parte 1	79
Figura 63: Codigo Salvar e Recuper dados ESP8266 12E com Firebase parte 2	80
Figura 64: Diagrama de Caso de Uso Usuario.	83
Figura 65: Diagrama de Caso de Uso Dispositivo.	84
Figura 66: Diagrama de Classe sistema	84
Figura 67: Plantio túnel baixo de plástico pelo Embrapa	85
Figura 68: Plantio campo aberto em canteiros com uso de cobertura de plástico e de irrigação ou gotejamento.	sistema 86
Figura 69: Projeto no Campo	87
Figura 70: Modelo de Arquitetura de IOT para o campo	88
Figura 71: Aplicação e Dispositivo em execução	90

LISTA DE TABELAS

Tabela 1: Cronograma de atividades TCC 2017-2018	47
--	----

LISTA DE ABREVIATURAS E SIGLAS

- IOT Internet of Things (Internet das Coisas).
- Embrapa Empresa Brasileira de Pesquisa Agropecuária.
- WEB Rede mundial de Computadores.
- Wi-Fi Wireless Fidelity.

DEDIC	CATÓRIA	5
AGRA	DECIMENTOS	6
LISTA	DE TABELAS	13
1.	INTRODUÇÃO	17
1.1.	OBJETIVOS	
1.2.	PÚBLICO-ALVO	
1.3.	JUSTIFICATIVA	19
2.	AGRICULTURA	20
2.1	AGRICULTURA DE PRECISÃO	
3.	MOBILE	266
3.1.	ARQUITETURAS SOFEA	
3.2.	REST E RESTFUL	
3.3.	ANDROID SDK	30
3.4	IONIC FRAMEWORK	31
3.5.	COMPONENTES E ESTRUTURAS DE PROJETO IONIC	333
3.6.	O QUE É LINGUAGEM SCSS	34
3.7.	O QUE É LINGUAGEM HTML	35
3.8.	O QUE É LINGUAGEM TYPESCRIPT	43
3.9.	CONEXÃO IONIC COM FIREBASE	51
3.10	CORDOVA FRAMEWORK	55
4.	ЮТ	556
4.1	PREPARANDO A ARDUINO IDE PARA USAR O ESP	62
4.2	CRIANDO O SCRIPT "HELLO WORLD"	65
4.3	O QUE É A EEPROM	66
4.4	SALVANDO VARIAVEL DO TIPO INT NA EEPROM	66
4.5	REALIZANDO LEITURA DA VARIAVEL INT NA EEPROM	667
4.6	SALVANDO VARIAVEL DO TIPO STRING NA EEPROM	68
4.7	REALIZANDO LEITURA DE VARIAVEL STRING NA EEPROM	
4.8	O QUE SÃO SENSORES	70
4.9	O QUE É O MODULO RTC DS3231	72
4.10	CONECTANDO ESP8266 12E COM A INTERNET	75
4.11	CONECTANDO ESP8266 12E COM WIFI	75
4.12	FIREBASE COM ESP8266 12E	77
4.13	SALVAR E BUSCAR DADOS PELO ESP8266 12E NO FIREBASE .	79

SUMÁRIO

5.	LEVANTAMENTO DOS REQUISITOS	
5.1.	LISTA DE EVENTOS	
5.2.	DIAGRAMA DE CASO DE USO	83
5.3.	DIAGRAMA DE CLASSE	83
5.4	PROPOSTA DE TRABALHO	85
6.	CONCLUSÃO	88
7.	CRONOGRAMA	91
8.	REFERÊNCIAS	92

1. INTRODUÇÃO

Com os avanços Tecnológicos aumentando exponencialmente novas soluções surgem se incorporando em várias áreas da sociedade e da indústria. Essas soluções vêm para resolver problemas ou adicionar melhorias nos processos das atividades, seja por uma máquina executando tarefa de construir um automóvel, um aplicativo em Smartfone realizando pagamento da fatura online maximizando o tempo da pessoa.

Uma das Inovações mais recentes em crescimento consta a Internet das Coisas¹ (IOT) se integrando a quase tudo como exemplo: uma geladeira inteligente que se comunica com o dispositivo móvel avisando o proprietário que um produto acabou, ou também uma máquina agrícola executando uma tarefa no campo ao qual foi destinada se comunicando através da internet (GILCHRIST, 2016).

Com a visão em direção ao campo alguns exemplos de uso do IOT como o GPS (Global Position System)² em maquinas Agrícolas para localização no campo, e o uso dos satélites capturando imagens para mapeamento de áreas a fim de filtrar e trazer resultados sobre a superfície do solo.

Outro Artigo Segundo Embrapa³ define sobre as utilizações da tecnologia no campo.

Entre as Tecnologias Agricultura de Precisão mais utilizada hoje no Brasil, e também no mundo, estão os monitores de colheita de grãos para gerar os mapas de produtividade, as ferramentas de direcionamento (barras de luz e piloto automático), e a semeadora/adubadora e adubadora /calcareadora para aplicação de insumos a taxas variadas ("Variable Rate Technology")⁴. Todas estas ferramentas são úteis para detectar, medir e controlar a variabilidade espacial.

A variabilidade espacial pode ocorrer devido a vários fatores, como manchas de solo, áreas com diferente disponibilidade de água ou nutrientes, camadas compactadas,

¹ Disponível em:< <u>https://aws.amazon.com/pt/iot/what-is-the-internet-of-things/</u> > Acesso em: 28/11/2017.

² Disponível em:< <u>https://www.gps.gov/</u> > Acesso em: 06/11/2017.

³ Disponível em:< <u>http://www.senar.org.br/agricultura-precisao/artigos-e-palestras/artigo-a-agricultura-de-precisao-como-ferramenta-para-o-produtor-rural</u> /> Acessado em 04/11/2017.

⁴ Disponível em: < <u>https://en.wikipedia.org/wiki/Variable_Rate_Technology</u> > Acesso em: 09/11/2017.

reboleiras de plantas daninhas ou pragas, ou ainda baixa qualidade das operações agrícolas.

Isso tudo reflete na produção e o mapa de produtividade é o registro destas variações. É por meio do mapa que o agricultor pode estudar e planejar a estratégia de investimento para cada região da sua propriedade.

O Desenvolvimento da Aplicação e do Protótipo de dispositivo foi em prol de contribuir no avanço e estudo para agricultura possibilitando agregar novos conceitos de usabilidade das tecnologias atuais para forma de pensar da agricultura do futuro.

1.1. OBJETIVOS

O objetivo deste trabalho foi analisar compreender e desenvolver uma aplicação hibrida de consumo de dados em nuvem que possa verificar os dados obtidos do protótipo de dispositivo, realizar análise dos dados criando gráficos, ter acesso em tempo real da execução das tarefas do protótipo de dispositivo e histórico do mesmo no campo teste, gerenciar o cadastro do campo teste e do dispositivo. A solução tem como finalidade criar novas formas de executar tarefas para o setor da Agricultura tornando algumas funções automatizadas, controlando e economizando recursos naturais no processo de produção, e a análise dos dados coletados para aprimoramento da solução desenvolvida e da Produção.

1.2. PÚBLICO-ALVO

A solução tem como alvo empresas Agrícolas que desejam automatizar e ter controle do fluxo de dados do progresso de seu campo ou lavoura em tempo real, gerando um melhor controle das atividades executadas e maior rendimento no desenvolvimento do produto.

1.3. JUSTIFICATIVA

Nos Dias atuais automatizar a execução de algumas atividades traz melhores resultados na produção, melhora o desempenho do colaborador para executar tarefas mais importantes que trazem impacto a empresa. A solução proporcionara um controle das atividades executadas, economia dos recursos naturais da produção afim de não gerar desperdício, e a integração da tecnologia no campo em benefício do aprimoramento das técnicas para agricultura.

2. AGRICULTURA

A Agricultura é uma cultura de subsistência utilizada por milhares de anos pelos homens como fonte de alimento, tendo como objetivo cultivar plantas transformando em alimentos, bebidas, medicamentos, energia e em muitas outras aplicações diversas. A origem se inicia em 10 mil anos atrás aproximadamente aonde o homem deixa de serem um Nômade dando origem as primeiras civilizações, esse ponto é chamado de Revolução Agrícola no Neolítico surgindo a Agricultura (MAZOYER,2008).

Na Agricultura Neolítica surgiram os sistemas Pastoreio e o cultivo derrubada-queimada, Pastoreio predominando em áreas de estepes e nas savanas, derrubada-queimada sendo usadas em áreas de florestas-temperadas e tropicais. Em áreas áridas sistemas Agrários hidráulicos foram desenvolvidos e usados em cultivos irrigados como nos vales do Nilo, em zonas tropicais úmidas também adotaram sistemas hidráulicos se adaptando ao estado do terreno para o cultivo.

Em áreas intertropicais realizaram desmatamento e o sistema de cultivo temporário de variedades foi adotado, usando ferramentas como enxada. Nas regiões temperadas na Europa após o desmatamento inicio os sistemas pós-florestais, depois disso começo a revolução Agrícola até os dias atuais.

No Século XIV na sociedade Asteca em Tenochtitlán a Agricultura era a base da economia, utilizava uma Técnica chamada de chiampas eram ilhas de cultivo, aonde colhiam espécies como milho, pimenta, tomate e cacau. O cacau era utilizado como moeda na sociedade para compra todo tipo de produto existente.



Figura 1: Astecas nas Chiampas. **Fonte:** https://dougnahistoria.blogspot.com.br/2015/09/astecas.html

Antes de falarmos sobre a atual Agricultura, olhando para o passado em meados de 1700 um agricultor e inventor inglês chamado Jethro tull teve seu importante papel do desenvolvimento Agrário. Viajou por vários países da Europa como França, Alemanha e Holanda, observando o processo de cultivo das espécies e a forma de aproveitamento do solo usada pelos camponeses, então inventou a semeadeira mecânica puxada por cavalo que abria o solo e depositava as quantidades necessárias de sementes aumentando a produtividade, se tornando base para evolução das maquinas subsequentes.



Figura 2: Semeadeira Mecânica de Jethro Tull. **Fonte:** https://fineartamerica.com/images/artworkimages/medium/1/tull-seed-drill-1701-granger.jpg

Em 1831, nos Estados Unidos um inventor chamado Cyrus McCormick também Agricultor e inventor criou uma ceifadeira mecânica (colheitadeira de cereais) puxada a cavalo que podia colher cereal por áreas extensas de campo mais rápido que homens utilizando foices.



Figura 3: Maquina Reaper de 1887 de McCormick. **Fonte:** https://www.granger.com/results.asp?image=0176647&screenwidth=1366

2.1. AGRICULTURA DE PRECISÃO

Com aperfeiçoamento das técnicas na agricultura e o crescente aumento da demanda por produtos de boa qualidade, os recursos natural ficando cada dia mais escasso, a economia da energia no processo das tarefas e produtos, o gerenciamento no cultivo das variedades de alimentos com menos insumos afim de um consumo racional vem a Agricultura de precisão (MOLIN, 2015).

Com o surgimento desse novo método foi possível cobrir um território maior de plantio, administrar grandes áreas com controle otimizado, aumentar a produção das lavouras e pomares, conseguindo um lucro superior aos métodos tradicionais de desenvolvimento das variedades.

A otimização dessas áreas são realizadas por maquinas agrícolas programados sendo os pilotos solicitados em momentos específicos para exercer determinadas tarefas.

As Maquinas possuem sistema com terminal embarcado que controla a navegação, sensores mostram os volumes ideais de insumos para as variedades, desempenho das maquinas nas atividades, sistema de internet e telecomunicações para salvar a coordenada de localização das mesmas e tarefa a qual estão executando em tempo real.



Figura 4: Painel de Maquina Agrícola.



Figura 5: Maquina pulverizando área de plantio. Fonte: http://jcrs.uol.com.br/_conteudo/2017/08/especiais/expointer_2017/581301-maquinas-agricolasestao-mais-inteligentes-e-proativas.html

Todo tipo de tecnologia criada com a finalidade de aperfeiçoar as tarefas no setor agrícola, economizando os recursos naturais, a energia e para gerar produtos de qualidade em beneficio para saúde é considerado uma pratica de Agricultura de precisão. Exemplo do uso de sensor de temperatura para climatizar estufa mantendo o clima favorável para o desenvolvimento da variedade cultivada.



Figura 6: Estufa climatizada. Fonte: http://faelandreatto09.wixsite.com/rrautomacao/fotos

3. MOBILE

Os dispositivos mobile ganharam destaque na ultima década, os celulares antes utilizados para somente realizarem ligações e com poucas funcionalidades que tinham receberam um upgrade, melhorias se transformando em smartphones potentes computadores de mão que cabem em nossos bolsos.

