INTERNET DAS COISAS EM SISTEMAS DE GERÊNCIA PARA AGRONEGÓCIO

MEDEIROS, Guilherme Schubert [1]
SENHORINHA, Márcio José Käms [2]

RESUMO

A tecnologia não para de evoluir, e atualmente possui-se muitas formas de automatizar processos e facilitar a produção de produtos. Com os microcontroladores e a internet das coisas pode-se melhorar a vida de muitos agricultores de pequeno e médio porte, tornando as irrigações inteligentes por exemplo e mostrando nível de minérios no solo. Usando a programação para desenvolver um *software* que envie informações pela internet para um *webservice* que coleta e mostra *online* as informações, tudo usando um microcontrolador conectado a alguns sensores, ou seja um protótipo, onde será efetuado um levantamento de tudo que irá ser necessário para o funcionamento adequado e dessa forma estar aprendendo algo novo, utilizando da metodologia exploratória para desenvolver este trabalho. Este é um trabalho complexo e que é de grande aprendizado principalmente para quem não possui muito conhecimento da área de programação e eletrônica, porém que pode ajudar muito a agroindústria.

Palavras-chave: IOT; Microcontroladores; Agronegócio.

ABSTRACT

Technology does not stop evolving, and currently there are many ways to automate processes and facilitate the production of products. With microcontrollers and the internet of things, the lives of many small and medium-sized farmers can be improved, making irrigation smarter, for example, and showing the level of minerals in the soil. Using programming to develop a software that sends information over the internet to a webservice that collects and displays the information online, all using a microcontroller connected to some sensors, that is, a prototype, where a survey of everything that will be necessary need for its correct functioning and thus be learning something new, using the exploratory methodology to develop this work. This is a complex job and it is a great learning experience, especially for those who do not have much knowledge in the area of programming and electronics, but it can help agribusiness a lot.

Keywords: *IOT; Microcontrollers; Agribusiness.*

1 INTRODUÇÃO

A agroindústria no Brasil é muito importante, pois é a principal forma de gerar economia no país, ou seja, quanto mais produtivo e eficaz o sistema agropecuário

[1]Graduando do Curso de Sistemas da Informação do Centro Universitário UNISOCIESC, guilhermeschu10@gmail.com;
[2]Márcio José Käms Senhorinha: Mestre em Eng de Produção, Centro Universitário UNISOCIESC, marcio.senhorinha@unisociesc.com.br;

Joinville – SC, 29 de Novembro de 2021.

se tornar, melhor se tornará as condições econômicas para o brasileiro, com os avanços da tecnologia, é possível utilizar de *hardwares* e *softwares* para auxiliar neste propósito. (FELEMA; RAIHER; FERREIRA, 2013)

A Internet é a maior rede que existe e junto com sua evolução, começou a se utilizar dessa rede para avançar com a tecnologia no meio da sociedade em diversas áreas, como o *e-commerce* permitindo a compra de algo da sua casa, os aplicativos de mensagens como o *WhatsApp* ou *Telegram*, dentre outros que possibilitam a conversa entre 2 ou mais pessoas que estejam em localizações totalmente distintas no mundo, com esse avanço veio também a IOT ou Internet das Coisas, permitindo ligar não só celulares,computadores e tablets a internet como também equipamentos eletrodomésticos, como a TV, geladeira, cafeteira, etc.

Dessa forma, automatizando os mesmos através da internet, ou seja programar a hora da cafeteira ligar para quando o indivíduo chegar do trabalho, o café já estará passado e pronto, ou a geladeira avisar quando estiver faltando algum alimento dentre outras possibilidades. (LEITE; MARTINS; URSINI, 2021)

Internet das Coisas, ou *Internet of Things* (IoT) é uma área da Tecnologia da Informação (TI) onde se estuda como conectar dispositivos a internet e torná-los mais inteligentes, automatizados e acessíveis de qualquer lugar, ou seja, não é apenas fazer alguns leds ligarem ou algo assim como muitos pensam (OLIVEIRA 2021).

Atualmente, com a evolução da internet a comunicação tornou-se possível e acessível utilizar microcontroladores e sensores para fazer análises de variáveis e determinar o que precisará ser otimizado ou não nas plantações de forma mais simples e centralizado tornando assim um ótimo investimento, pois além do custo não ser alto, irá trazer muitas informações para que o produtor consiga destinar sua estratégias de negócio.

2 INFRAESTRUTURA

A infraestrutura na área da Tecnologia da Informação mostra como que os servidores, rede, cabeamento estão implementados, podendo ser *On-premise* ou em *Cloud*, a diferença é que no modelo *on-premise* fica tudo no local, ou seja a empresa

acaba custeando tudo, como servidores, *switches*, *backup*, profissionais para gerenciar e dar manutenção no ambiente, etc.

No modelo de *cloud* a empresa paga pelo que ela utiliza, e acaba economizando em alguns aspectos por exemplo, o empresário adquire um servidor onde as configurações de *hardware* são ótimas, porém na maioria do tempo acaba ficando ocioso, no modelo de *cloud* seria configurado o servidor de acordo com sua necessidade e principalmente utilizá-lo em todo o seu potencial, tornando assim muitas vezes um investimento em vista dos outros custos envolvendo uma infraestrutura *on-premise*, como ar condicionado, energia, *nobreaks*, etc.

