UNIV

UNIVERSIDADE POSITIVO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

CAMILA SILVA RODY

CAROLINA MAYUME YASUE

MATHEUS LOURENÇO STELLA

E-CO Vaso Inteligente

CURITIBA

2020

UNIVERSIDADE POSITIVO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO
CAMILA SILVA RODY
CAROLINA MAYUME YASUE
MATHEUS LOURENÇO STELLA

E-CO Vaso Inteligente

Trabalho apresentado como requisito parcial na disciplina de Conclusão de Curso da Universidade Positivo para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: MSc. Leandro Vasconcelos dos Reis

CURITIBA
2020

RESUMO

Desde o advento da primeira revolução industrial, a influência tecnológica refinou as técnicas de plantio provendo mais recurso para os meios urbanos. Porém, para realizar tais processos, muitas técnicas acabam por serem insustentáveis, seja pela acessibilidade, custo ou por afetar até a própria flora. Com ciência dos argumentos prévios, este documento traz uma abordagem para solucionar tais impasses, unindo a tecnologia e ecologia de forma sustentável e acessível por meio de um vaso expansível. O vaso inteligente tem como objetivo automatizar a irrigação e possibilitar pessoas no meio urbano a plantar com qualidade e adquirir alimentos orgânicos sem agredir a natureza, além de economizar e tornar o ato de plantar acessível. A intenção é demonstrar todo o processo da criação de um produto que comporte um ou mais vasos de plantio, tornando-o inteligente com conhecimentos de *hardware* e *software*, e garantir o cuidado básico sobre o cultivo desde o plantio até a colheita das hortaliças e dos temperos. Este processo será realizado através de um sistema de monitoramento automático, capturando dados sobre a saúde da(s) planta(s), como a necessidade de irrigação e coleta de dados em tempo real de umidade do solo e temperatura do ambiente. Com a irrigação automatizada não há mais a preocupação com o cuidado da(s) planta(s), encaixando-se bem no contexto urbano no qual o tempo é escasso. Construído com componentes de *hardware* baratos e fáceis de repor, o vaso inteligente é de fácil manutenção para que a solução satisfaça a automação como um todo sem dispensar a interação humana com o ato de plantio. O produto une a harmonia entre o uso intuitivo do produto através da interface do *software* com o usuário que acessa suas próprias plantas. A interface exibe de maneira trabalhada os dados para o usuário tomar decisões com base nos mesmos para satisfazer a rotina de cuidado que se queira obter, caso contrário, por ser uma automação, o vaso se responsabiliza pela decisão e faz o cuidado básico diário. Assim, na aplicação *web*, é possível ter a visualização dos dados de saúde da planta por meio de gráficos e valores de umidade, temperatura e quantidade de regas que são obtidos e gravados em tempo real, e observar o estado do(s) vegetal(ais) através da captura de imagens. Com todos os fatores citados, o projeto E-CO Vaso Inteligente tem como sua principal proposta incentivar o plantio

caseiro, buscando automatizar o processo, e integrar o plantio com a tecnologia já utilizada pelo consumidor e ampliar o alcance ao acesso e consumo à alimentos orgânicos para a sociedade.

Palavras-chave: *IoT*, Plantas, Sustentabilidade, Controle, Irrigação.

ABSTRACT

Since the advent of the first industrial revolution, technological influence has refined planting techniques by providing more resources for urban areas. However, to carry out such processes, many techniques turn out to be unsustainable, whether due to accessibility, cost or even affecting the flora itself. Aware of the previous arguments, this document brings an approach to solve such impasses, uniting technology and ecology in a sustainable and accessible way through an expandable vessel. The smart pot aims to automate irrigation and enable people in the urban environment to plant with quality and purchase organic food without harming nature, in addition to saving and making the act of planting accessible. The intention is to demonstrate the entire process of creating a product that includes one or more planting pots, making it intelligent with knowledge of hardware and software, and ensuring basic care about cultivation from planting to the harvest of vegetables and plants. spices. This process will be carried out through an automatic monitoring system, capturing data on the health of the plant (s), such as the need for irrigation and data collection in real time of soil moisture and ambient temperature. With automated irrigation there is no longer a concern with the care of the plant (s), fitting well in the urban context in which time is scarce. Built with inexpensive and easy to replace hardware components, the smart pot is easy to maintain so that the solution satisfies automation as a whole without dispensing with human interaction with the act of planting. The product unites the harmony between the intuitive use of the product through the software interface with the user who accesses his own plants. The interface displays the data in an elaborate way for the user to make decisions based on it to satisfy the care routine that one wants to obtain, otherwise, because it is an automation, the vessel is responsible for the decision and performs basic daily care. Thus, in the web application, it is possible to view the health data of the plant by means of graphs and values of humidity, temperature and amount of watering that are obtained and recorded in real time, and observe the state of the plant (s) (a) by capturing images. With all the factors mentioned, the E-CO Vaso Inteligente project has as its main proposal to encourage home planting, seeking to automate the

process, and integrate planting with the technology already used by the consumer and expand the reach and access to organic food for the society.

Keywords: *IoT, Plants, Sustainability, Control, Irrigation.*

LISTA DE FIGURAS

LISTA DE QUADROS

SUMÁRIO

1. INTRODUÇÃO

De acordo com dados do IBGE (2017), houve, com o passar dos anos, um aumento do número de produção orgânica no Brasil. Isso mostra que existe uma tendência crescente no interesse e procura por produtos orgânicos. Parte do crescimento da tendência deve-se ao fato de que a sociedade passou por uma fase de extrema industrialização, desenvolvendo uma necessidade de retorno à origem e simplicidade, com a maior apreciação da origem dos alimentos, de acordo com o SEBRAE (2018).

Ainda neste ponto, várias organizações, como a ONU (2020), enfatizam a preocupação global e enfática com a sustentabilidade e preservação do meio ambiente, abordando questões como a de uma alimentação mais saudável.

Segundo o Harvard Law Review (2018), há um aumento no uso da tecnologia como hábito em tarefas auxiliares na rotina do indivíduo.

De acordo com a tabela de renda média disponibilizada pelo IBGE (2020), é possível concluir que a renda média salarial do brasileiro no primeiro trimestre é de R$2.323,00 reais, isso relacionado à classe trabalhadora regular. Porém a Veja Investe (2020) mostra que atualmente o salário mínimo em 2020 (dois mil e vinte), será em torno de R$1.045,00 reais. Levando em conta esses fatores e observando essa característica, produtos orgânicos mostram-se pouco acessíveis à renda dos brasileiros de carteira assinada, sem contar aqueles que não são regularizados e recebem ainda menos. Além disso, a baixa produção de produtos orgânicos para o mercado acarreta no alto custo e o consequente desconhecimento do consumidor sobre o produto, e as características do produtor favorecem também esse encarecimento e a limitação na produção, segundo J.P. Santiago (2020). A questão de espaço físico no meio urbano e escassez de tempo livre também afetam o trabalhador moderno, impossibilitando em sua maioria o cultivo orgânico *indoor* eficiente.

A iniciativa do projeto E-CO Vaso Inteligente propõe um cultivo *indoor*, que tenha um custo baixo e que supra as necessidades básicas do consumidor. Assim, suas principais funções são a irrigação e o monitoramento. A irrigação automatizada

retira uma preocupação com o cuidado da planta, encaixando-se no contexto onde o tempo é escasso, além de trazer acessibilidade ao público com problemas de coordenação motora. O monitoramento agrega os dados capturados acerca da saúde da planta, como por exemplo, a umidade e histórico de rega. Os dados podem ser acessados remotamente, mantendo o sistema sempre disponível para a consulta de informações.

1.1 Justificativa

O projeto que toma inspiração de iniciativas como o Farmbot (2015) e tutoriais encontrados na *web* sobre o assunto, surgiu através de um trabalho acadêmico realizado pelo grupo dos autores, no ano letivo de 2019. O projeto levou em consideração os 17 Objetivos de Desenvolvimento Sustentável *(17 ODS)*, que são uma série de metas apresentadas pela ONU (2015), e que tem o intuito de aliviar diversos problemas relacionados com o desenvolvimento sustentável até o ano de 2030. Assim, dentro dessa série de metas, o projeto engloba os 5 objetivos:

- 2 Fome Zero e Agricultura Sustentável *(alcançar a segurança alimentar e melhoria da nutrição desenvolvendo uma agricultura sustentável*).
- 11 Cidades e comunidades sustentáveis (*tornar as cidades e os assentamentos humanos inclusivos, seguros, resilientes e sustentáveis*),
- 12 Consumo e produção responsáveis (*assegurar padrões de produção e de consumo sustentáveis*).
- 15 Vida terrestre (proteger, recuperar e promover o uso sustentável dos ecossistemas terrestres, gerir de forma sustentável as florestas, *combater a desertificação, deter e reverter a degradação da terra e deter a perda de biodiversidade*).
- 17 Parcerias e meios de implementação (*Fortalecer os meios de implementação e revitalizar a parceria global para o desenvolvimento sustentável*).

A partir daí, levando em consideração a mudança de hábitos da população mundial, em aspectos como a busca por uma alimentação mais saudável, junto com a crescente preocupação com fatores de conservação ambiental, isso unido a dificuldades da vida moderna (espaço, tempo, custo, etc) e uma lacuna presente no mercado nacional por um produto semelhante, foi percebida uma oportunidade de continuação do projeto. A partir da crescente mudança de hábitos das pessoas ao redor do mundo, as mesmas buscam por uma alimentação saudável e com melhor custo benefício. O principal desafio do projeto é proporcionar hortaliças/temperos saudáveis, frescas, mais baratas e sem agrotóxicos.

A proposta se atrela a ideia de fornecer acessibilidade de diversas formas, uma delas através dos componentes comuns e acessíveis, com estrutura simples e foco em cultivo *indoor*, sem alterar a qualidade de funcionamento do produto. Outra, disponibilizando uma área administrativa intuitiva e que aproxima o processo ao usuário.

Dessa forma, o produto em questão consegue proporcionar segurança, acessibilidade e qualidade à produção caseira pois as hortaliças e temperos serão livres de agrotóxicos.

A acessibilidade entra também de forma a considerar quem não possui coordenação motora, geralmente idosos ou pessoas portadoras transtornos neurológicos ou distúrbios do sistema nervoso que afetem o sistema motor. Ela confere praticidade em situações difíceis como em longos períodos de distanciamento, e na dificuldade de encontrar uma solução para manter plantas saudáveis.

1.2 Objetivos

1.2.1 Objetivo Geral

Desenvolver uma aplicação *web* automatizada integrada com componentes de *hardware* para auxiliar no cultivo de plantas *indoor.*

1.2.2 Objetivos Específicos

Com o intuito de atingir o objetivo principal, alguns objetivos específicos são necessários.

- Desenvolver um projeto que possibilite o controle sobre o vaso de forma fácil através da interface física e digital, através da aplicação *web*.
- Desenvolver uma integração entre o vaso e a plataforma *web* onde o usuário tenha acesso e consiga modificar os parâmetros.
- Desenvolver atribuições do microcontrolador para que o mesmo fique responsável pela irrigação de forma controlável.
- Realizar uma análise para registrar a adaptação do projeto às plantas.

2. PLANEJAMENTO

O projeto tem como proposta incentivar o plantio *indoor* e busca automatizar o processo ao integrar o plantio através de tecnologias comuns na sociedade. Porém, para se atingir o resultado esperado, alguns processos devem ser seguidos para que o desenvolvimento ocorra sem adversidades.

Através do *hardware* e seus componentes será realizado todo o sistema de irrigação. A parte lógica será desenvolvida a partir da codificação do conjunto de processos que serão necessários para o projeto operar. Já a parte física será disposta da junção dos componentes que se atrelam de forma orgânica.

2.1 Definição do escopo

Neste tópico será definido o escopo do projeto, desde requisitos funcionais até os não funcionais. No Quadro 1 e no Quadro 2 é apresentado uma disposição dos mesmos definidos e uma breve descrição de cada.

Quadro 1 - Requisitos Funcionais

Requisitos Funcionais		
Número	**Requisitos**	**Descrição**
RF01	O vaso deve captar as condições da planta	O vaso deve ser capaz de identificar se a planta está precisando de água ou não por meio do *hardware* e *software*.

RF02	O vaso deve regar a planta se for necessário	O vaso deve regar a planta se o solo estiver seco ou se o usuário agendar no sistema um horário fixo para a rega.
RF02	O *software* deve permitir criar visualizações para as hortaliças/temperos	Permite ao usuário criar os perfis dos vasos, configurar e alterar certos dados de acordo com a necessidade de cada planta.
RF03	Visualizar informações	Permite ao usuário visualizar informações do vaso.
RF05	Visualização gráfica	Concebe a apresentação gráfica dos dados do sensor de umidade.
RF06	Alertas	Envia um alerta ao usuário, informando sobre o que está acontecendo na automação, seja desde a falta de água até umidade baixa.

Fonte: AUTORES.

Quadro 2 - Requisitos Não Funcionais

Requisitos Não Funcionais		
Número	**Requisitos**	**Descrição**
RNF01	*Backend* da aplicação em Laravel.	O *backend* não será realizado em outra linguagem.
RNF02	As plataformas deverão ser intuitivas.	As plataformas deverão ser de fácil usabilidade.
RNF03	Disponibilidade 24/7 do sistema.	O sistema estará disponível 24 horas por dia 7 dias por semana.
RNF04	A forma de acesso dos apps será via *web.*	O *software* apenas será acessado via *web*, pelo site oficial.
RNF05	Plataforma alvo será *web.*	O sistema deverá ser disponível na plataforma *web.*
RNF06	Suporte ao Usuário via *web.*	O suporte será realizado a partir do *site web.*

Fonte: AUTORES.

