

UNIVERSIDADE FEDERAL DE SANTA CATARINA  
CENTRO DE CIÊNCIAS, TECNOLOGIAS e SAÚDE  
DEPARTAMENTO DE COMPUTAÇÃO  
CURSO ENGENHARIA DE COMPUTAÇÃO

Nicolas Beraldo

***Reinforcement learning* aplicado no controle de variáveis de uma casa de vegetação**

Araranguá

2022

Nicolas Beraldo

***Reinforcement learning* aplicado no controle de variáveis de uma casa de vegetação**

Trabalho Conclusão do Curso de Graduação em Engenharia de Computação do Centro de Ciências, Tecnologias e Saúde da Universidade Federal de Santa Catarina como requisito para a obtenção do título de Bacharel em Engenharia de Computação  
Orientador: Prof. Roderval Marcelino, Dr.

Araranguá

2022

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Beraldo, Nicolas

Reinforcement learning aplicado no controle de  
variáveis de uma casa de vegetação / Nicolas Beraldo ;  
orientador, Roderval Marcelino, 2022.

22 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Campus Araranguá,  
Graduação em Engenharia de Computação, Araranguá, 2022.

Inclui referências.

1. Engenharia de Computação. 2. casas de vegetação. 3.  
machine learning. 4. reinforcement learning. 5. internet of  
things. I. Marcelino, Roderval. II. Universidade Federal  
de Santa Catarina. Graduação em Engenharia de Computação.  
III. Título.

Nicolas Beraldo

***Reinforcement learning* aplicado no controle de variáveis de uma casa de vegetação**

Este Trabalho Conclusão de Curso foi julgado adequado para obtenção do Título de Bacharel em Engenharia de Computação e aprovado em sua forma final pelo Curso de Graduação de Engenharia de Computação

Araranguá, 24 de março de 2022.

---

Profa. Analucia Schiaffino Morales, Dra.  
Coordenador do Curso

**Banca Examinadora:**

---

Prof. Roderval Marcelino, Dr.  
Orientador  
Universidade Federal de Santa Catarina

---

Profa. Analucia Schiaffino Morales, Dra.  
Avaliadora  
Universidade Federal de Santa Catarina

---

Prof. Bruno Pansera Espindola, Dr.  
Avaliador  
Instituto Federal Catarinense

# *Reinforcement learning* aplicado no controle de variáveis de uma casa de vegetação

Nicolas Beraldo\*

Roderval Marcelino†

2022, Março

## Resumo

Com a crescente demanda por alimentos e a piora das condições climáticas o uso de casas de vegetação tem se tornado comum, entretanto o controle dessas casas, na grande maioria, tem sido feito por modelos de controladores que necessitam de ajuste manuais, como os clássicos. O objetivo deste projeto é reunir as pesquisas mais recentes sobre o controle das variáveis de um ambiente controlado a fim de criar um protótipo autônomo para gerenciar uma casa de vegetação. A partir de estudos bibliográficos foram encontrados diversos métodos para a otimização do uso dos recursos naturais e artificiais e como aproveitá-los ao máximo para o melhor desenvolvimento dos produtos vegetais. No decorrer deste documento será apontado o que foi descoberto das bibliografias, as suas singularidades, como foram aplicadas e o porquê de serem usadas. Serão descritos os passos do desenvolvimento de uma casa de vegetação doméstica, os algoritmos envolvidos na aquisição e armazenamento de dados, treinamento de algoritmos de tomada de decisões e tomada de ações dentro da casa de vegetação. Para o algoritmo de tomada de decisão foi usado o algoritmo de *reinforcement learning*, que recentemente tem crescido em popularidade dentro da área de *machine learning*, a fim de testar sua capacidade fora dos ambientes comumente usados, onde as variáveis de entrada e saída são definidas com exatidão, já que esse algoritmo normalmente se aplica em jogos eletrônicos, que possuem regras, ações e consequências bem definidas. Ao fim do trabalho estarão detalhados os motivos da implementação de *reinforcement learning*, neste caso, não ter convergido para um modelo funcional, por causa do pouco tempo disponível para treinamento, do mau planejamento da casa de vegetação entre outros, e como os evitar em trabalhos futuros.