Os smartphone atuais possuem varias funcionalidades nativas como tela sensíveis ao toque chamadas touchscreen, funções do próprio celular em realizar chamadas e agendas, bluetooth tendo a identificação do equipamento possibilitando se conectar a outros dispositivos e outros smartphones para se compartilhar dados como fotos, conexão com a internet por sinal de wi-fi, posicionamento global como GPS, reconhecimento de voz, câmera fotográfica e vídeo. Além das funcionalidades nativas do dispositivo existem inúmeros aplicativos disponibilizados gratuitamente ou para compra no acervo das empresas, como Google Play Store⁵ e IOS App Store⁶.

Vários sistemas operacionais mobiles foram surgindo pelas grandes corporações como Symbian OS da Nokia⁷, Android da Google⁸, RIM da Blackberry⁹, Windows Phone da Microsoft¹⁰, IOS da Apple¹¹ entre outros.

Os aplicativos disponibilizados no Store das empresas são desenvolvidos especificamente para o Sistema Operacional que roda no Dispositivo, cada Sistema Operacional sua arquitetura e sua linguagem de programação a qual os aplicativos podem ser executados. No caso do Android os aplicativos são em formato APK e são programados em linguagem Java¹², usando IDE Android Studio¹³ para criação e implementação, IDE significa Integrated Development Environment em português chamado Ambiente de desenvolvimento integrado.

⁵ Disponível em: < <u>https://play.google.com/store?hl=pt_BR</u> > Acesso em: 13/03/2018.

⁶ Disponível em: < <u>https://www.apple.com/br/ios/app-store/</u> > Acesso em: 13/03/2018.

⁷ Disponível em: < <u>https://www.nokia.com/pt_int</u> > Acesso em: 13/03/2018

⁸ Disponível em: < <u>https://www.android.com/</u> > Acesso em: 13/03/2018.

⁹ Disponível em: < <u>https://us.blackberry.com/</u> > Acesso em: 13/03/2018

¹⁰ Disponível em: < <u>https://www.microsoft.com/pt-br</u> > Acesso em: 13/03/2018.

¹¹ Disponível em: < <u>https://www.apple.com/br/ios/ios-11/</u> > Acesso em: 13/03/2018.

¹² Disponível em: < <u>https://www.java.com/pt_BR/</u> > Acesso em: 13/03/2018.

¹³ Disponível em: < <u>https://developer.android.com/studio/index.html?hl=pt-br</u> > Acesso em: 13/03/2018

Além do Android Studio existem frameworks para desenvolvimento hibrido ao qual o programador pode criar um único código compilar e gerar executáveis que rodam em mais de um sistema operacional. Esses frameworks precisam das versões dos sistemas operacionais para gerar o APP, exemplo dos frameworks mais conhecidos como IONIC¹⁴ e REACTJS¹⁵.

3.1. ARQUITETURAS SOFEA

Desenvolvimento de aplicações para plataforma mobile utilizando modelo SOFEA¹⁶ ao qual possui a View é constituída por programação componentes HTML, JavaScript e CSS, todo processamento é executado na View somente os dados são trafegados, e fazendo requisição de dados do Serviço RESTful ao qual pode possuir vários serviços disponíveis para consumo de dados, a aplicação RESTful pode ser construída usando qualquer tipo de linguagem como Codeigniter em PHP ou SpringBoot em Java, ou Também utilizar Serviços de Banco de Dados chamado Back-End-As-A-Service que provem serviços Embutidos como o Firebase. O Serviço por sua vez acessa o banco de dados em Nuvem empacota esses dados em formatos JSON JavaScript Object Notation – Notação de Objetos JavaScript um formato rápido e pratico para se consumir dados, retornando para o Serviço e depois para aplicação que fez requisição.

¹⁴ Disponível em: < <u>https://ionicframework.com/</u> > Acesso em: 13/03/2018.

¹⁵ Disponível em: < <u>https://reactjs.org/</u> >Acesso em: 13/03/2018.

¹⁶ Disponível em: < <u>https://dzone.com/articles/service-oriented-front-end-architecture-sofea</u> >Acesso em: 10/09/2018.



Figura 7: Arquitetura SOFEA com aplicação web utilizando serviço Rest API. Fonte: Autoria Própria.



Figura 8: Aplicação Mobile utilizando serviço Restful integrado ao banco de Dados (back-end-as-a-Service). Fonte: Autoria Própria.





3.2. REST E RESTFUL

Rest é um termo criado por Roy Fielding¹⁷ ao qual desenvolveu uma arquitetura de software para uso de sistemas que operassem em rede, o uso dessa arquitetura facilita a transição de dados ou conteúdos que são solicitadas entre as aplicações pela internet. Rest em inglês quer dizer Representational State Transfer para português "Transferencia de Estado Representacional", toda aplicação web é vista como conjunto de recursos e a Representação do estado são os dados, recursos que podem ser enviados e recebidos.

O Protocolo usado para a transferência desses recursos é o HTTP Hyper Text Transfer Protocol, significa Protocolo de Transferencia de HiperTexto que permite a transferência de dados na rede de computadores e a internet, usa a URL com endereço do site para comunicar com Servidor Web ou Serviço que mantém os dados.

Quando falamos sobre REST estamos abordando o conceito do modelo e as características dessa Arquitetura, já quando falamos sobre um sistema que tem como base os conceitos de arquiterura REST é chamado de RESTful (DIAS, 2016).

Para fazer a comunicação e transmissão dos dados usamos o Protocolo de HTTP, os dados podem ser transmitidos em formato JSON, com endereço do Serviço RESTful junto com as chaves e senhas da conta a aplicação também precisamos chamar o métodos para obter os dados. Métodos como GET que traz os dados do banco, POST que inclui um novo dado, PUT altera um registro já existente e DELETE que exclui um registro já criado no banco.

¹⁷ Disponível em: < <u>https://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm</u> > Acesso em: 10/09/2018.

Esse modelo de arquitetura traz benefícios como visibilidade, confiabilidade e escalabilidade. Outra questão é que o usuário não precisa saber como os dados estão alocados no servidor somente consumir as informações, isso deixa a aplicação separada do serviço tornando ambas independentes do tipo de tecnologia usada para seu desenvolvimento.

3.3. ANDROID SDK

O Android SDK¹⁸ consiste em um kit de desenvolvimento de software para Android que inclui ferramentas de desenvolvimento, e bibliotecas para gerar app, uma ferramenta open source mantida pela Empresa Google. A base do Kernel ao qual o Android utiliza é composta pelo Linux, e seu código é programado em linguagem Java.

Quando incluímos na lista do Android sdk versões do sistema operacional devemos aceitar os termos de uso e licença que são obrigatórias. Para criar um App precisamos da versão especifica do sistema operacional que conterá todas as bibliotecas necessárias para que a aplicação seja executada com sucesso no dispositivo.

Utilizamos a ferramenta SDK Manager¹⁹ que gerencia todos os pacotes e versões do Android que incluímos na lista, nele podemos incluir excluir e atualizar versões dos sistemas Operacionais disponíveis.

Android SDK Manager			
Packages Tools			
SDK Path: C:\Users\john\AppData\Local\Android\android-sdk			
Packages			
👾 Name	API	Rev.	Status
Android 8.0.0 (API 26)			
🗌 🚔 SDK Platform	26	2	👼 Installed
🔲 🌆 Android TV Intel x86 Atom System Image	26	7	😿 Installed
🔲 🌆 Android Wear Intel x86 Atom System Image	26	4	👼 Installed
🔲 💵 Google Play Intel x86 Atom System Image	26	7	👼 Installed
🗌 🔢 Google APIs Intel x86 Atom_64 System Image	26	8	👼 Installed
🔲 🌆 Google APIs Intel x86 Atom System Image	26	8	😿 Installed
Sources for Android SDK	26	1	😿 Installed
> 🗌 🔂 Android 7.1.1 (API 25)			
> 🗌 🔂 Android 7.0 (API 24)			

Figura 10: Android SDK Manager. Fonte: Autoria Própria.

¹⁸ Disponível em: < <u>https://www.androidpro.com.br/blog/android-studio/android-sdk/</u>>Acesso em: 11/09/2018.

¹⁹ Disponível em: < <u>https://developer.android.com/studio/command-line/sdkmanager.html</u>>Acesso em: 13/03/2018.

3.4. IONIC FRAMEWORK

lonic²⁰ é um framework de desenvolvimento de aplicação hibrida que utiliza comando de linha ionic para criar projetos, seu processamento de back end tem base no Cordova que é responsável por criar e fazer gerencia de aplicações nativas, para usar o framework precisamos instalar outras ferramentas necessárias que se integram a ele para poder ser executado.

A instalação do Node js que serve como um Servidor web tendo objetivo de ajudar a desenvolver sistemas escaláveis e realizar varias conexões simultâneas, o Node js usa JavaScript e V8 JavaScript Engine. Junto com a instalação do Node vem integrada a ferramenta de gerenciamento de pacotes chamada NPM Node Package Manager (Gerenciador de Pacotes Node). O NPM é um repositório online com ele usamos comando de linha para instalarmos pacotes, gerenciamos versões e dependências no projeto lonic.

Com terminal Prompt aberto inserimos o comando "npm install –g ionic cordova" assim instalando o lonic e o Cordova na maquina, na instalação é necessária instalar a linguagem Typescript com o comando "npm install –g typescript", o "-g" inserido dentro do comando de linha declara para instalação que essas tecnologias serão utilizadas de forma global para qualquer ação dentro do sistema operacional. TypeScript será a linguagem usava no desenvolvimento do projeto lonic, a linguagem é um "filho do JavaScript" criada para facilitar o desenvolvimento de aplicações, programa-se em TypeScript e compila para JavaScript, funcionando em qualquer navegador de internet. A linguagem é orientada a objetos e suporta plug-ins criados em JavaScript.

Depois que todas as instalar todas as tecnologias necessárias, dentro do diretório ou pasta a qual vai ser criado o projeto executamos o comando de linha lonic "ionic start NomeProjeto blank", "blank" significa que o projeto vai ser criado em branco somente com a classe padrão "home", assim que executar o comando será instalado todos os componentes plug-ins e diretivas bibliotecas do lonic automaticamente, algumas solicitações de particularidades de configuração do projeto a ser criado será exibida caso queria instalar algo adicional, essa situação vai da versão atual que o framework estiver.

³¹

²⁰ Disponível em: < <u>https://ionicframework.com/docs/</u>>Acesso em: 11/09/2018.



Figura 11: Criando Projeto Ionic. Fonte: Autoria Própria.

Para acessar a aplicação e certificar que esta executando entramos no projeto com comando de linha "cd NomeProjeto", após entrar no projeto o comando a ser acionado é "ionic serve", nesse momento os códigos estarão sendo compilado gerando uma aplicação web, sendo acessada no endereço local no navegador "localhost:8100.





Figura 13: Projeto Ionic em execução. Fonte: Autoria Própria.

3.5. COMPONENTES E ESTRUTURAS DE PROJETO IONIC

As aplicações desenvolvidas em lonic são criadas por blocos de construção de alto nivel chamadas Componentes²¹.

Os componentes permitem que uma interface sejá desenvolvida rapidamente. Existem varios tipos de componentes como botôes, alertas, cartões, checkbox, inputs, navegação, lista entre outros.

Na Estrutura o nivel principal para desenvolvimento se localiza dentro da pasta src em pages que contem as classes, dentro contem os arquivos HTML, SCSS e TYPESCRIPT.

²¹ Disponível em: < <u>https://ionicframework.com/docs/components/</u> >Acesso em: 28/07/2018.



Figura 14: Estrutura de projeto Ionic. Fonte: Autoria Própria.

3.6. O QUE É LINGUAGEM SCSS

O SCSS²² é uma linguagem de extensão do CSS para se programar de forma mais estruturada dando dinâmica para a programação dos estilos.

As classes são criadas no arquivo SCSS e aplicadas no arquivo HTML.

Existem varios atributos de CSS para criar a aparencia de uma aplicação, dentre eles alguns como cores de fundo, tamanho das fontes de texto, alinhamento de texto, alinhamento de icones, cores de icones, cores de texto, bordas entre outros.

²² Disponível em: < <u>http://sass-lang.com/documentation/file.SASS_REFERENCE.html</u> >Acesso em: 28/07/2018.



Figura 15: Arquivo SCSS. Fonte: Autoria Própria.

Background-color: Esse atributo é responsavel por aplicar cor de fundo na tela.

Border-top: Esse atributo cria uma borda acima do componente ao qual foi aplicado.

Color: O atributo insere cor no texto ou icone ao qual foi destinado.

Font-size: O atributo é responsavel por aplicar o tamanho da fonte do componente.

Text-align: Esse atributo ajusta o alinhamento do texto para o posicionamento desejado.

3.7. O QUE É LINGUAGEM HTML

O HTML do ingles que significa Hypertext Markup Language para portugues Linguagem de Marcação de Hipertexto é uma linguagem executada nas paginas interpretada pela maioria dos navegadores de internet, sua versão atual é HTML5²³ que recebeu melhorias com novos elementos, mudanças de sintaxe e integrações.