2.2 Cronograma de atividades

De acordo com o que foi passado nas normativas da disciplina para atingir a meta do TCCI, foi necessário a criação de um cronograma onde foi estipulado todas as etapas, este pode ser visualizado no Quadro 3.

Quadro 3 - Cronograma

	2020										
Atividades	**F**	**M**	**A**	**M**	**J**	**J**	**A**	**S**	**O**	**N**	**D**
Definição e aprovação do tema de TCC	✓	✓									
Desenvolvimento da proposta e planejamento	✓	✓	✓								
Revisão bibliográfica e análise		✓	✓	✓	✓						
Especificação técnica e modelagem			✓	✓	✓						
Desenvolvimento do projeto	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	
Hardware Parte 1	✓	✓	✓								
Software Parte 1		✓	✓	✓	✓						
Teste Parte 1					✓						
Entrega do documento de TCC para qualificação					✓						

Apresentação do projeto para banca de qualificação					✓						
Hardware Parte 2						✓	✓	✓			
Software Parte 2						✓	✓	✓			
Teste Parte 2							✓	✓	✓	✓	
Entrega do documento de TCC para banca final										✓	
Banca final										✓	
Entrega do documento de TCC corrigido											✓

Fonte: AUTORES.

2.3 Orçamento

Uma breve lista é apresentada na Quadro 4 sobre os materiais usados na construção do projeto. Alguns materiais podem ser adaptados ou substituídos por alternativos.

Quadro 4 - Materiais

Item	Descrição	Qtd.	Unid.	Preço Unit.	Preço Total
NodeMCU ESP8266	Placa com microcontrolador	1	Um	R$ 49,90	R$ 49,90

Raspberry PI ZeroW	Placa com microcontrolador	1	Um	R$ 179,90	R$ 179,90
Câmera para Raspberry PI	Componente câmera digital para microcontrolador	1	Um	R$ 200,00	R$ 200,00
Válvula solenóide	Componente Válvula Solenóide	1	Um	R$ 32,90	R$ 32,90
Bomba d'água submersível 3v-6v	Componente bomba d'água para microcontrolador	1	Um	R$ 18,90	R$ 18,90
Transistores	Transistores tipo BD137	2	Dois	R$ 0,20	R$ 0,40
Diodo	Diodo Zener, regulador de tensão	2	Dois	R$ 0,20	R$ 0,40
Jumpers	Cabos de cobre.	1	Um	R$ 26,90	R$ 26,90
Placa de prototipação 830 pontos	Placa de prototipação.	1	Um	R$ 18,90	R$ 18,90
Sensor de umidade	Componente sensor de umidade do solo para microcontrolador	2	Dois	R$ 10,90	R$ 21,80
Hub fonte ajustável	Fonte ajustável 3v-5v	1	Um	R$ 5,00	R$ 5,00
TOTAL					**R$ 494,80**

Fonte: AUTORES.

3. DESENVOLVIMENTO

O desenvolvimento do vaso inteligente demanda por conhecimentos básicos sobre plantio, sustentabilidade, física elétrica, componentes elétricos, *hardware* e *software*. Este capítulo abordará a composição, orientação e esquemáticas, a elaboração e construção do *hardware* (microcontroladores e componentes) e da aplicação *web* que compõe o projeto proposto.

Baseado "faça você mesmo", do inglês "*do it yourself*" (DIY,2020), o projeto vaso inteligente E-CO, é fundado sobre uma estrutura de *hardware* que orquestra todo o funcionamento da irrigação. Possuindo componentes como bomba d'água e sensores de umidade é possível criar um sistema cujo qual pode-se obter dados de umidade e fazer irrigações.

Através do *software* é obtido as informações necessárias para fazer as tomadas de decisão sobre a irrigação com base na umidade do solo, satisfazendo o cuidado básico de hortaliças ou plantas. Os dados são obtidos pelo *software backend* que consome as funções básicas criadas no *software* do *hardware* e, assim, transferidos para uma interface *web* qual o usuário possa monitorar por meio de métricas, histórico e imagens, dados estes previamente manipulados no *backend* para criar uma consistência sobre o processo de irrigação, fácil leitura e identificação de ações.

As métricas são baseadas na umidade, tempo e vezes de rega. O histórico é gerado a cada ação reproduzida pelo *hardware*. Uma vez programado, o mesmo possui ações básicas que serão acionadas pelo backend. Já as imagens, são capturadas por outro tipo de *hardware*, de forma desacoplada, garantindo assim, a visualização de como a planta está fisicamente. O uso da câmera é baseado em uma das funções do projeto Farmbot (2015) que, consiste em desde o plantio até a colheita cuidar e monitorar também com o auxílio de câmeras.

Assim, E-CO Vaso Inteligente é capaz de realizar toda a rotina de irrigação e monitoramento considerando manutenibilidade por usar *hardware* e componentes de fácil acesso e, a disponibilidade que é garantida através da aplicação *web* desenvolvida.

3.1 Fundamentação teórica

3.1.1 Cultivo

A fundamentação tem o foco no cuidado de hortaliças, temperos e plantas através da utilização de componentes elétricos.

Para a elaboração do projeto proposto, é necessário ter um nível básico de conhecimento sobre plantio, para que a irrigação seja a principal característica a se ater.

Através da empresa de cultivo Faz Verde (2016), que trata da irrigação de plantas, é possível ter noções básicas sobre cultivo como por exemplo, o volume de cada rega em função da variação de temperatura, a frequência a água deve ser aplicada e etc. Analisar e conhecer tais características sobre cultivo básico possibilita criar uma solução que possa automatizar todo o processo de cultivo de forma mais precisa.

A Faz Verde (2016) evidencia que há diversos fatores que devem ser levados em conta para manter a saúde das plantas. Assim, é necessário a observação das condições de temperatura do dia, o conhecimento das características da espécie da planta que está sendo cuidada para evitar que fiquem ora demasiadamente ora precariamente irrigadas. O solo deve estar sempre úmido, mas não encharcado e muito molhado.

Logo, com base nessas informações é possível entender os processos básicos e necessários para manter a planta irrigada e também seu solo úmido.

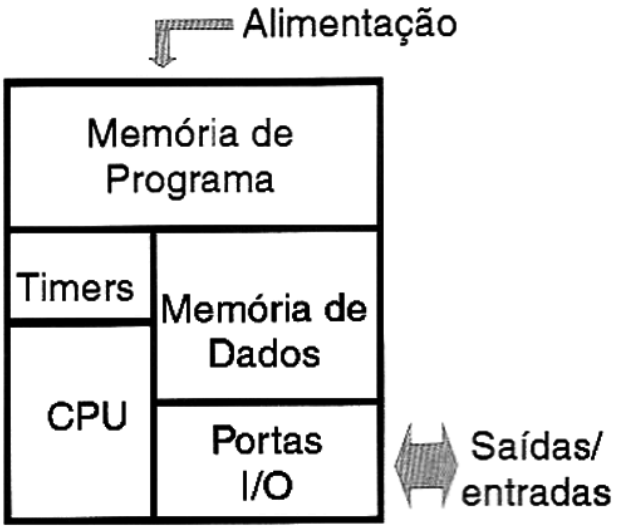
3.1.2 Hardware e componentes

3.1.2.1 Microcontroladores

Um microcontrolador é um computador inserido em um *chip* de circuito integrado. Ele é constituído de memória, processador e interfaces de entrada-saída (i/o) sendo programados para executarem uma tarefa específica.

Por ser formado de um circuito integrado, o mesmo traz a ideia de que é possível reunir num componente um circuito completo que exerça determinada função como representado na Figura 1. (BRAGA, 2020).

Figura 1 - Estrutura do microcontrolador em uma placa de circuito



Fonte: (BRAGA, 2020)

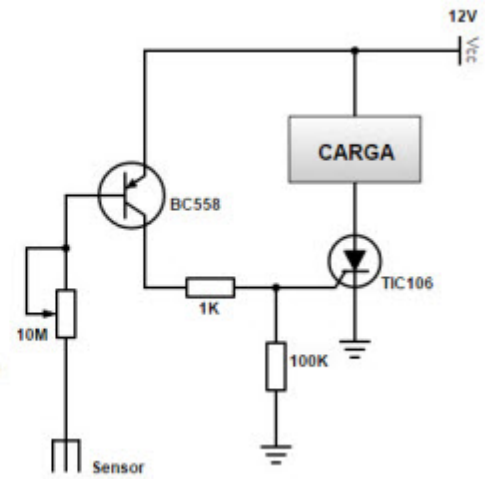
Os microcontroladores são mais usados para compor placas com circuito, exemplo de placas como essas são o Arduino Uno, Raspberry PI, NodeMCU ESP8266 que, são compostos pelas características de serem programados para executar alguma tarefa e por possuir interfaces de entrada-saída, é possível acoplar demais componentes a serem controlados. (BRAGA, 2020)

3.1.2.2 Sensor de umidade

Para satisfazer a irrigação automatizada, é muito importante saber a umidade do solo, pois é através dela que pode se saber se o solo está propício a receber mais, menos ou nenhuma quantidade de água. (SOARES, 2016)

Como citado no tópico que diz respeito à Microcontroladores, é necessário um microcontrolador para controlar e executar tarefas sob respectivos componentes, e é o sensor de umidade que tem a característica de ser um componente capaz de medir a umidade do solo, criando dados mais precisos com a programação. Esse componente mede a umidade relativa da área de contato através da água em contato propagando energia que será detectada pelo microcontrolador. Quanto mais molhado, mais a energia irá se propagar, sinalizando o quão úmido o substrato está. Há vários tipos de sensores de umidade, alguns mais sofisticados em seu material e outros mais simples, como pode-se observar na Figura 2, junto com a esquemática elétrica do sensor. (SOARES, 2016)

Figura 2 - Esquema elétrico do sensor de umidade



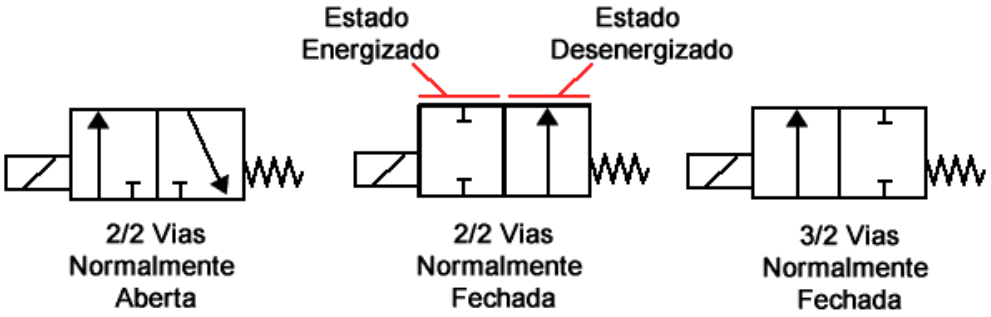
Fonte: (SOARES, 2016)

3.1.2.3 Válvula solenóide

A válvula solenóide é uma válvula eletromecânica controlada através de uma tensão específica do tipo de válvula, seja de 5 (cinco) *volts* ou 12 (doze) *volts*, que são os mais comuns no mercado. A válvula acionada tem o objetivo de controlar o fluxo de gases e líquidos, interrompendo ou liberando o fluxo. (SILVEIRA, 2018)

O componente principal da válvula solenóide é a bobina elétrica, que em posição de repouso tampa o pequeno orifício que o fluxo passa. Quando a válvula é energizada, a corrente elétrica que circula na bobina cria um campo magnético exercendo uma força sobre o êmbolo que é puxado em direção ao centro da bobina fazendo com que um dos orifícios do componente se abra, realizando o princípio básico da válvula que é fechar a abrir o fluxo. (SILVEIRA, 2018)

Figura 3 - Esquema do funcionamento da válvula solenóide



Fonte: (SILVEIRA, 2018)

3.1.2.5 Bomba D'água

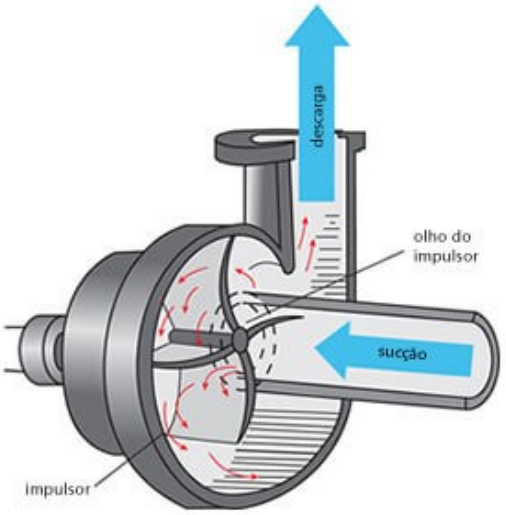
Há diversos tipos de bombas d'água, mas ater-se ao seu princípio básico de funcionamento é especificamente necessário para a elaboração deste projeto.

Tanto as bombas de aquário quanto os demais modelos tem a função de criar pressão quando ligadas e, através da sucção feita, o líquido será puxado circulando através de tubos e conexões, como representado na Figura 4. (AQUÁRIOS SOBRINHOS, 2019)

É com esse princípio básico que será desenvolvido este projeto. Pode-se usar

tanto uma bomba específica para arduino como uma bomba d'água normal, achada em pet-shops e relativos.