Desse modo, o ideal seria fazer um *trade-off* analisando tudo o que a empresa precisa para manter o ambiente funcionando, analisar o custo de uma eventual parada do ambiente por conta de algum servidor ou equipamento e desse modo verificar o custo de manter em nuvem ou local.

2.1 *HARDWARE*

Hardware seria tudo que pode-se tocar, existem vários tipos de *hardwares*, como o processador, placa mãe, *Hard Disk Drive (HDD)/Solid State Drive (SSD)*, microcontroladores e sensores, e combinando alguns deles se compõe servidores, computadores entre outros, o computador por exemplo, precisa de placa-mãe, memória RAM, armazenamento e processador.

O que controla esses componentes são os sistemas operacionais como *Windows, Mac, Linux, Unix, Android,* ou em caso de sistemas embarcados ao invés de um sistema operacional, seria apenas um *software* sempre que o *hardware* utilizado ligar esse software irá ser executado.

Com o passar do tempo, o *hardware* veio evoluindo, um exemplo disso é o armazenamento que hoje temos disponível terabytes o que se compararmos com o início das soluções de armazenamento é umas 1000 vezes maior e com um custo muito menor também, hoje além de armazenamento em HDD, temos o SSD que é muito mais veloz que um HDD por conta de ser eletrônico e não magnético, esse é apenas um exemplo da evolução dos hardwares, mas os processadores, memórias e outros também tiveram seu desenvolvimento. (HERNANDES, 2021)

2.1.1 Microcontroladores

Os microcontroladores existem desde meados da década de 70, e com a evolução dos sistemas embarcados, começaram a assumir tarefas que eram feitas de forma manual e transformá-las em tarefas automatizadas. (OLIVEIRA 2021)

A forma de se utilizar seria por interfaces de entrada e saída com dispositivos elétricos como por exemplo relés, são muito utilizados nas indústrias e é uma grande aliada da Indústria 4.0 e atualmente se expandiram para demais segmentos do mercado, inclusive em ambientes domésticos, transformando as casas inteligentes.

Os microcontroladores são diferentes dos microprocessadores que necessitam de periféricos para poder utilizar, como teclados, monitores e mouse. A principal diferença é que eles são destinados a usos diferentes, microprocessadores ou como um mini computador onde precisa de um sistema operacional, precisa dar o *boot*, pode rodar alguns serviços como um servidor *web*, ou usar para tarefas do dia a dia de um usuário comum, como acessar e-mail e navegar na internet.

Os microcontroladores são destinados principalmente para usos sem interação ou acesso, ou seja, é desenvolvido um *software*, onde é feito *upload* desse para o microcontrolador e toda vez que ele desligar e ligar a única coisa na memória dele vai ser rodar o *software* criado. (NERYS, 2021)

Para este trabalho foi utilizado o microcontrolador ESP32, pois o mesmo possui a quantidade suficiente para fazer a conexão dos sensores e já possui o módulo WIFI integrado.

2.1.2 Sensores

Os sensores são componentes eletrônicos ou elétricos, os mesmos se conectam às interfaces de entrada e saída do microcontrolador para que possa coletar as informações necessárias para o trabalho, existem vários sensores e cada um tem sua finalidade. Existem os LEDs que são utilizados para informar algo ao usuário como por exemplo quando uma bateria estiver carregada ou se está passando energia no circuito, tem os relés que podem ser usados para acionar equipamentos, como motores, eletrodomésticos, enfim, existem muitos sensores no mercado, será utilizado alguns para este trabalho (OLIVEIRA 2021). Os seguintes sensores serão utilizados:

a) sensor de umidade do solo;
b) sensor de temperatura;
c) relé.

Além dos sensores, será utilizado 2 placas solares de 6V conectadas em série fornecendo no total 12V e 3 baterias recarregáveis de 4.2V conectadas em série também fornecendo o total de 12,6V para alimentar o circuito, além de outros componentes eletrônicos.

2.1.3 Servidor

A arquitetura cliente-servidor é a mais utilizada atualmente, ela será implementada no projeto, os microcontroladores serão clientes, com os usuários acessando a página *Web*, já o servidor irá fornecer os dados para o cliente. Nesta arquitetura não importa qual *hardware* usa o servidor, podendo ser até mesmo um computador pessoal, porém quando é utilizado em produção ou seja com milhares de clientes requisitando informações do servidor, é imprescindível que o mesmo seja dedicado para esse serviço, o que envolve um *hardware* mais específico. (TANENBAUM 2013)

Para este trabalho será utilizado o servidor *WebService* como máquina virtual utilizando o *software* VirtualBox, onde será instalado a distribuição CentOS e será configurado para servir a página *web* onde o cliente terá acesso pela internet.