O HTML contem tags que são marcações de pagina responsaveis pela identação das informações, marcações como titulos, rodapes, links de endereços de pagina, botões, paragrafos, fotos, icones entre outros.

Dentro do HTML do projeto no lonic as tags que são tratadas como componentes possuem seu proprio "nome" de tags lonic.

²³ Disponível em: < <u>https://www.w3schools.com/html/html5_intro.asp</u> >Acesso em: 28/07/2018.

• dis	spositivo.html 🔹
1	<ion-header></ion-header>
2	<ion-navbar color="primary"></ion-navbar>
3	<pre><ion-title></ion-title></pre>
4	<pre><ion-icon class="custom-iconcloud" name="cloud"></ion-icon></pre>
5	CLOUD FIELD
6	
7	
8	
9	
10	<ion+content class="background_cor" padding=""></ion+content>
11	<pre><div "ngif="cadastroacionado"></div></pre>
12	<ion-list></ion-list>
13	<ion-item></ion-item>
14	<pre><ion-label class="alinhado_centro"><ion-icon class="custon-iconcog" name="cog"></ion-icon>CADASTRO DE DISPOSITIVO</ion-label></pre>
15	
16	
17	
18	<ion-list></ion-list>
19	<ion-item></ion-item>
20	<ion-label>DATA</ion-label>
21	
22	<ion-item></ion-item>
23	<pre><ion-datetime [(ngnodel)]="dispositivoObjeto.data_criacao" displayformat="DD/MM/YYYY"></ion-datetime></pre>
24	
25	<ion-item></ion-item>
26	<pre>(ion-label>Nome Dispositivo:</pre>
27	
28	<ion-item></ion-item>
29	<pre><loninput <="" [(ng%odel)]="dispositivoObjeto.nome_dispositivo" name="dispositivoObjeto.nome_dispositivo" placenoider="insina o Nome" pre="" type="text"></loninput></pre>
30	
31	101-1157
32	(100-1157)
33	<pre><ion-label>con-label>con-label>conton lock (click)="buttonprogramacao()">Escolha a Programacao(/button></ion-label></pre>
2.4	(/100-1708)

Figura 16: Arquivo HTML parte 1. Fonte: Autoria Própria.

<ion-reader>: O cabeçalho é um componente pai que contém os componentes Navbar e Title. Fica fixo na parte superior da tela de interface que contem a legenda do titulo, podendo possuir botões de navegação ou tambem execultar outras ações.

<ion-navbar>: O Navbar é como uma barra de ferramentas de navegação que vem com um botão de retorno para outra pagina anterior. O Navbar deve sempre estar dentro de uma tag header para que fique acima dos conteudos da pagina.

Color: é um atributo usado para preencher com cores o componente. As cores de fabrica no framework são primary, secondary, danger, light e dark.

<ion-tittle>: Tittle é um componente que informa o titulo da barra de navegação ou de ferramentas.

<ion-icon>: O componente icone pode ser usado sozinho ou dentro de outros componentes, existe uma lista de icones, cada icone tem uma aparencia para cada plataforma que está sendo executada. A diretiva de class é referente à classe de SCSS que está agindo na pagina lonic HTML dando aparencia aos componentes. O elemento name aponta qual icone será exibido.
<ion-content>: Consiste na área de conteudo da aplicação aonde serão organizadas todas as informações. O elemento padding serve para criar a area de preenchimento, o espaço entre o conteudo do elemento e sua borda.

<ion-list>: O componente List é um elemento de interface lista ao qual pode ser incluido dentro textos, botões e icones e outros componentes como Item.

<ion-item>: Um componente item pode conter imagens, texto entre outras coisas, o item é inserido dentro de um componente List.

<ion-label>: É um componente igual um rótulo dentro deles podem ser inseridos outros componentes como ion-input, ion-checkbox ou texto entre outros.

: È um elemento HTML que faz com que a cor do texto fique destacada, resaltando em negrito.

<ion-datatime>: O componente Datetime exibe uma interface de datas e horas para o usuario. O Selecionador apresenta colunas roláveis para escolher anos, meses, dias, horas e minutos individualmente. Nas diretivas de displayFormat se faz o ajuste do formato que a Data será exibida. Uma diretiva ng-model associa o valor de entrada do elemento HTML como exemplo um input a variavel do escopo da aplicação.

<ion-input>: O Componente input serve para fazer entrada de dados como texto, senha, números, email, telefone e url. Dentro dele usa diretiva HTML Type declarando que o tipo dele é texto. O elemento placeholder serve para deixar automatico um valor a ser exibido no componente na interface da aplicação.



Figura 17: Arquivo HTML parte 2. Fonte: Autoria Própria.

<button>: Os botões podem ser criados por texto ou icones, e receberem atributos de css.
O componente ion-button informa que o botão é um componente lonic, e o metodo de
chamada (click)="NomeDaFuncao()", realiza a chamada da função que foi declarada no
arquivo Typescript.

<div>: é uma tag HTML que serve para dividir o codigo em partes e alterar o posicionamento dos componentes dentro da interface, como tambem aceita classe de scss parar fazer alteração da aparencia.

nglf: é uma diretiva angular utilizada em elementos HMTL junto com uma condição definida para que seja ou não exibido seu conteudo.

ngFor: é uma Diretiva angular utilizada em elementos HTML para listar objetos ou variaveis contidas em um array, ou tambem em uma lista.

async : Significa que a chamada dos objetos a serem exibidos vai ser assincrona, se tiver qualquer mudança no banco de dados, de uma alteração, novo objeto, delete ele retornará a nova chamada com os valores atualizados. {{programacao.nome_programacao}} : Esse codigo está exibindo um o atributo nome_programação do objeto programação, a notação de chaves duplas '{{}}' serve para exibir os dados de objetos ou variaveis vindos do escopo de programação do arquivo Typescript.

 : é um elemento HTML que realiza quebra de linha.



Figura 18: Arquivo HTML parte 3. Fonte: Autoria Própria.

<ion-select>: é um componente de seleção de valores para que o usuario possa escolher. O usuario ao tocar no componente de seleção abre uma caixa de dialogo contendo as opções para selecionar. As opções de seleção são atribuidas nos <ion-option>, dentro do "value" é atribuido o valor que a opção fornece.

O dispositivo.html X	
117	<pre>cion-item></pre>
118	Estado Operacional:
119	
120	<ion-item></ion-item>
121	<pre><ion-input [(ngnodel)]="alterar_estado_operacional" class="colorformblack" placeholder="{{alterar_estado_operacional}}</pre" type="text"></ion-input></pre>
122	name="alterar_estado_operacional">
123	
124	
125	
126	<ion-list></ion-list>
127	<pre><ion-item>(ion-label>(button ion-button block (click)="buttonprogramacao()">Escolha a Programacao(/button>(/ion-label>)</ion-item></pre>
128	
129	<pre><dv *nglf="buttonprogramacaoactonado"></dv></pre>
130	<pre><lon-item 'ngtor="let programacao or programacoes async"></lon-item></pre>
131	<pre><div class="divsepara"></div></pre>
132	<010>
133	<pre>cbutton lcn-button color="dark" (button lcn-button color="dark")</pre>
134	(click)= escullioAlterar(programaca0.nome_programaca0.rey,programaca0.tempo_irrigaca0.programaca0.tempo_veri=icanumidabe) ">Abicionar 0 https://www.abicionar
135	S/ DUCCON
130	
130	<pre>chest ong/thmets/sciong/throgramacau.none_programacau// chest</pre>
130	(otronal Tames de jeriarelar//stanas//seasabers temes (sejarral))
140	cpear of Barrento op Thi Titleteory and of the delenged the field of t
141	<pre>cstrong/Tempo de verificação de humidade*(/strong)/(orogramação tempo verificahumidade))</pre>
142	studio.
143	
144	<pre><div class="divsepara"></div></pre>
145	
146	
147	
148	<pre><ion-list></ion-list></pre>
149	<pre>(ion-item)</pre>
150	ID da Programação:
151	
152	<ion-item></ion-item>
153	<ion-label class="colorformblack">{{alterar_id_programacao}}</ion-label>
154	
155	<pre><ion-item></ion-item></pre>

Figura 19: Arquivo HTML parte 4. Fonte: Autoria Própria.

 disposit 	tvohtni •
159	<pre>Nome da Programação:</pre>
160	
161	<pre><ion-item>cion-label class="colorformblack">{{alterar_nome_programacao}}</ion-item></pre>
162	<pre><ion-item>Tempo de Irrigação:</ion-item></pre>
163	<pre><ion-item>(ion-label class="colorformblack">{{alterar_tempoirrigacao}}</ion-item></pre>
164	<pre><ion-item>Tempo de intervalo de verificação de humidade:</ion-item></pre>
165	<pre><ion-item><ion-label class="colorFormblack">{{alterar_tempohumidade}}</ion-label></ion-item></pre>
166	<pre><div class="divsepara"></div></pre>
167	
168	<pre><div padding=""></div></pre>
169	<pre><buildeduction (click)="retornar_cadastro()" class="colorFormblack" color="secondary">Retornar</buildeduction></pre>
170	 button ion-button color="danger" class="colorFormblack"
171	(click)="alterar_salvardispositivo(alterar_id,alterar_nome_dispositivo,alterar_id_programacao,alterar_nome_programacao,
172	alterar_estado_operacional,alterar_data_criacao,alterar_tempoirrigacao,alterar_tempohumidade)">Alterar
173	
174	
175	<pre><ion-list><ion-list><ion-label class*"alinhado_centro"=""><ion-licon class*"custon-iconcog"="" name*"cog"=""></ion-licon>REGISTRO DE DISPOSITIVO</ion-label></ion-list></ion-list></pre>
176	<ion-list></ion-list>
177	<pre><ion-item '="" *ngfor*'let="" async="" dispositivo="" dispositivos="" of="" =""></ion-item></pre>
178	<div class="divsepara"></div>
179	<div></div>
180	<pre><ion-icon class="custom-id" name="key"></ion-icon>: ((dispositivo.key)) -</pre>
181	<pre>Dispositive: ({dispositive.nome_dispositive})</pre>
182	<pre></pre>
183	<pre>(br>ID Programação: {(dispositivo.id_programação))</pre>
184	<pre>(br>Nome Programacao: {{dispositive.nome_programacao}}</pre>
185	<pre>(dr>ID Cultivo ((dispositive.id_cultive))</pre>
186	<pre>(dr>Nome Variedade://strong> ((dispositivo.nome_variedade))</pre>
187	(or)(strong))empo de Irrigéceo:(/strong) {(dispositive.temp_irrigéceo)}
100	<pre>(dr)(dtrong)tempo de intervalo de verificação de numicade/strong> {(dispositivo.tempo_numicade)}</pre>
109	(or) (strong) data crisecovaterates () strong > ((uspositive costed crisecov)
101	(involution on bottom color content (click) bittom disperiing (inputition bay disperiing content of participation of the second se
191	<pre>dution ich-putch color= stoneary (click)= arterar_oispositivo(dispositivo(aspositivo, mome_dispositivo, dispositivo, astano constituanti di anonzamante dispositivo ante processitivo di cultivo dispositivo attano constituanti di anonzamante dispositivo ante processitivo di cultivo di cultivo di cultivo di cultivo di anonzamante dispositivo di cultivo di cultivo di cultivo di di cultivo di cultivo di di cultivo di cultivo</pre>
103	discontinuo ana unidada discontinu discontinu tano induces discontinu tano unidade vibetana
194	Cadays
195	(div_lass="divseara")/(div)
196	(/ion-item)
197	(/ion-list)chr)
198	(/ion-content)

Figura 20: Arquivo HTML parte 5. Fonte: Autoria Própria.

CLOUD FIELD							
CADASTRO DE DISPOSITIVO							
DATA							
15/09/2018							
Nome Dispositivo:							
Insira o Nome							
ESCOLHA A PROGRAMAÇÃO							
ID							
Nome							
Nome							
Nome Tempo Irrigação:							

Figura 21: Tela Dispositivo parte 1. Fonte: Autoria Própria.

← CLOUD FIELD	
	REGISTRO DE DISPOSITIVO
Contention of the second state of the second s	
-:: -LLCpuzbdE72rNztXJTc - Dispositivo: Red ribbon Estado Operacional: Ativado ID Programação: -LC-SHjV8vXBSql4-P02 Nome Programação: teste3 ID Cultivo: -LLDAxejn2Jxw11924h0 Nome Variedade: rucula Tempo de Irrigação: 3 Tempo de intervalo de verificação de humidade: 3 Data criação/Alteração: 31/8/2018 DELETAR ALTERAR	

Figura 22: Tela Dispositivo parte 2. Fonte: Autoria Própria. 41

← CLOUD FIELD						
CALTERAR CADASTRO DE DISPOSITIVO						
Data:						
30/8/2018						
ID da Dispositivo:	Alterar Estado					
-LLCndJ5s2bdlzYjClKM	Operacional:					
Nome da Dispositivo:	O Desativado					
horizon_00	CANCEL OK					
Nome Variedade:						
limao rosa						
ID Cultivo:						
-LLCr0MSIDhzRONtiRKV						
Alterar Estado Operacional:						

Figura 23: Tela Dispositivo parte 3. Fonte: Autoria Própria.