Figura 4 - Esquema do funcionamento da bomba d'água



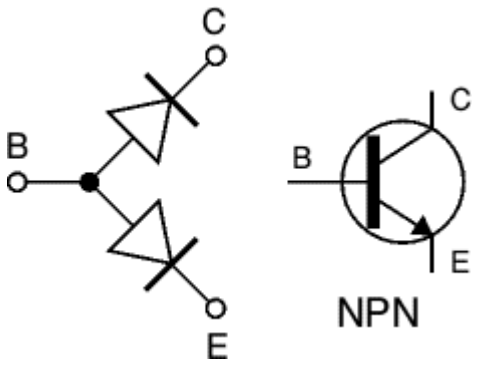
Fonte: (AQUÁRIOS SOBRINHOS, 2019)

3.1.2.6 Transistores

Os transistores são dispositivos semicondutores que ampliam sinais ou podem agir como interruptores. Isto ocorre pelo controle da corrente ou da tensão que é aplicada em um eletrodo. Existem dois tipos de transistores, NPN e PNP, que possuem três eletrodos, conhecidos como B (base), C (coletor) e E (emissor). Esses dois tipos de transistores possuem mais eletrodos que um diodo.

A corrente passa da base para o emissor quando se conecta a base ao polo positivo e o emissor ao polo negativo. Essa corrente é chamada de corrente de base. Quando a corrente passa da base para o emissor, ao mesmo tempo passa corrente do coletor para o emissor. Então a corrente base pode ser usada para ajustar a corrente do coletor. Na Figura 5 demonstra o esquema do transistor. (HEVES, 2020)

Figura 5 - Esquema do transistor



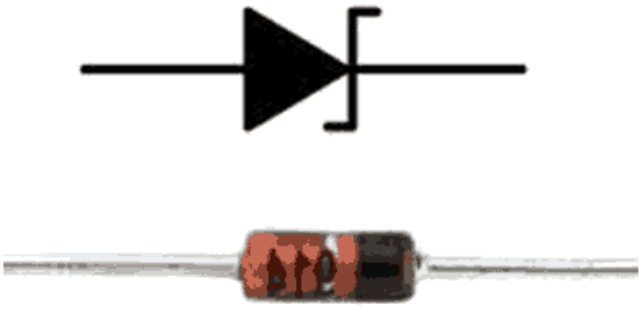
Fonte: HEVES, 2020.

3.1.2.4 Diodo

O diodo é um componente elétrico que permite a passagem de corrente elétrica em apenas um sentido, devido a sua polarização. Pode-se resumir sua funcionalidade à uma válvula, uma vez que há apenas um sentido de passagem.

Os diodos podem exercer diversas funções dependendo do objetivo inserido no esquema elétrico, como função de tensão, regulador de tensão (Zener), emissão de luz (LED) etc. Na Figura 6 é possível observar como é o esquema do diodo Zener. (ATHOS ELECTRONICS, 2020)

Figura 6 - Esquema do diodo Zener



Fonte: ATHOS ELECTRONICS, 2020.

3.2 Material e métodos

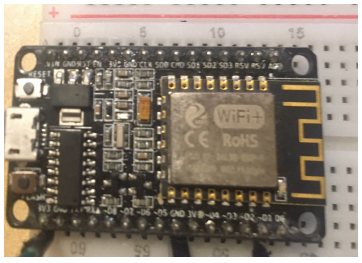
Neste tópico será abordado desde a construção do *hardware* até a solução tratada pelo *software* com ênfase nos materiais, processos e métodos selecionados. Os itens utilizados possuem características específicas pautadas na preferência dos autores deste projeto, considerando o custo benefício de cada elemento e também a acessibilidade dos recursos.

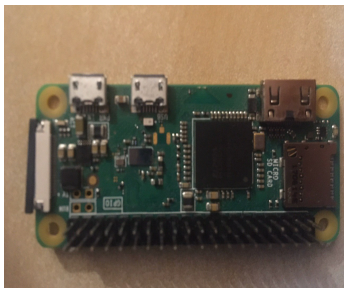
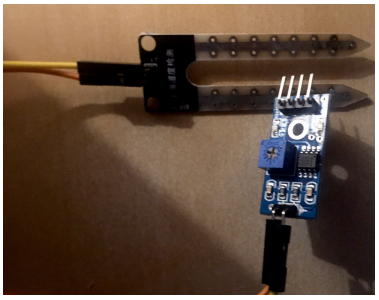
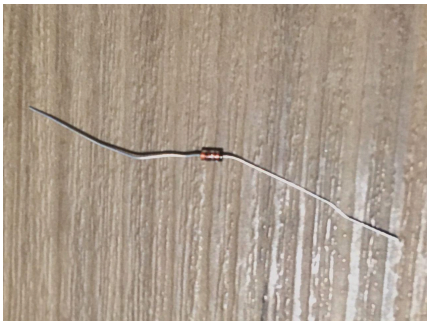
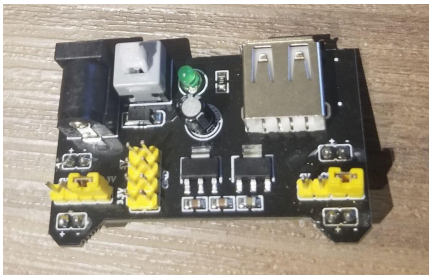
3.2.1 Materiais do hardware

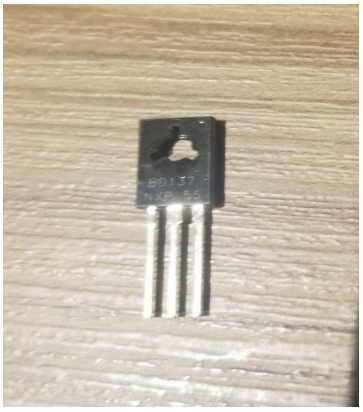
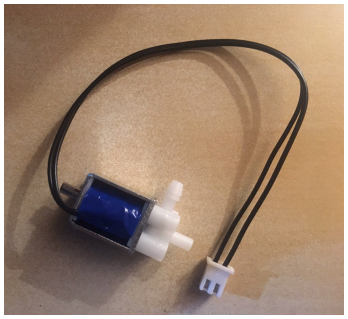
No Quadro 5 é possível visualizar os tipos de componentes e microcontroladores escolhidos para a composição da automação e suas breves descrições das principais funções de cada um sobre o projeto.

Para a composição do *hardware* responsável pela automação, serão utilizados os seguintes componentes e materiais:

Quadro 5 - Componentes

Material/Componente	Descrição/Função
	Microcontrolador placa NodeMCU ESP8266 3v com módulo *Wi-Fi* que irá reproduzir ações através de comandos vindos do *software backend* que passa pelo Raspberry Pi os dados gerados.

	Microcontrolador placa Raspberry Pi Zero W, com pinagem macho e entrada para cabo *falt*. Será utilizado como um HUB ou concentrador das informações capturadas pelo NodeMCU e que também controlará componentes de câmera.
	Sensor de umidade do solo higrômetro que irá ser fincado ao solo para que através do contato com a água irá disparar voltagens que poderão ser convertidas a um outro tipo de dado.
	Diodo Zener para controlar a tensão que irá passar, protegendo o microcontrolador de altas cargas.
	Fonte ajustável 3v-5v para alimentar a placa de prototipação de acordo com a voltagem dos componentes. Alimentação via fonte DC 12v ou USB

	Transistor BD 137 que controla a corrente tanto da válvula solenóide como da bomba d'água, via microcontrolador no emissor.
	Válvula Solenóide 5v 3 vias para abrir e fechar o fluxo de água através da codificação de comandos no microcontrolador.
	Bomba d'água submersível 3v-6v para fazer a pressão da água no reservatório liberada pela válvula solenóide.

Fonte: AUTORES.

3.2.2 Materiais do software

Para a implementação da aplicação *web* do projeto, foi necessária a utilização de linguagens de programação, servidor e banco de dados.

3.2.2.1 Linguagens de programação

I. Laravel

Laravel é um *framework* baseado em PHP e Symphony que é usado para a construção de aplicações *web*. (LARAVEL, 2020)

É com ele que é realizado todo o *backend*, classes e conexão com o banco de dados, além do consumo de bibliotecas para uma maior manipulação. O tipo de desenvolvimento *backend* faz a comunicação entre a base de dados e o navegador. (STEWART, 2019)

II. Vue JS

O Vue JS é um *framework* de JS (javascript) que é capaz de criar maiores interações sobre a UX e a estrutura de HTML, trazendo a reatividade aos componentes visuais ao programar. (VUEJS, 2020)

Essa tecnologia que é aplicada nos componentes visuais constituídos no *software*, além do consumo de bibliotecas para maior manipulação.

III. TailwindCSS

O TailwindCSS é um *framework* utilitário baseado em SASS e CSS3. Com ele, é possível abstrair estruturas e diminuir a preocupação com a questão visual como, *grid*, *flex* e técnicas semelhantes. (TAILWINDCSS, 2020)

É usado para compor partes do projeto referentes ao *design* e à disposição, e trazer a disponibilidade do uso como *app* através de seus recursos.

IV. SASS

O SASS é um *framework* baseado em CSS3 que traz uma convenção para a estrutura do CSS. (SASS, 2020)

É usado neste projeto de forma a agregar características específicas à UI (*user interface*) dos componentes usados no projeto.

V. HTML5

O HTML é uma ferramenta de hipertexto de marcação, a estrutura mais básica da *web* para estruturar o conteúdo de uma página. (MDN, 2019)

Essa ferramenta define a estrutura do site construído.

3.2.2.2 Servidor e banco de dados

I. MariaDB

MariaDB é um banco estruturado baseado em SQL. (MARIADB FOUNDATION, 2020)

Essa tecnologia de banco de dados será usada para guardar e orquestrar todas as informações coletadas da automação pela aplicação.

II. Heroku

O Heroku é uma plataforma em nuvem que oferece o serviço de deploy de aplicações com suporte à várias linguagens de programação. (HEROKU, 2020)

É nele que será hospedado uma parte da aplicação.

III. Web Server

Web server é um conceito de criar um servidor local para acesso aos dados disponibilizados pelo mesmo qual o *software* captura. Dessa forma, é possível fazer sistemas alheios consumirem os dados disponíveis pelo *endpoint.* (MDN, 2019)

Neste projeto, o *web server* é aplicado no microcontrolador através do consumo de bibliotecas que disponibiliza de forma fácil essa função para o código do mesmo.

3.2.3 Construção do hardware

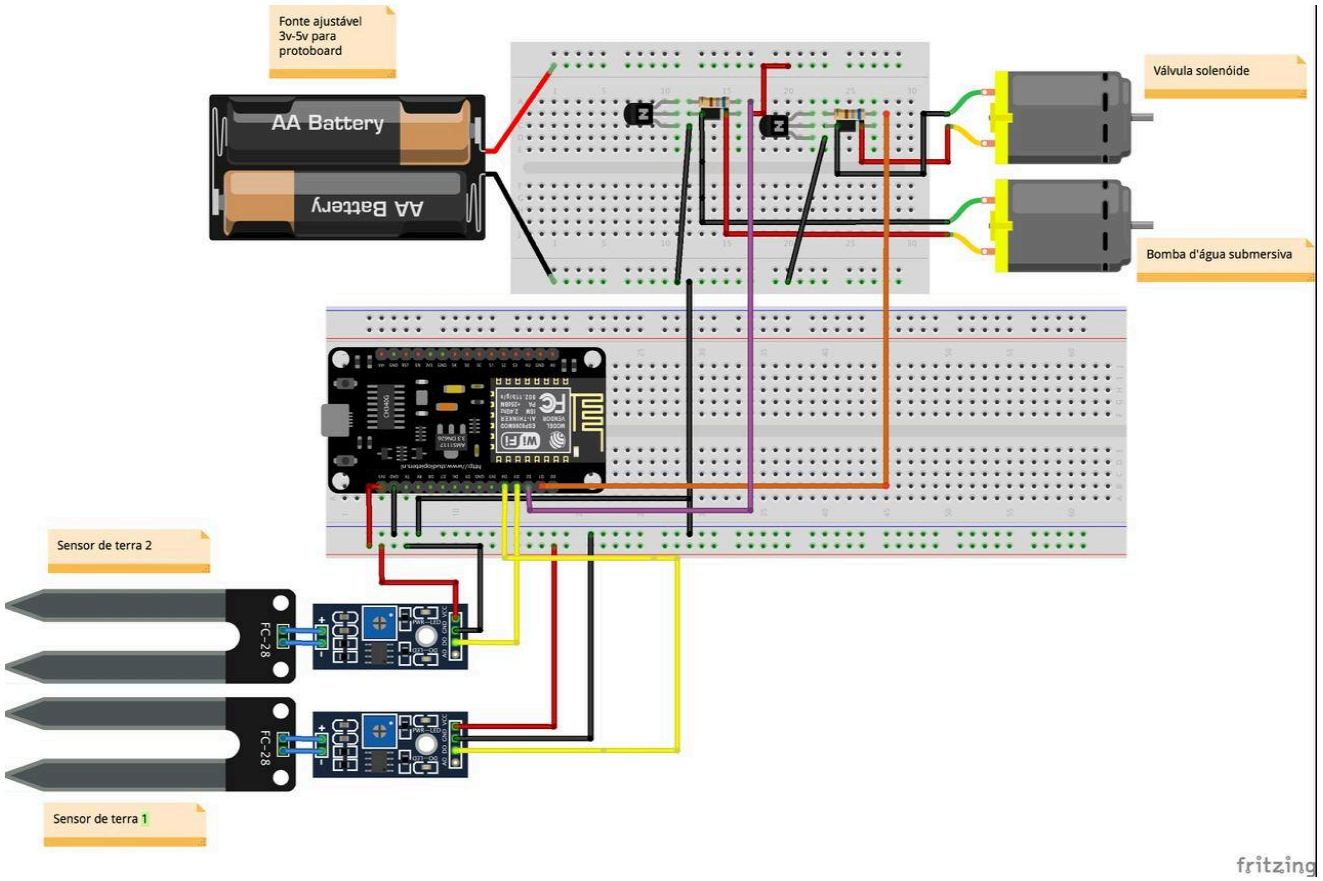
Compreendendo a ideia do projeto, os componentes e sua relação um com o outro, é possível começar a executar a montagem de cada parte do *hardware*.

