

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)
CURSO DE ENGENHARIA DE COMPUTAÇÃO

ANDRÉ OTAVIO PEDROFEZA DE OLIVEIRA
ARTHUR MIGUEL SEEHAGEN DE OLIVEIRA
FELIPE ALVES BARBOZA

**HIDRONUTRIDOR
DISPOSITIVO REGULADOR DE NUTRIÇÃO PARA PLANTAS
HIDROPÔNICAS**

OFICINA DE INTEGRAÇÃO 2 – RELATÓRIO FINAL

CURITIBA

2021

ANDRÉ OTAVIO PEDROFEZA DE OLIVEIRA
ARTHUR MIGUEL SEEHAGEN DE OLIVEIRA
FELIPE ALVES BARBOZA

**HIDRONUTRIDOR
DISPOSITIVO REGULADOR DE NUTRIÇÃO PARA PLANTAS
HIDROPÔNICAS**

Relatório Final da disciplina Oficina de Integração 2, do curso de Engenharia de Computação, apresentado aos professores que ministram a mesma na Universidade Tecnológica Federal do Paraná como requisito parcial para obtenção da aprovação na disciplina.

Orientador: Prof. Dr. César Manuel Vargas Benítez
Prof. Dr. Heitor S. Lopes

CURITIBA

2021

RESUMO

. HIDRONUTRIDOR

DISPOSITIVO REGULADOR DE NUTRIÇÃO PARA PLANTAS HIDROPÔNICAS. 51 f. Oficina de Integração 2 – Relatório Final – Curso de Engenharia de Computação, UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR). Curitiba, 2021.

Apresentamos neste projeto o desenvolvimento de um dispositivo que regula a solução nutritiva de uma estufa de alface hidropônica. Através do monitoramento dos fatores: *pH*, temperatura e concentração de sais da água, o dispositivo automaticamente administra a adição - por meio da ativação de bombas peristálticas - de adubo e das soluções reguladoras de *pH* para que as condições ideais de cultivo da alface hidropônica sejam mantidas. Além disso, através de uma interface *Web*, o usuário pode: cadastrar uma nova estufa, calibrar os sensores de *pH* e condutividade, visualizar as leituras dos sensores da estufa em tempo real e estabelecer o horário de funcionamento da bomba hidráulica que circula a solução pela estufa.

Palavras-chave: Alface, Estufa Hidropônica, Hidroponia, Aplicação *Web*, *ESP32*

LISTA DE FIGURAS

FIGURA 1 –	Two Points Calibration	16
FIGURA 2 –	Sensor e módulo de condutividade	17
FIGURA 3 –	Diagrama de blocos do sistema	19
FIGURA 4 –	Estrutura mecânica: estágio inicial	20
FIGURA 5 –	Estrutura mecânica: injetores	21
FIGURA 6 –	Estrutura mecânica e reservatório principal	21
FIGURA 7 –	Recipientes auxiliares	22
FIGURA 8 –	Esquemático do circuito: versão inicial	23
FIGURA 9 –	Esquemático do circuito: versão final	24
FIGURA 10–	Esquemático do circuito: bombas hidráulicas	25
FIGURA 11–	Diagrama de Transição de Estados do <i>Firmware</i>	26
FIGURA 12–	Diagrama de blocos de <i>software</i>	27
FIGURA 13–	Diagrama de casos de uso.	28
FIGURA 14–	Diagrama de sequencia para login.	29
FIGURA 15–	Diagrama de sequencia para logout.	29
FIGURA 16–	Diagrama de sequencia para criar estufa.	30
FIGURA 17–	Diagrama de sequencia para deletar estufa.	30
FIGURA 18–	Diagrama de sequencia para selecionar estufa.	31
FIGURA 19–	Diagrama de sequencia para alterar tempos de funcionamento da bomba principal.	31
FIGURA 20–	Diagrama de sequencia para calibrar sensores.	32
FIGURA 21–	Diagrama de Entidade-Relacionamento.	33
FIGURA 22–	<i>Sitemap</i> da Interface <i>Web</i>	34
FIGURA 23–	Tela de Login	37
FIGURA 24–	<i>Dashboard</i>	38
FIGURA 25–	Diálogo de nova estufa	39
FIGURA 26–	Tela de calibragem	40
FIGURA 27–	Diálogo de calibragem de <i>pH</i>	41
FIGURA 28–	Opção de ajuste de horário de funcionamento da bomba principal	42
FIGURA 29–	Calibragem de <i>pH</i> : solução ácida	43
FIGURA 30–	Calibragem de condutividade	43
FIGURA 31–	Cronograma detalhado	46
FIGURA 32–	Diagrama de Gantt	47

LISTA DE TABELAS

TABELA 1 – Horas gastas por membro	45
TABELA 2 – Materiais e custos	48

LISTA DE SIGLAS

ADC	<i>Analog to Digital Converter</i>
DOM	<i>Document Object Model</i>
UI	<i>User Interface</i>
PVC	Policloreto de vinila
DTE	Diagrama de Transição de Estados

LISTA DE SÍMBOLOS

cm	Centímetro
S	Siemens
V	Volts

SUMÁRIO

1 INTRODUÇÃO	13
1.1 MOTIVAÇÃO	13
1.2 OBJETIVOS	13
1.2.1 Objetivo geral	13
1.2.2 Objetivos específicos	13
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 ESP32	15
2.2 SENSORES	15
2.2.1 Sensor de pH	15
2.2.2 Sensor de condutividade	17
2.3 POSTGRESQL	18
2.4 VUE	18
2.5 VUETIFY	18
2.6 NODEJS	18
3 METODOLOGIA	19
3.1 VISÃO GERAL	19
3.2 PROJETO MECÂNICO	20
3.3 PROJETO DE HARDWARE	22
3.4 PROJETO DE SOFTWARE	27
3.4.1 Modelagem	28
3.4.2Interface	33
3.4.3Integração	34
4 EXPERIMENTOS E RESULTADOS	37
5 CRONOGRAMA E CUSTOS DO PROJETO	45
5.1 CRONOGRAMA	45
5.2 CUSTOS	45
6 CONCLUSÕES	49
6.1 CONCLUSÕES	49
6.2 TRABALHOS FUTUROS	49
REFERÊNCIAS	51

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

A motivação de nosso projeto advém da necessidade de se realizar, em uma estufa hidropônica, ajustes periódicos da concentração de nutrientes e do *pH* da solução nutritiva das plantas. Muitos agricultores não realizam tal controle de forma eficiente, deixando, muitas vezes, de ajustar os parâmetros da solução no momento apropriado ou acabam por administrar a quantidade incorreta de adubo e soluções corretoras de *pH*. Eventualmente, este ajuste inapropriado acarreta no desenvolvimento insatisfatório e morte de plantas, trazendo prejuízos ao agricultor. Por meio de nosso projeto, desejamos, portanto, auxiliar o trabalho do agricultor através da automatização do controle da solução nutritiva.

1.2 OBJETIVOS

1.2.1 OBJETIVO GERAL

Automatizar a regulagem de *pH* e concentração de nutrientes da solução nutritiva de uma estufa de alface hidropônica.

1.2.2 OBJETIVOS ESPECÍFICOS

O projeto deve permitir ao usuário acessar uma interface *Web* na qual poderá:

- Realizar *login* e *logout* no sistema.
- Visualizar, em tempo real, as leituras dos sensores de *pH*, condutividade e temperatura da solução nutritiva da estufa.
- Cadastrar, selecionar e remover uma estufa.

- Calibrar os sensores de condutividade e pH da estufa.
- Ajustar o horário de funcionamento da bomba hidráulica da estufa.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 ESP32

ESP32 é uma série de microcontroladores de baixo custo e com baixo consumo de energia. Possui diversos recursos integrados ao chip, como Wi-Fi e Bluetooth, e possui uma boa capacidade de processamento, sendo capaz de rodar um sistema operacional como o RTOS com tranquilidade.

A série ESP32 emprega um microprocessador *Tensilica Xtensa LX6* com duas variações, dual-core e single-core de 32 bits, operando em 160 ou 240 MHz. Possui 520 KB de SRAM, inclui uma antena integrada, amplificador de potência, receptor de baixo ruído amplificado, filtros, gerenciamento de energia dos módulos, três UARTs, dois ADCs de 12 bits, além de diversos outros recursos. ESP32 foi criado e desenvolvido pela *Espressif Systems*, uma empresa Chinesa com sede em Xangai.

Esses fatores o tornam adequado para o desenvolvimento de produtos de IoT, suprimindo todos os requisitos presentes no nosso projeto. A versão empregada foi o módulo ESP-WROOM-32, que conta com o processador dual-core.

2.2 SENSORES

2.2.1 SENSOR DE *pH*

Para a calibragem dos sensores de *pH*, nós utilizamos o método da calibração por dois pontos (*Two Point Calibration*) (ADAFRUIT, 2021). Como leituras de referência para o sensor de *pH*, utilizamos duas soluções tampão de *pH* 4 e 7.

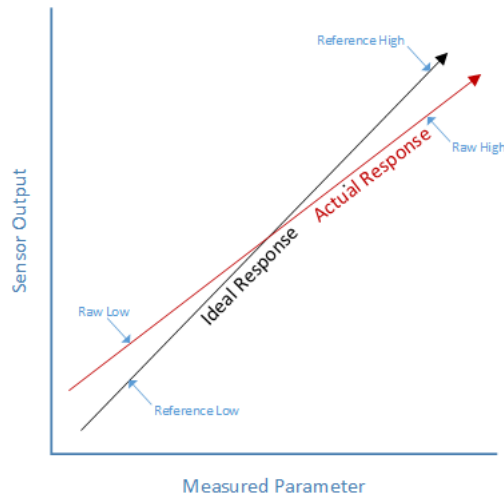


Figura 1: Two Points Calibration

Fonte: (ADAFRUIT, 2021)

Sendo S_1 o valor de referência da solução de pH baixo, S_2 o valor de referência da solução de pH alto, V_1 o valor lido pelo sensor para a S_1 e V_2 o valor lido para a solução S_2 , o cálculo dos coeficientes a e b utilizados na calibragem foi realizado da seguinte forma:

$$aV_1 + b = S_1 \quad (1)$$

$$aV_2 + b = S_2 \quad (2)$$

Isolando a em (1):

$$a = \frac{S_1 - b}{V_1} \quad (3)$$

Isolando b em (2):

$$b = S_2 - aV_2 \quad (4)$$

Substituindo (4) em (3):

$$a = \frac{S_1 - (S_2 - aV_2)}{V_1} = \frac{S_1 - S_2 + aV_2}{V_1}$$

Portanto:

$$a = \frac{S_2 - S_1}{V_2 - V_1} \quad (5)$$

$$b = S_2 - \frac{(S_2 - S_1)}{V_2 - V_1} V_2 \quad (6)$$

De posse dos coeficiente a e b , e do valor de pH V lido pelo ADC do *ESP32*, podemos

calcular S , o valor de pH corrigido, por meio da seguinte relação:

$$S = aV + b \quad (7)$$

Substituindo (5) e (6) em (8), obtemos:

$$S = \left(\frac{S_2 - S_1}{V_2 - V_1} \right) (V - V_2) + S_2$$

2.2.2 SENSOR DE CONDUTIVIDADE

O sensor de condutividade realiza a medição em micro-siemens e é ajustado pelo seu valor ADC.