← CLOUD FIELD
ESCOLHA A PROGRAMAÇÃO
ADICIONAR Nome:Leste3 Tempo de irrigação:3 Tempo de verificação de humidade:3
ADICIONAR Nome:teste2 Tempo de irrigação:8 Tempo de verificação de humidade:10
ADICIONAR Nome:teste1 Tempo de irrigação:10 Tempo de verificação de humidade:10
ID
-LC-5HjV8vXBSql4-P02

Figura 24: Tela Dispositivo parte 4. Fonte: Autoria Própria.

3.8. O QUE É LINGUAGEM TYPESCRIPT

A linguagem Typescript é uma programação orientada a objetos que fornece classes, interfaces e é fortemente tipada. Foi desenvolvida pela empresa Microsolft²⁴, Typescript possibilita criar aplicações de forma pratica e quando executada transforma seu codigo para linguagem javascript podendo ser executada na maioria dos navegadores de internet.

O projeto é divido entre Camada Model ou modelo, camada de visão ou view e camada de service ou serviços.

A Orientação a objeto é um conceito de compreende realizar a representação de um objeto do mundo real para dentro de um sistema. O objeto pertence a um tipo de classe, ele precisa de uma identificação um numero ou valor para ser único dentro de um sistema, como a identificação de uma pessoa o RG que significa Registro geral que contem uma identificação unica.

Esse Objeto precisão de seus atributos que são caracteristicas como por a cor dos olhos de uma pessoa ou o nome dessa pessoa. Esse objeto tambem possui metodos que atraves dele realiza ações diversas dentro da aplicação ao qual foi destinada. Exemplos de um metodo para um objeto da classe pessoa como metodo falar, andar, olhar, pegar ou dormir.

™ dispo	sitivo.model.ts ×							
1	<pre>1 export interface DispositivoModel{</pre>							
2								
3	key?: string;							
4	nome_dispositivo:string;							
5	nome_programacao:string;							
6	<pre>id_programacao:string;</pre>							
7	<pre>tempo_irrigacao:string;</pre>							
8	<pre>tempo_humidade:string;</pre>							
9	estado_operacional:string;							
10	id_cultivo:string;							
11	nome_variedade:string;							
12	<pre>data_criacao:string;</pre>							
13								
14	}							

Figura 25: Arquivo Dispositivo model Ts. Fonte: Autoria Própria.

²⁴ Disponível em: < <u>https://www.microsoft.com/pt-br</u> >Acesso em: 28/07/2018.

A Camada model ou modelo é responsavél por declarar a classe do objeto contendo os seus atributos.

key: a identificação do objeto dispositivo.

nome_dispositivo: nome do dispositivo.

nome_programação: nome da programação.

id_programação: identificação do objeto programação

estado_operacional: situação atual do dispositivo.

data_criacao: data de criação.

import { Component } from '@angular/core'; import { NavController } from 'ionic-angular'; import { AlertController } from 'ionic-angular'; import { AngularFireDatabase,AngularFireList } from 'angularfire2/database'; import { Observable } from 'rxjs/Observable'; import 'nxjs/add/operator/map'; import { DispositivoModel } from '../../model/dispositivomodel/dispositivo.model'; import { DispositivoService } from '../../providers/dispositivo-service/dispositivo-service'; import { ProgramacaoService } from '../../providers/programacao-service/programacao-service'; 10 11 12 @Component({
 selector: 'page-dispositivo', 13 templateUrl: 'dispositivo.html' 14 3) 15 export class DispositivoPage { 16 17 programacoes : Observable(any[]); 18 programacoesRef: AngularFireList<any>; 19 dispositivos : Observable<any[]>; 28 dispositivoRef: AngularFireList(any); 21 22 dispositivoObjeto : DispositivoModel = { 23 24 nome dispositivo: " nome_programacao: '', id_programacao: '',
tempo_irrigacao: '', 25 26 27 •• tempo_humidade: 28 estado_operacional: 'Aguardando', 29 id_cultivo: " 30 nome_variedade: ", data_criacao: 31 32); 33 alterar_id : any = ""; alterar_nome_dispositivo : any = ""; alterar_estado_operacional : any = ""; alterar_id_programacao : any = ""; alterar_nome_programacao : any = ""; alterar_data_criacao : any = ""; 34 35 36 escolhida_programacao:any =""; escolhida_idprogramacao:any = ""; escolhida_tempoirrigacao:any =""; escolhida_tempohumidade:any = ""; 37 38 alterar_id_cultivo:any = ""; alterar_tempohumidade:any =""; alterar_tempoirrigacao:any = ""; alterar_tempohumidade:any = ""; 39 48

Figura 26: Arquivo Dispositivo Pagina Ts parte 1. Fonte: Autoria Própria.

O Arquivo Dispositivo Ts faz parte da Camada de Visão que compõe o conjunto de arquivos HTML, SCSS e TYPESCRIPT.

São importadas as bibliotecas de Component para os componentes que integrão o IONIC, o NavController é responsavel pela navegação das paginas e das variaveis pelas paginas, AlertController é responsavel por criar os componentes alert.

As bibliotecas AngularFireDatabase e AngularFireList são responsavel pela conexão ao banco de dados Firebase²⁵ e tratamento dos dados recebidos. As bibliotecas Observable²⁶ é necessaria para criar os objetos do tipo Observable que receberam os dados tratados vindos do Banco de dados.

A importação das classes Dispositivo Model para criar novos objetos do tipo Dispositivo, a importação da classe Dispositivo Service para que possam ser realizado a transação dos dados entre a aplicação e o banco. A importação da Programação Service é necessaria para que possa ter acesso à lista de objeto do tipo programação.

@component: Declara pelo "selector" qual SCSS está sendo chamada e pelo "templateUrl" qual Pagina HTML está sendo chamada.

programacoes: Observable<any[]> : Criar um objeto do tipo Observable para que possa receber as informações tratadas e exibir as informações na pagina HTML.

programacoesRef: AngularFireList<any> : Cria um objeto do tipo AngularFireList como uma lista que recebe as informações vindas do banco, trata as informações realizando mapamento de identificações, depois é encaminhada para um objeto do tipo Observable.

dispositivoObjeto : DispositivoModel : Instancia um objeto da classe dispositivo.

any : Declara uma variavel de um tipo qualquer para que possa ser utizada no escopo da programação. A variavel do tipo any pode receber qualquer tipo de dado string, int, boleano entre outros.

²⁵ Disponível em: < <u>https://firebase.google.com/?hl=pt-br</u> >Acesso em: 28/07/2018.

²⁶ Disponível em: < https://angular.io/guide/observables >Acesso em: 28/07/2018

TS dispo	sitivo.ts ×
42	
43	dia:any;
44	mes:any;
45	ano:any;
46	data:any;
47	buttonprogramacaoacionado: boolean = false;
48	cadastroacionado : boolean = true;
49	alteraracionado : boolean = false;
50	
51	
52	
53	constructor(public navCtrl: NavController , public alertCtrl : AlertController, public firebase: AngularFireDatabase,
54	public dispositivoservice: DispositivoService, public programacaoservice: ProgramacaoService){
55	
56	<pre>this.dispositiveObjeto.data_criacao = new Date().toISOString();</pre>
57	
58	<pre>this.programacoesRef = this.programacaoservice.listarProgramacaoService();</pre>
59	<pre>this.programacoes = this.programacoesRef.snapshotChanges().map(changes => {</pre>
60	<pre>return changes.map(c => ({ key: c.payload.key,c.payload.val() }));</pre>
61));
62	
63	<pre>this.dispositivoRef = this.dispositivoservice.listarDispositivoService();</pre>
64	<pre>this.dispositivos = this.dispositivoRef.snapshotChanges().map(changes => {</pre>
65	<pre>return changes.map(c => ({ key: c.payload.key,c.payload.val() }));</pre>
66	3);
67	
68	3
69	

Figura 27: Arquivo Dispositivo Pagina Ts parte 2. Fonte: Autoria Própria.

boolean: Cria uma variavel do tipo Booleana que pode receber o valor de falso ou verdadeiro para ser executada dentro do escopo da programação.

O Metodo Construtor da classe é responsavel por criar os componentes necessarios ou objetos que a classe precisa, como no codigo das classes importadas ou bibliotecas realizando uma injeção de objetos que serão utilizadas dentro da aplicação.

Date: criar um objeto do tipo Date²⁷ que informa dados de ano, mes, dia, e horas. A função toISOString²⁸ retorna o valor como cadeia de caracteres da data no formato ISO do ingles

Organization for Standardization.

Usando o objeto da classe Programação Service chamando o metodo listarProgramacaoService busca a lista completa de coleções de objeto situado no banco de dados, passando o resultado para o objeto programacoesRef.

O objeto programacoesRef passa por um mapeamento das identificação e dos atributos atraves da função snapshotChanges²⁹ e a map³⁰, passando o resultado para o Objeto dispositivos.

- ²⁸ Disponível em: < <u>https://msdn.microsoft.com/pt-br/library/ff925953(v=vs.94).aspx</u> >Acesso em: 28/07/2018.
- ²⁹ Disponível em: < <u>https://firebase.google.com/docs/reference/js/firebase.database.DataSnapshot</u> >Acesso em: 28/07/2018.
- ³⁰ Disponível em: < <u>https://github.com/angular/angularfire2/blob/master/docs/rtdb/lists.md</u> >Acesso em: 28/07/2018.

²⁷ Disponível em: < <u>https://msdn.microsoft.com/pt-br/library/cd9w2te4(v=vs.94).aspx</u> >Acesso em: 28/07/2018.





Ao invocar o metodo criar_dispostivo executa o metodo create do objeto alertCtrl que foi injetado no construtor, ele cria um componente Alert, o componente recebe o titulo pela função setTitle, recebe botões pela função addButton, executando sua ação dentro do handler.A função present execução o componente alert criado.

getDate(): Obtem o dia atual, faz parte da classe Date.

getUTCMonth(): Obtem o mês atual, faz parte da classe Date.

getFullYear(): Obtem o ano atual, faz parte da classe Date.

A chamada de função salvarDispositivoService do Objeto dispositivoservice passa o objeto dispositivoObjeto para a camada de serviço, que é responsavel por salvar o novo objeto no banco.

O metodo limpar atualiza os dados do objeto para que o proximo objeto que for criado no formulario estejá pronto para receber as novas informações.



Figura 29: Arquivo Dispositivo Pagina Ts parte 4. Fonte: Autoria Própria.

O metodo deletar_dispositivo recebe a mesma estrutura de codigo que o metodo criar_dispositivo, as novas mudanças no codigo são o atributo id que vem da pagina HTML contendo a identificação do objeto selecionado, passando a identificação para a camada de serviço usando o metodo excluirDispositivoService do objeto dispositivoservice para excluir o objeto no banco de dados.

O metodo alterar_salvardispositivo recebe todas as informações que foram alteradas no formulario da pagina HTML, passando para o metodo editarDispositivoService a identificação e o objeto dispositivoObjeto para a camada de serviço para efetivar a alteração do objeto.



Figura 30: Arquivo Dispositivo Pagina Ts parte 5. Fonte: Autoria Própria.

O metodo alterar_dispositivo é responsavel por receber os dados do objeto selecionado na lista de registros no HTML, o valor booleano da variavel cadastroacionado passa para falso e o valor da variavel alteraracionado passa para verdadeiro. Essa mudança faz com que o formulario de cadastro fique invisivel para o usuario e deixe visivel para o usuario o formulario de alteração do cadastro. O valor do objeto selecionado na lista de registros é passado para as variaveis temporarias no formulario de alteração de cadastro no HTML.

A função retornar_cadastro limpa todos os campos do formulario de alteração de registro, e altera o valor das variaveis booleanas deixando o formulario de cadastro disponivel novamente e o formulario de alteração de cadastro invisivel.

A função escolhido que age no formulario cadastro e a função escolhidoAlterar que executa no formulario alterar cadastro são metodos que obtem os valores dos atributos dos objetos programação que pertencem a classe programação.

Os atributos de identificação e nome do objeto programação são integrados aos atributos do objeto dispositivo.