Para o começo do projeto, construir o *hardware* é a primeira parte para contemplá-lo, pois é o mesmo que irá satisfazer a automação.

Nesta primeira etapa, será usada a placa de prototipação para comportar a placa NodeMCU. Uma vez o NodeMCU conectado, por ser composto de pinos macho, quando ligado na USB, irá energizar tanto a placa e seus pinos como as entradas próximas aos mesmos na placa de prototipação. Além da energia do NodeMCU, também será utilizado a fonte ajustável (3v-5v) que alimenta uma segunda placa de prototipação para energizar os componentes alheios.

Para que os componentes válvula solenóide e bomba d'água possam ser ligados e controlados, é necessário notar que os tipos desses componentes atuam em voltagens um pouco superiores ao que o microcontrolador consegue suportar. Portanto, para ligar ambos e controlá-los dinamicamente serão utilizados um transistor e de um diodo. O transistor manterá o fluxo de energia dentro do circuito, atuando com os pinos base, emissor e coletor. No pino coletor é realizada a conexão com a fonte de energia, e no pino base do transistor é conectado o GND do NodeMCU. Já no emissor é conectado a porta de leitura, onde se pode obter dados através do transistor. No coletor do transistor é conectado o componente (bomba d'água ou válvula solenóide) que está ligado em conjunto a um diodo tipo Zener, que lida com as cargas que passam até o componente. Entre a conexão de dados e o transistor encontra-se um resistor de 510 (quinhentos e dez) *ohms*, mas pode ser também usado o de 500 (quinhentos) *ohms* para evitar sobrecargas na placa. Note essas características na Figura 7 abaixo.

Figura 7 - Circuito completo



Fonte: AUTORES.

Com toda a montagem concluída é possível testar o funcionamento dos componentes interligados.

Ao engatar a válvula solenóide na saída de água continua, vindo da bomba d'água imersa no recipiente, é possível bombear água para as plantas através de uma fina mangueira enquanto o sensor de umidade segue identificando os níveis de umidade da terra.

3.2.4 Construção do software

3.2.4.1 NodeMCU

Com a lógica de programação aplicada no arduino *IDE* é possível observar o comportamento proveniente da construção do *hardware* sobre todo o circuito dos componentes.

Com a linguagem de programação C é possível controlar todos os componentes. Assim, se torna factível a automação do controle de tensão gradual da bomba d'água e da válvula solenóide através dos transistores.

No caso da bomba d'água, a intensidade do motor é aumentada ou diminuída, dando mais pressão ou retendo-a completamente. Já a válvula solenóide também possui o mesmo efeito, liberando ou bloqueando saída de líquidos. Além destes componentes, os dados capturados pelos sensores de umidade são processados pelo código de programação. Todos os comportamentos são abstraídos em uma função tipada em C.

Por meio do *Wi-Fi* que o NodeMCU dispõe, é criado um processo de configuração na rede local. Uma vez conectado, o microcontrolador faz o uso das seguintes bibliotecas para disponibilizar o *web server* na rede:

- ArduinoJson.h: para a criação de arquivo JSON.
- ESP8266WiFi.h: para criar a conexão de *Wi-Fi*.
- WiFiClient.h: para criar um cliente *Wi-Fi*.
- ESP8266WebServer.h: para criar um *web server* para o NodeMCU.
- ESP8266mDNS.h: para lidar com DNS no NodeMCU.

O *web server* do NodeMCU tem como o seu *hostname* o nome de "planta" no DNS padrão da rede local. Assim, as rotas são definidas para a execução das operações de cada componente.

Com os dados de parâmetro para fazer os componentes funcionarem em conjunto para a rotina de irrigação e a porcentagem da umidade, é criado uma

request (requisição) para comunicar a porcentagem da umidade atual. A *request* referente à umidade sempre será executada, pois é necessária para mostrar um dado de análise e também para tomar as decisões de forma precisa sobre a rotina de irrigação.

O sistema de irrigação é composto de dois comportamentos. O primeiro é o automático, que rega duas vezes ao dia caso a umidade esteja baixa, e o outro é o de rega manual, feita pelo usuário. se encontra disponível publicamente a base do código NodeMCU para o acionamento dos componentes[1], que complementa a construção abordada neste tópico e justifica a mesma, disponibilizado via Gitlab.

3.2.4.2 Raspberry Pi

Além das demais funcionalidades orquestradas pelo NodeMCU, o Raspberry Pi tem como responsabilidade capturar imagens pelo componente de câmera acoplado à ele. A mesma é conectada via cabo *flat* na placa e via *shell script.* É programada para enviar imagens da planta para o *backend*.

O sistema do Raspberry Pi é bem simples de instalar. Para fazê-lo é necessário a gravação da imagem de OS *linux*, ou qualquer distribuição, no microSD, que, por sua vez, será inserido em uma das entradas para microSD que o modelo Raspberry Pi Zero W dispõe.

Com o uso de uma conexão via rede entre um computador e o Raspberry Pi, é possível ter acesso ao sistema *Linux instalado.* Com isso é possível comportar a interação com qualquer linguagem de programação, bibliotecas e etc, de maneira intuitiva e conveniente.

O Raspberry Pi comporta também o *backend* que se comunica com um servidor cujo qual comporta o banco de dados. Essa comunicação do *backend* com o banco e a interface, além de acontecer em formato de API, no caso da interface, também dispõe de um *túnel ssh* para que a conexão do que está no *hardware* aconteça com demais serviços/interfaces.

---

[1] Disponível em: <https://gitlab.com/camilasrody/iot-plant-nodemcu-esp8266>

O objetivo desse componente é capturar imagens da planta para aproximar o usuário visualmente da mesma.

3.2.5 Desenvolvimento web

3.2.5.1 Backend

No Laravel é usado o modelo padrão do *framework* que disponibiliza um *design pattern* completo de *MVC (Model; View; Controller;)* orientando seguir tal modelo da maneira mais simplificada e performática. Deve-se atentar que no caso desse projeto, o "V" de *view*, que seria a interface, está integralmente desacoplada em um projeto de *frontend* consolidado.

Nele é criado classes de interação (*Plants*; *Users*; *Indicators*; *Logs*) que serão os objetos a serem manipulados nos *controllers*. Em cada *controller* existe uma espera por requisições sobre os dados enviados do *web server* do NodeMCU. Esta parte do projeto é responsável por trabalhar com lógicas de processos a serem executadas na interface sobre a planta como, irrigar quando a umidade está baixa, verificar a umidade, disponibilizar programação de horários e desativação da rotina de irrigação, criar histórico resumido por meio de *logs* das atividades ocorridas e não ocorridas etc. Além disso, também executa validações sobre o código da planta, que seria a licença para liberar o acesso a plataforma.

As licenças definem se aquele vaso existe no sistema. O vaso é sempre previamente registrado e relacionado à classe de plantas (*Plants*) que faz a relação com a classe de usuários (*Users*) para associar a mesma ao novo usuário registrado.

O usuário é registrado automaticamente após a validação, entrando no registro do banco de dados SQL associado a planta já cadastrada no sistema. Então dados como nome, sobrenome etc, são editáveis após o cadastro.

O sistema de rega automática possui uma integração de informações climáticas de acordo com a geo-localização da planta, assim como a umidade do ambiente. Coletando indicadores para prover informações (gráficos) para o usuário.

Além de realizar todas essas funcionalidades, o *backend* também é responsável por retornar as imagens retiradas pelo componente de câmera do próprio Raspberry Pi.

A configuração de conexão com o *Wi-Fi* também estará contida neste projeto. Por estar no *controller*, o mesmo tem como finalidade de repassar os dados capturados da interface para a configuração de *Wi-Fi* do NodeMCU.

Então pode-se concluir que o *backend* faz funções que contemplam o projeto de *hardware* como, irrigar, controlar a irrigação, notificar, criar histórico, liberar acessos e guardar as informações geradas pelo NodeMCU e tratá-las para a melhor experiência do usuário com a interface. Se encontra disponível publicamente a base do código Solução lógica de *software backend* com Laravel[2], que complementa a construção abordada neste tópico e justifica a mesma, disponibilizado via Gitlab.

3.2.5.2 Servidor

O servidor é uma máquina na nuvem onde é possível hospedar projetos de *software*, sendo assim uma das formas de tornar o projeto online e a espera de acessos.

Para hospedar o banco de dados, é utilizado o servidor Heroku, que possui conceitos como *twelve factors*, uma nova metodologia de *software* como serviço, garantindo a melhor interação com as aplicações além de ser uma plataforma aberta.

No Heroku é configurado através do recurso de addons que, são serviços disponíveis pelo mesmo, o banco de dados. É possível configurar tanto pela interface quanto pela interface de linha de comando (CLI) este recurso de hospedar o banco. Com o acesso ao servidor criado, pela linha de comando, apenas com o comando "*heroku addons:create jawsdb-maria*" (HEROKU, 2020), o banco de dados que será criado e passará a ser acessível dentro daquela instância criada no Heroku. Assim, com a disponibilidade do mesmo, é possível pelo *backend* acessar o banco na nuvem realizando a população dos dados no mesmo.

---

[2] Disponível em: <https://gitlab.com/camilasrody/planta-server>

3.2.5.3 Banco de dados

O banco de dados usado neste projeto foi o MariaDB, mas qualquer banco baseado em SQL pode se encaixar. Por ter essa característica de ser implementado todas as regras de ACID (atomicidade, consistência, isolação e durabilidade), isso cria uma consistência sobre os dados e contempla a estrutura dos mesmos ainda que manipulados. Configurado no servidor Heroku, o banco de dados apenas fica a espera de popular os dados das tabelas criadas de acordo com a modelagem baseada em UML. No *backend* é realizado *migrations* que cria o banco e popula com os dados obtidos realizando esse acesso a máquina do Heroku e a aplicação de banco de dados, conectando-se normalmente através dos dados de acesso ao banco.

3.2.5.4 Frontend

Para contemplar a interação do ciclo de plantio, é necessário trazer o usuário mais próximo dos processos. Pensar em elementos de interação de interface são essenciais para compor todo o fluxo de usabilidade da solução. Por isso, a mesma foi concebida com a ideia de ser desacoplada do *backend* (Laravel) com uma estrutura de projeto bem organizada e performática.

A fim de iniciar o projeto de interface, é necessário a escolha de gerenciadores de pacote, bibliotecas de consumo por API *frameworks* de estilos e de manipulação de DOM (*Document Object Model*) . Com esse leque de tecnologias que é construído cada parte da interface, é possível elaborar desde animações e ações até mesmo gráficos que manipulam visualmente os dados obtidos pela aplicação *backend*.

A estrutura de arquivos da interface é completamente convencionada sobre a estrutura que o CLI do VueJS fornece, por isso, o controle sobre os componentes visuais fica mais resiliente e de fácil manutenção. Além disso, o framework VueJS traz reatividade para os comportamentos realizados na interface, o que além de trazer uma experiência melhor para o usuário, também traz performance

A construção desse projeto é baseada em *dashboard*, é nele onde as informações e interações mais importantes serão contempladas. O *dashboard* é constituído por componentes de visualização de acréscimo ou decréscimo sobre a umidade em formato de números e gráficos para melhor visualização do estado atual da umidade da terra e, também seu histórico, cruzando dados já registrados com os atuais. Também é possível orquestrar o estado e funcionamento da bomba d'água e da válvula solenóide através de botões tipo switch, trazendo fluidez no uso desse componente.

Configurar horários para rega é uma das funcionalidades mais importantes, pois é pela mesma que o usuário é capaz de alterar a programação padrão de irrigação. Também é disponibilizado a interação com o clima a partir de geolocalização, comparando a umidade do local com a umidade da planta

Além disso, também é disposto um sub-menu interativo onde é possível ter acesso às páginas que permitem a visualização do histórico de atividades do *hardware* e de imagens capturadas da planta pelo Raspberry Pi.

No menu do usuário, em configurações, é possível a alteração dos dados básicos do mesmo como: nome, sobrenome, idade, foto; assim como sair da aplicação ou conectá-la ao Twitter. O Twitter vem como uma integração que leva os dados básicos que estão contidos na interface para uma outra rede, criando mais um ponto de verdade em relação às mesmas.

As telas iniciais são simples, compostas por elementos que esperam *input* do código do vaso (licença) e a configuração do *Wi-Fi* do ambiente. Seu *design* foi inspirado no minimalismo e na visualização como um app para ser objetivo.

Todas essas características e disposição dos elementos na tela foram realizados com o *framework* CSS, TailwindCSS. Demais reatividades ou interações com o HTML foram realizadas com o *framework* de JS, VueJS. Ambos dispondo de um código mais limpo e compreensível pelo seu *design pattern* (design padronizado).

O TailwindCSS traz consigo considerações de *design* ético, simplicidade, facilidade de uso e conceitos de CSS como, *grid*, *flex-box* etc. O VueJS, não muito diferente também traz consigo todas essas características e ainda mais. Ele leva em

conta toda a estrutura, modificando o DOM com os melhores conceitos usados considerando os componentes *web*.

No entanto, para que todos os dados apareçam na interface é usado o Axios que é uma dependência que gerencia os dados recebidos, e quais APIs estão sendo consumidas. Também, essa dependência disponibiliza as informações requisitadas pelos elementos da interface.

Mas para gerenciar todas essas dependências do projeto, é utilizado o programa de linha de comando Yarn. O Yarn é mais leve comparado ao Npm. Usando-o, é criado um arquivo *package.json* onde fica listado todas as dependências usadas no projeto.