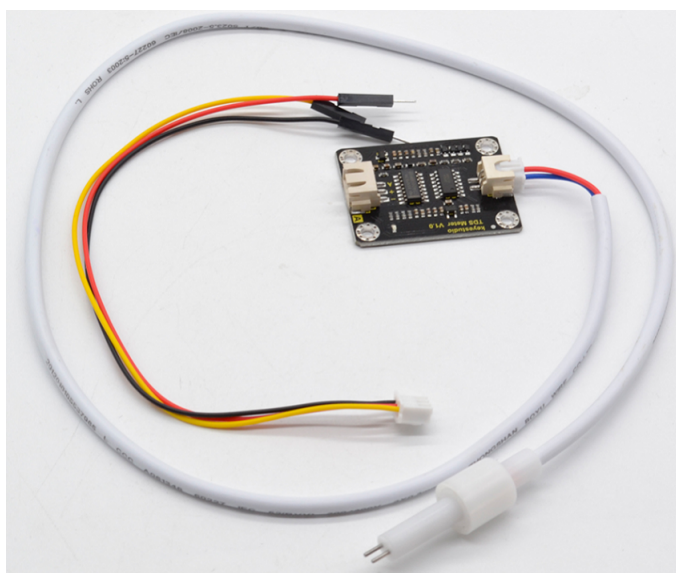


Figura 2: Sensor e módulo de condutividade

Fonte: (KEYESTUDIO, 2021)

Para sua regulagem é necessário obter o valor de referência de uma solução padrão S_1 - para o nosso caso, a solução de referência possui valor de condutividade elétrica igual a $1413 \mu S/cm$. Com ele é possível comparar com o que o sensor está realmente medindo: S_2

Para ajustarmos o erro é necessário realizar uma conta simples:

$$V = S_1/S_2 \quad (8)$$

Ao obtermos o V , o *back-end* manda uma mensagem para o *esp32*

sinalizando que deve-se multiplicar a leitura por V para obter o resultado real.

2.3 POSTGRESQL

O sistema de banco de dados relacional escolhido foi o Postgres. A equipe já possuía familiaridade com o mesmo e o sistema possui instruções SQL simples e permite facilidade de instalação local em qualquer máquina, além de ser *open-source*.

2.4 VUE

Vue é um *framework open-source* de *JavaScript*. Foi escolhido como ferramenta para se construir o *front-end* do portal *Web* por apresentar curva de aprendizagem relativamente rápida e por demonstrar bons resultados em termos de eficiência e *design*. Sua *DOM* virtual e interoperabilidade com outras bibliotecas permite a construção de aplicações *Single-Page* interativas e sofisticadas. (VUEJS, 2021)

2.5 VUETIFY

O *Vuetify* é um *UI framework* para *Vue.js* voltado para a construção de interfaces *Web*; contém diversos componentes que seguem os princípios do *Material Design*, uma linguagem de *design* criada pela *Google*. Por ser de fácil utilização e apresentar bom desempenho, esta foi a biblioteca utilizada para criação da *UI* da *Webpage*. (VUETIFY, 2021)

2.6 NODEJS

O *NodeJS* é um *framework open-source* de *JavaScript*. Similar ao *Vue* porém com uma abordagem diferente: *back-end*. Junto à uma grande quantidade de bibliotecas disponíveis, o *NodeJS* possui escalabilidade para gerenciamento de rotas e *end-points*, além de facilitar a comunicação com o banco de dados. Outra vantagem foi poder unir a comunicação com o *front-end* e o *esp32*, pois é possível realizar a comunicação via *WebSockets* em paralelo com as requisições vindas do portal *web*.

3 METODOLOGIA

3.1 VISÃO GERAL

Demos início ao desenvolvimento do projeto identificando, primeiramente, quais elementos eram necessários para que resolvêssemos o problema que iríamos solucionar. O cultivo de plantas hidropônicas exige que certos parâmetros da solução nutritiva sejam mantidos dentro de uma faixa de valores. Para o nosso caso em específico, a concentração de nutrientes necessária para o cultivo de alface, medida por meio da condutividade elétrica da solução, situa-se dentro da faixa de $1300\mu S/cm$ a $1500\mu S/cm$ e a faixa de pH recomendada é de 5.5 a 6.5. O monitoramento destes parâmetros requer sensores de pH e condutividade; a temperatura também afeta os valores de condutividade elétrica da solução, portanto, um sensor de temperatura é igualmente necessário (FLORAMAX, 2021).

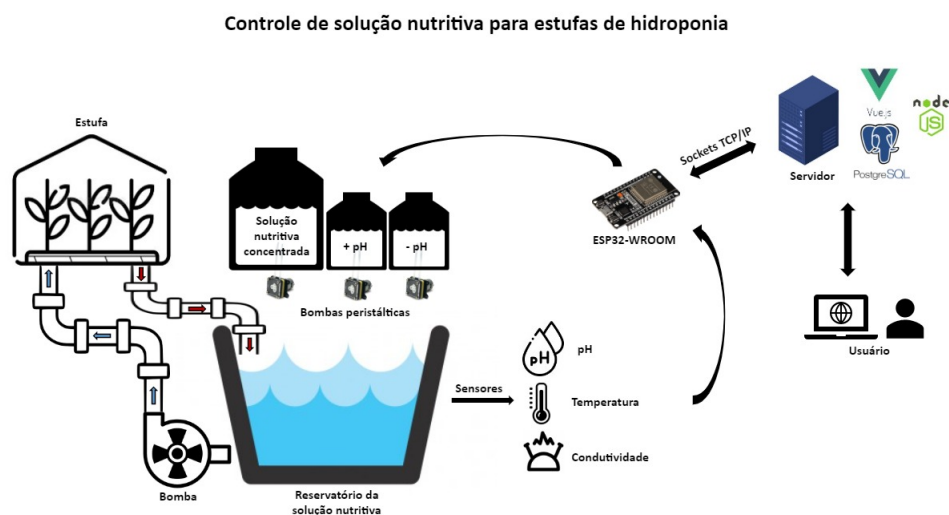


Figura 3: Diagrama de blocos do sistema

Fonte: Autoria própria

Após identificarmos os parâmetros a serem regulados e monitorados, iniciamos a conceptualização do *software* a ser integrado e dos protocolos de comunicação que seriam utilizados. Decidimos por dividir o desenvolvimento de nosso projeto em três frentes: *front-end*, *back-end* e *hardware+estrutura-mecânica*.

Tendo definido os detalhes acima, prosseguimos com a aquisição dos sensores e componentes. Os sensores foram escolhidos tendo como base o preço e compatibilidade com microcontroladores. Optamos por fazer a estrutura de nossa estufa com canos de PVC, por ser este um material relativamente barato, versátil, de fácil manipulação e impermeável.

3.2 PROJETO MECÂNICO

A estrutura mecânica de nosso projeto foi desenvolvida com base em estufas de hidroponia reais e construída com canos de PVC. Nos 5 canos que acomodam as mudas de alface, 30 furos foram perfurados no total - 6 furos em cada cano.

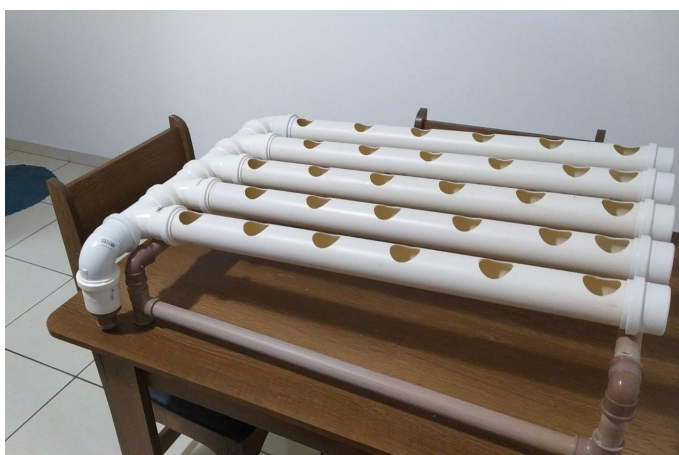


Figura 4: Estrutura mecânica: estágio inicial

Fonte: Autoria própria

Cada um dos canos está conectado a injetores que promovem o fluxo da solução nutritiva para as plantas. A solução, por sua vez, é armazenada no reservatório principal e bombeada para a estrutura por meio de uma bomba de aquário *Sarlo SB1000C*. Após circular pela estrutura, a solução é drenada para o reservatório principal para que seja novamente bombeada.



Figura 5: Estrutura mecânica: injetores

Fonte: Autoria própria



Figura 6: Estrutura mecânica e reservatório principal

Fonte: Autoria própria

Ao reservatório principal, estão conectados reservatórios auxiliares que

armazenam os nutrientes e as soluções reguladoras de pH , os quais são administrados na solução nutritiva por bombas peristálticas controladas pelo microcontrolador *ESP32*.



Figura 7: Recipientes auxiliares

Fonte: Autoria própria

Imersos na solução do reservatório principal, estão os sensores de pH , condutividade e temperatura, onde realizam suas leituras e transmitem os dados coletados para o microcontrolador.

3.3 PROJETO DE HARDWARE

O projeto da parte eletrônica iniciou após termos definido o modelo de microcontrolador que viemos a utilizar. Escolhemos o *ESP32* pois - além de possuímos experiência prévia com ele - é capaz de estabelecer comunicação *Wi-Fi* e apresenta bom desempenho.

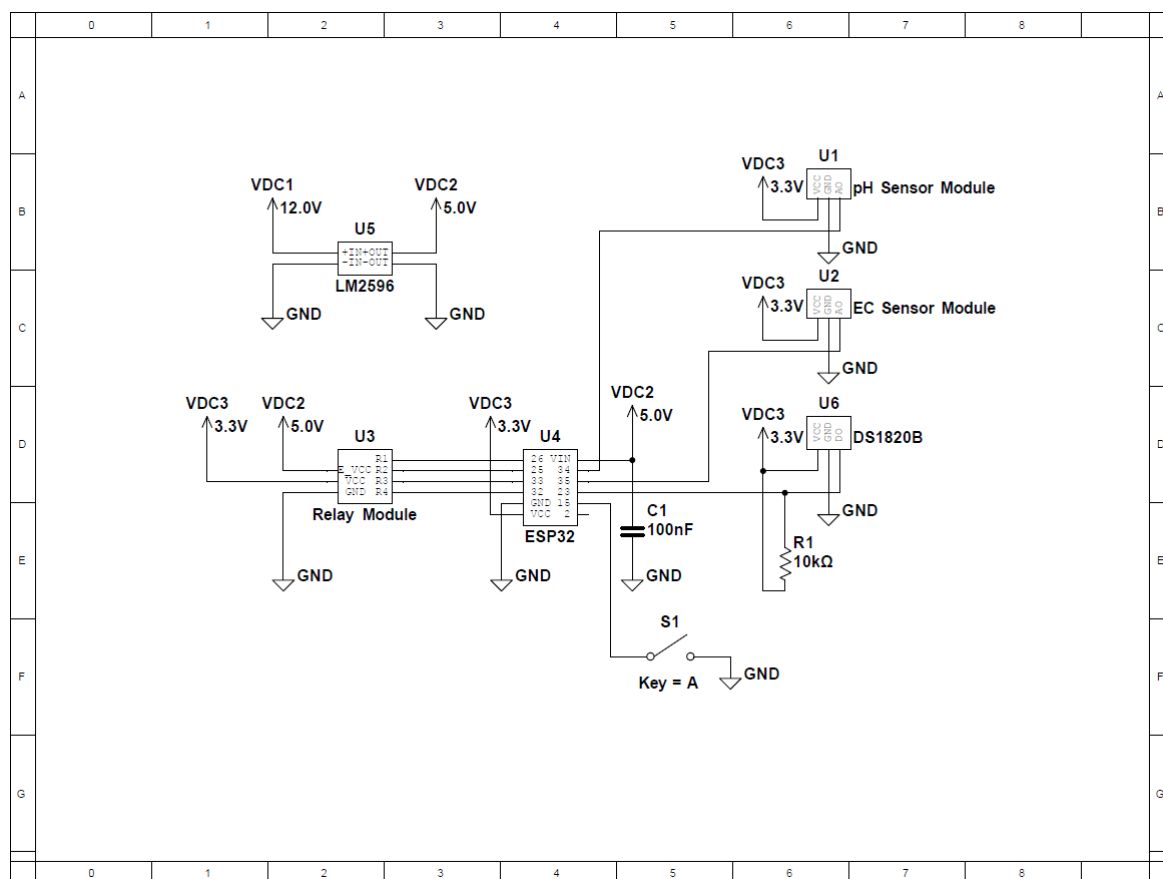


Figura 8: Esquemático do circuito: versão inicial

Fonte: Autoria própria

No diagrama acima, a tensão de alimentação VDC1, fornecida por uma fonte chaveada de 12 V, alimenta todo o sistema. As bombas peristálticas são alimentadas com estes 12V, e o regulador de tensão LM2596 transforma VDC1 na tensão VDC2 de 5V, a qual irá alimentar o pino Vin do ESP32 e a parte lógica do módulo de relés de 4 canais. Com a intenção de reduzir ruídos provenientes das bombas e das bobinas dos relés, optamos por utilizar um módulo de relés optoacoplado.