**Palavras-chaves:** casa de vegetação; machine learning; reinforcement learning; internet of things.

---

\*nicolas.beraldo@outlook.com

†roderval@yahoo.com.br

# Reinforcement learning applied to controlling variables in a greenhouse

Nicolas Beraldo\*

Roderval Marcelino†

2022, Março

## Abstract

With the growing demand for food and the continuous worsening of climatic conditions, the use of greenhouses has become common, however, the control of these houses, in the vast majority, has been done by controller models that need manual adjustment, such as the classics. The objective of this project is to gather the most recent research on the control of variables in a controlled environment in order to create an autonomous prototype to manage a greenhouse. From bibliographic studies, several methods were found to optimize the use of natural and artificial resources and how to make the most of them for the best development of the fruit. In the course of this document it will be pointed out what was discovered from the bibliographies, their singularities, how they were applied and why they were used. It will also describe the steps of the development of a domestic greenhouse, the algorithms involved in the acquisition and storage of data, training of decision-making algorithms and taking actions inside the greenhouse. For the decision-making algorithm, the reinforcement learning algorithm was used, which has recently grown in popularity within the machine learning area, in order to test its capacity outside commonly used environments, where the input and output variables are precisely defined, since this algorithm is usually applied in electronic games, which have well-defined rules, actions and consequences. At the end of the work, we will have the reasons for the implementation of reinforcement learning, in this case, not having converged to a functional model, which points the author believes are the main points of failure and how to avoid them in other works.

**Key-words:** greenhouse; machine learning; reinforcement learning; internet of things

---

\*nicolas.beraldo@outlook.com

†roderval@yahoo.com.br

## Introdução

A sociedade muito se preocupa com o futuro da humanidade e do planeta terra, assim frequentemente se ouve sobre como melhorar a eficiência do uso dos recursos naturais disponíveis e como diminuir os gastos desnecessários. Também existe a preocupação com a produção de alimentos que se tornarão escassos e limitados, devido ao crescimento exacerbado da população a qual se estima que será de aproximadamente 10 bilhões de pessoas em 2050, se esperando um crescimento de 70% na demanda de alimentos (PARADA et al., 2021). Logo é necessário tomar cuidado tanto com o gasto dos recursos naturais limitados, como também com a crescente demanda de alimentos. Por isso a pesquisa sobre casas de vegetação integradas com as tecnologias 4.0 tem aumentado significativamente com o passar dos anos (TAO et al., 2021; FILHO, 2018), a fim de aumentar as taxas de produção e diminuir os recursos desperdiçados.

Os estudos sobre casas de vegetação têm se espalhado pelos diferentes campos de pesquisa, cada um com focos diferentes, se estendendo desde a biologia das plantas até a física dentro das casas de vegetação. Na área da Engenharia de Computação os estudos são focados na utilização de Internet das Coisas (*do inglês "IoT"*) para monitorar e controlar diversas variáveis das casas de vegetação, fazendo uso de modelos matemáticos preditivos (LIAO et al., 2021; LIN; ZHANG; XIA, 2021; MOSHARAFIAN et al., 2021; PARADA et al., 2021; BONE, 2019; ACHOUR et al., 2020) ou de modelos de *Machine Learning* (ML) para realizar a tomada de decisão sem a necessidade de interações humanas (WANG; ZHANG, 2018; LIU et al., 2020; FEI; XIAO-LONG; YONG, 2021; MOON; Eek Son, 2021). Ao se usar da análise de padrões ou modelos para o controle do ciclo de vida das plantações, é possível diminuir os gastos e perdas de produto por erro ou descaso humano.