Esta aplicação *frontend* (interface), é hospedada no Netlify, um serviço que suporta a mesma agindo como servidor, já disponibilizando-a na *internet* com considerações de SEO (motor de busca otimizado). Disponível publicamente a base do código Interface *web* para exibir dados e funcionalidades[3], que complementa a construção abordada neste tópico e justifica a mesma, disponibilizado via Gitlab.

Na Figura 8 é possível visualizar a tela inicial de inserção do código de ativação/licença do vaso.

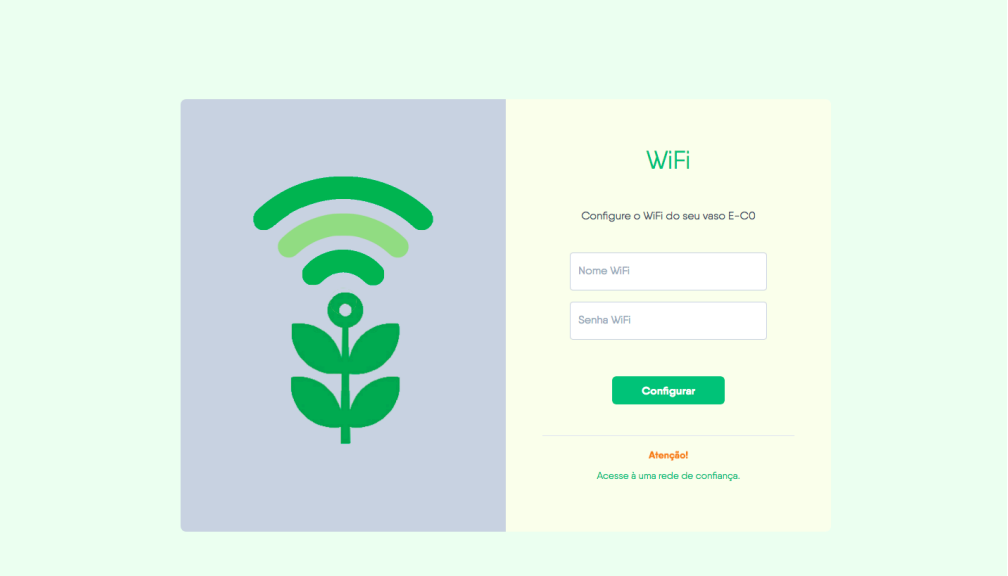
Figura 8 - Inserir o código

Fonte: AUTORES.

[3] Disponível em: <https://gitlab.com/camilasrody/planta-app>

Na Figura 9 é possível visualizar a tela para configurar o *Wi-Fi* da residência.

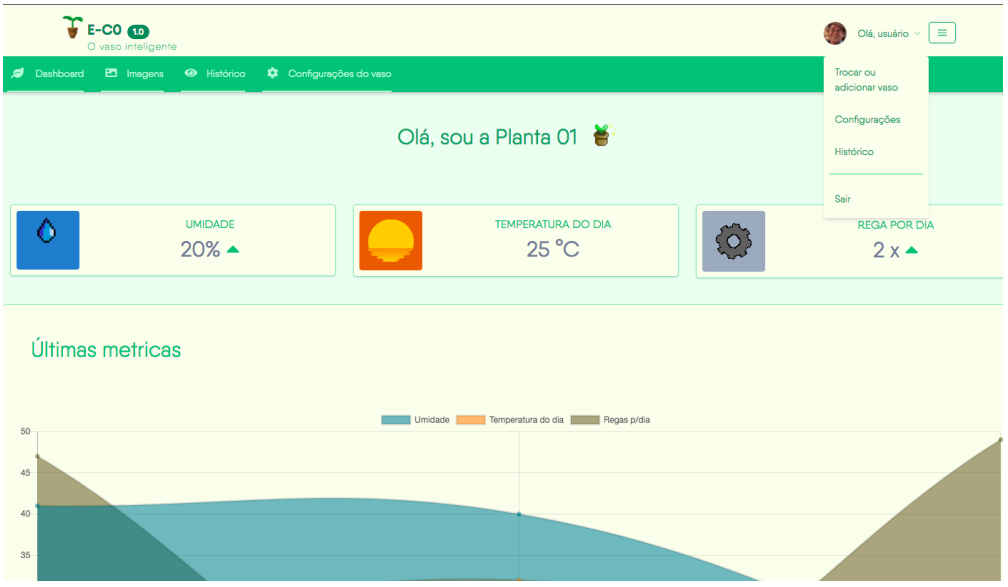
Figura 9 - Interface de configuração da rede *Wi-Fi*



Fonte: AUTORES.

Na Figura 10 observamos os dados que são retornados e mostrados em formato de gráfico.

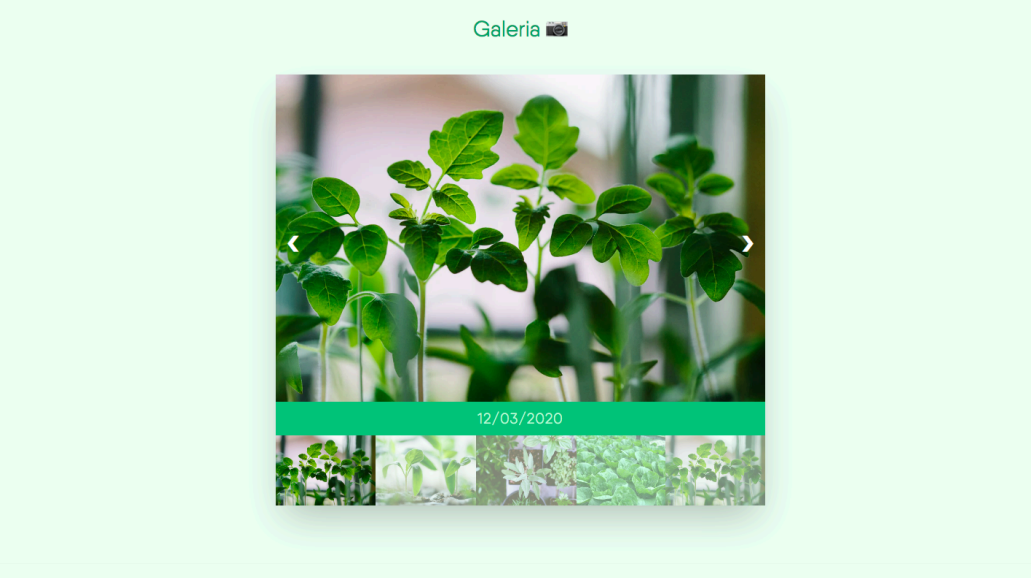
Figura 10 - Interface da *dashboard*



Fonte: AUTORES.

Na Figura 11 apresenta a galeria de imagens que dispõe as imagens guardadas após serem capturadas.

Figura 11 - Interface da galeria



Fonte: AUTORES.

Na Figura 12 apresenta a interface onde o usuário irá colocar as suas informações pessoais e optar por adicionar uma foto ao perfil.

Figura 12 - Interface de configuração dos dados

Fonte: AUTORES.

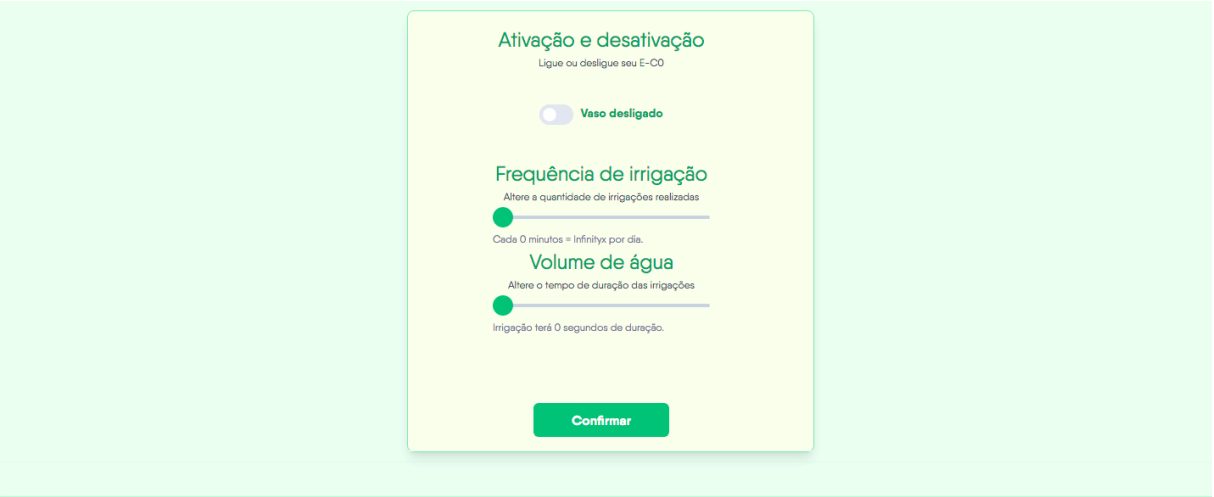
A Figura 13 apresenta o histórico das atividades, provenientes da parte de *hardware*, que guarda as informações referentes à umidade, à temperatura e ao acionamento da bomba d'água e da válvula solenóide.

Figura 13 - Interface do histórico de atividades

Fonte: AUTORES.

A Figura 14 mostra a interface das configurações da planta que contém as opções: ligar/desligar o vaso, quantidade de rega que o usuário pode definir e o controle do volume de água para a irrigação.

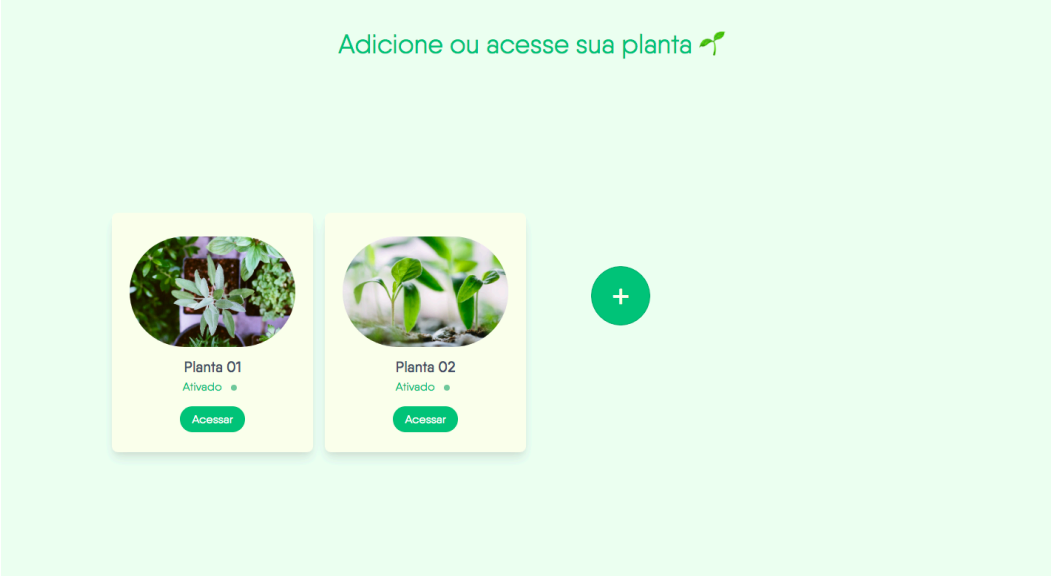
Figura 14 - Interface da configuração da planta



Fonte: AUTORES.

A interface da Figura 15 mostra a quantidade de vasos cadastrados para o mesmo usuário e a possibilidade de expansão para mais vasos.

Figura 15 - Interface de escolha ou de adição de um vaso



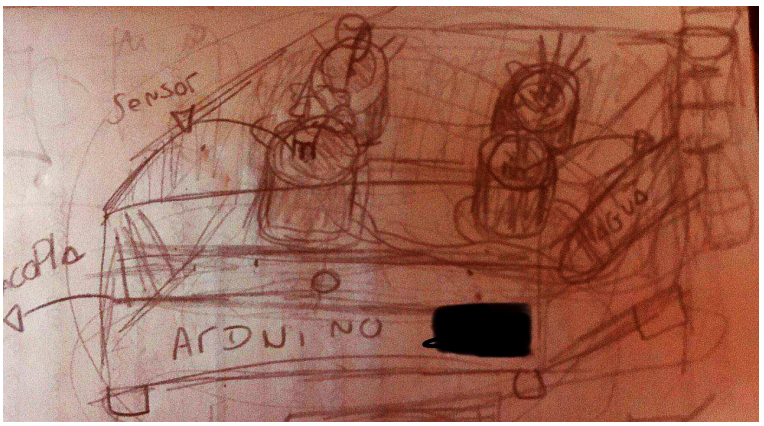
Fonte: AUTORES

3.2.6 Construção do design do produto

Para sintetizar visualmente o esquema do produto, foram desenvolvidos alguns esboços. A ideia é envolver uma carcaça ecológica e eficiente para comportar toda a automação e estrutura requerida para realizar o proposto.

O exemplo da Figura 16, esboçado pela equipe, ilustra plantas em um recipiente feito a partir de caixas, o qual há compartimentos planejados para manter o fluxo de irrigação e monitoramento. Nessa fase, não há o compartimento para a câmera, pois o foco desse esboço é demonstrar a função principal que deve ser garantida.