2.2 *SOFTWARE*

Diferentemente do *hardware* o *software* seria o que não podemos tocar, existem tipos de *softwares* como o que é desenvolvido para se manipular o *hardware*, como sistemas operacionais, tem os que são desenvolvidos para aplicação como aplicativos de celulares, jogos, entre outros tipos.

Os *softwares* dentro da área da tecnologia é o que mais se necessita de atualização, pois se ficar obsoleto, pode causar problemas de segurança ou performance, sendo necessário manutenção e atualização das tecnologias, comparado ao *hardware* que tem um avanço mais lento em seu desenvolvimento e

leva mais tempo para que seja necessário a troca para algo mais novo no mercado. (BASNAYAKA, 2021)

2.2.1 API

A interface de programação de aplicações ou simplesmente API é um conjunto de normas e definições que facilitam a comunicação entre aplicações, ou seja mesmo com linguagens de programação diferentes é possível a troca de informações, por exemplo se tem um *software* criado em java que disponibiliza uma API para coletar informações desse sistema, pode assim usá-la em um sistema que não foi escrito em java e ter essas informações. Isso facilita o trabalho de muitos programadores, já que não precisa ficar fazendo tudo do zero ou seja se é necessário usar uma funcionalidade de GPS no *software*, é só utilizar a API do Google Maps ou de algum outro *software* de mapa sem precisar programar uma funcionalidade de mapa do zero. (OLIVEIRA 2016)

O NodeJS foi desenvolvido por Ryan Dahl e sua principal diferença dentre as outras linguagens de *backend* é que ele é *single thread* e trabalha de forma não bloqueante, tornando assim as requisições mais rápidas e não acaba sofrendo de *deadlock*. É utilizado o JavaScript para se programar em Node por isso o nome de NodeJS, o JS é de javascript, foi usado o motor V8 que é o mesmo usado no navegador Google Chrome para ser desenvolvido e é amplamente usado no mercado atualmente em vários tipos de aplicações, como aplicações em tempo real, desenvolvimento de API etc (PEREIRA, 2014).

Será usado o NodeJS para desenvolver a API e toda a parte de *backend* necessária para este trabalho, devido às vantagens que ele entrega e por conta de ser utilizado Javascript que é uma linguagem leve e tranquila de se utilizar.

2.2.2 *FrontEnd*

Frontend é um termo utilizado na programação, indica aquilo que o cliente vê, ou seja a parte visual do código, quando é aberto um site por exemplo, os banners, menu, GIFs e tudo que você consegue ver é o *frontend*, em aplicativos de celulares

segue o mesmo conceito também, ou seja tudo que você consegue visualizar é *frontend*. (SOUTO 2019)

E para desenvolver o *frontend* existem *frameworks* para agilizarem o trabalho, por exemplo o react, angular ou vue que são os 3 mais famosos e utilizados no mercado atualmente, e para este trabalho será utilizado o *react* como *framework* para desenvolver o *frontend*.

React nada mais é que um *framework frontend* que trabalho com o modelo de componentização, foi desenvolvida pela equipe do facebook e torna a construção visual muito mais simples e reutilizável, pois você quebra em componentes as partes visuais sendo assim mais fácil a manutenção e reutilização do conteúdo, pois por exemplo se existe um formulário com vários inputs de texto, ao invés de ficar copiando e colando várias tags, cria-se um componente de input que recebe as props que seriam as propriedades, sendo muito mais simples customizar o componente e para dar manutenção (LINS, 2021).

3 METODOLOGIA

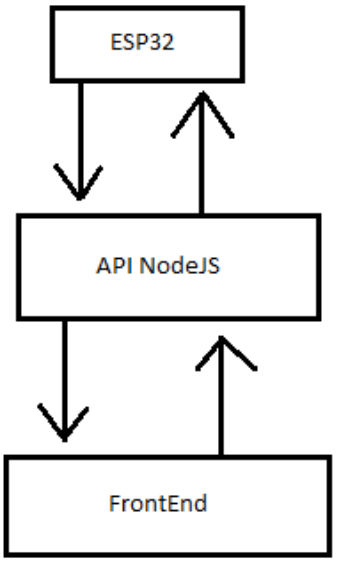
Para este trabalho foi realizada uma pesquisa de modo exploratório, obtendo a coleta das informações teóricas e com base na coleta das mesmas, colocando em prática através da criação de um protótipo com seus respectivos testes, registrando todo o procedimento, para análise dos acertos e das dificuldades encontradas durante sua realização.

Efetuou-se testes em emulador para validar o protótipo, após testes, foram adquiridos os componentes para efetuar a montagem do mesmo.

4 DESENVOLVIMENTO

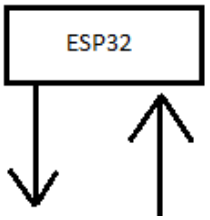
A Figura 1, demonstra um fluxograma do funcionamento do *software* desde a requisição do microcontrolador.