A tensão VDC3 de saída gerada pelo regulador interno do microcontrolador alimenta os módulos dos sensores e é utilizada como tensão de referência para o módulo de relés.

Contudo, descobrimos um grande problema ao fazer os testes de hardware seguindo este circuito. Os sensores de pH e condutividade quando imersos na solução nutritiva simultaneamente, interferem um no outro. Isso se dá pelo fato de ambos gerarem uma tensão elétrica na solução e seus terras estarem no mesmo

potencial. A solução encontrada para contornar este problema pode ser vista no diagrama a seguir.

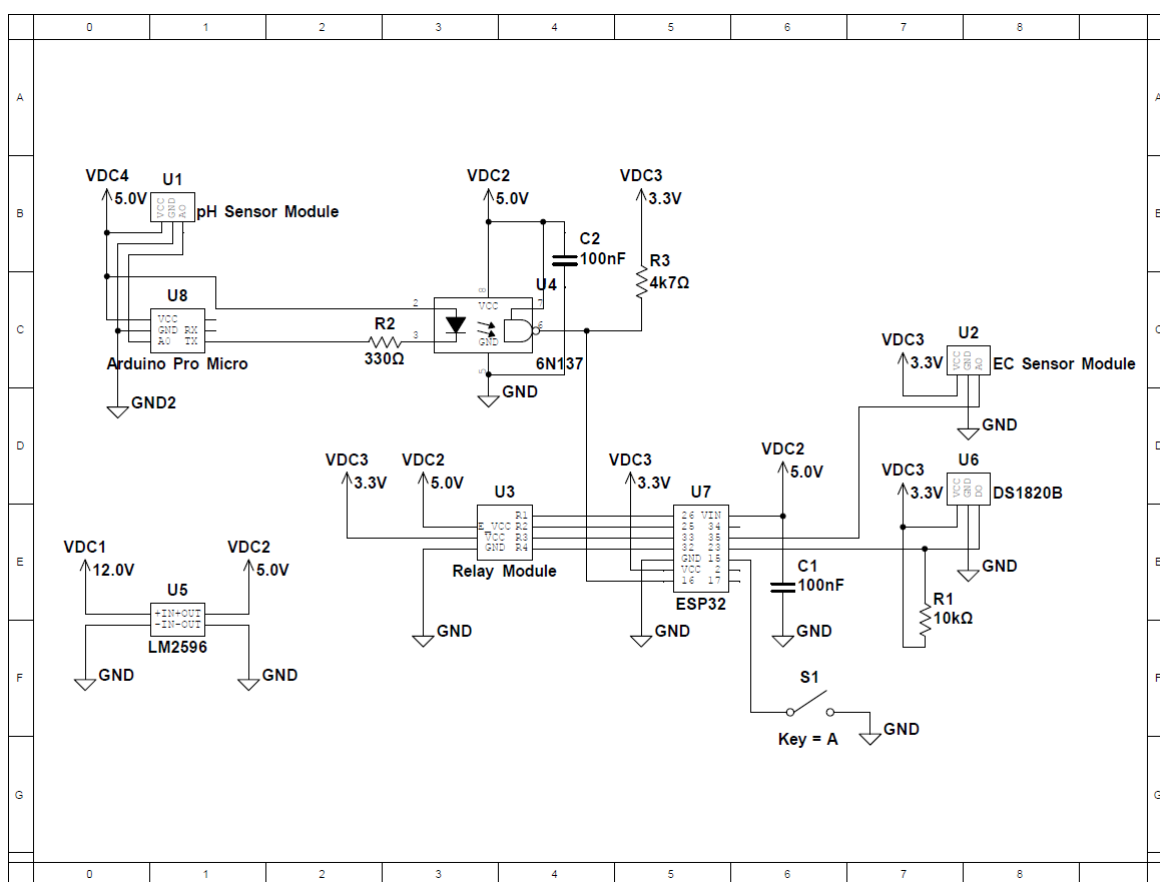


Figura 9: Esquemático do circuito: versão final

Fonte: Autoria própria

Para isolar os sensores um do outro utilizamos um microcontrolador adicional, o Arduino Pro Micro. Este ficou encarregado apenas de fazer as leituras do sensor de pH e enviá-las para o ESP32 via comunicação serial (UART). O Arduino e o sensor de pH foram completamente isolados do ESP, sendo alimentados por uma fonte separada de 5V. A comunicação entre eles também foi isolada galvanicamente através de um optoacoplador de alta velocidade, o 6N137.

O esquemático da parte de potência do circuito pode ser visualizado a seguir. Como estamos utilizando um módulo de relés comprado pronto, não fizemos a representação da sua parte lógica.

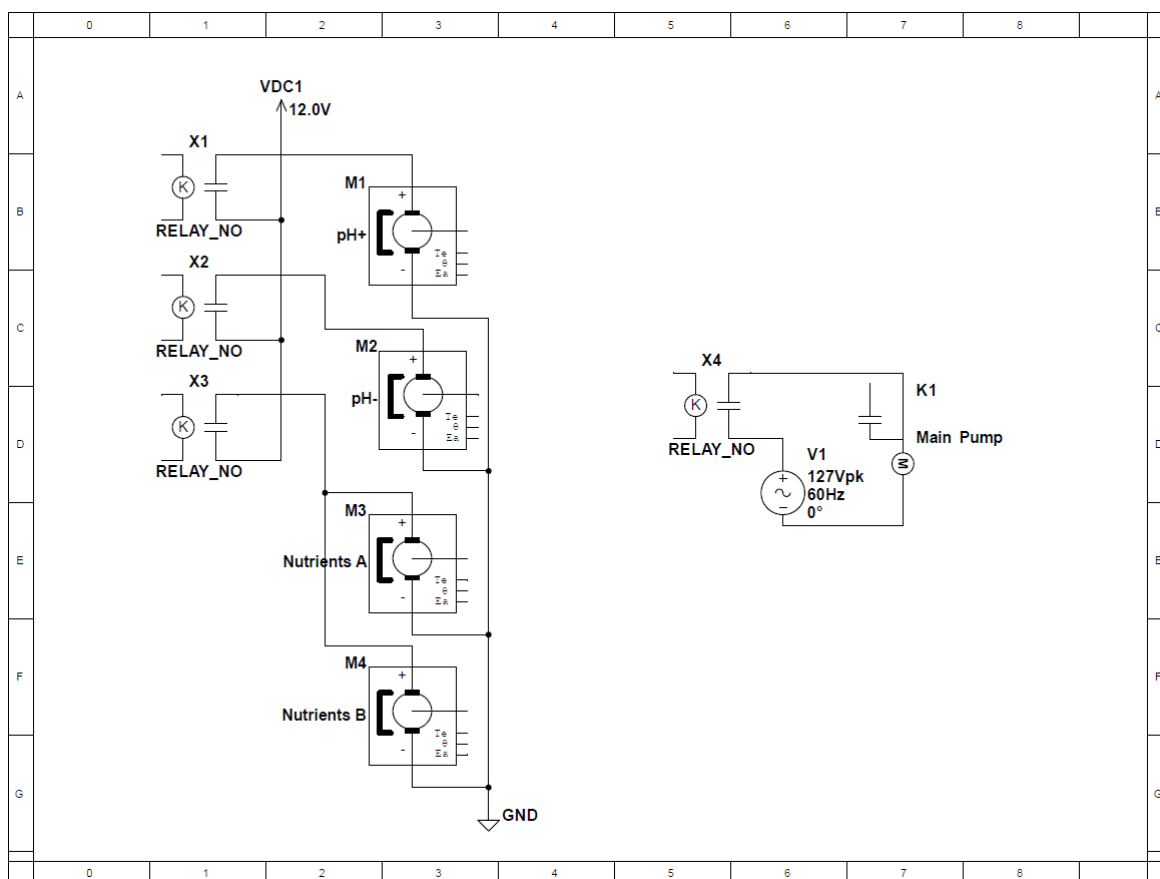


Figura 10: Esquemático do circuito: bombas hidráulicas

Fonte: Autoria própria

Temos então um conjunto de quatro relés: um aciona a bomba peristáltica que insere a solução redutora de pH, um aciona a bomba que insere a solução aumentadora de pH, um aciona a bomba principal que faz a solução nutritiva circular pelo sistema e o último aciona em paralelo as duas bombas de nutrientes, já que os dois nutrientes devem ser adicionados sempre ao mesmo tempo e nas mesmas quantidades. É importante notar que os nutrientes da marca *PLANTPAR*, que utilizamos, são distribuídos em dois pacotes distintos e que, por recomendação do fabricante, a solução nutritiva preparada com estes deve ser mantida em recipientes separados para prevenir a precipitação de cálcio. A bomba principal é alimentada por uma tensão alternada de 127V, enquanto as bombas peristálticas são alimentadas por uma tensão contínua de 12V.