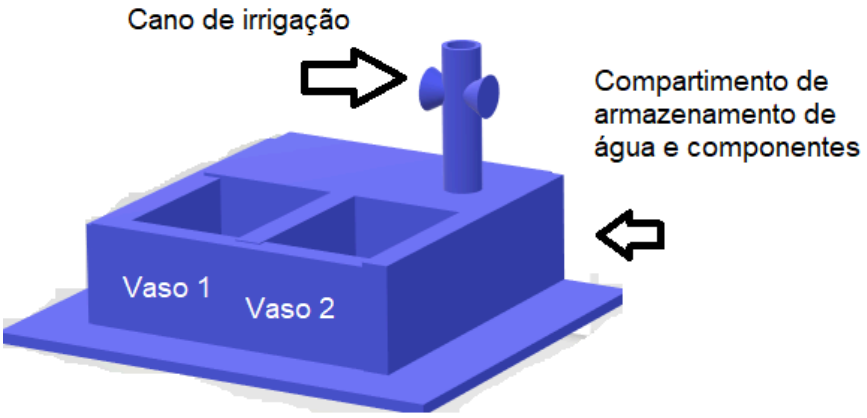
Figura 16 - Esboço Inicial



Fonte: AUTORES.

Observando a Figura 17, o mesmo representa o segundo esboço realizado pelos autores, e demonstra como poderia ser vislumbrada a disposição do *hardware* com a planta.

Figura 17 - Esboço da estrutura



Fonte: AUTORES.

A Figura 18 consolida a estrutura sobre o *design* do produto, proporcionalmente adaptado, considerando todos os componentes e deixando-o o mais compacto possível, foi decidido um formato agradável.

A estrutura é definida de forma vertical logo, colocar o *hardware* em uma perspectiva elevada garante a proteção dos mesmos contra curtos e eventuais empasses relacionado a hidráulica. Além disso, por ter esta característica, disponibiliza mais espaço para colocar vasos de diversos tamanhos. A estrutura possui uma base onde é possível colocar vasos de pequeno à médio porte e, caso precise comportar vasos maiores, a base serve para garantir a sustentação de todo o mecanismo elaborado com *hardware* e componentes.

Através do uso de mangueiras na solenóide, há a distribuição de água através de pequenos furos para causar o efeito de esguichar que ajuda a manter certo controle na quantidade de água dada. Já a fonte de água torna-se a melhor opção para o usuário. Dependendo do local colocado, é possível tanto optar por conectar a mangueira da bomba à torneira, tornando a fonte mais abundante ou optar por um reservatório.

A ideia é uma pequena caixa que pode se adaptar à qualquer superfície próxima aos vasos de plantas. Com a função de ter simples manutenção uma vez que há fácil acesso aos componentes.

Figura 18 - Modelo final da estrutura

Fonte: AUTORES

4. MODELAGEM

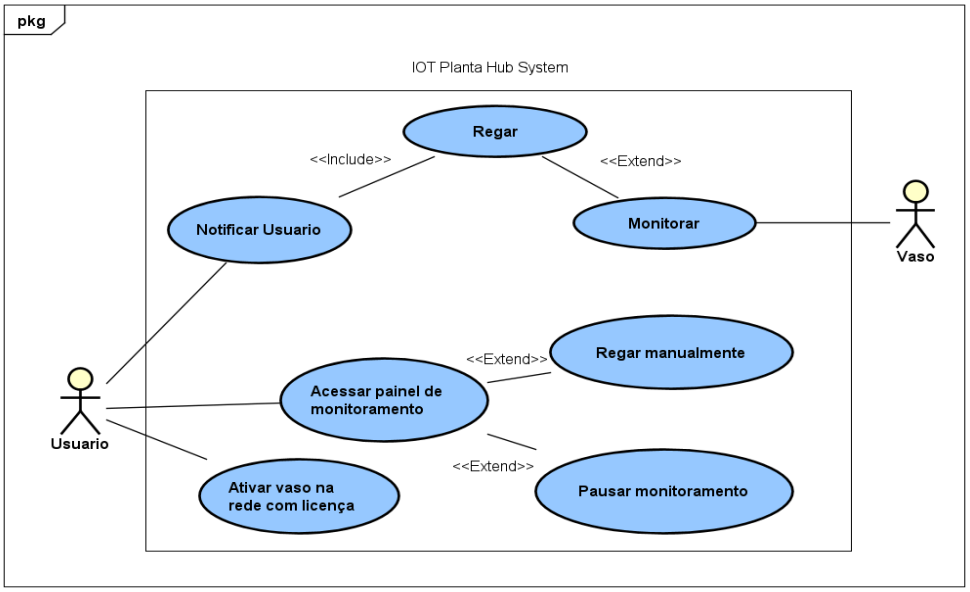
4.1 Caso de uso

Neste tópico serão abordados os casos de uso que compõem tanto o sistema quanto o fluxo como um todo, como é evidenciado no caso de uso da Figura 19. Nele é ilustrado dois agentes que irão interagir com o sistema, o usuário e o vaso.

O usuário pode ser notificado para realizar a rega e também monitorar o vaso de planta pelo sistema. O mesmo também tem acesso ao painel com informações e por ele pode tomar a decisão de desativar a rega automática realizada pelo sistema ou optar por configurar por si próprio a rega, o que é nomeado de ‘rega manual’. Também é possível que o usuário ative a licença, pois apenas dessa forma, o mesmo possuirá acesso às interações e dados.

Quanto ao vaso, ele pode se monitorar. Isso ocorre, pois para a geração do histórico, o sistema precisa sempre estar verificando a si mesmo para notificar o usuário ou guardar a informação de sua atividade. Além disso, como o vaso é programado para regar, seja customizado pelo usuário ou pelo padrão, o mesmo tem a interação da rega no caso de uso por esta razão.

Figura 19 - Caso de uso



Fonte: AUTORES.

Quadro 6 - Especificação do Caso de Uso "Ativar o vaso"

Nome do UC	Ativar vaso	
Descrição	Este caso de uso é utilizado para descrever como será conectado o vaso a plataforma.	
Pré-condições	Este caso de uso pode iniciar somente se: 1. O usuário estar com a licença previamente ativada (logado). 2. O vaso estiver operando.	
Pós-condições	Após o fim normal deste caso de uso o sistema deve: Redirecionar o usuário a tela principal, mostrando as informações	
Ator Primário	Usuário	
Fluxo de Eventos Principal	Ativar	1. O usuário abre a aplicação. 2. O usuário localiza o código disponível no vaso. 3. Insere o código na tela principal. 4. O usuário ganha acesso às informações.
Fluxos de Exceção	E1. O código está incorreto.	1. O sistema não encontra o vaso. 2. O sistema retorna "Código incorreto".

Fonte: AUTORES.

Quadro 7 - Especificação do Caso de Uso "Visualizar informações"

Nome do UC	Visualizar informações	
Descrição	Este caso de uso é utilizado para mostrar ao usuário as informações atuais do vaso.	
Pré-condições	Este caso de uso pode iniciar somente se: O usuário estar conectado ao vaso.	
Pós-condições	Após o fim normal deste caso de uso o sistema deve: Mostrar as informações na tela principal.	
Ator Primário	Usuário	
Fluxo de Eventos Principal	Visualizar	1. O usuário acessa as informações 2. O usuário analisa.

Fonte: AUTORES.

Quadro 8 - Especificação do Caso de Uso "Visualizar lista de plantas"

Nome do UC	Visualizar lista de plantas
Descrição	Este caso de uso é utilizado para mostrar quantas vasos o usuário tem conectado.
Pré-condições	Este caso de uso pode iniciar somente se: 1. O usuário estar com a licença previamente ativada. 2. O usuário estiver na Página principal.
Pós-condições	Após o fim normal deste caso de uso o sistema deve: Mostrar os outros vasos conectados.

Ator Primário	Usuário	
Fluxo de Eventos Principal	Visualizar	1. O usuário abre um menu secundário. 2. Visualiza os vaso conectados. 3. Seleciona o que deseja saber mais informações. 4. É redirecionado ao menu do vaso.
Fluxos de Exceção	E1. Há apenas um vaso conectado .	O sistema retorna um único vaso conectado.

Fonte: AUTORES.

Quadro 9 - Especificação do Caso de Uso "Criar perfil de vaso"

Nome do UC	Visualizar lista de plantas	
Descrição	Este caso de uso é utilizado para criar ou modificar um perfil do vaso pelo usuário.	
Pré-condições	Este caso de uso pode iniciar somente se: O usuário estiver na página principal.	
Pós-condições	Após o fim normal deste caso de uso o sistema deve: Confirmar e gravar os dados.	
Ator Primário	Usuário	
Fluxo de Eventos Principal	Cadastrar	1. O usuário seleciona o menu 2. Seleciona a opção de "Criar novo perfil" 3. Cadastra as informações

Fluxos de Exceção	E1. Sem informações suficientes	1. O usuário não adiciona todas as informações necessárias 2. O sistema não permite ao usuário criar 3. Retorna ao usuário “Inserir todas as Informações necessárias”

Fonte: AUTORES.

Quadro 10 - Especificação do Caso de Uso “Alerta sobre falta d'água”

Nome do UC	Alerta sobre falta d'água	
Descrição	Este caso de uso é utilizado para alertar o usuário sobre falta de abastecimento de água no vaso.	
Pré-condições	Este caso de uso pode iniciar somente se: Se o vaso tiver conexão com a *internet*. Se o vaso estiver associado a algum *smartphone*.	
Pós-condições	Após o fim normal deste caso de uso o sistema deve: Mandar uma notificações caso o abastecimento seja cortado.	
Ator Primário	Usuário	
Fluxo de Eventos Principal	Alertar	1. O sistema detecta que o nível do reservatório baixou. 2. Envia uma notificação ao usuário. 3. O sistema verifica a si mesmo de 30 em 30 minutos para mostrar ao usuário se o nível de água abaixou ou aumentou.

Fluxos de Exceção	E1. Falta de luz	1. O sistema não pode notificar o usuário.
	E2. Sem conexão com a *internet*	1. O sistema notifica que a conexão foi perdida para o usuário.

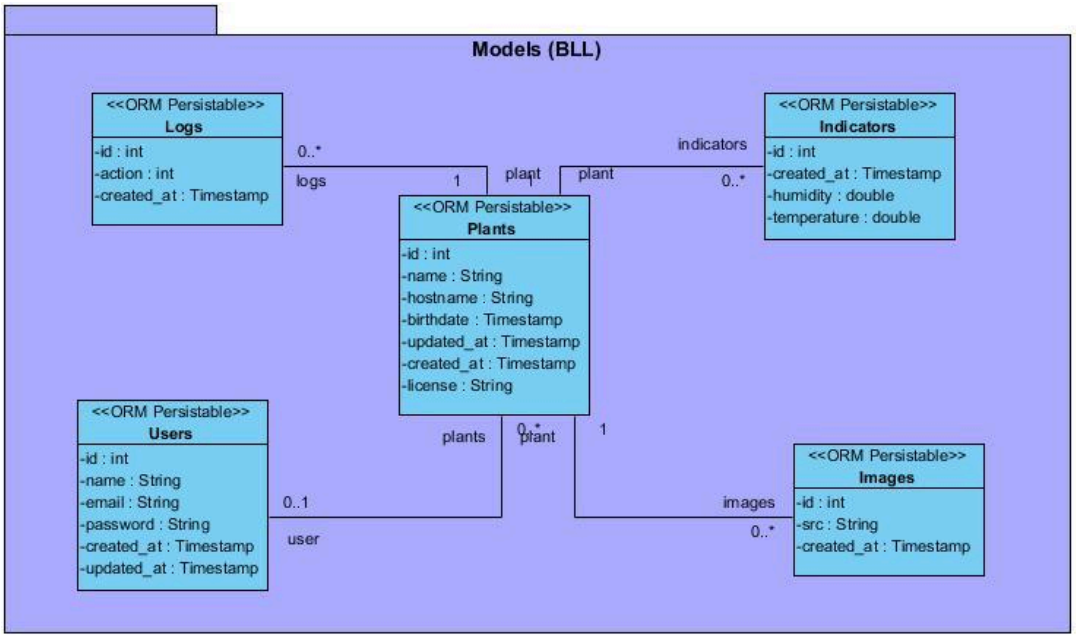
Fonte: AUTORES.

4.2 Diagrama de classe

As modelagens foram realizadas de acordo com o fluxo do sistema, entregando uma visão sobre como o mesmo será estruturado, quais atores e ações serão realizadas além de mostrar os estados da aplicação e a estrutura dos dados perante ao processo proposto no *software*.

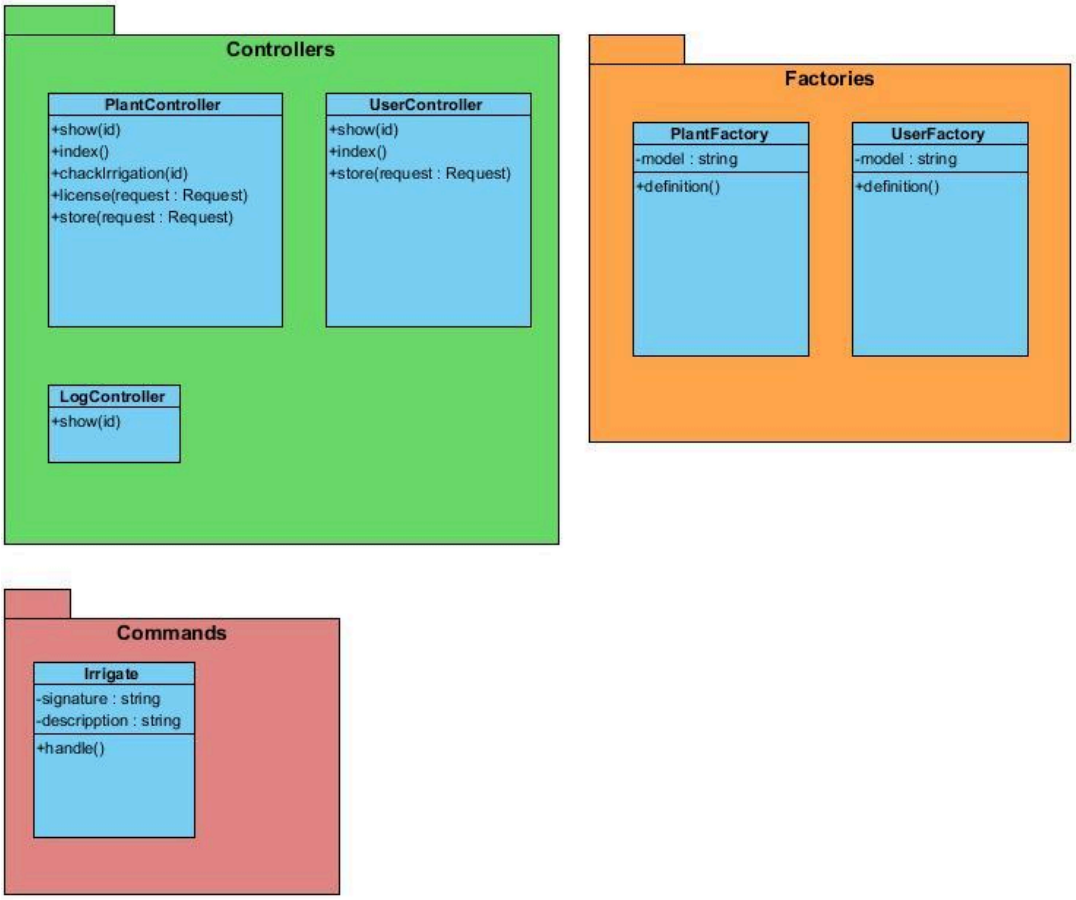
No diagrama de classes nas Figura 20 e 21, é possível notar a estrutura de como os dados iram ser dispostos, tratados e armazenados.