Figura 1 - Fluxograma do funcionamento do *software*

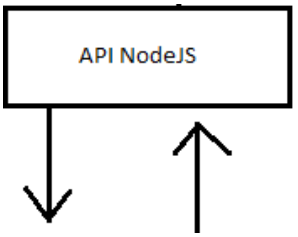


Fonte: o autor, 2021

Detalhando por partes e mais tecnicamente o fluxograma acima, tudo inicia no microcontrolador:



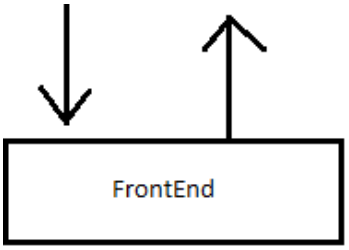
Para esse projeto foi escolhido o ESP32, ele é configurado na rede WIFI e atua como um cliente HTTP, onde o mesmo envia requisições a cada 15 segundos usando o verbo PUT do protocolo HTTP com os valores dos sensores.



Então, quando a requisição chega na API que foi criada utilizando o runtime javascript NodeJS o mesmo verifica se o protocolo é POST ou PUT, se for POST é feito uma inserção no banco de dados, se for PUT, é feito a atualização dos valores, essa API suporta requisição usando o GET do protocolo HTTP também, onde pode

se usar para obter os valores de todos os sensores ou apenas de um sensor específico que pode ser consultado passando a ID do mesmo.

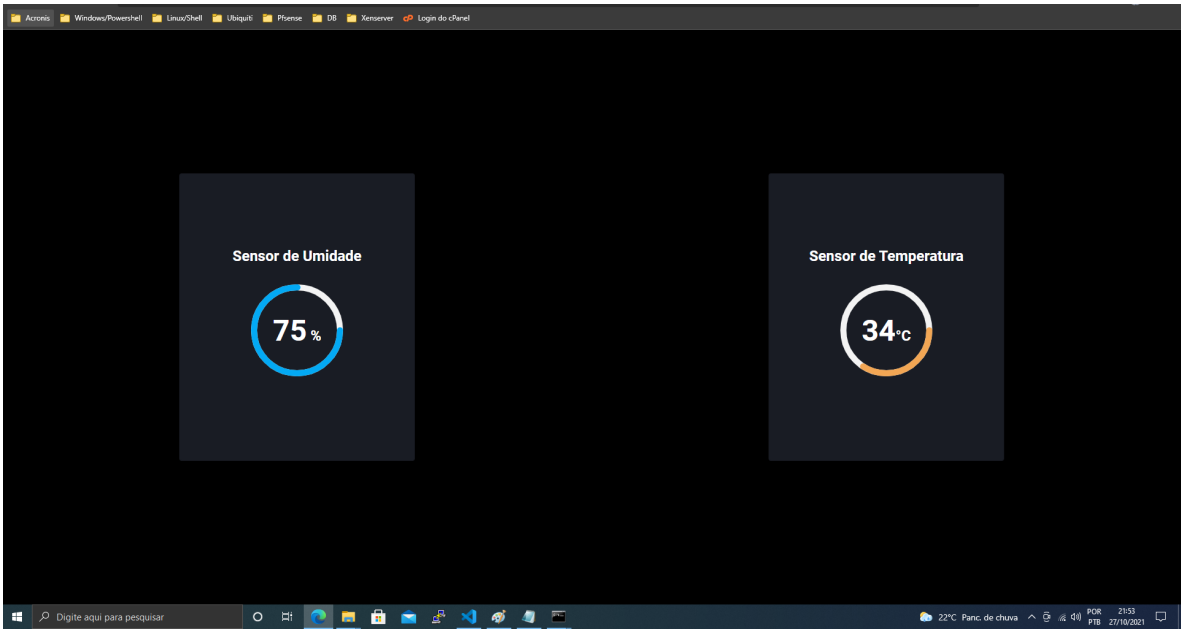
O banco de dados utilizado no projeto foi o PostgreSQL por conta de sua popularidade e confiabilidade no mundo relacional dos bancos de dados, foi utilizado o *querybuilder* Knex.JS para fazer as operações que envolvem o banco de dados.



Já o *frontend* foi construído utilizando o *framework ReactJS* que por sua alta aderência e utilização no mercado o torna um *framework* muito confiável, além de sua documentação ser muito completa e tranquila de se utilizar.