O metodo buttonprogramacao serve para que alterne o valor booleano da variavel buttonprogramacaoacionado sejá trocado, para que seja visivel a lista das programações no formulario de cadastro somente quando o usuario desejar escolher ou alterar.

```
s dispositivo-service.ts X
     import { Injectable } from '@angular/core';
 1
 2
     import { AngularFireDatabase, AngularFireList } from 'angularfire2/database';
 3
     import { DispositivoModel } from '../../model/dispositivomodel/dispositivo.model';
 4
 5
     @Injectable()
 6
      export class DispositivoService{
 7
       dispositivoRef: AngularFireList<any>;
 8
       constructor(public firebase: AngularFireDatabase) {
 9
10
11
12
       salvarDispositivoService(dispositivoObjeto : DispositivoModel){
13
         this.firebase.list('dispositivo/cadastro/dispositivo_unidade/').push(dispositivoObjeto);
14
         return;
15
        }
16
        editarDispositivoService(id: any, dispositivoObjeto : DispositivoModel ){
17
18
          this.dispositivoRef = this.firebase.list('dispositivo/cadastro/dispositivo_unidade/');
 19
          this.dispositivoRef.update( id , dispositivoObjeto );
20
         return:
21
        }
22
23
       excluirDispositivoService(id: any){
         this.dispositivoRef = this.firebase.list('dispositivo/cadastro/dispositivo_unidade/');
24
25
         this.dispositivoRef.remove(id);
26
         return;
27
        3
28
 29
        listarDispositivoService(){
30
          this.dispositivoRef = this.firebase.list('dispositivo/cadastro/dispositivo_unidade/');
31
         return this.dispositivoRef;
32
        }
33
 34
```



A camada de serviço é responsavel por fazer a integração do fluxo de dados entre a aplicação e o banco de dados.

Dentro dela está os metodos criar, listar, alterar, e excluir que recebem os objetos da camada View ou visão que é pagina invocando os serviços. Essa estrutura fornece segurança e integridade da aplicação, além de facilitar a troca dos caminhos de transferencia de informações ou conexão com o banco de dados escolhido, sem precisar alterar a camada View da aplicação.

O caminho informado no objeto firebase declara a localização da coleção desejada dentro do banco de dados.

O metodo push do objeto firebase salva o objeto dispositivoObjeto no banco de dados.

O metodo update passa a identificação do objeto e o objeto para alterar os atributos no banco.

A função remove passa a identificação do objeto para que sejá excluido da coleção o objeto. Se for executado a função sem o atributo de identificação toda a coleção localizada no banco de dados será excluida.

A função list é responsavel por fazer uma lista com todas as informações ou objetos encontrados no caminho que foi informado.

3.9. CONEXÃO IONIC COM FIREBASE

Firebase³¹ é um banco de dados e uma plataforma de ferramentas que provem serviços para o desenvolvimento de aplicações Mobile, WEB e IOT. O Firebase fornece sua utilização como BaaS³² a sigla significa backend as a service um banco de dados na Nuvem³³ que provem serviços desde autenticação de identidade de usuario, armazenamento e manipulação de informações, notificações, grafico de uso de conexão dos serviços entre outros. Para que a Arquitetura Arquitetura SOFEA funcione é necessario que uma aplicação API de serviço REST faça a ponte de ligação entre a aplicação do usuario e o banco de dados, no caso do banco de dados Firebase que provem serviços não é necessario uma API porque o proprio disponibiliza um serviço REST para a aplicação do usuario.

São fornecidos varios serviços como:

-Realtime Database: Banco de dados de possibiliza sincronização dos dados em tempo real para a aplicação. As regras de segurança podem ser configuradas sobre o acesso dos dados.

-Authentication: fornece serviço de autenticação por contas como Facebook³⁴, Google³⁵, Twitter³⁶ entre outros.

-Storage: Possibilita salvar dados de imagem dos usuarios.

-Notifications: Envia notificações para os usuarios.

³¹ Disponível em: < <u>https://firebase.google.com/docs/database/?hl=pt-br</u>>Acesso em: 28/07/2018.

³² Disponível em: < <u>https://www.infoq.com/br/news/2016/07/backend-pros-e-contras</u> >Acesso em: 28/07/2018.

³³ Disponível em: < <u>https://www.rackspace.com/pt/library/what-are-cloud-databases</u> >Acesso em: 28/07/2018.

³⁴ Disponível em: < <u>https://www.facebook.com</u> >Acesso em: 28/07/2018.

³⁵ Disponível em: < <u>https://www.google.com.br/intl/pt-BR_br/about/</u>>Acesso em: 28/07/2018.

³⁶ Disponível em: < <u>https://about.twitter.com/pt.html</u>>Acesso em: 28/07/2018.



Figura 32: Interface Firebase parte 1. Fonte: Autoria Própria.

Após acessar o Firebase com a senha do perfil de usuario da conta Google, e criar um projeto acesse o icone de ferramentas no topo da interface e toque em configurações do projeto. Essa tela conterá informações privadas do projeto como chaves de API entre outras informações. Abaixo de todas as informações encontram-se as chaves de API para cada tipo de projeto entre eles para aplicações IOS, ANDROID e WEB. A chave para o projeto IONIC é a chave de aplicativo WEB.



Ao acessar o icone para WEB, abre um caixa de dialogo contendo o script de programação para conexão como as chaves de API e URL do banco.



Figura 34: Interface Firebase parte 3. Fonte: Autoria Própria.

Para ter acesso às informações contidas no banco pela aplicação do usuario sem a autenticação é necessario alterar as configurações de segurança informando a condição True para Read do ingles que significa leitura e Write que significa escrita. Para realizar a alteração basta acessar Database e depois em Regras.



53

O Armazenamento de dados dentro do Realtime Database organiza as informações em uma estrutura árvore³⁷ JSON, as bases de dados são nos niveis da árvore sendo tratadas como coleções. Dentro das coleções são salvos as informações em formato JSON, cada um possuindo sua identificação e seus atributos.



Figura 36: Arquivo App module Ts. Fonte: Autoria Própria.

No projeto lonic dentro da raiz do projeto existe o arquivo App module Ts esse arquivo é responsavel por declarar todas as classes das paginas, plugins de componentes adicionais, classes dos serviços e conectores de banco de dados. A biblioteca de conexão do Firebase com lonic se chama AngularFire2³⁸ disponivel no GITHUB³⁹, para realizar a instalação da biblioteca basta usar a seguinte comando de linha no terminal prompt "npm install firebase angularfire2 --save", esse comando deve ser executado atraves do terminal prompt na pasta do projeto ionic. A Chave de API deve ser inserida como na Figura para declarar as informações de conexão com o banco.

³⁷ Disponível em: < <u>http://www.ufjf.br/jairo_souza/files/2009/12/EDI-06.ED_.Arvores.pdf</u> >Acesso em: 28/07/2018.

³⁸ Disponível em: < <u>https://github.com/angular/angularfire2</u> > Acesso em: 28/07/2018.

³⁹ Disponível em: < <u>https://github.com/</u> >Acesso em: 28/07/2018.

3.10. CORDOVA FRAMEWORK

CORDOVA é um Framework open source para desenvolvimento de aplicações mobile multiplataforma, significa que um mesmo código que o programador escreva pode ser executado em outros sistemas operacionais chamado de desenvolvimento hibrido. O Cordova pega o código implementado em Linguagem HTML, JAVASCRIPT e CSS converte e empacota criando um executável que funciona em sistemas operacionais ANDROID, IOS e WINDOWS. Os comandos para gerar ações no projeto criado são executados em comandos de linha no terminal prompt exatamente no diretório do projeto, comandos como exemplo "cordova platform add android" que adiciona a plataforma android dentro do projeto junto com os recursos necessários para poder executavel para poder instalar o sistema no dispositivo.

Quando se executa o projeto o código começa a sofre um processo de compilação e tradução/transpilação, isso significa que o Cordova pega os Códigos escritos em Typescript convertendo para Javascript junto com HTML e CSS, os Browsers não executam linguagem Typescript somente Javascript. Assim como aplicações compiladas que possuem plug-ins com recursos nativos só funcionaram com em dispositivo Mobile.



Figura 37: Processo de Compilação utilizando Cordova. Fonte: Autoria Própria.

4. IOT

A "Internet of Things" ou IOT em língua portuguesa Internet das coisas é um conceito que integra diversas tecnologias e plataformas a internet a qual temos além das maquinas e notebooks conversando entre si temos também dispositivos como celulares, carros, vídeos games, geladeiras, robôs, casas, impressoras, maquinas agrícolas e muitas outras coisas físicas, que se conectam a rede e trocam dados.



Figura 38: Tecnologias que integram o IOT Fonte: https://www.embarcados.com.br/iot-contexto-geral-perspectiva-parte-2/

O conceito vem bem antes da internet com as empresas que utilizavam RFID⁴⁰ Radio Frequency Identification em suas lojas para controle de etiquetas de roupas e acessórios que possibilitava controle de suas mercadorias.

⁴⁰ Disponível em:< <u>https://www.tecmundo.com.br/tendencias/2601-como-funciona-a-rfid-.htm</u> > Acesso em: 19/03/2018.

O surgimento da tecnologia RFID vem da Segunda guerra quando exercito precisava ter controle dos armamentos e identificar qual nave no radar era amigo ou inimigo. Após a Guerra com os avanços tecnológicos foi liberada ao mercado possibilitando empresas diversas utilizarem o serviço Radio frequência em seus negócios.

Para comunicação na rede utilizamos protocolos IP significa Internet Protocol, o que possibilita o acesso a sites, vídeos, fotos e sistemas entre muitos outros benefícios disponibilizados na Web.

Atualmente usamos o Protocolo vigente Ipv4 na sua versão 4, mas devido a escassez de IPs disponíveis pelo grande crescimento de objetos que precisam estar conectados foi desenvolvido um novo protocolo surgiu o IpV6 possibilitando uma infinidade de IPS disponíveis. Junto com IOT vem outro novo conceito chamado "Big Data" ⁴¹ que é a geração de grandes volumes de dados, vários dispositivos conectados a internet gerando informações de todos os tipos como notificações de Facebook⁴², whatshapp⁴³, email, lojas virtuais, sites entre outros tipos de gerações de dados. Todo esse grande volume de dados precisara ser analisados para compreender seu padrão e obtendo como resposta as informações necessárias.

No IOT a coisa ou objeto possui um conjunto de funcionalidades e de características, sendo elas físicas e virtuais. Definindo o IOT como um sistema complexo que possui nove funcionalidades que se dividem em três conjuntos são elas características, relações e interface.

O Conjunto das características são os atributos do próprio objeto, sendo o processamento á capacidade computacional para gerir suas aplicações e requisições, o endereçamento que é a localização de poder ser encontrado na rede, a identificação a qual todo objeto tem sua própria identificação na rede, e sua localização sendo seu posicionamento no mapa geográfico.

⁴¹ Disponível em:< <u>https://www.sas.com/pt_br/insights/big-data/what-is-big-data.html</u> > Acesso em: 19/03/2018.

⁴² Disponível em < https://www.facebook.com/ > Acesso em: 19/03/2018.

⁴³ Disponível em:< <u>https://www.whatsapp.com/?l=pt_br</u> > Acesso em: 19/03/2018.

O Conjunto das relações referente à capacidade de se relacionar com outros objetos IOT, a comunicação entre os objetos de enviar e receber mensagens, a cooperação no processo das atividades, o sensoriamento permitindo que possa coletar dados do meio ambiente pelos sensores presentes no seu hardware, e a atuação que é a capacidade de operar no meio físico a qual foi destinada.

O Conjunto da interface se diz respeito à interação do objeto ou coisa com o usuário, podendo o usuário consumir os dados gerados pelo dispositivo, alterar ações e atividades que o mesmo executa no ambiente.

Um dos dispositivos para prototipação de automação, robótica e objetos IOT conhecidos no mercado é o microcontrolador ARDUINO²⁰. O Arduino Uno é uma placa de hardware de projeto livre para desenvolvimento eletronico, constituído por processador, memorias, entradas e saídas para conectar outros componentes externos, conseguindo interagir com o ambiente usando seu hardware e software.

Os componentes são diversos como resistores, leds, display de cristal liquido, sensores, módulos de internet entre muitos outros, cada componente tem sua programação e sua prototipação. Os desenvolvedores de aplicações em Arduino possuem uma grande comunidade que compartilham seus projetos com os códigos e diagramas de circuito podendo ser executados ou alterados por outros programadores ajudando no aprendizado e entendimento das tecnologias.

O Arduino Uno se baseia no Microcontrolador Atmel do modelo Atmega328, seus atributos de hardware possuem um processador Atmega328p-pu, 8 bits, velocidade de Clock de até 20 Mhz, sua alimentação de 1,8 a 5,5 Volts, memoria de Flash de 32 Kbytes, memoria Ram de 2 Kbytes, 23 pinos de entrada e saída, 6 canais de Conversor analógico digital e mais PWM de 6 canais.

²⁰ Disponível em < https://www.arduino.cc/ > Acesso em: 19/03/2018.