Figura 20 - Diagrama de Classes - Parte 1



Fonte: AUTORES.

Figura 21 - Diagrama de Classes - Parte 2



Fonte: AUTORES.

A Classe *Logs* é referente ao histórico de atividades. A automação irá ser realizada sobre a rotina de irrigação e cuidado com a planta, como por exemplo, se está úmido, se está seco, se foi acionado ou não a irrigação, se a programação foi mudada etc.

Já a de Indicadores é relacionada à umidade que será o indicador para que as demais atividades possam realizar a automação, além de exibi-la para o usuário. É com base nela que o *backend* irá mandar os comandos necessários para a automação. A temperatura também faz parte da classe, pois por meio de integrações com bibliotecas de temperatura com base na localização que seja bem precisa, compõe ainda mais toda a funcionalidade, sendo trabalhado para a aplicação de interface criando notificações.

A classe *plants* é referente à ideia de que o usuário pode ter várias plantas caso queira expandir.

Os campos do usuário não serão obrigatórios, mas caso preenchidos, a interface terá um perfil rápido onde disponibilizará esses dados pessoais para si mesmo após autenticado.

A classe *Images* representa a intenção de se armazenar fotos tiradas pela câmera do Raspberry Pi, a qual o usuário irá ter acesso.

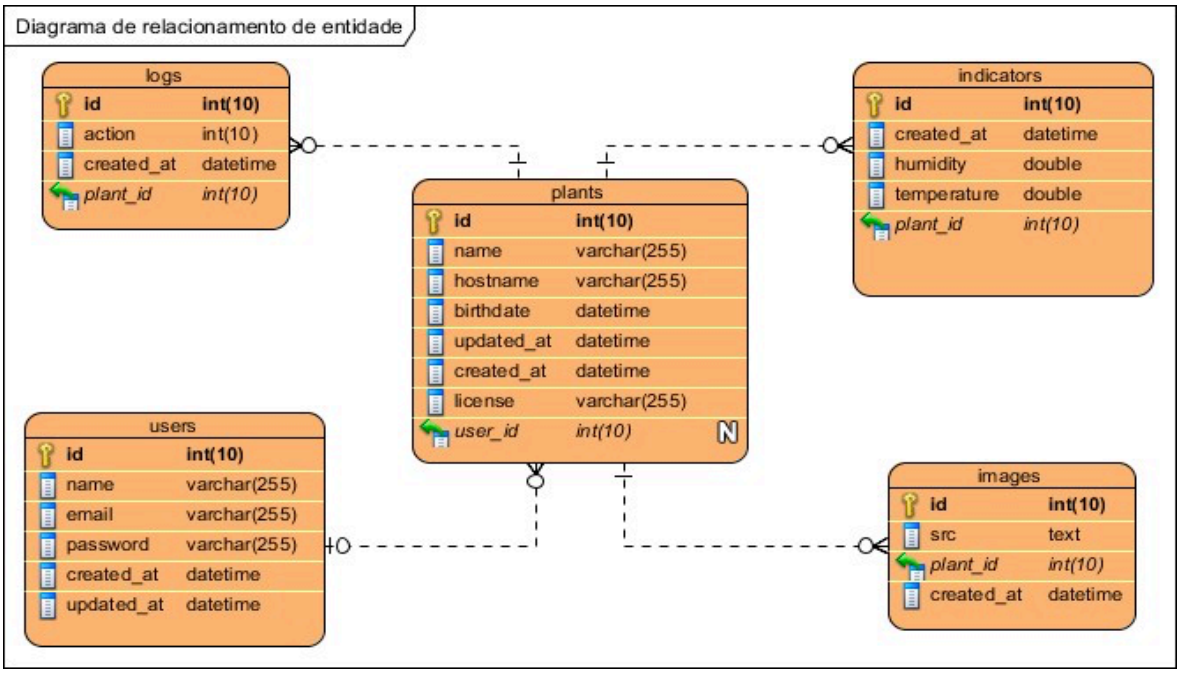
As classes de controllers entram para realizar todas as funções de comunicação necessária entre o *backend* e o *frontend* assim como trabalhar as informações que chegam do *web server* do NodeMCU.

A Factory está relacionada a estrutura a ser recebida para popular o banco de dados e também, tem em sua função criar instâncias podendo ser usado em qualquer parte do projeto.

Já a classe Irrigation, é um comando de *background* agendado que é rodado para realizar verificar a umidade e então tomar a decisão de rega. Esse comando é responsável por checagem e irrigação de todas as plantas nas bases de todos os usuários.

Abaixo no diagrama de entidade-relacionamento na Figura 22, é possível visualizar como os dados do banco de dados iram se dispor de forma que favoreça a programação no *framework* Laravel das classes, com um formato semelhante mostrando com mais detalhes o tipo e o cruzamento desses dados e armazenando-os.

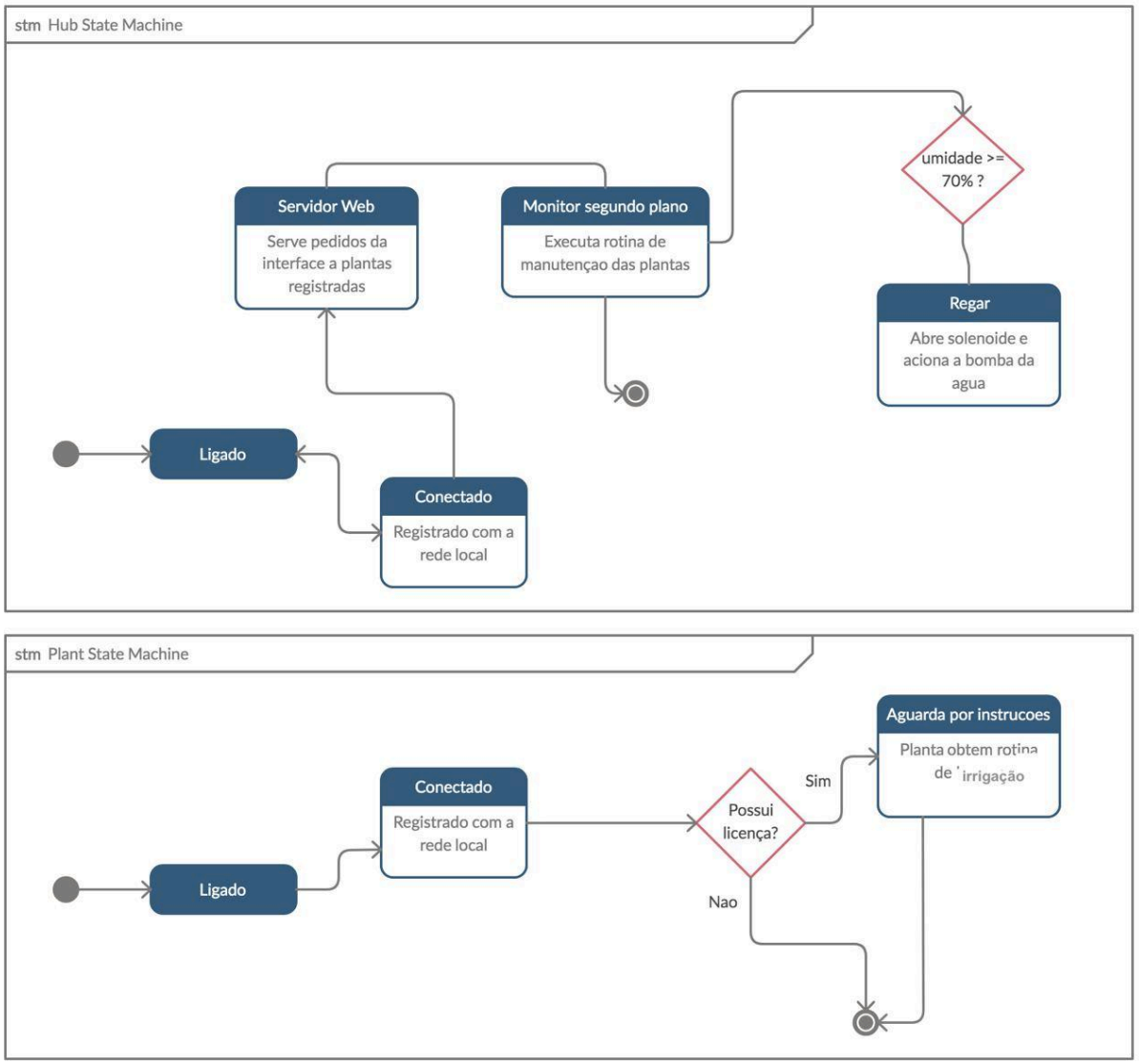
Figura 22 - Diagrama de Entidade Relacionamento (ER)



Fonte: AUTORES.

A Figura 23 representa o diagrama de estado que detalha o estado da aplicação sobre as rotinas propostas a serem realizadas no sistema como um todo.

Figura 23 - Diagrama de estado



Fonte: AUTORES.

No diagrama de estado, nota-se toda a rotina do *software* e também é possível notar que o mesmo sempre verifica se a porcentagem é maior que 70%, valor este que é fixo para melhor compreensão sobre a umidade da terra.

No mesmo diagrama subsequente é verificado por meio da licença que, seria o código do vaso, se é válido ou não, permitindo ao usuário assim a interação com o sistema.

5. RESULTADOS OBTIDOS

5.1 Testes e validações

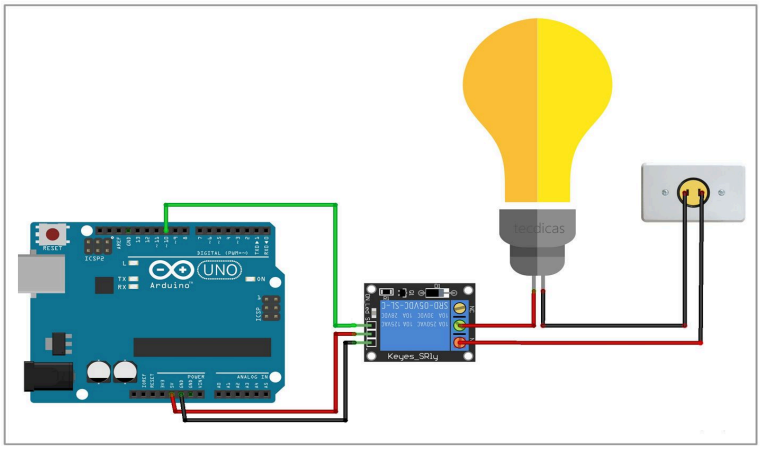
5.1.1 Testes com relé

Alguns testes e validações foram realizados durante a construção do *hardware*. No início da montagem do NodeMCU foi percebido que o primeiro esquema não satisfazia o objetivo pois o módulo relé usado inicialmente se tratava de apenas um isolado, precisando usar mais de um por ser composto por dois componentes. O problema enfrentado é que para comportar um "relé" simples separado, precisaria de uma nova fonte de energia, e a intenção é simplificar o *hardware*. Então, foi optado por usar o “relé” de dois canais, pois além de fazer o funcionamento esperado, protege o NodeMCU por ter isolamento entre o circuito.

No teste de ambos com apenas o NodeMCU como alimentação, foi observado que os relés ligavam, mas os componentes não. Por meio de mais pesquisas a respeito sobre o assunto de como realizar a interação com o “relé”, obteve-se o conhecimento sobre o sistema a ser obtido.

Como se buscava usar a fonte *USB* do próprio NodeMCU, ela devia alimentar tanto a placa do “relé” quanto às entradas de energias correspondentes aos componentes conectados na entrada do "relé". Porém, com algumas estratégias realizadas no circuito, o problema foi resolvido entendendo o princípio do "relé" e a alimentação de energia acompanhando a ideia e a estrutura do esquema na Figura 24.

Figura 24 - Esquema de "relé" com lâmpada



Fonte: (MESSINA, 2019)

5.1.2 Testes com transistor

Com a procura de uma estrutura que fosse mais simples e mais barata, foram feitas pesquisas sobre os melhores esquemas e componentes para controlar a bomba e a válvula.