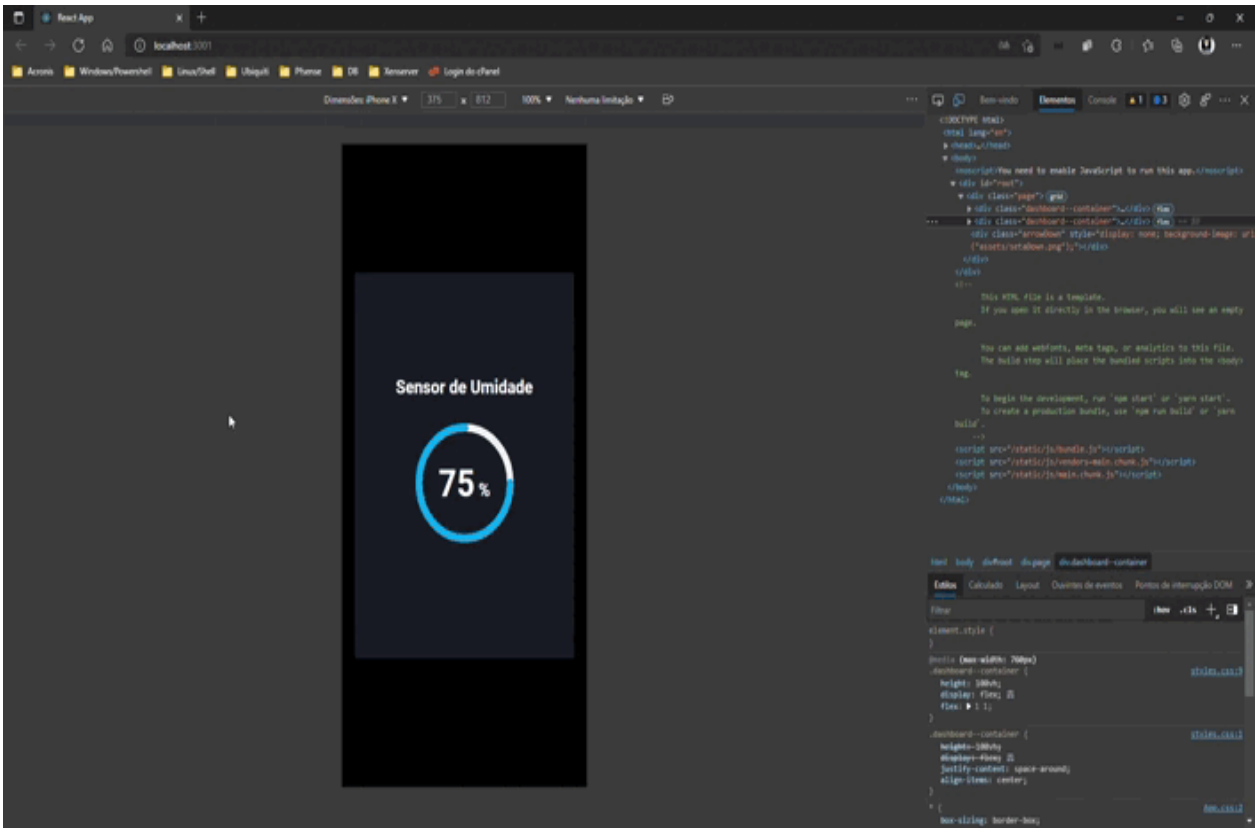
Foi desenvolvido uma *Single Page Application (SPA)* onde possui 2 *cards* com indicadores de temperatura e umidade do solo, sendo atualizada a cada 15 segundos, foi desenvolvido utilizando CSS3, para construir a base do *layout* foi utilizado *grid* e *flex-box*, facilitando assim a configuração do *design* responsivo, onde pode se acessar pelo celular sem que o *layout* “quebre”. As Figuras 2 e 3, são imagens do *frontend* da aplicação no *desktop* e no celular respectivamente.

Figura 2 - Imagem do *Desktop*



Fonte: o autor, 2021

Figura 3 - Imagem do *Mobile*/Celular



Fonte: o autor, 2021

4.1 DESENVOLVIMENTO *HARDWARE*

Sobre a parte eletrônica do projeto, antes de mostrar o fluxograma do projeto, detalha-se sobre alguns componentes que foram utilizados, além do tipo de ligação escolhida.

As baterias escolhidas foram as baterias de *lithium-ion (LI-ION)* 18650 por conta de terem uma tensão maior do que as pilhas comuns e serem muito utilizadas na eletrônica para criar *pack* de baterias onde seja necessário fornecer energia para algo, além da tecnologia utilizada que seria a *lithium* ser a mais indicada no caso de baterias recarregáveis, pois duram mais.

A placa solar como o nome diz é uma placa que gera energia através da luz do sol, sendo uma energia limpa ou seja não causa danos ao meio ambiente, a placa solar escolhida para o projeto foi a de 6 Volts, porém como usando apenas 6

Volts no auge do sol seria muito pouco para carregar 3 baterias de 4.2V, então optou-se por utilizar 2 placas ligadas em série somando assim 12V quando o sol estiver bem intenso.

O Controlador de Tensão, como o nome sugere, é um componente que recebe uma tensão na entrada com a possibilidade de ser ajustado de acordo com a necessidade de tensão na saída, nesse projeto, se utiliza 2 , um para controlar a tensão da placa solar e outro para controlar a alimentação do ESP32.

Battery Management System ou BMS é um componente que possui um sistema de controle e proteção das baterias recarregáveis, fazendo com que elas não sobrecarreguem e possam até explodir ou pegar fogo nesses caso ou descarregue demais, fazendo com que elas não possam mais serem utilizadas.

Já o Relé é um interruptor eletromecânico, que funciona exatamente como um interruptor de luz, porém é acionado por energia elétrica, muito utilizado para controlar quando equipamentos devem ser ligados ou desligados.