O comportamento do *firmware* desenvolvido está representado no diagrama de transição de estados (DTE) na figura 11

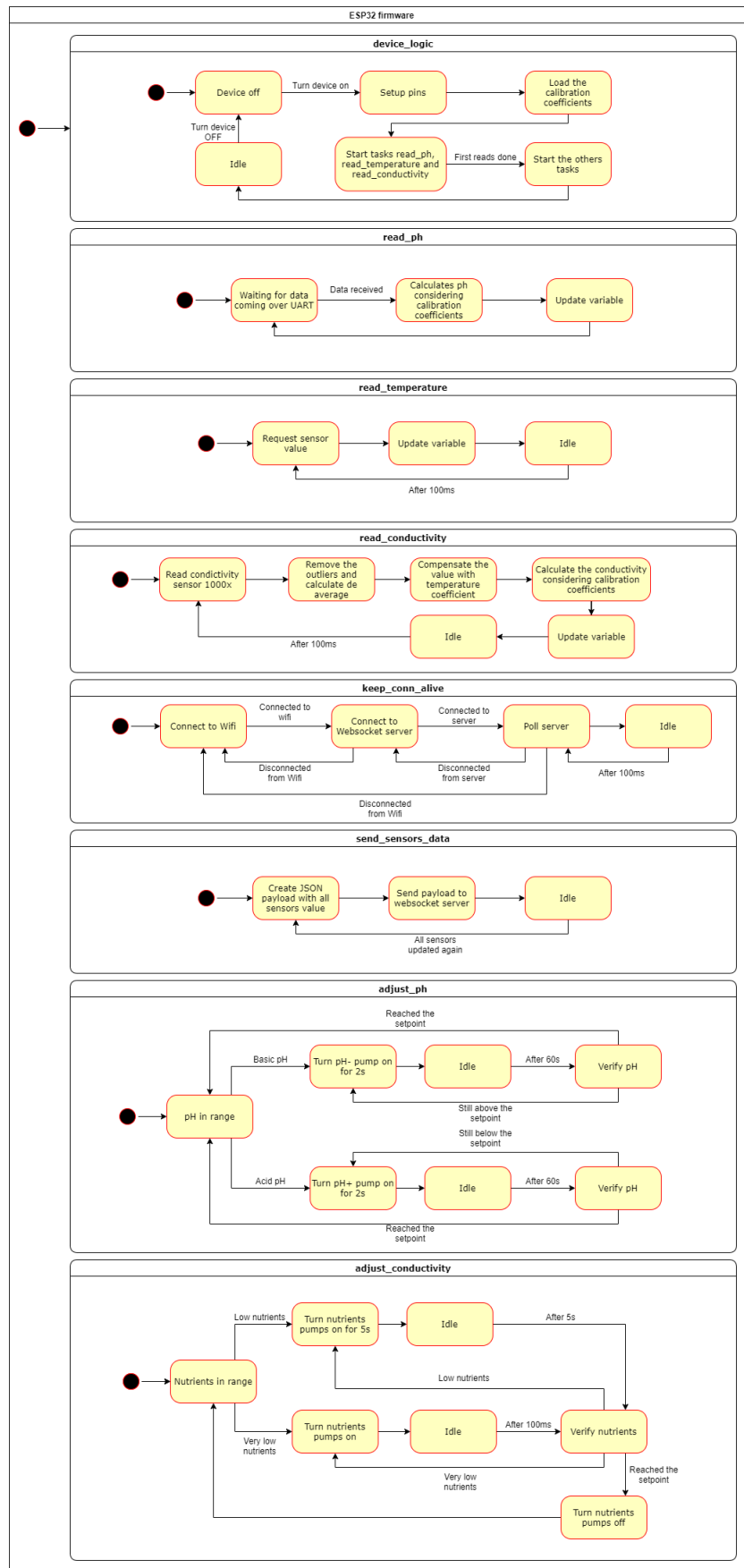


Figura 11: Diagrama de Transição de Estados do *Firmware*

Fonte: Autoria própria

A codificação do *firmware* foi feita em linguagem C/C++, utilizando o *framework* Arduino e empregando o sistema operacional de tempo real RTOS. Foram definidas algumas tarefas, para fazer as leituras dos sensores e filtrar os ruídos dos sinais analógicos, manter a conexão com o *Wi-Fi* e o servidor ativos, tratar os comandos recebidos do servidor, fazer o envio das leituras via *WebSockets* e realizar os ajustes de *pH* e nutrientes na solução.

Existem quatro possíveis comandos que podem ser recebidos do servidor: um comando para ligar ou desligar a bomba principal, um para ligar ou desligar os ajustes na solução nutritiva, um para receber e salvar no sistema de arquivos os coeficientes de calibração calculados para o sensor de *pH* e outro para o coeficiente do sensor de condutividade.

3.4 PROJETO DE SOFTWARE

O projeto de *software* pode ser dividido em 2 componentes principais: Servidor e Interface Web. Foram utilizados *frameworks* em *JavaScript* a fim de facilitar o desenvolvimento: *NodeJS* (FOUNDATION, 2021) para o *back-end* e *VueJS* para a interface web. O diagrama de blocos, figura 12 demonstra melhor a comunicação

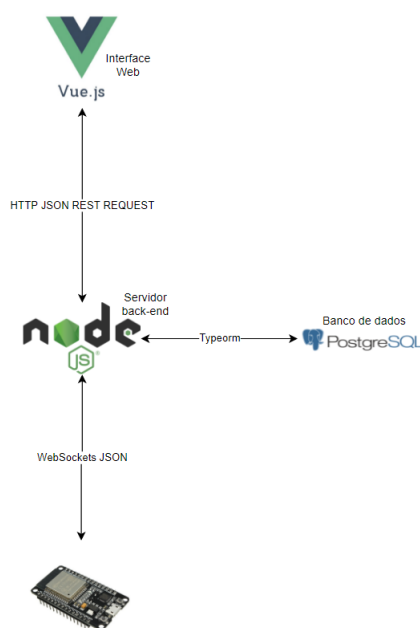


Figura 12: Diagrama de blocos de *software*.

Fonte: Autoria própria

3.4.1 MODELAGEM

A partir dos diagramas de casos de uso e sequência foi possível codificar os serviços do servidor de forma que abrangesse toda a aplicação.

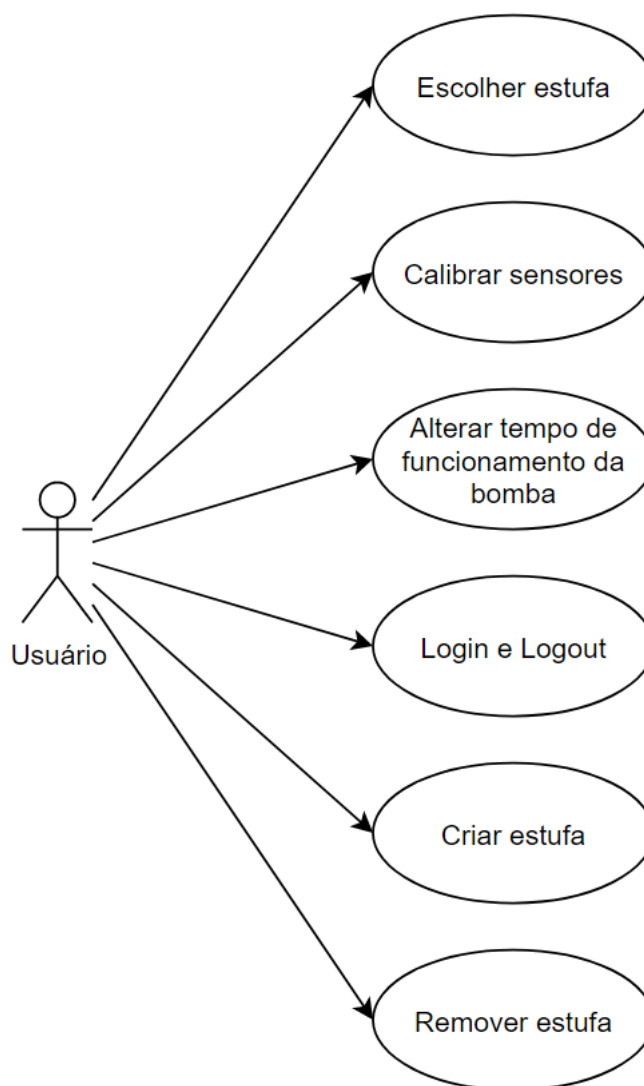


Figura 13: Diagrama de casos de uso.

Fonte: Autoria própria

Tendo em mente apenas um usuário final, o diagrama de casos de uso - Figura 13 - foi pensado com base nas funcionalidades mais necessárias, sem necessidade de aumentar a complexidade do portal.

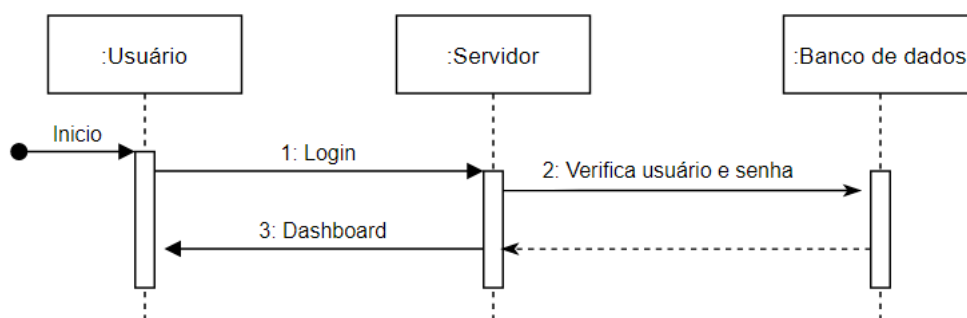


Figura 14: Diagrama de sequencia para login.

Fonte: Autoria própria

Ao fazer login, o servidor compara o usuário e senha com as informações armazenadas no banco de dados e libera o acesso à pagina principal. Vide figura 14

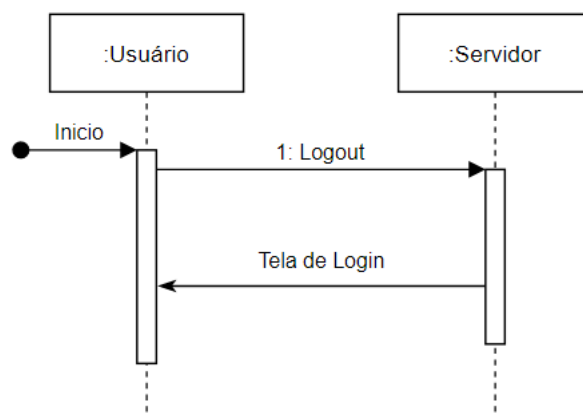


Figura 15: Diagrama de sequencia para logout.

Fonte: Autoria própria

Diferente do login, para realizar o logout o servidor apaga o *token* armazenado ao adentrar na aplicação e retorna o usuário à tela de login, impedindo seu acesso às telas principais. Vide figura 15

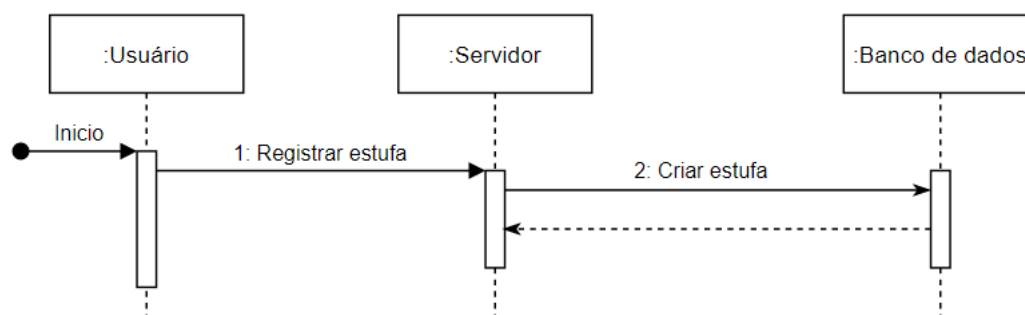


Figura 16: Diagrama de sequencia para criar estufa.