A fim de substituir o relé, foram realizados testes com componentes semelhantes, um deles seria o transistor BD137. Após muitas pesquisas, foi concluído que seria melhor aproveitável tê-lo no projeto além das demais características, pois o efeito que o transistor tem de diminuir e aumentar a tensão permite persistir tal efeito sobre a potência dos componentes acoplados a ele. Ou seja, passa com mais ou menos intensidade de água pela bomba d'água e solenóide.

Ao contrário do “relé” que apenas interrompe a corrente agindo somente como um interruptor, função que o transistor também é capaz de simular além da previamente citada. Mas para que as altas cargas não afetem o microcontrolador, foi pesquisado qual tipo de diodo era preciso utilizar. Há vários tipos de diodo, mas no caso do esquema realizado, o necessário seria o diodo de regular tensão. O diodo Zener protege o microcontrolador da placa de circuito NodeMCU de altas cargas.

5.2 Resultados gerais

Até o presente momento, o projeto E-CO Vaso Inteligente obteve resultados em parte dos objetivos a serem alcançados. No Quadro 11 é apresentado de forma resumida os resultados obtidos com base nos objetivos definidos no projeto.

Quadro 11 - Resultados obtidos

Objetivo	Realizado
Geral	● O *hardware* completamente funcional e pronto. Seu algoritmo também reproduz o que é necessário para se trabalhar com os dados obtidos dos componentes. ● O Raspberry atua como câmera, tirando fotos a cada dia. ● O *backend* em Laravel, trabalha com as informações coletadas do *software* em C do *hardware* e trabalha as mesmas para reproduzir em formato de dados mais simples e dinâmicos em gráficos. Além disso, também orquestra qual o momento dos componentes ligarem, como devem ligar e agir. ● O *frontend* possui todos os efeitos de UI (*user interface*) para tornar a experiência visualmente agradável através de animações, consideração de cores e posição dos componentes. Também há considerações de UX (*user experience)* que graças a reatividade do Vue JS, proporciona uma experiência bem fluída e rápida.

	● Banco de dados em SQL, hospedado no servidor recebe todas as informações trabalhadas no *backend* antes do mesmo ser passado para a interface *frontend*. ● Integração no *frontend* com twitter também ocorre apenas para trazer os dados básicos de configuração do usuário. ● O *backend* está acoplado ao *hardware* do Raspberry Pi.
Experiência do usuário	Há fluidez e reatividade nos componentes visuais criados. Botões de interação e animações complementam a jornada do usuário ao utilizar o *software frontend*. Todas as requisições realizadas neste sistema são funcionais. Além de ter um ambiente de configuração dedicada aos dados pessoais do usuário como foto, email etc. Interfaces realizadas que mostram dados em *cards* e em gráficos dinâmicos.
Integração	Para se obter a integração entre o vaso e a plataforma foi desenvolvido a conexão entre os componentes de *hardware* e *software*. Consequentemente tornando a comunicação passível de troca de informação.
Microcontrolador	Todo o esquema do NodeMCU já satisfaz as tarefas a serem realizadas, controlando via I/O (*input/output*) os componentes acoplados ao mesmo. O NodeMCU com sucesso, atende às ações programadas no mesmo e requisitadas pelo *software web*. No Raspberry Pi, após sua configuração é possível capturar imagens a cada vinte e quatro (24) horas da planta acoplada à solução de *hardware*.

Componentes	Com a troca realizada do relé por transistor possibilitou maior dinamismo e mais assertividade no controle dos componentes do *hardware*. Além disso, garantiu segurança perante a curtos que possam atingir a placa com o transistor. Todos os componentes foram interligados com sucesso. A bomba d'água passa o cano até a entrada sempre aberta da solenóide que distribui água para as plantas de acordo com a saída que é aberta quando acionado. Os sensores de umidades também acompanham mas independentes, gerando dados de acordo com a umidade da terra. Também foi acrescentado uma fonte ajustável 3v(*volts*) -5v(*volts*) para alimentar todos os componentes pois o NodeMCU se limita apenas a 3v(*volts*).
Análise das plantas	No *backend* métricas são geradas de acordo com os dados capturados do *web server* criado no *hardware* por uma biblioteca em C. São manipuladas e registradas no banco de dados antes de serem levadas para a aplicação *frontend.*

Fonte: AUTORES.

Dessa forma, para o final do projeto, o E-CO Vaso Inteligente formou um protótipo funcional que consegue proporcionar o necessário para o cultivo e irrigação, com o intuito de garantir uma maior praticidade. Com isso, espera-se que a produção orgânica ganhe maior impulsão em uma sociedade de produção em massa.

Assim, consequentemente, almeja-se a ampliação do alcance sobre o uso de alimentos orgânicos e com isto promover algo acessível para promover o plantio caseiro de maneira fácil e prática, quebrando certas barreiras iniciais existentes nessa prática (talento, mobilidade física e outros).

Logo, espera-se que o produto exposto neste trabalho, sirva para encorajar o consumo consciente, levando os valores da sustentabilidade para o máximo de lares possíveis.

6. CONCLUSÃO

O presente trabalho de conclusão de curso permitiu compreender sobre o processo de construção de um cultivo *indoor* sustentável. Diante disso, o principal fator levado em consideração na criação e implementação do projeto, foi a sustentabilidade (preservação e conservação dos recursos naturais).

Presente no diálogo social como um dos mais importantes recursos para a garantia de uma vida futura no planeta, a sustentabilidade se encontra presente como tema principal dos 17 objetivos de desenvolvimento sustentável da ONU, que foi uma das bases para elaboração deste projeto. A iniciativa do vaso ecológico se encaixa nos objetivos 12 (segurança no consumo alimentar e aumento da nutrição com uma agricultura sustentável) e no objetivo 15 (proteção, recuperação e incentivo do uso sustentável dos ecossistemas terrestres, e o impedimento da perda de biodiversidade).

Além disso, existe uma tendência mercadológica que demonstra que a alimentação saudável e a procura por produtos orgânicos tende a aumentar ano a ano, tornando o vaso ecológico uma opção comercial viável, já que o projeto busca promover o cultivo caseiro, resolvendo os problemas mais aparentes no cultivo de plantas (quando regar e quantidade de água necessária).

Para tanto, foram investigadas soluções de baixo custo (placa de arduino e similares), para aumentar uma possível comercialidade futura do produto, visando o custo benefício do produto final. Para a fundamentação teórica, foram reunidas informações sobre a aparelhagem de construção (bomba de água, microcontroladores, válvulas e sensores), para investigar e garantir a melhor forma possível de funcionamento e efetividade do vaso ecológico.

Diante de todos os fatores ressaltados neste documento, o projeto do vaso inteligente se conclui com êxito. É comercialmente viável considerando a concorrência e suas funcionalidades principais satisfazem seu objetivo, a irrigação. Além disso, há a facilidade de integração com a câmera, caso o usuário queira optar por um modelo mais robusto, caso contrário, existe a opção de um modelo mais

simples. Isso mostra a adaptabilidade do produto independente do tipo de modelo que o usuário busca, seja para uma ou mais plantas.

REFERÊNCIAS BIBLIOGRÁFICAS

AQUÁRIOS SOBRINHO. **Guia de montagem de um aquário de água doce - Parte 2.4: Custos das bombas do aquário**, 2019. Disponível em: <https://www.aquariossobrinho.com/post/gmaadbomba>. Acesso em: 30/05/2020.

HEVES, John. T**ransistors**, 2020 Diponível em: <https://electronicsclub.info/transistors.htm> Acesso em: 20/11/2020.

BRAGA, Newton C. **O básico sobre os Microcontroladores – parte 1 (MIC139)**, 2020. Disponível em: <https://newtoncbraga.com.br/index.php/eletronica/52-artigos-diversos/13263-o-basico-sobre-os-microcontroladores-parte-1-mic139>. Acesso em: 23/05/2020.

ATHOSS ELECTRONICS. **Diodo Zener e susas aplicações – O que é um diodo Zener?**, 2020. Disponível em: <https://athoselectronics.com/diodo-zener-e-suas-aplicacoes/>. Acesso em: 20/11/2020

DOITYOURSELF. **DIY Home Improvement Information**, 2020. <https://www.doityourself.com/>. Acesso em: 14/06/2020

FARMBOT. **Open-Source CNC Farming**, 2015. Disponível em: <https://farm.bot/>. Acesso em: 15/05/2020.

FAZ VERDE. **Regar Plantas: Aprenda como, quando e quantas vezes são necessárias**, 2016. Disponível em: <https://www.fazverde.com.br/regar-plantas/>. Acesso em: 23/05/2020.

FAZEDORES. **Você sabe o que é DIY?** <https://blog.fazedores.com/voce-sabe-o-que-e-diy/>. Acesso em: 14/06/2020.

FILIPEFLOP. **O maior portal maker do Brasil**, 2020. Disponível em: <https://www.filipeflop.com/>. Acesso em: 15/05/2020.

Harvard Law Review. **Issues**. Volume 131, n° 6, 2018. Disponível em: <https://harvardlawreview.org/issues/volume-131-issue-6/>. Acesso em: 10/05/2020.

HEROKU. **Cloud Application Platform,** 2020. Disponível em: <https://www.heroku.com> . Acesso em: 20/08/2020.

MDN. **HTML: Linguagem de Marcação de Hipertexto,** 2019. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML> . Acesso em: 20/08/2020.

MDN. **O que é um servidor web? (web server),** 2019. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Learn/Common_questions/o_que_e_um_web_server> . Acesso em: 20/08/2020.

IBGE - Instituto Brasileiro de Geografia e Estatística. **Censo agropecuário 2017: resultados preliminares**. Censo agropec., Rio de Janeiro, v. 7, 2017. Disponível em: <https://biblioteca.ibge.gov.br/visualizacao/periodicos/3093/agro_2017_resultados_preliminares.pdf>. Acesso em: 9/05/2020.

IBGE; SIDRA. **Pesquisa Nacional por Amostra de Domicílios Contínua trimestral**, 2020. Disponível em: <https://sidra.ibge.gov.br/tabela/5431>. Acesso em: 15/06/2020.

LARAVEL. **The PHP Framework for Web Artisans,** 2020. Disponível em: <https://laravel.com> . Acesso em: 20/08/2020.

Lua. **Mu Design,** 2020. Disponível em: <https://mu-design.lu/lua#lua-intro> . Acesso em: 20/08/2020.

MARIADB FOUNDATION. **MariaDB Server: The open source relational database,** 2020. Disponível em: <https://mariadb.org> . Acesso em: 20/08/2020.

MESSINA, Ana Paula. **Como ligar uma lâmpada no Arduino Uno**, 2019. Disponível em: <https://tecdicas.com/como-ligar-uma-lampada-no-arduino-uno/>. Acesso em: 17/06/2020.

ONU - Nações Unidas Brasil. **Conheça os novos 17 Objetivos de Desenvolvimento Sustentável da ONU**, 2015. Disponível em: <https://nacoesunidas.org/conheca-os-novos-17-objetivos-de-desenvolvimento-sustentavel-da-onu/>. Acesso em: 10/05/2020.

SANTIAGO J.P., **Novo salário mínimo de 2020 será de R$ 1.045. Veja o que dá para comprar com esse valor,** 2020. Disponível em: <https://valorinveste.globo.com/objetivo/organize-as-contas/noticia/2020/01/15/novo-salario-minimo-de-2020-sera-de-r-1045-veja-o-que-da-para-comprar-com-esse-valor.ghtml>. Acesso em: 15/06/2020.

SASS. **CSS with superpowers,** 2020. Disponível em: <https://sass-lang.com> . Acesso em: 20/08/2020.

SEBRAE. **5 tendências no mercado de alimentação saudável: o setor cresceu 98% nos últimos anos no Brasil. Saiba como empreender no mercado de alimentação saudável**. Disponível em: <https://www.sebrae.com.br/sites/PortalSebrae/semanadomei2019/5-tendencias-no-mercado-de-alimentacao-saudavel,5240da3e777ba610VgnVCM1000004c00210aRCRD>. Acesso em: 9/05/2020.

SILVEIRA, Cristiano B. **Como Funciona a Válvula Solenóide e Quais São os Tipos Existentes?** 2018. Disponível em: <citisystems.com.br/valvula-solenoide/>. Acesso em: 23/05/2020.

SOARES, Nelson V. **Sensor de umidade**, 2016. Disponível em: <https://blog.render.com.br/eletronica/sensor-de-umidade/>. Acesso em: 23/05/2020.

STEWART, Lauren. **Front End vs Back End Development**, 2019. Disponível em: <https://www.coursereport.com/blog/front-end-development-vs-back-end-developme n t-where-to-start>. Acesso em: 20/09/2020.

STROSKI, Pedro Ney. **Conheça o relé eletromecânico**, 2018. Disponível em: <https://www.electricalelibrary.com/2018/08/06/conheca-o-rele-eletromecanico/>. Acesso em: 30/05/2020.

TAILWINDCSS. **Rapidly build modern websites without ever leaving your HTML**, 2020. Disponível em: <https://tailwindcss.com>. Acesso em: 20/08/2020.

VALOR INVESTE. **Novo salário mínimo de 2020 será de R$ 1.045. Veja o que dá para comprar com esse valor,** 2020. Disponível em: <https://valorinveste.globo.com/objetivo/organize-as-contas/noticia/2020/01/15/novo-salario-minimo-de-2020-sera-de-r-1045-veja-o-que-da-para-comprar-com-esse-valor. ghtml>. Acesso em: 15/06/2020.

VUEJS. **The Progressive JavaScript Framework,** 2020. Disponível em: <https://vuejs.org> . Acesso em: 20/08/2020.

HEROKU. **JAWSDB Maria**, 2020 Disponível em: **<**https://devcenter.heroku.com/articles/jawsdb-maria>. Acessado em: 20/11/2020