Figura 39: Microcontrolador Arduino. Fonte: https://www.embarcados.com.br/iot-contexto-geral-perspectiva-parte-2/



Figura 40: Componentes adicionais. Fonte: Autoria Própria. O ESP8266 12E NODEMCU é um modulo tendo integrado em sua placa WI-FI 2.4 GHz, Wireless 802.11 b/g/n, um processador Tensilica Lx106 de 32 bits rodando de 80 Mhz a 160Mhz, memoria Ram de 20 kb, memoria flash de 4 Mb disponivel para uso, protocolos de rede lpv4, Tcp e Http. As linguagems de programação disponiveis que podem ser usadas na placa são Lua, C++ e Javascript. Para ligar o modulo a uma Maquina utiliza um cabo micro-usb, podendo usar a Arduino IDE para implementar a programação.

A integração com WI-FI proporciona a possibilidade de desenvolvimento de muitos projetos como automação residencial, robótica, Internet das coisas e tambem integrar o modulo com outros microcontroladores como o Arduino.

A placa possui 30 pinos cada um para uma função espefica, podendo conectar componentes como sensores e outros modulos diversos.

A velocidade Serial em que os codigos são executados na placa é de 115200, com alimentação de corrente eletrica de 5 volts para funcionar, mas para uso interno todos os componentes nela inseridos precisam operar em 3.3 volts para que não danifique ou queime a mesma.



Figura 41: Placa Esp8266 12e Especificações. **Fonte**: http://www.settorezero.com/wordpress/contents/2016/11/NodeMCU_devkit_pinout.png.

Os pinos são:

GND: Para fazer a terra.

VCC: Alimentação de 3.3v.

GPIO: Pinos de entrada ou saída de propósito geral.

RST: Para resetar o funcionamento da placa.

RX: Receber dados.

TX: Transmitir dados.

CIK: Clock da interface SPI.

SD0: MISO da interface SPI.

SD1: MOSI da interface SPI.

CMD: Cs Seleção de chip da interface SPI.

A0: Entrada analógica de 10 bits ADC 0-1 Volts.

EN: Ativar módulo. Ativo alto.

RSV: reservado (não use).

VIN: Pode alimentar a placa com 5 volts exata.

Os Botões:

RST: Ao ser acionado ele reinicia a placa.

FLASH: Ao ser acionado ele realiza gravações de programa na placa, mas ao utilizar a Arduino IDE esse procedimento é realizado de forma automática não precisando ser realizado.

Esses Componentes serão utilizados no desenvolvimento do protótipo de dispositivo, o sensor de humidade realizará a verificação das condições de humidade ideal do solo, a válvula solenoide será responsável pela irrigação, o Módulo de Internet Wifi realizará a comunicação das atividades concluídas e status salvando as informações no banco de dados.

Para implementação do software que será executado na placa utiliza o Arduino IDE um editor de código com endentação automática, compilando e carregando o programa para placa através do USB. A programação é de fácil compreensão podendo ser implementada em C ou C++, tendo a função Setup () a qual inicia a programação e a configurações que serão usadas, e a função Loop () que executa repetindo os blocos de código ou até desligar a placa pela execução da programação.



Figura 42: Diagrama de conexão dos componentes eletronicos. Fonte: Autoria Própria.

4.1. PREPARANDO A ARDUINO IDE PARA USAR O ESP

Antes de criar o script de programação precisamos preparar a IDE Arduino.

Com a IDE aberta:

1° Entre em Arquivos, depois em Preferencias e adicione o link.

🔊 sketch_jul30a Arduino 1.8.5									
Arquiv	Arquivo Editar Sketch Ferramentas Ajuda								
T I	Novo	Ctrl+N							
1	Abrir	Ctrl+0							
1	Abrir Recente	•							
9	Sketchbook	•							
E	Exemplos	•	run once:						
F	Fechar	Ctrl+W							
9	Salvar	Ctrl+S							
9	Salvar como	Ctrl+Shift+S	in repeatedly:						
(Configuração da página	Ctrl+Shift+P							
I	mprimir	Ctrl+P							
F	Preferências	Ctrl+Vírgula							
5	Sair	Ctrl+Q							

O link "http://arduino.esp8266.com/stable/package_esp8266com_index. json" em Urls Adicionais.



Figura 44: Passo 1.1 instalar biblioteca ESP8266. Fonte: Autoria Própria.

2° Em Ferramentas, depois em Placa e Gerenciador de placas.

Procure o nome Esp8266 e instale.

🥸 sketch_jul30a Arduino	1.8.5		and can be a supervised the	State of Concession, Name of Street, or other
Arquivo Editar Sketch Fe	rramentas) Ajuda	-		
Sketch_jul30a void setup() { // put your setu }	Autoformatação Ctrl+ T Arquivar Sketch Corrigir codificação e recarregar Monitor serial Ctrl+ Shift+ Plotter serial Ctrl+ Shift+ WiFi101 Firmware Updater Viente Structure	M		
void loop() {	Placa: "NodeMCU 1.0 (ESP-12E Module)"		∆ Gereneiador de Placas	
<pre>// put your main }</pre>	Hash Size: "4M (IM SPIFFS)" Debug port: "Disabled" Debug Level: "Nenhum" IwIP Variant: "v2 Lower Memory" CPU Frequency: "80 MHz" Upload Speed: "115200" Erase Flash: "Only Sketch" Porta Obter informações da Placa		Placas Arduino AVR Arduino Yún Arduino/Genuino Uno Arduino Duemilanove or Diecimila Arduino Nano Arduino/Genuino Mega or Mega 2560 Arduino Mega ADK Arduino Leonardo Arduino Leonardo ETH	
	Programador: "USBasp" Gravar Bootloader	_	Arduino/Genuino Micro Arduino Esplora	

ierenciador de Plac	as a constant and a c	×
Todos	• [esp8266]	
sp8266 by ESP826 acas incluídas nes eneric ESP8266 Mi 0, ESPresso Lite 2 OD-WIFI-ESP8266 leMos D1 mini Pro ystems gen4 IoD i nline help ore info	In Community versão 2.4.1 INSTALLED se pacote: odule, Generic ESP8285 Module, ESPDuino (ESP-13 Module), Adafruit Feather HUZZAH ESP8266, ESPresso Lite .0, Phoenix 1.0, Phoenix 2.0, NodeMCU 0.9 (ESP-12 Module), NodeMCU 1.0 (ESP-12E Module), Olimex (-DEV), SparkFun ESP8266 Thing, SparkFun ESP8266 Thing Dev, SweetPea ESP-210, WeMos D1 R2 & mini, . WeMos D1 mini Lite, WeMos D1 R1, ESPino (ESP-12 Module), ThaiEasyElec's ESPino, WifInfo, Arduino, 4D Range, Digistump Oak. Fed	

Figura 46: Passo 2.1 instalar biblioteca ESP8266. Fonte: Autoria Própria.

3° Apos instalar as placas do Esp8266 entre em Ferramentas, depois em placas e selecione na lista o modelo Generic Esp8266 module.

😳 sketch_jul30a Arduino	1.8.5		and one first a countries	C. Bullet Bask
Arquivo Editar Sketch Fe	rramentas) Ajuda			
sketch_jul30a	Autoformatação Arquivar Sketch Corrigir codificação e recarregar	Ctrl+T		
<pre>void setup() { // put your setu</pre>	Monitor serial Plotter serial	Ctrl+Shift+M Ctrl+Shift+L		
1	WiFi101 Firmware Updater			
<pre>void loop() { // put your main }</pre>	Placa: "NodeMCU 1.0 (ESP-12E Mod Flash Size: "4M (1M SPIFFS)" Debug port: "Disabled" Debug Level: "Nenhum" IwIP Variant: "v2 Lower Memory" CPU Frequency: "80 MHz" Upload Speed: "115200" Erase Flash: "Only Sketch" Porta Obter informações da Placa	ule)"	Arduino Pro or Pro Mini Arduino NG or older Arduino Robot Control Arduino Robot Motor Arduino Gemma Adafruit Circuit Playground Arduino Yún Mini Arduino Industrial 101 Linino One Arduino Uno WiFi	
	Programador: "USBasp" Gravar Bootloader		ESP8266 Modules	
L			Generic ESP8285 Module Generic ESP8285 Module ESPDuino (ESP-13 Module) Adafruit Feather HUZZAH ESP8266	

4° Depois de realizar esses procedimentos, verifique a velocidade serial em Upload Speed deixando 115200, tambem verifique a porta USB de conexão utilizada na maquina.

	Autoformatação	Ctrl+T			
	Arquivar Sketch				
sketch_jul30a	Corrigir codificação e recarregar				
roid setup() { // put your setu	Monitor serial	Ctrl+Shift+M			
	Plotter serial	Ctrl+Shift+L			
	WiFi101 Firmware Updater				
<pre>void loop() { // put your main }</pre>	Placa: "NodeMCU 1.0 (ESP-12E Module)"				
	Flash Size: "4M (1M SPIFFS)"	,			
	Debug port: "Disabled"	,			
	Debug Level: "Nenhum"				
	IwIP Variant: "v2 Lower Memory"	,	8		
	CPU Frequency: "80 MHz"	,			
	Upload Speed: "115200"	,			
	Erase Flash: "Only Sketch"		-		
	Porta: "COM5"			Portas seriais	
	Obter informações da Placa		1	COM5	
	Programador: "USBasp" Gravar Bootloader				

Figura 48: Passo 4 instalar biblioteca ESP8266. Fonte: Autoria Própria.

4.2. CRIANDO O SCRIPT "HELLO WORLD"



A função Void setup executa a velocidade Serial, e outras funcionalizadades que o programador precisa para que a placa execute suas atividades.

A função loop é utilizada para executar funções de rotina se repetindo continuamente.

Serial.println("Hello World!!"): é utilizado para exibir as informação por linha no Monitor.

4.3. O QUE É A EEPROM

EEPROM⁴⁴ é uma memória Flash não volátil que o microcontrolador Esp8266 12E possui, que possui um armazenamento de 4 mb. Podendo fazer muitas gravações de informações que são necessarias quando a placa está inicializando o sistema na função Setup ou em suas ações da função loop.

Para salvar as informações e consumi-las é necessario incluir a biblioteca EEPROM nos includes do codigo.

4.4. SALVANDO VARIAVEL DO TIPO INT NA EEPROM

😎 eepromintsalvar Arduino 1.8.5
Arquivo Editar Sketch Ferramentas Ajuda
eepromintsalvar §
<pre>void setup() { Serial.begin(115200); EEPROM.begin(4096);</pre>
<pre>int valor = 1; EEPROM.write(0 , valor); EEPROM.commit(); }</pre>
<pre>void loop() { Serial.println("Salvo com sucesso!!"); }</pre>

Figura 50: Codigo Salvar int na EEPROM. Fonte: Autoria Própria.

⁴⁴ Disponível em: < <u>https://arduino-esp8266.readthedocs.io/en/latest/libraries.html#eeprom</u> >Acesso em: 28/07/2018.

EEPROM.begin(): é usado para carregar a memoria ROM da placa para ser utilizada,podendo ser carregada minimo 4 ao maximo de 4096kb que ela possui.

EEPROM.write(0,1): Para se realizar a gravação simples de dados basta usar o codigo, sendo o primeiro valor dentro do parentese o ponteiro do local na memoria e o segundo valor dentro do parentese o valor a ser salvo.

EEPROM.commit(): Para efetuar os dados que foram gravados na memoria.

4.5. REALIZANDO LEITURA DA VARIAVEL INT NA EEPROM



Figura 51: Codigo Leitura simples na EEPROM. Fonte: Autoria Própria.

EEPROM.read(0): Para Leitura realizar a leitura o valor atribuido dentro do parenteses se refere ao ponteiro indicando o lugar na memoria ao qual a variavel se encontra.

4.6. SALVANDO VARIAVEL DO TIPO STRING NA EEPROM



Figura 52: Codigo Salvando variavel String na EEPROM. Fonte: Autoria Própria.

toCharArray(): Essa função converte a variável String em Char, o primeiro parâmetro é a variável que receberá a conversão e o segundo parâmetro é o tamanho que a variável convertida terá.

lenght(): Calcula o tamanho que a variavel possui.

EEPROM.put(addressid, arrayToStore) realiza um a gravação da informação para dentro da memoria com o ponteiro da memoria inicial local aonde vair começar a ser realizado a gravação da informação e a informação que foi passada para a variavel arrayToStore, nesse momento caracter por caracter vai ser gravado na memoria.

EEPROM.end(): Utilizada para fechar a sessão de acesso a memoria.

4.7. REALIZANDO LEITURA DE VARIAVEL STRING NA EEPROM



Figura 53: Codigo Leitura de variavel String na EEPROM. Fonte: Autoria Própria.

Para realizar a leitura da variavel String na EEPROM seguimos o mesmo conceito abortado para salvar, utilizasse uma variavel String para receber o valor da busca.

EEPROM.get(addressId, arrayToStore): O primeiro parametro contem o endereço de memoria inicial e o tamanho que a string ocupa na memoria no caso a arrayToStore.