Fonte: Autoria própria

Ao dar início à aplicação, o usuário deve realizar o cadastro de uma estufa, a qual será armazenada no banco de dados e passará a armazenar os dados dos sensores. Vide figura 16

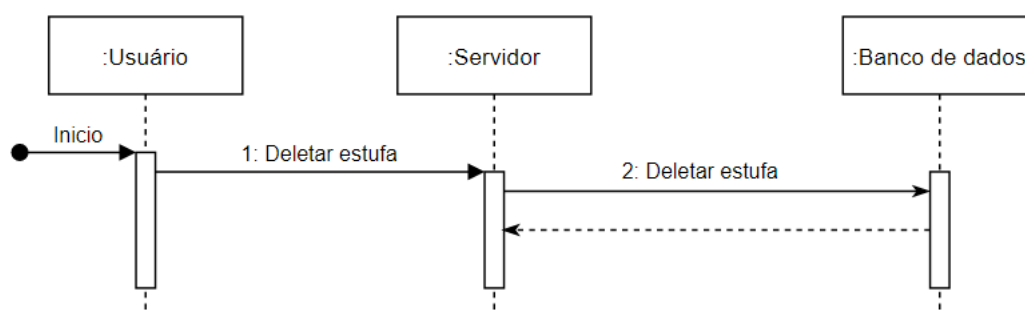


Figura 17: Diagrama de sequencia para deletar estufa.

Fonte: Autoria própria

Similar a criação de uma estufa, a ação de deletar segue a mesma linha de raciocínio, porém removendo o registro do banco de dados. Vide figura 17

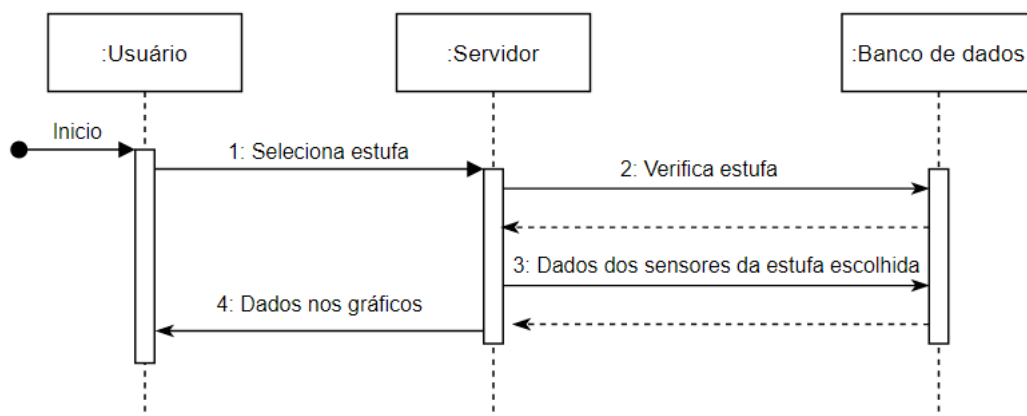


Figura 18: Diagrama de sequencia para selecionar estufa.

Fonte: Autoria própria

Como múltiplas estufas podem ser cadastradas, na tela principal deve ocorrer a seleção da estufa cujos dados serão apresentados. Ao selecionar a estufa, o servidor verifica sua existência e retorna os cem últimos dados das leituras dos sensores da estufa correspondente. Vide figura 18

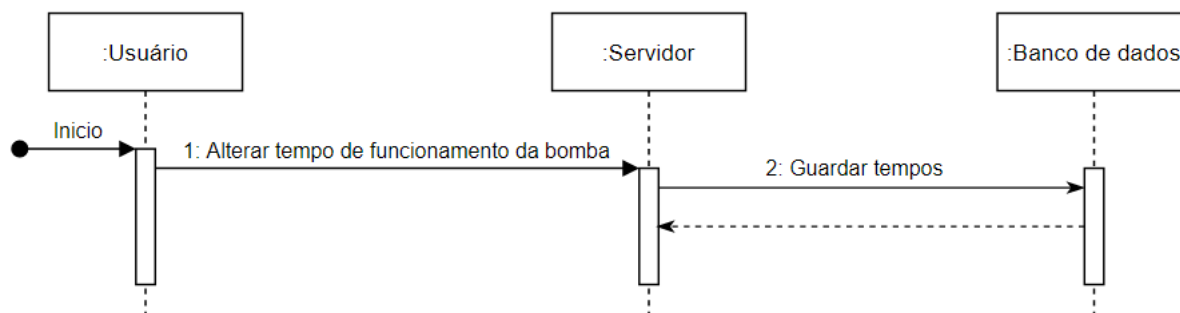


Figura 19: Diagrama de sequencia para alterar tempos de funcionamento da bomba principal.

Fonte: Autoria própria

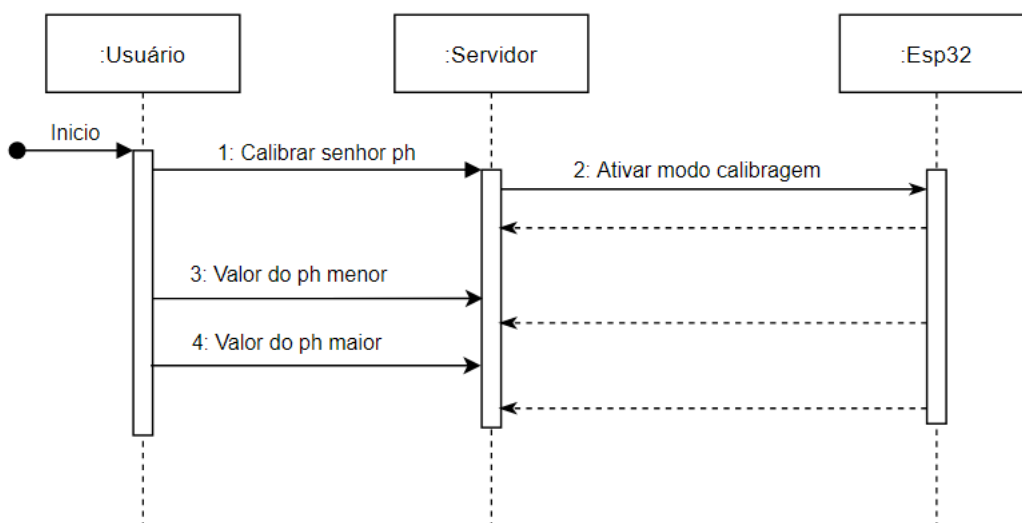


Figura 20: Diagrama de sequencia para calibrar sensores.

Fonte: Autoria própria

A calibragem dos sensores é de suma importância para realizar a regulagem dos nutrientes e *pH* de forma precisa. Ao observar o diagrama de sequência, percebe-se que o servidor envia uma mensagem para o *ESP32* comunicando-o do início da calibragem. Isso se deve ao fato de que o microcontrolador não deve regular a solução nutritiva com base nas próximas leituras - que serão as de calibragem. O servidor então recebe constantemente as leituras do *ESP32* e espera as últimas leituras se demonstrarem estáveis. Vide figura 20 Ao realizar isso para as duas soluções de referência de *pH*, o servidor termina a calibragem e realiza o cálculo dos novos coeficientes.

Outro fator importante foi a modelagem do banco de dados, feita a partir de um diagrama de Entidade-Relacionamento - figura 21. Foi criado junto ao *back-end* utilizando a biblioteca *typeorm*.

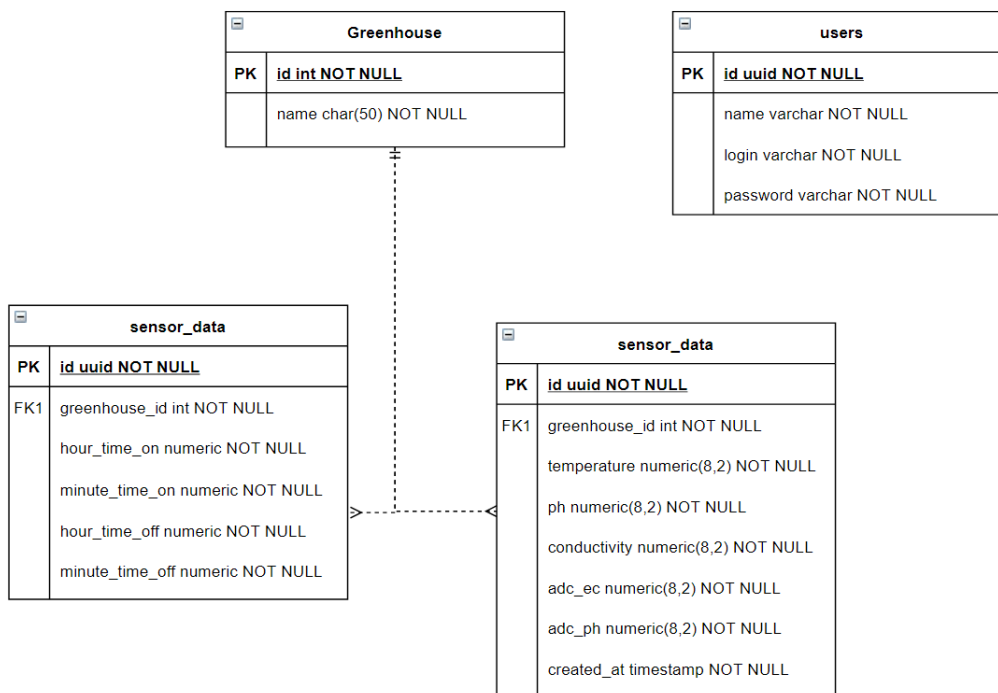


Figura 21: Diagrama de Entidade-Relacionamento.

Fonte: Autoria própria

3.4.2 INTERFACE

Pensando na usabilidade do usuário, a interface criada é limpa e descritiva. Garantimos que todos os parâmetros necessário para se monitorar a estufa fossem apresentados na *Dashboard* e que as funcionalidades mais relevantes, relacionadas à calibragem e ajuste de horários, estivessem presentes na tela de *Calibragem*. A interface foi codificada em *JavaScript* por meio do *Vue.js*, o que possibilitou sua rápida criação, pois utilizamos um *framework* para *Vue.js* chamado *Vuetify* que possui diversos componentes prontos. O *front-end* foi desenvolvido utilizando-se da ferramenta *Vue CLI* (VUECLI, 2021), que permite configurar rapidamente o *Vue ecosystem* e facilita a instalação de *plugins* e bibliotecas. A figura 22 exhibe o *layout* da interface *Web*.

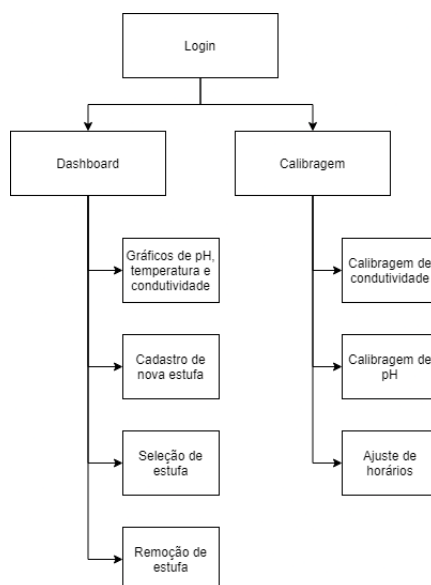


Figura 22: Sitemap da Interface Web

Fonte: Autoria própria

3.4.3 INTEGRAÇÃO

A partir do conhecimento acumulado de várias disciplinas, foi possível realizar uma integração de forma coesa.

A comunicação entre o *back-end* e *ESP32* foi estabelecida por meio *WebSockets* com mensagens do tipo *JSON*. O microcontrolador manda mensagens periodicamente em intervalos de 1,5 segundos, contendo os dados crus e tratados lidos pelos sensores de: *ph*, condutividade e temperatura. Esses dados são armazenados no banco de dados e a partir de uma requisição do *front-end* são dispostos em tela. Como o projeto realiza várias tarefas, como calibragem dos sensores por exemplo, a comunicação deve ser retroativa, ou seja, dependendo das ações tomadas no portal, o *ESP32* deve deligar ou ligar a bomba de aquário e as bombas peristálticas. Com isso em mente, o padrão abaixo para os objetos *JSON* foi estabelecido:

Esp32 enviado ao servidor os dados lidos.

```

1 {
2   "estufa_id": 1,
3   "temperature": 19.5,
4   "conductivity": 1413,

```

```

5     "ph": 6.02,
6     "adc_ph": 584,
7     "adc_ec": 1581
8 }

```

Em que *adc_ph* e *adc_ec* são os valores não tratados das leituras dos sensores, usados posteriormente para calibragem.