A Válvula Solenóide é uma válvula que é ativada por energia elétrica, parecido com o relé, só que neste caso é para controlar o fluxo de água ou seja controlar quando deve deixar a água passar e quando não deve.

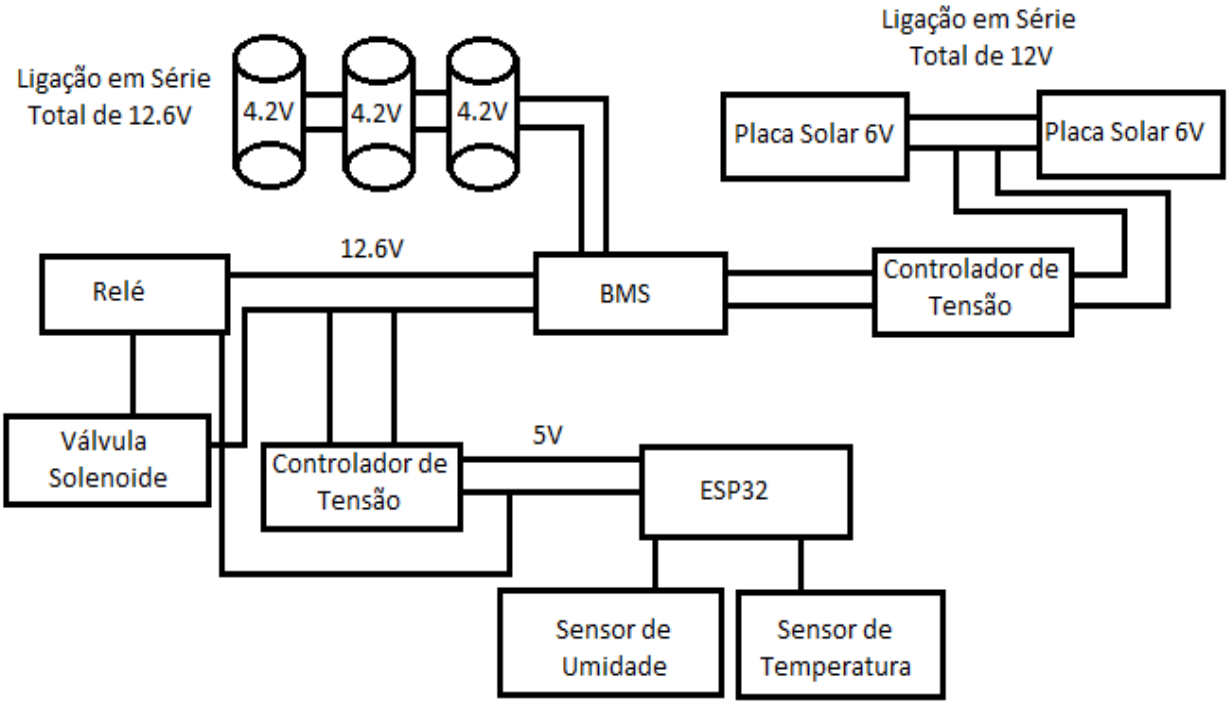
O ESP32 é o microcontrolador utilizado para controlar e enviar os dados dos sensores, foi escolhido por conta de ter um número maior de portas analógicas.

Os sensores usados para este projeto são de dois tipos, o sensor de umidade do solo e o sensor de temperatura.

Na elétrica/eletrônica existem dois tipos de ligação, em série e em paralelo, a ligação em série soma a tensão dos componentes ou seja se duas pilhas de 1.5V forem ligadas utilizando esse tipo no final terei 3V, já em paralelo o que soma é a corrente ou seja usando o mesmo exemplo de duas pilhas de 1.5V cada pilha possuindo 500mAh no final eu terei 1.5V porém com 1000mAh.

Na Figura 4, é possível observar o fluxograma do protótipo.