4.8. O QUE SÃO SENSORES

Sensores são agentes que servem para capturar dados informando ao circuito eletrônico da placa microcontroladora que ocorreu uma mudança no meio fisico, ele pega as mudanças ocorridas convertendo-as em valores que a programação consegue entender e tomar decisões. O valor adquirido pelo do sensor antes de ser atribuido a uma variavel passa por uma conversão que atraves da biblioteca do proprio componente traduz o sinal para um valor númerico que os programadores possam compreender.

Exemplo de sensores: Sensor de temperatura, umidade do ar, som, umidade do solo e outros.

Os valores capturados pelo sensor podem ser salvos em variaveis e ou utilizados para ativar outros componentes como atuadores, os atuadores podem recebem esses dados e pelo sistema modificarem as ações no meio fisico. Exemplo de atuadores: Relés, motores, Válvulas, Solenóides entre outros.

Exemplo de sensor utilizado nesse projeto: Sensor de umidade do solo.



Figura 54: Sensor umidade solo. Fonte: Autoria Própria. Especificações do Sensor:

-Tensão de operação entre 3,3 volts até 5 volts.

-Saida Digital TTL(D0) e Analógica(A0).

Pinos:

- GND: Terra.
- VCC: Alimentação 3,3volts até 5 volts.
- A0: Pino Analógico.



Figura 55: Codigo Sensor Solo. Fonte: Autoria Própria.

Na programação é declarada uma variavel do tipo int global acima da função setup de inicialização que será responsavel por receber os dados pelo pino analogico A0 da placa.

analogRead(sensorSolo): A função captura os dados do sensor passados pelo pino A0 convertendo e atribuindo o resultado para variavel. Logo abaixo dentro dos lfs o valor é comparado por outros valores fixos ficticios estabelecidos para teste no código para exibir o resultado do estado que o solo se encontra.

4.9. O QUE É O MODULO RTC DS3231

O Rtc⁴⁵ ou real time clock em inglês significa relogio de tempo real que possibilita registrar informações de data e hora para que o usuario possa utilizar junto com outros componentes. O Rtc DS3231 apos alimentado com as informações trabalha por conta propria computando a data e a hora, ele possui um socket para bateria de 3 volts em sua placa que serve para que mesmo que o microcontrolador que estiver conectado a ele pare de enviar alimentação de energia o Rtc passa a ser alimentado pela bateria operando normalmente.

O modulo trabalha com informações como anos, meses, dias, horas, minutos, segundos e pode operar com formato de 12 horas ou 24 horas.



Figura 56: Modulo Data RTC DS3231. Fonte: Autoria Própria.

⁷²

⁴⁵ Disponível em: < https://www.arduino.cc/en/Reference/RTC >Acesso em: 28/07/2018.
Os pinos são: VCC - Entrada de 3.3 volts ou 5 volts. GND - terra. Chip: ds3231 Chip de memória: AT24C32 pino 32k para acesso a memoria. SCL e SDA: comunicação por protocolo I2C que é a comunicação entre dispositivos, o I2C trabalha no modelo Mestre e escravo. Aonde o microcontrolador é o mestre e faz requisições de dados para o RTC sendo o escravo que retorna com a resposta.

SQW: Usado para acesso a disparos de alarmes do Rtc.

As bibliotecas são disponibilizadas por varios fornecedores, para cada tipo de microcontrolador precisa encontrar a biblioteca ideal para o funcionamento na programação.

Abaixo uma lista com o nome e o repositorio de algumas bibliotecas disponiveis:

-RtClib localizado no endereço https://github.com/adafruit/RTClib.

-DS3231 localizado no endereço https://github.com/NorthernWidget/DS3231.

-Rtc3231 localizado no endereço https://github.com/Makuna/Rtc.

-Ds3231_simple localizado no endereço https://github.com/sleemanj/DS3231_Simple.

Após baixar a biblioteca da fonte escolhida abra a Arduino IDE, clique em sketck depois em incluir biblioteca, depois em adicionar biblioteca zip e aguardar o processo de carregamento da biblioteca dentro da IDE.



No codigo da imagem foi inserido o include da biblioteca Rtc3231, na linha abaixo instanciado um objeto do tipo RTC_DS3231 que é responsavel por fazer as chamadas de função da biblioteca RTCLib. O função rtc.begin() faz a inicialização de acesso ao modulo, e a função rtc.adjust(DateTime(2018,7,20,12,10,0)) é responsavel por salvar as informações de data e hora para que o modulo possa operar, seguindo os valores dentro do parenteses do DateTime os primeiros 3 numeros são ano, mes e dia, e os ultimos 3 numeros são hora, minuto e segundos.

🥺 recuperardadosmodulodata Arduino 1.8.5	and state in a longer land
Arquivo Editar Sketch Ferramentas Ajuda	
recuperardadosmodulodata	
<pre>#include <rtclib.h></rtclib.h></pre>	
RTC_DS3231 rtc;	
<pre>void setup() {</pre>	
<pre>Serial.begin(115200);</pre>	
<pre>rtc.begin();</pre>	
}	
<pre>void loop() {</pre>	
<pre>DateTime agora = rtc.now();</pre>	
<pre>Serial.println("");</pre>	
<pre>Serial.println("Ano:");</pre>	
<pre>Serial.println(agora.year());</pre>	
<pre>Serial.println("Mes:");</pre>	
<pre>Serial.println(agora.month());</pre>	
<pre>Serial.println("Dia:");</pre>	
<pre>Serial.println(agora.day());</pre>	
<pre>Serial.println("Hora:");</pre>	
<pre>Serial.println(agora.hour());</pre>	
<pre>Serial.println("Minuto:");</pre>	
<pre>Serial.println(agora.minute());</pre>	
<pre>Serial.println("Segundo:");</pre>	
<pre>Serial.println(agora.second());</pre>	
<pre>Serial.println("");</pre>	
delay(1000);	
}	

Figura 58: Codigo Recuperando dados do Rtc ds3231. Fonte: Autoria Própria.

No Codigo da imagem segue os mesmo principios que o script anterior, primeiro chama a função pelo objeto rtc.begin() para carregar o acesso ao modulo, dentro da função loop o comando DateTime agora = rtc.now() faz com que o objeto agora do tipo DateTime receba os valores do objeto rtc pela função now() que passa os dados de data e hora do exato tempo presente. Os comandos de função year() exibem o ano, month() o mes, day() o dia, hour() a hora, minute() o minuto, second() o segundo. A função delay(1000) aguarda o tempo de 1 segundo para retornar a função loop para assim conseguir pegar a data e hora do exato tempo presente que foi processado pelo modulo Rtc.

4.10. CONECTANDO ESP8266 12E COM A INTERNET

O Microcontrolador conectado a internet proporciona varias possibilidades de desenvolvimento de internet das coisas, podendo realizar transferencias de informações que são geradas pelos sensores ao detectarem situações no meio fisico ou para que os atuadores tomen ações no meio fisico. Outra possibilidade com o microcontrodor conectado com a internet é de salvar informações no banco de dados e tambem poder salvar em sua memoria novas informações de funcionamento que estão no banco de dados.

4.11. CONECTANDO ESP8266 12E COM WIFI

Para instalar a biblioteca ESP8266 primeiro abra a Arduino IDE, acesse Arquivos depois em preferencias, dentro de preferencias encontre o campo URLS Adicionais para Gerenciadores de placas, dentro deste campo adicione o seguinte caminho de pagina da internet "http://arduino.esp8266.com/versions/2.4.1/package_esp8266com_index.json".

Entre em Ferramentas depois gerenciador de placas, digite na busca Esp8266 e instale a nossa placa dentro da IDE. Agora deve escolher a versão correta para que a placa ESP8266 funcione corretamente.

Abaixo o codigo de conexão da ESP8266 12E com a internet.

```
conectandoWifiesp | Arduino 1.8.5
Arquivo Editar Sketch Ferramentas Ajuda
        Ð
  conectandoWifiesp
#include <ESP8266WiFi.h>
#define WIFI_SSID "teste"
#define WIFI_PASSWORD "12345678"
void setup()
{
    Serial.begin(115200);
    WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
    Serial.print("Conectando a internet...");
    while (WiFi.status() != WL CONNECTED) {
          Serial.print(".");
          delay(500);
    3
    Serial.println();
    Serial.println( "Conectado!!" );
    Serial.println("IP Adquirido:");
    Serial.println( WiFi.localIP());
}
void loop(){
```

Figura 59: Codigo Conectando ESP8266 12E com Wifi. Fonte: Autoria Própria.

O codigo apresenta include da biblioteca ESP8266WIFI para que possamos ativar a conexão do microcontrolador com a internet, foi declarado Define WIFI_SSID ao qual possui o nome do roteador que será feita a conexão e o WIFI PASSWORD contendo a chave de do roteador liberar а internet. Ο acesso para comando Wifi.begin(WIFI SSID, WIFI PASSWORD) usa o objeto Wifi com função de iniciar a conexão com roteador passando os atributos de WIFI_SSID nome do roteador e WIFI_PASSWORD senha do roteador.

O laço de repetição while é executado até que a conexão com a internet tem sido realizada com sucesso.

Seguindo para o comando Wifi.localIP() o objeto Wifi faz chamada da função localIP() retornando o endereço local disponivel que o microcontrolador adquiriu, sendo esse endereço adquirido automaticamente após a chamada de função Wifi.begin().

4.12. FIREBASE COM ESP8266 12E

O Firebase é um banco de dados em nuvem do tipo não relacionam que favorece a integração de muitas aplicações de Web e internet das coisas. Sua interface disponibiliza Serviços e ferramentas variadas que o usuario pode agregar dependendo de suas necessidades. As bases de dados são abordadas como coleções de dados todas em formato do tipo json, sua estrutura de base de dados segue um estilo de modelo de arvore ao qual um pai vai ter seus filhos e seus filhos outros filhos ou podendo possuir coleções cada uma com sua chave gerada automaticamente pelo banco.

A biblioteca do Firebase Arduino está localizada no endereço "https://github.com/firebase/firebase-arduino", precisa ser adicionada dentro da Arduino IDE em incluir bibliotecas para ser utilizada as funções de conexão, salvar, busca, atualizar e excluir informações. A documentação da biblioteca Firebase Arduino⁴⁶ fornece muitas funções para a transação de dados, para cada tipo de dado uma função especifica para manipular a informação.



Figura 60: Estrutura de Bases de dados e coleções no Firebase. Fonte: Autoria Própria.

⁴⁶ Disponível em: < <u>http://firebase-arduino.readthedocs.io/en/latest/</u> >Acesso em: 28/07/2018.



Figura 61: Codigo Conectando ESP8266 12E ao Firebase. Fonte: Autoria Própria.

O código segue partes do script da figura conectando com wi-fi adicionado os pacotes da biblioteca Firebase, declarando um define FIREBASE_HOST com o endereço da base de dados fornecida pelo banco de dados FIREBASE e o define FIREBASE_AUTH com a chave segredo do banco FIREBASE. O comando Firebase.begin(FIREBASE_HOST) utiliza a função begin() do objeto Firebase passando o atributo FIREBASE_HOST para que se inicie a conexão com o banco de dados.

4.13 SALVAR E BUSCAR DADOS PELO ESP8266 12E NO FIREBASE



Figura 62: Codigo Salvar e Recuperar dados ESP8266 12E com Firebase parte 1. Fonte: Autoria Própria.

```
andodadosWifiespFirebase | Arduino 1.8
rquivo Editar Sketch Ferramentas Ajuda
             + +
 salvandodadosWifiespFirebase
void loop() {
    if(x == 0){
   StaticJsonBuffer<200> jsonBuffer;
   JsonObject& valueObject = jsonBuffer.createObject();
   valueObject["nome"] = "joão paulo";
valueObject["rua"] = "piratininga";
   valueObject["cidade"] =
                                   "assis";
   String chave = Firebase.push( "/alunos" , valueObject );
   if (Firebase.failed()){
      Serial.print("setting /number failed:");
      Serial.println(Firebase.error());
      return; }
    else{
      Serial.println("Salvo aluno com sucesso!! ID:");
      Serial.println(chave);
      Serial.println("Iniciando busca de dados no Firebase pela chave adquirida:");
      String nomeGet = Firebase.getString("/alunos/" + chave + "/" + "nome");
      String ruaGet = Firebase.getString("/alunos/" + chave + "/" + "rua");
String cidadeGet = Firebase.getString("/alunos/" + chave + "/" + "cidade");
      String ruaGet =
      Serial.println("Resultado da Requisição getString da Coleção Alunos:");
      Serial.println("Nome:");
      Serial.println(nomeGet);
      Serial.println("Rua:");
      Serial.println(ruaGet);
      Serial.println("Cidade");
      Serial.println(cidadeGet);
      Serial.println("Concluido busca!!");
      x = 1:
```



O codigo segue partes do script da figura conectando com firebase adicionando uma variavel int X que será responsavel por certificar que as informações serão salvas com sucesso no banco de dados.