Já na comunicação retroativa, foram estabelecidos os seguintes comandos:

Ativação de desativação das bomba peristálticas.

```

1 {
2     "type": "adj",
3     "cmd": "on"
4 }

```

```

1 {
2     "type": "adj",
3     "cmd": "off"
4 }

```

Ativação de desativação da bomba principal.

```

1 {
2     "type": "pump",
3     "cmd": "on"
4 }

```

```

1 {
2     "type": "pump",
3     "cmd": "off"
4 }

```

Dados necessários para calibrar os sensores.

```

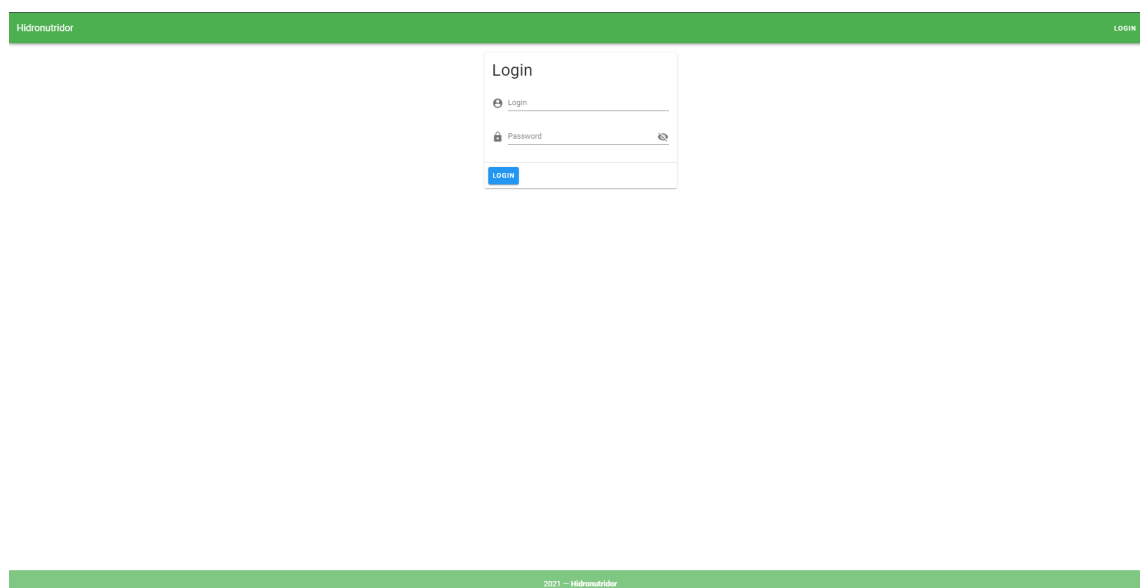
1 {
2     "type": "coef_ph",
3     "a": -0.229,
4     "b": 22.1592

```

```
5 }  
  
1 {  
2   "type": "coef_ec",  
3   "a": 0.83  
4 }
```


4 EXPERIMENTOS E RESULTADOS

O portal *Web* que projetamos é consistido de 3 páginas. Ao acessar o *website*, o usuário é confrontado com uma tela de *login* - vide figura 25. As informações de *login* são providenciadas ao usuário de forma *hardcoded*.



The screenshot shows a web browser displaying the login page of a website named 'Hidronutridor'. The page layout includes a green header bar at the top with the site name on the left and 'LOGIN' on the right. The main body is white and features a central login form. This form is titled 'Login' and contains two input fields: one for the username (preceded by a user icon) and one for the password (preceded by a lock icon and followed by a toggle icon). Below these fields is a blue button labeled 'login'. At the bottom of the page, there is a green footer bar with the text '2021 - Hidronutridor'.

Figura 23: Tela de Login

Fonte: Autoria própria

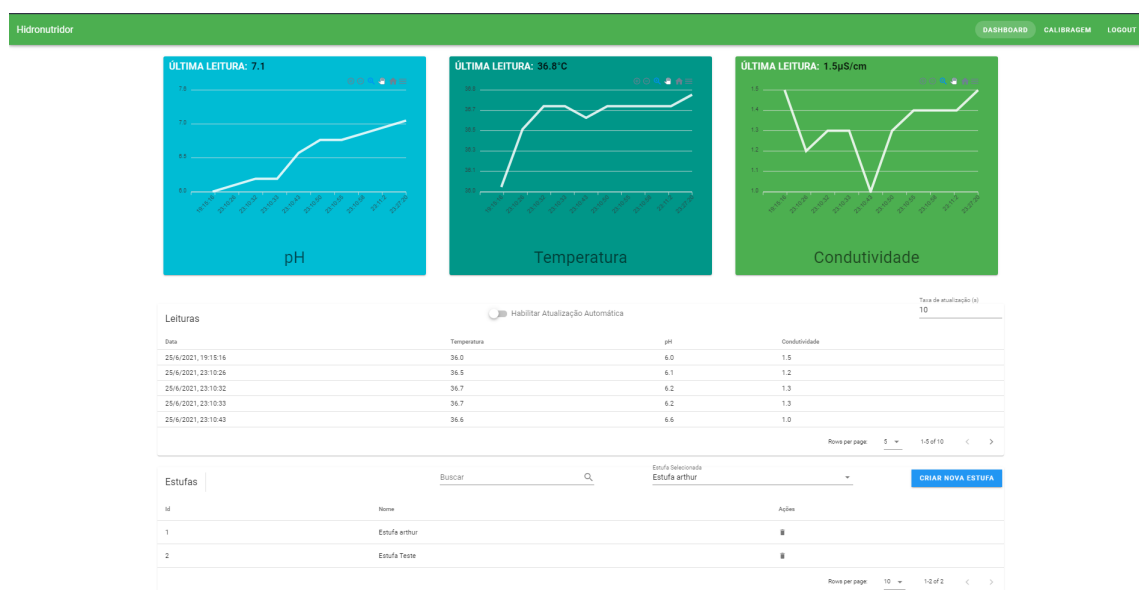


Figura 24: Dashboard

Fonte: Autoria própria

Após realizar o *login* o usuário é direcionado à uma *dashboard* - figura 24. Depois de selecionar uma estufa, são apresentados na *dashboard* gráficos das leituras dos sensores em função do tempo. Ainda nesta tela, estão presentes duas tabelas que contém, respectivamente, as leituras dos sensores e a lista de estufas cadastradas no sistema. O usuário pode optar por criar e remover uma estufa e, caso habilite a atualização em tempo real, controlar a taxa de atualização dos gráficos.

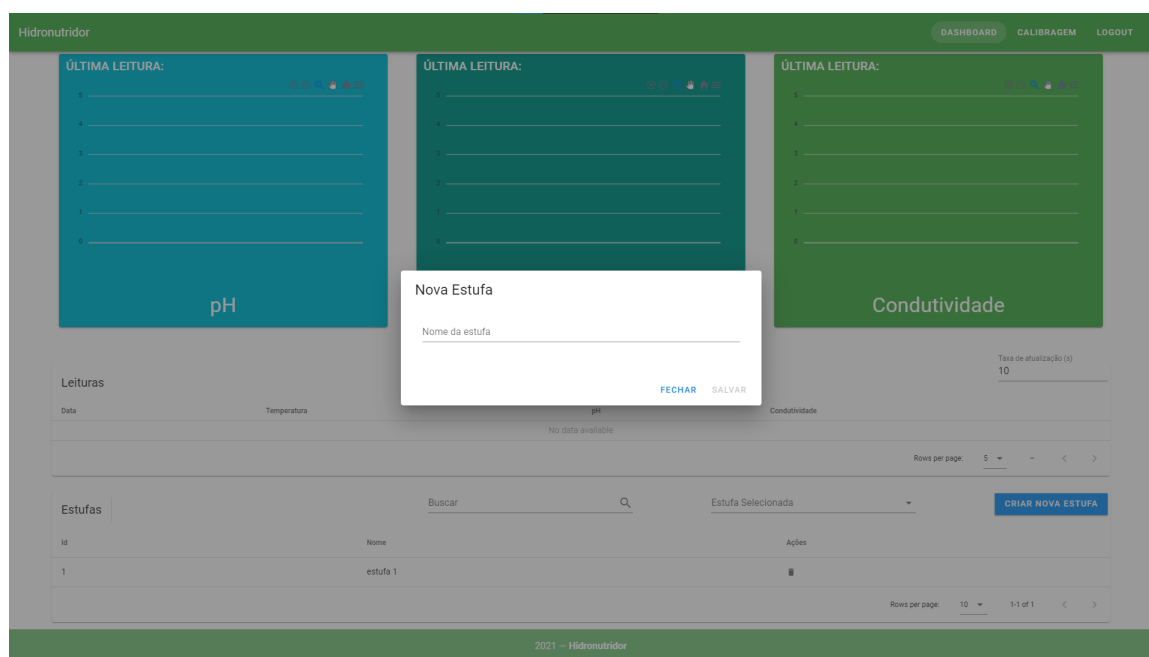


Figura 25: Diálogo de nova estufa

Fonte: Autoria própria

O usuário também tem acesso a uma tela de calibragem de sensores e configuração de horários - vide figuras 27 e 28. Nesta tela o usuário pode visualizar as leituras dos sensores em tempo real, calibrar os valores de referência dos sensores de *pH* e condutividade e configurar os horários de ativação da bomba principal da estufa selecionada.

Horário de Ativação	Horário de Desativação	Ações
06:30	07:30	

Figura 26: Tela de calibragem

Fonte: Autoria própria

A calibragem de *pH* ocorre em dois passos. Quando o usuário clica em calibrar, um diálogo de calibragem é apresentado informando-o para que insira o sensor de *pH* na solução de referência de *pH* ácido. Após confirmar a inserção, o usuário deve informar o valor de referência da solução e clicar em salvar. Assim que o botão de salvar é pressionado, o *back-end* recebe o valor informado e começa a verificar as leituras do sensor. A partir do momento que as leituras estabilizam, o *back-end* calcula a média das 5 últimas leituras e emite uma resposta de confirmação ao *front-end*, permitindo que o usuário prossiga para o próximo passo. Ao clicar em continuar, de forma semelhante ao processo de calibragem do *pH* ácido, o usuário primeiramente insere o sensor na solução tampão básica, informa o valor de referência da solução e espera a estabilização das leituras. Quando o *back-end* terminar de computar as médias das leituras, ele realizará o cálculo dos coeficientes de calibragem - apresentados na seção de fundamentação teórica deste relatório - e os enviará para o microcontrolador, concluindo o processo.

A calibragem do sensor de condutividade ocorre em apenas um passo. Ao clicar no botão de editar do campo de calibragem de condutividade, um diálogo informa ao usuário para que insira o sensor de condutividade na solução de referência. Em seguida, o usuário deve informar o valor de referência da solução.

Ao clicar em salvar, o valor informado é enviado ao *back-end*, que aguarda a estabilização das leituras e calcula o coeficiente de ajuste.