Figura 4 - Fluxograma do Protótipo



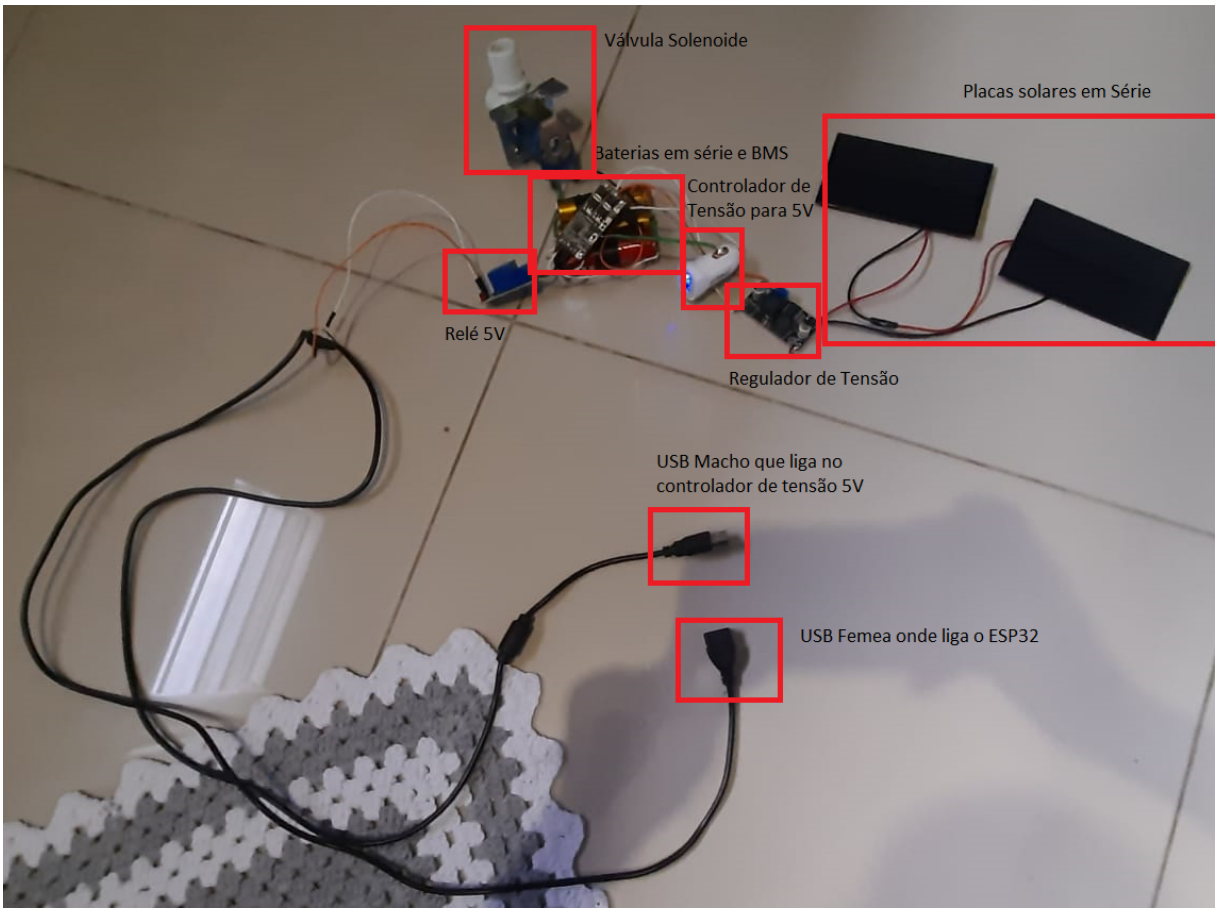
Fonte: o autor, 2021

Para o protótipo foram utilizadas três baterias 18650 cada uma com 4,2V ligadas em série fornecendo um total de 12,6V, optou-se por essa ligação com o objetivo de carregar as baterias mais rapidamente com as placas solares que também foram ligadas 2 placas de 6V em série, ou seja, irá carregar usando 12V quando o sol estiver bem potente e, também para que possa usar a saída das baterias para alimentar a válvula solenoide sem ter problemas de acabar queimando.

As placas solares são ligadas em um controlador de tensão que permite passar apenas 12,6V onde é conectado no BMS e do BMS as baterias para carregar, quando tem sol o circuito utiliza a energia solar e quando não tem utiliza a energia das baterias, saindo do BMS é ligado o relé e a válvula solenoide de 12V e dessa saída é ligado outro controlador de tensão que regula para 5V para que o ESP32 possa usar, essa saída de 5V é ligado também o relé que é alimentado com 5V.

Abaixo, na Figura 5, pode-se observar a imagem do protótipo concluído.

Figura 5 - Protótipo



Fonte: o autor, 2021

Na Tabela 1, é possível observar os componentes utilizados na montagem do protótipo, bem como seus valores.

Tabela 1 - Componentes utilizados

Componente	Valor
40 Fios Fêmea-macho Protoboard	14,90 R$
BMS para baterias de Lítio 18650 4s 30a	25,00 R$
Microcontrolador ESP32	57,56 R$
Protoboard 400 Pontos	15,90 R$
3 Baterias Recarregável 18650 4800mah 4.2V	54,00 R$

2 Painéis Solares 6v 1w 200ma	53,90 R$
40 Fios Fêmea-Fêmea Protoboard	14,90 R$
3 Conversores DC-DC Step Up E Down Xl6019	68,99 R$
Módulo relé 5V	13,90 R$
Válvula Solenoide 12V	34,90 R$
Frete	47,17 R$
Sensor de Umidade do Solo	9,00 R$
Sensor de Temperatura	15,00 R$
Total	425,12 R$

Fonte: o autor, 2021

CONCLUSÃO

Para concluir este projeto foi necessário muito estudo principalmente na área da eletrônica e do mundo *maker,* que são as áreas que não possui-se muito contato durante os estudos, foi um grande aprendizado, principalmente na parte eletrônica, com este projeto foi possível fazer uma rega inteligente e totalmente integrado e praticamente em tempo real, onde o agricultor pode verificar como está o solo de suas plantações e a temperatura no campo.