Na linha com o comando StaticJsonBuffer<200> jsonBuffer : StaticJsonBuffer⁴⁷ é responsavel por criar o tamanho em bytes que o documento objeto json precisará para receber as informações.

O comando JsonObject& valueObjetc = jsonBuffer.createObject() : cria o objeto valueObject do tipo JsonObject⁴⁸, recebendo o objeto jsonbuffer com a capacidade de bytes informadas.

O comando valueObject["nome"] = "joão paulo": é responsavel por agregar os atributos para do objeto como a string contendo a string "joão paulo".

O Comando String chave = Firebase.push("/alunos", valueObject) : executa o metodo criar com a função Firebase.push() passando os atributos do caminho da base de dados e o objeto json para ser salvo, a variavel String chave recebe o id que é gerado automaticamente pelo banco de dados.

⁴⁷ Disponível em: < <u>https://arduinojson.org/v5/doc/memory/</u> > Acesso em: 28/07/2018.

⁴⁸ Disponível em: < <u>https://arduinojson.org/v5/api/jsonobject/</u> > Acesso em: 28/07/2018.

Firebase.failed() : é usada para verificar se ocorreu falha na conexão do banco ao tentar realizar transação de informações.

Firebase.error() : informa o erro que ocorreu no processo de conexão.

Firebase.getString() : executa a Busca de dados do tipo String do atributo desejado dentro do objeto no banco de dados.

O Comando String nomeGet = Firebase.getString("/alunos/" + chave + "/" + "nome") : executa o metodo busca com a função Firebase.getString() passando os atributos do caminho da base de dados, a variavel chave que contém a id do objeto no banco de dados e o nome do atributo que deseja ser solicitado, a variavel nomeGet recebe a informação que foi retornada do banco de dados.

5 LEVANTAMENTO DOS REQUISITOS

Realizado o levantamento de requisitos foi obtido a Lista de Eventos das atividades no Campo desde o controle das variedades, controle do campo, controle dos cultivos chegamos ao Diagrama de Classe e Casos de Uso provisório que precisara ser estudado e melhorado no decorrer do projeto. O levantamento ajudara na implementação da aplicação e do protótipo de dispositivo que contribuirá melhorando as atividades exercidas na Horta.

5.1 LISTA DE EVENTOS

- criar cultivo.
- consultar cultivo.
- alterar cultivo.
- excluir cultivo.
- criar variedade.
- consultar variedade.
- alterar variedade.
- excluir variedade.
- criar campo.
- consultar campo.
- alterar campo.
- excluir campo.
- criar dispositivo.
- consultar dispositivo.
- alterar dispositivo.

- excluir dispositivo.
- criar programação.
- consultar programação.
- alterar programação.
- excluir programação.
- criar status.
- consultar status.
- consultar status histórico.

5.2 DIAGRAMA DE CASO DE USO

Abaixo a imagem do Diagrama de Casos de Uso demonstrando as atividades do Usuário.





Figura 65: Diagrama de Caso de Uso Dispositivo. Fonte: Autoria Própria.

5.3 DIAGRAMA DE CLASSE

Abaixo a imagem do Diagrama de Classes obtido através da Lista de Eventos.



5.4 PROPOSTA DE TRABALHO

O Objeto de estudo para o desenvolvimento do projeto será o cultivo de alface em um campo teste, pois a mesma é de fácil manejo cultivado e comercializado no Brasil, e em outros países. Fatores como clima, o solo, a água e a infraestrutura são necessários para o inicio do plantio.

Existem diversos grupos de cultivares de alface no mundo, dentre eles os mais conhecidos são Crespa, lisa, Americana, Romana e Mimosa.

Sendo a Alface do Grupo Roxo e Vermelho fazer parte de todos os grupos citados acima, menos o tipo cultivar Americana. A Alface roxa e vermelha é caracterizada pela apresentação de pigmentação vermelha nas suas folhas, e sua época de plantio atinge o ano todo.

Foram estudados dois tipos de Modalidade de Cultivo dentre elas Plantio em túnel baixo e Plantio a campo aberto em canteiros com uso de cobertura de plástico e sistema de irrigação ou gotejamento.

O conceito de uso do Plantio em túnel baixo consiste em ter arcos longos com filme plástico que cobrem todo o canteiro possuindo em media 3 metros entre os arcos, e 2 metros da altura do solo.

Isso garante proteção das cultivares em tempos com muita chuva e do sol intenso, isso aumenta a qualidade e aumento da produtividade.



Figura 67: Plantio túnel baixo de plástico pelo Embrapa. Fonte: http://www.embrapa.br

O conceito do Plantio a campo aberto em canteiros com uso de cobertura de plástico e sistema de irrigação por irrigação ou gotejamento, como já afirma o nome do conceito usa a pratica do plantio em campo aberto com os canteiros coberto por um plástico com nome mulching podendo ser utilizado por até quatro plantios, essa modalidade aumenta a produtividade, mas em períodos secos a colheita fica desidratada.



Figura 68: Plantio campo aberto em canteiros com uso de cobertura de plástico e sistema de irrigação ou gotejamento. Fonte: www.negreira.com.br

No estudo será usado o conceito dos dois métodos, o plantio de campo aberto com cobertura de plástico com uso de irrigação por gotejamento junto com o Plantio em Túnel Baixo. Teremos duas Fileiras de sementes lado a lado ou duas fileiras de replantio de alface lado a lado em campo teste no vaso retangular, que terá um túnel baixo protegendo o campo e o dispositivo de chuva ou sol intenso.

O projeto desenvolvido consiste em ter um sistema web que mantenha o campo, as variedades, o cultivo, assim como o dispositivo e a programação do dispositivo que serão responsáveis pelo monitoramento, e desenvolvimento da variedade escolhida para o cultivo na horta. O sistema tem acesso a todos os dados de humidade e irrigação executada no campo, assim como datas do ciclo de semeadura e replantio.

Utiliza o Esp8266 12e para criar o protótipo do dispositivo que realiza o monitoramento através dos sensores e salva as informações em nuvem utilizando o banco de dados Firebase que possui um serviço Restful integrado.

Abaixo uma previa da representação do projeto, no sistema web imagem 1 faz o controle criando o campo as variedades e o cultivo, adiciona o dispositivo com a programação no cultivo desejado, e instala o equipamento no campo, o dispositivo imagem 2 realizará as tarefas que foram programadas, conversando com o banco de dados Firebase imagem 3 para salvar as informações, o equipamento também se atualiza se houver mudança na sua programação, por fim o usuário poderá consumir os dados no sistema que são gerados pelo dispositivo que monitora a horta.



Figura 69: Projeto no Campo. Fonte: Autoria própria.

Para o projeto conseguir funcionar corretamente foi estudada uma arquitetura de IOT para o sistema e os dispositivos pudessem trabalhar de forma conjunta, a arquitetura de IOT além de ser uma arquitetura tambem propoem a forma como deve ser exercida a forma de trabalho para todos os elementos nela juntos trabalhem e obtenha sucesso.



Figura 70: Modelo de Arquitetura de IOT para o Campo. Fonte: Autoria própria.

A imagem da Arquitetura desmonstra a conexão do sistema por varios caminhos como o roteador acessando pelo o sistema atraves de um Notebook ou Celular, ou tambem acessando o sitema por um celular se comunicação por Torre de celular que por sua vez faz ligação a base de dados do Banco na Nuvem. Como demonstrado os usuarios podem acessar o sistema por multiplas plataformas.

A Fabrica de Dispositivos é o local que se encontra a linha de montagem dos dispositivos e tambem a manutenção, após o dispositivo ser criado no sistema com suas identificações a linha de montagem recebe suas informações inseri os dados e o envia para o local que foi destinado para realizar suas atividades.

6 CONCLUSÃO

No desenvolvimento inicial do projeto foi realizado o levantamento de requisitos de agricultura no setor de horta, possibilitando alcanlar os casos de uso que o sistema de gerenciamento do campo necessitava para ser implementado, assim como tambem as informações para o desenvolvimento do prototipo de dispositivo para agricultura.

Foram estudados camponentes eletronicos, tecnologias e conceitos para criar a aplicação e o prototipo de dispositivo que ambas integrão ao campo teste, executando tarefas e coletando os dados tornando as atividades mais precisas em sua execução.

A aplicação realizou o gerenciamento das variedades, do campo, do cultivo, do dispositivo e da programação do dispositivo com sucesso. O Sistema terá continuidade realizando melhorias em sua estrutura como tambem em desenvolver a aréa de login e segurança do todo. Foi de uso fundamental e satisfatorio a integração da aplicação com framework IONIC e o banco de Dados Firebase, facilitando o desenvolvimento rapido do sistema sem ter a necessitadade de criar uma aplicação Rest para ponte de conexão entre as duas tecnologias já que o Firebase provém serviços Rest.

O dispositivo ou maquina obteve exito em suas atividades, coletando dados pelo sensor de umidade do solo, coletando dados fornecidos pelo modulo de data que fornece data e horas. O estudo de aproveitamento do uso da memoria flash do modulo ESP favoreceu a agregação de dados particulas de identidade, nome, programação necessarios para se criar um Objeto IOT, foram desenvolvidos dois tipos de codigos o primeiro de agregação de atributos e diretrizes de execução de atividades do objeto.

Dispositivo, o segundo codigo final para o processo de atividades, os codigos fonte originais não foram anexado ao trabalho por conta de principios empresariais futuros disponibilizando somente os codigos de pesquisa do uso das bilbiotecas separadamente. Foi agregado junto aos códigos nas ultimas semanas de teste do dispositivo o fator de qualidade de uso da memoria flash da maquina ESP8266 que são contadores de uso no ato do salvamento de informações na memoria, a memoria precisa ser testada para obter parametro de tempo de uso sobre a vida util até que fique obsoleta. A memoria foi fundamental nesse projeto para transforma-lo em um objeto IOT.

Uma das influencias para o desenvolvimento deste projeto de IOT na questão de uso da memoria da maquina foi imspirado em uma serie chamada Westworld ficticia, as quais existem varias maquinas do tipo Androids e todas elas possuem diretrizes de atividades, identificações e narrativas.

O ponto que não foi integrado ao dispositivo foi o atuador valvula solenoide, por questões financeiras e de tempo. A falta que o atuador proporcionou teve que ser suprida realizando irrigação manualmente para ser testado o dispositivo, sendo necessario futuramente na continuação do projeto integrar a valvula e realizar novos estes obtendo resultados para melhorar toda a aplicação.



Figura 71: Aplicação e Dispositivo em execução. Fonte: Autoria própria.

7 CRONOGRAMA

Cronograma	NOV	DEZ	JAN	FEV	MAR	ABR	MAI	JUN	JUL	AGO	SET
Descrição do projeto Proposto											
Levantamento de requisitos											
Cronograma											
Estudo das Tecnologias											
Desenvolvimento da Aplicação Hibrida											
Armazenamento e Visualização dos dados											
Desenvolvimento do protótipo de dispositivo											
Testes											
Entrega do Projeto											
Apresentação para banca											

Tabela 1: Cronograma de atividades TCC 2017-2018Fonte: Imagem do Windows programa Excel – Autoria Própria

8 REFERÊNCIAS

ABREU, LUIS Typescript – O Javascript Moderno para Criação de Aplicações. Editora FCA, 2017.

BANZI, MASSIMO Primeiros passos com o Arduino Editora Novatec, 2011.

CAMDEN, RAYMOND K. Apache Cordova In Action Editora Manning, 2016.

DIAS, EMÍLIO Desmistificando Rest com Java Editora Algaworks, 2016.

FACCIONI FILHO, MAURO Internet das Coisas Editora Unisulvirtual Palhoça, 2016.

GILCHRIST, ALASDAIR Industry 4.0 The Industrial Internet of Things Editora APRESS, 2016.

JAVED, ADEEL Criando projetos com Arduino para Internet das Coisas Editora Apress Novatec, 2016.

JUSTIN, JOYCE Learn Ionic 2 Editora Apress, 2017.

KARVINEN, KIMMO Make Getting Started with Sensors Editora Maker Media, 2014.

MAZOYER, MARCEL História das agriculturas no mundo Editora UNESP, 2008.

MAZZA, LUCAS HTML5 e CSS3 Domine a web do futuro Editora Casa do Código, 2012

MCROBERTS, MICHEL Arduino Básico Editora Novatec, 2011.

MOLIN, J. P. Agricultura de Precisão Editora Graphium Editora de Textos, 2015.

OLIVEIRA, SÉRGIO Internet das coisas com ESP8266, Arduino e Raspberry pi Editora Novatec, 2017.

PRESSMAN, R. S. Engenharia de Software ed. São Paulo: AMGH Editora LTDA, 2011.

SEBRAE Alface Saiba como cultivar hortaliças para colher bons negócios Editora PlanoMídia, 2011.

VERAS, MANOEL Computação em Nuvem Editora Brasport, 2015