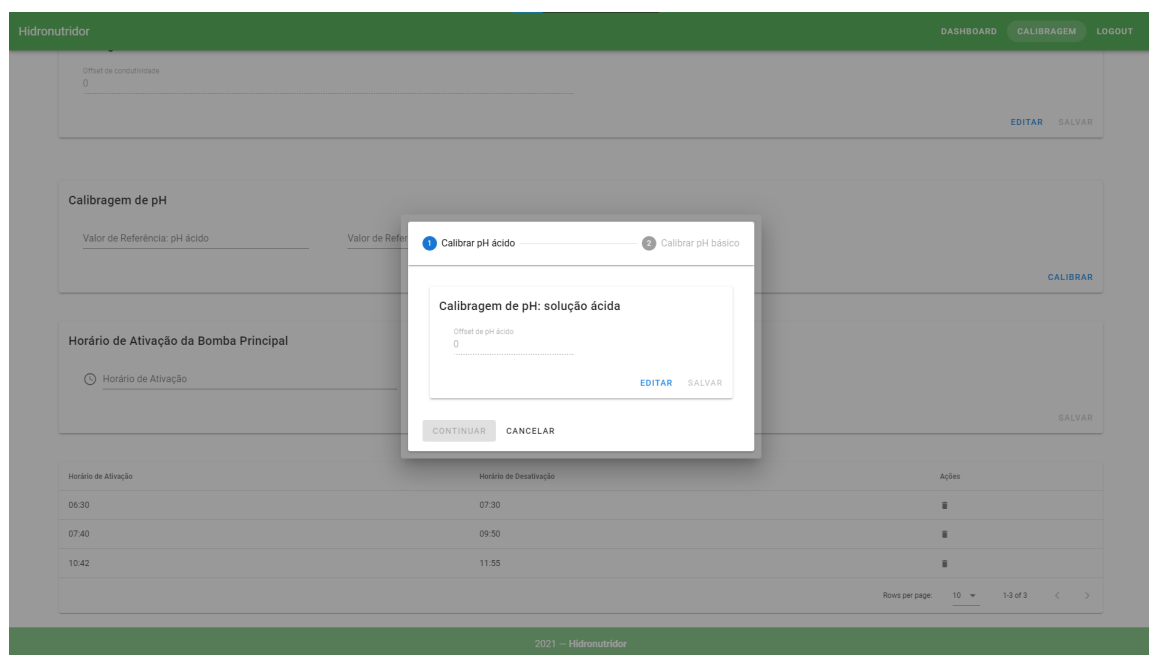


Figura 27: Diálogo de calibragem de pH

Fonte: Autoria própria

Como mencionado acima, ainda na tela de calibração é oferecida ao usuário a possibilidade de estabelecer o horário de funcionamento da bomba principal da estufa:

Hidronutridor DASHBOARD CALIBRAGEM LOGOUT

Offset de condutividade
0
EDITAR SALVAR

Calibragem de pH
Valor de Referência: pH ácido Valor de Referência: pH básico
CALIBRAR

Horário de Ativação da Bomba Principal
Horário de Ativação Horário de Desativação
SALVAR

Horário de Desativação	Ações
07:30	
09:50	
11:55	

Rows per page: 10 1-3 of 3

2021 - Hidronutridor

Figura 28: Opção de ajuste de horário de funcionamento da bomba principal

Fonte: Autoria própria

Nossa interface Web e o sistema como um todo demonstraram funcionamento satisfatório. Todas as funcionalidades foram testadas e executaram de forma esperada após efetuarmos algumas correções.

Os testes de calibragem de condutividade e *pH* demonstraram que as leituras dos sensores estão sendo apresentadas corretamente. Os dados são transmitidos pelo *ESP32* de forma confiável ao *back-end* a cada segundo e os coeficientes calculados são condizentes.



Figura 29: Calibragem de pH : solução ácida

Fonte: Autoria própria



Figura 30: Calibragem de condutividade

Fonte: Autoria própria

Para os testes de regulagem de concentração de nutrientes, inicialmente

enchemos o reservatório principal com água limpa e preparamos, nos reservatórios auxiliares, as soluções com o nutrientes. O microcontrolador, ao receber as leituras de *pH*, detectou a necessidade de se corrigir a acidez da água efetuou a ativação da bomba peristáltica que administra o a solução ácida. Em seguida, corrigido o *pH*, o dispositivo iniciou a ativação das bombas conectadas às soluções com os nutrientes. Foi possível acompanhar na interface *Web* a evolução ao longo do tempo das leituras de condutividade e *pH*. Quando o valor de *pH* medido igualou-se a 6, a bomba da solução ácida foi desativada. À medida que os valores de condutividade medidos aproximaram-se de $900\mu S/cm$, a intensidade de ativação das bombas reduziu, como esperado, para que os nutrientes não fossem adicionados em excesso. Ao atingir a marca de $1400\mu S/cm$, as bombas dos nutrientes foram desativadas. Durante todo o processo, as leituras do sensor permaneceram relativamente constantes em torno dos $18^{\circ}C$.

A propósito de testes, deixamos a estufa funcionando ao longo de 48 horas seguidas. Inicialmente, detectamos um comportamento indesejável em relação ao horário de funcionamento programado para bombas e, posteriormente, verificamos que tal comportamento era causado por um erro nas estruturas condicionais utilizadas pelo *back-end* para a emissão dos comandos de ativação. Depois de aplicarmos correções, verificamos que a bomba hidráulica foi ligada e desligada nos horários programados e que os parâmetros da solução nutritiva permaneceram constantes, comprovando o funcionamento do dispositivo.

5 CRONOGRAMA E CUSTOS DO PROJETO

5.1 CRONOGRAMA

A distribuição das horas do projeto foi pensada a partir dos pontos fortes de cada integrante: Felipe no *back-end* e banco de dados, Arthur no *front-end* e André no *hardware* e *firmware*. Desta forma, a equipe dividiu as tarefas conforme entregáveis semanais. Às horas previstas para cada atividade, foi acrescentada uma margem de erro de 30% a mais sobre cada atividade. Com isso em mente, o projeto foi estimado em 268 horas e realizado em 266.

A partir da tabela abaixo pode-se perceber que a divisão entre os membros ocorreu de forma coesa.

Membro	Horas planejadas	Horas gastas
André	80	79
Arthur	98	107
Felipe	89	79

Tabela 1: Horas gastas por membro

Na figura 31 estão as divisões por entregáveis de forma explícita. E para entender as datas respectivas de cada entrega foi feito um diagrama de Gantt dividido dia a dia, vide figura 32.

5.2 CUSTOS

A fim de diminuir os gastos, a maioria dos itens foi adquirida pelo mercado livre. Listamo-os na tabela 2. Por mais que o custo final seja elevado, os itens se mostraram necessários para a conclusão do projeto.

Area	Activity / Deliverable	Responsible	Assistant	Status	Duration	Error margin	Total	Data Início	Data Fim	Time Taken
Project	Project Inception	All		Done				6/23/2021	6/25/2021	
Project	Project specification	All		Done				6/26/2021	6/30/2021	
Project	Make Requirements Definition	André		Done	2	0,6	2,6	04/07/2021	04/07/2021	3
Project	Make Schedule	Felipe		Done	3	0,9	3,9	03/07/2021	04/07/2021	4
Project	Make Budget	André		Done	2	0,6	2,6	03/07/2021	04/07/2021	1
Project	Write Project charter	Arthur	Felipe	Done	4	1,2	5,2	02/07/2021	04/07/2021	3
Project	Prepare project charter presentation	Felipe		Done	1,5	0,45	1,95	04/07/2021	05/07/2021	2
Entrega Plano do Projeto										
Blog	Set up blog	Arthur		Done	3	0,9	3,9	05/07/2021	06/07/2021	1,5
Blog	Write blog posts	Arthur		Done	2	0,6	2,6	06/07/2021	09/07/2021	2
Project	Purchase and Gather Materials	André		Done	3	0,9	3,9	01/07/2021	05/07/2021	2
Server/Web	Set up programming space and tools	Felipe	Arthur	Done	3	0,9	3,9	05/07/2021	08/07/2021	3,5
Server/Web	Create web model	Arthur		Done	3	0,9	3,9	06/07/2021	09/07/2021	4
Entregável 1: Site/Blog de acompanhamento										
Hardware	Assemble peristaltic pump to reservoirs	André		Done	2	0,6	2,6	11/07/2021	12/07/2021	2
Hardware	Assemble main pump to bucket	André	Arthur	Done	2	0,6	2,6	13/07/2021	14/07/2021	2
Hardware	Assemble sensors to bucket	André	Felipe	Done	1	0,3	1,3	13/07/2021	14/07/2021	1
Mockup	Create mockup with pvc pipes	André	Arthur	Done	10	3	13	12/07/2021	17/07/2021	8
Mockup	Put real seeds to grow on mockups	André		Done	1	0,3	1,3	17/07/2021	20/07/2021	0
Server/Web	Learn Vue.js	Arthur		Done	6	1,8	7,8	06/07/2021	09/07/2021	10
Blog	Update blog	Arthur		Done	1	0,3	1,3	06/07/2021	09/07/2021	1
Entregável 2: Projeto e montagem da estrutura mecânica										
Firmware	control actuators	André		Done	4	1,2	5,2	17/07/2021	20/07/2021	3,5
Hardware	Test sensors	André		Done	2	0,6	2,6	17/07/2021	20/07/2021	1
Hardware	Test Water Pump and peristaltic pump	André		Done	2	0,6	2,6	22/07/2021	24/07/2021	1
Hardware	Test main pump waterflow from bucket	André		Done	2	0,6	2,6	22/07/2021	24/07/2021	2
Hardware	Test main pump with microcontroller	André		Done	2	0,6	2,6	24/07/2021	26/07/2021	2
Hardware	Test peristaltic pump with microcontroller	André	Felipe	Done	2	0,6	2,6	24/07/2021	26/07/2021	2
Hardware	Calibrate sensors	André		Done	3	0,9	3,9	24/07/2021	26/07/2021	3,5
Hardware	Circuit assembly on protoboard	André		Done	5	1,5	6,5	24/07/2021	26/07/2021	4
Blog	Update blog	Arthur		Done	1	0,3	1,3	06/07/2021	09/07/2021	1
Entregável 3: Projeto eletrônico e testes de hardware										
Server/Web	Model server's software	Felipe		Done	4	1,2	5,2	14/07/2021	16/07/2021	3
Firmware	Model firmware software	André		Done	6	1,8	7,8	14/07/2021	16/07/2021	10
Server/Web	Create database, tables and relations	Felipe	Arthur	Done	3	0,9	3,9	14/07/2021	16/07/2021	3
Server/Web	Define API endpoints and returns	Felipe	Arthur	Done	3	0,9	3,9	20/07/2021	22/07/2021	2
Firmware/Server	Define communication protocol	André	Felipe	Done	2	0,6	2,6	20/07/2021	22/07/2021	2
Server/Web	Test web and server communication	Arthur	Felipe	Done	6	1,8	7,8	26/07/2021	27/07/2021	6,5
Server/Web	Create Server communication with database	Felipe		Done	4	1,2	5,2	28/07/2021	29/07/2021	4
Server/Web	Create web login interface	Arthur		Done	5	1,5	6,5	16/07/2021	20/07/2021	6,5
Server/Web	Create web main page	Arthur		Done	8	2,4	10,4	17/07/2021	28/07/2021	21
Server/Web	Create Server micro-services	Felipe		Done	8	2,4	10,4	17/07/2021	28/07/2021	10
Blog	Update blog	Arthur		Done	1	0,3	1,3	06/07/2021	09/07/2021	1
Entregável 4: Projeto do software com UML										
Hardware	Configure microcontroller with localhost	André		Done	4	1,2	5,2	30/07/2021	02/08/2021	5
Server/Web	microcontroller	Felipe	Arthur	Done	12	3,6	15,6	30/07/2021	02/08/2021	8
Firmware	Peristaltic pumps software	André	Arthur	Done	3	0,9	3,9	03/08/2021	07/08/2021	4
Firmware	Main pump software	André	Felipe	Done	2	0,6	2,6	03/08/2021	07/08/2021	2
Firmware	WebSockets communication software	André	Felipe	Done	5	1,5	6,5	03/08/2021	07/08/2021	6
Firmware	WiFi configuration portal software	André		Done	5	1,5	6,5	03/08/2021	07/08/2021	4
Server/Web	Set up server and web on local machine	Felipe	Arthur	Done	4	1,2	5,2	30/07/2021	03/08/2021	3
Firm./Server/Web	and web	All		Done	8	2,4	10,4	30/07/2021	03/08/2021	12
Blog	Update blog	Arthur		Done	1	0,3	1,3	06/07/2021	09/07/2021	1,3
Entregável 5: Integração mecânica + hardware + software										
Server/Web	Database tests	Arthur	Felipe	Done	4	1,2	5,2	04/08/2021	06/08/2021	2
Server/Web	Web interface with real time data test	Arthur	Felipe	Done	4	1,2	5,2	06/08/2021	08/08/2021	4
Evaluation	Final test	All		Done	6	1,8	7,8	08/08/2021	10/08/2021	5
Evaluation	Prepare Final Report	All		Done	6	1,8	7,8	10/08/2021	16/08/2021	8
Blog	Update blog	Arthur		Done	1	0,3	1,3	06/07/2021	09/07/2021	1
Demonstração do funcionamento do protótipo + Relatório Técnico										
Evaluation	Record video	All			6	1,8	7,8	16/08/2021	17/08/2021	
Evaluation	Edit video	Felipe			5	1,5	6,5	18/08/2021	20/08/2021	
Vídeo de 15-20 minutos com a apresentação e demonstração do projeto										
Final Evaluation										
	Total Time				268,5	80,55	349,05			