Sobre os custos, foi um custo muito alto, porém a justificativa desse custo é por conta de um dos controladores de tensão ter vindo com problemas, sendo necessário adquirir outro, a grande maioria dos componentes foram adquiridos nacionalmente por conta da entrega ser mais rápida, ou seja comprando em sites chineses como Aliexpress ou Banggood acaba sendo mais barato, muitos dos componente são para montar em bancada, como os *jumpers* e *protoboard* o que não é necessário para produzir em escala quando já estiver homologado, as baterias 18650 foram compradas, porém é muito fácil de reciclar as mesmas em descartes de notebooks, retirando as baterias de notebooks com defeito.

O custo mais alto será apenas em um dos equipamentos, pois irá precisar apenas de um sensor de umidade para alguns metros de terra e uma válvula solenóide para liberar a água para toda ou grande parte da plantação, então

analisando tudo isso, dá pra economizar uma boa quantia e o retorno de investimento acaba se pagando muito rapidamente com o preço que o agricultor irá economizar no desperdício de água e mão de obra.

Durante o projeto houve alguns contratempos para efetuar os testes como o tempo não ajudar, pois acabou chovendo durante muitas semanas, impossibilitando o andamento do projeto, pois para seguir a construção do protótipo seria necessário testar o BMS e a carga das baterias, com isso acabou fazendo com que atrasasse a conclusão do mesmo, outro contratempo foi por conta do desconhecimento na área da eletrônica sendo necessário muita literatura e buscas na internet para conseguir compreender e dar seguimento no protótipo muitas das vezes.

Duas das baterias acabaram entrando em curto na hora de montar o *pack* de baterias por conta de ter feito a ligação errada, justamente por conta de não ser tão habituado a área da eletrônica, onde foi necessário a troca das baterias.

Com este projeto espera-se que consiga fazer com que agricultores de pequeno e médio porte possam melhorar sua produção e com isso fomentar o mercado da agroindústria gerando uma quantidade e qualidade de alimentos maiores, facilitando o monitoramento de suas plantações com o mesmo podendo verificar pelo celular como estão suas plantações.

REFERÊNCIAS

BASNAYAKA. **INTRODUCTION TO COMPUTER SOFTWARE**. Disponível em: https://www.academia.edu/24804336/INTRODUCTION_TO_COMPUTER_SOFTWARE. Acesso em: 14 jun. 2021.

FELEMA, João; RAIHER, Augusta Pelinski; FERREIRA, Carlos Roberto. Agropecuária brasileira: desempenho regional e determinantes de produtividade. **Revista de Economia e Sociologia Rural**, [S.L.], v. 51, n. 3, p. 555-573, set. 2013. FapUNIFESP (SciELO). http://dx.doi.org/10.1590/s0103-20032013000300008.

HERNANDES, Sediane Carmem Lunardi. **Conceitos de Hardware e Software**. Disponível em: http://paginapessoal.utfpr.edu.br/sedianec/disciplina-de-organizacao-e-arquitetura-de-computadores/arquivos/Aula%2001b%20-%20Conceitos%20de%20Hardware%20e%20Software.pdf. Acesso em: 14 jun. 2021.

LINS, Gabriel de Souza. **UTILIZANDO REACTJS PARA O DESENVOLVIMENTO DE UM SISTEMA DE ALOCAÇÃO E RESERVA DE SALAS NO CAMPUS DA UFC EM QUIXADÁ**. 2019. 36 f. TCC (Graduação) - Curso de Curso de Graduação em Sistemas de Informação, Universidade Federal do Ceará, Quixadá, 2019. Disponível em: http://www.repositorio.ufc.br/bitstream/riufc/49762/1/2019_tcc_gdeslins.pdf. Acesso em: 14 jun. 2021.

NERYS, José Wilson Lima. **MICROPROCESSADORES E MICROCONTROLADORES**. Disponível em: http://inf.ufes.br/~zegonc/material/Introducao_a_Computacao/Microprocessadores%20-%20Parte%201.pdf. Acesso em: 14 jun. 2021.

OLIVEIRA, C. A gestão arquivística de documentos como apoio à publicação de dados governamentais abertos. Acervo - Revista do Arquivo Nacional, v. 29, n. 2, p. 168-178, 2016. Disponível em: http://hdl.handle.net/20.500.11959/brapci/45232. Acesso em: 14 jun. 2021.

PEREIRA, Caio Ribeiro. O problema das arquiteturas bloqueantes. In: PEREIRA, Caio Ribeiro. **Aplicações web real-time com Node.js**. São Paulo: Casa do Código, 2014. p. 10-12. Disponível em: https://play.google.com/books/reader?id=Wm-CCwAAQBAJ&hl=pt&pg=GBS.PT9. Acesso em: 14 jun. 2021.

SOUTO, Mario. **O que é front-end e back-end?** 2019. Disponível em: https://www.alura.com.br/artigos/o-que-e-front-end-e-back-end. Acesso em: 14 jun. 2021.

TANENBAUM, Andrew Stuart. Servidores. In: TANENBAUM, Andrew Stuart. **Organização Estruturada de Computadores**. 6. ed. São Paulo: Pearson Education do Brasil Ltda, 2013. p. 29.