Figura 31: Cronograma detalhado

Fonte: Autoria própria

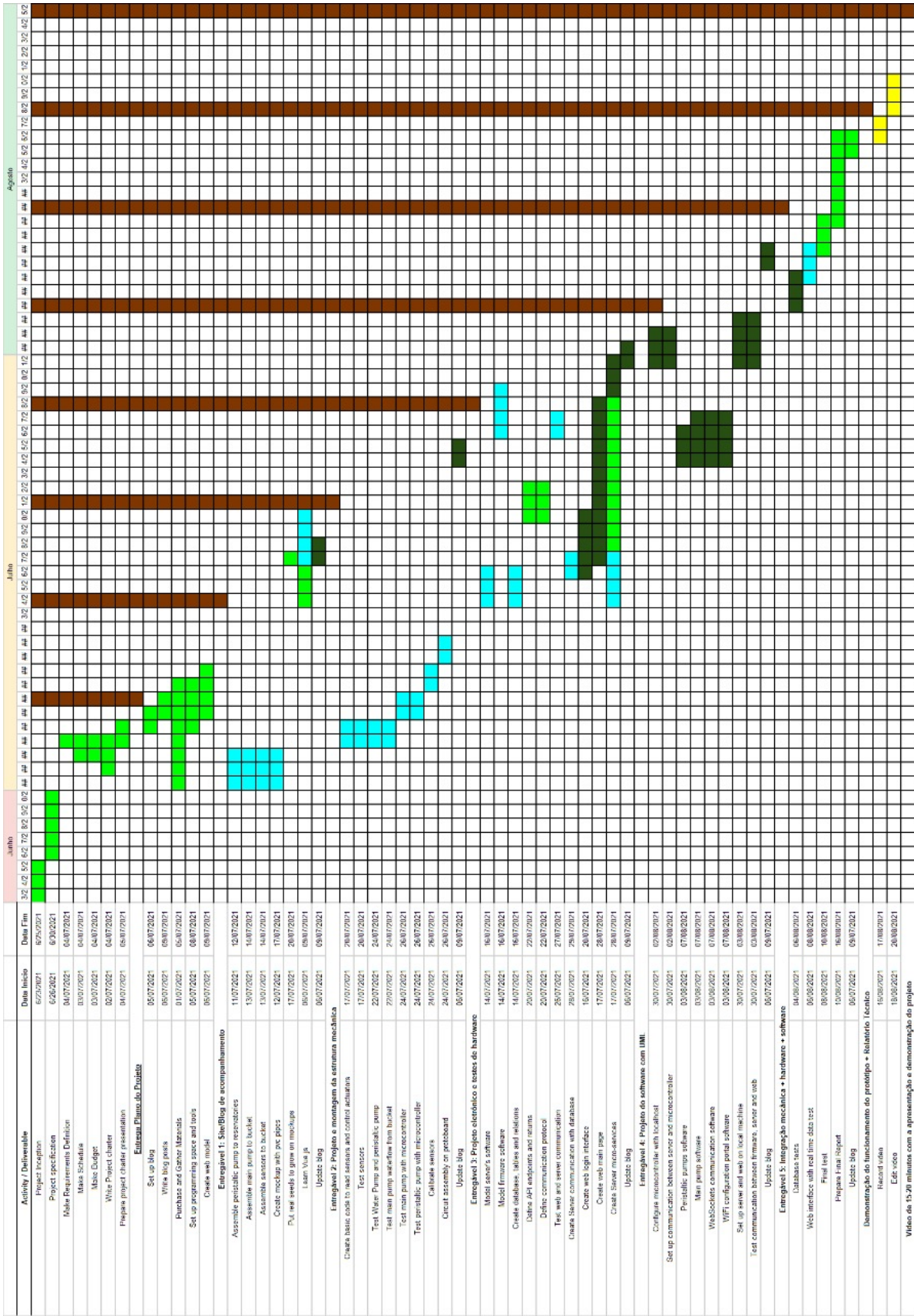


Figura 32: Diagrama de Gantt

Fonte: Autoria própria

Item	Qnt.	Valor Unit.	Valor Total
Sensor de pH + Módulo BNC - PH900	1	238,90	238,90
Sensor de temperatura - DS18B20	1	34,90	34,90
Sensor de condutividade - Keyestudio TDS Meter v1.0	1	119,00	119,00
Canos PVC , cola, mangueira 1/2" , materiais p/ montagem mecânica	1	260,00	260,00
ESP32 - WROOM Dev. KIT	1	56,99	59,99
Bomba dosadora peristáltica 12v - Fortek	4	78,90	315,60
Bomba de aquário - Sarlo SB1000C	1	78,90	78,90
Reservatório de 14L	1	30,00	30,00
Regulador de tensão LM2596	1	15,41	15,41
Módulo relé 4 canais optoacoplado	1	31,90	31,90
Kit 10 injetores de solução nutritiva 4x7mm	1	29,90	29,90
Mangueira 100% silicone 3x5mm - 3m	1	45,00	45,00
Solução condutividade 1413uS/cm - 250mL	1	78,90	78,90
Solução tampão pH 4,0 e 7,0 - 250mL de cada	1	36,90	36,90
Kit solução reguladora de pH - Power Up & Down - 1L de cada	1	105,60	105,60
Kit solução nutritiva, sementes de alfaca, espuma fenólica	1	126,92	126,92
Fonte estabilizada 12V 10A	1	28,80	28,80
TOTAL			R\$1636.62

Tabela 2: Materiais e custos

6 CONCLUSÕES

6.1 CONCLUSÕES

Ao decorrer do projeto foram encontradas várias dificuldades a serem superadas, das quais a mais impactante foi a interferência entre sensores: o sensor de pH e condutividade não podiam realizar leituras ao mesmo tempo. Com isso resolvido, o projeto ocorreu de forma coesa, a equipe se dividiu de forma que as tarefas foram bem distribuídas e o aprendizado entre membros foi compartilhado. Entretanto, mesmo com as tarefas divididas, por medidas de segurança devido à Covid-19, a equipe manteve o distanciamento social e a comunicação foi em boa parte feita de forma assíncrona, o que dificultou algumas integrações, principalmente entre o servidor e o *ESP32*. Após os meses de desenvolvimento, pode-se dizer que o resultado foi satisfatório para a equipe, o *HidroNutridor*TM se mostrou um produto pensado para o mercado, com real aplicação na indústria de cultivo hidropônico.

6.2 TRABALHOS FUTUROS

Pensando na maior adequação do projeto ao mercado, as principais mudanças seriam na redução do valor do produto, pois, embora seja um protótipo, seu custo foi elevado. Porém, com os mesmos materiais utilizados para construção desta estufa hidropônica poderia ser feita uma similar com capacidade maior que a atual.

Outra preocupação da equipe se dá no quesito da escalabilidade do projeto, pois o desenvolvimento se deu em torno de apenas uma estufa, mas a maioria dos cultivadores possuem no mínimo duas. Pensando nisso, a equipe planeja utilizar o módulo *RS 485* com protocolo *Modbus*, que permite a comunicação de várias estufas em grandes distâncias, sem interferência de motores.

Para uma escala mediana, as necessidades do projeto também escalariam consideravelmente, pois são utilizadas bombas mais potentes e reservatórios de mais de mil litros de água. Entretanto, a regulação do pH e soluções nutritivas ainda poderiam se dar por bombas peristálticas.

Para o servidor, a principal mudança seria a alocação em nuvem e replicação local, para preservar os dados caso ocorra uma perda de conexão. Se mostra importante também aumentar o armazenamento para o banco de dados. Já no portal, uma alteração se daria no design para ser mais inclusivo com todas as idades - uma vez que mais de 50% dos produtores rurais brasileiros tem acima 45 anos de idade (DATASEBRAE, 2021) - com tutoriais de cada funcionalidade e opção de fontes maiores.

REFERÊNCIAS

ADAFRUIT. **Two Point Calibration**. 2021. Disponível em: <<https://learn.adafruit.com/calibrating-sensors/two-point-calibration>>. Acesso em: 16 de agosto de 2021.

DATASEBRAE. **Perfil do Produtor Rural**. 2021. Disponível em: <<https://datasebrae.com.br/perfil-do-produtor-rural/>>. Acesso em: 17 de agosto de 2021.

FLORAMAX. **Factors Affecting the EC Value in Hydroponics**. 2021. Disponível em: <<https://www.floramax.com/factors-affecting-the-ec-value/>>. Acesso em: 17 de agosto de 2021.

FOUNDATION, O. **About Node.js**. 2021. Disponível em: <<https://nodejs.org/en/about/>>. Acesso em: 16 de agosto de 2021.

KEYESTUDIO. **KS0429 keyestudio TDS Meter V1.0**. 2021. Disponível em: <https://wiki.keyestudio.com/KS0429_keyestudio_TDS_Meter_V1.0>. Acesso em: 17 de agosto de 2021.

VUECLI. **Vue CLI Overview**. 2021. Disponível em: <<https://cli.vuejs.org/guide/>>. Acesso em: 17 de agosto de 2021.

VUEJS. **What is Vue.js?** 2021. Disponível em: <<https://vuejs.org/v2/guide/index.html>>. Acesso em: 15 de agosto de 2021.

VUETIFY. **What is Vuetify?** 2021. Disponível em: <<https://vuetifyjs.com/en/introduction/why-vuetify/getting-started>>. Acesso em: 15 de agosto de 2021.