As casas de vegetação tem se tornando mais populares, tão qual atualmente se estima que ocupem 3,64 milhões de hectares de terra (LIN; ZHANG; XIA, 2021). Por se tratar de um ambiente fechado e controlado está começando a popularização de casas de vegetação urbanas, com a intenção de diminuir o trajeto necessário para o transporte dos alimentos, onde se estima que haja uma perda de 30% dos produtos (IDDIO et al., 2020). Além disso há a ocorrência de pragas e doenças que são responsáveis por 20% a 40% das perdas durante o cultivo (AKHTER; SOFI, 2021). Assim o uso de casas de vegetação tem se popularizado mais por conseguirem mitigar essas perdas e em paralelo melhorar a qualidade e quantidade dos produtos, enquanto diminuiu a utilização de recursos naturais e diminui os custos de operação devido ao menor desperdício.

Neste trabalho foi desenvolvido uma casa de vegetação com o mínimo de interações humanas necessária, para o controle de diversas variáveis como umidade do solo e ar, temperatura, radiação solar, entre outras. Para o controle dessas variáveis será feito o uso de técnicas de ML populares para otimizar o uso dos recursos e o ciclo de vida da planta, a fim de obter o melhor produto com o menor gasto e em maior quantidade. Além disso se buscou criar uma rede IoT de sensores e que as análises decisórias sejam feitas com computação local de um processador ARM (*Acorn RISC Machine*), neste caso um *Raspberry Pi 4* que será responsável por realizar o armazenamento dos dados coletados, e o pré-processamento e a análise do modelo de ML implementado. Os sensores e atuadores serão conectados aos *ESPs 32* com módulos para a coleta de dados e ativação de relés, se aproveitando do baixo consumo de energia dessa tecnologia e da conectividade Wi-Fi já implementada nos seus controladores.

## 1 Estado da Arte

Com base na literatura encontrada no **Science Direct** a busca por trabalhos na área de casas de vegetação foi feita. Usando a palavra-chave: “*greenhouse*” foram encontrados aproximadamente 300.000 resultados, para diminuir o escopo da busca para um número plausível foi usado a ferramenta de busca avançada para poder limitar os resultados. Assim a busca para “*greenhouse AND IoT AND machine learning*” retornou 690 resultados, para tornar este trabalho o mais recente foi utilizado o filtro de anos, nos limitando a 270 artigos, foram selecionados 10 trabalhos do ano de 2021 que incorporem o monitoramento e controle de variáveis das casas de vegetação através de análises e processamentos para tomada de decisões.

Também foi feita a busca por “*greenhouse AND X*” onde “X” era substituído por diferentes variáveis a serem controladas, como esperado cada busca retornou uma grande quantidade de resultados mesmo utilizando o filtro por ano, assim foram lidos diversos resumos para encontrar artigos que poderiam trazer maior benefício para este trabalho e assim foram selecionados até 3 artigos por variável para um maior esclarecimento de como controlar tal variável. Durante a pesquisa se percebeu que há mais trabalhos sobre o controle da radiação solar, umidade e irrigação do que sobre outras variáveis como nutrientes no solo e quantidade de CO<sub>2</sub> e O<sub>2</sub> no ambiente.

Nos trabalhos [Liao et al. \(2021\)](#), [Parada et al. \(2021\)](#), [Bone \(2019\)](#) os autores tinha o objetivo de aumentar a eficiência do uso d’água para irrigação. Eles deixam claro que fizeram uso de modelo matemáticos preditivos, quais variáveis influenciam o modelo e como o modelo foi desenvolvido. Em [Bone \(2019\)](#) o autor foi o único a definir quais dispositivos, sensores e atuadores foram utilizados, também realizou o desenvolvimento de uma *dashboard* para monitoramento remoto e realizou o controle de outras variáveis além das relacionadas com a irrigação. O autor teve sucesso ao desenvolver um dispositivo de monitoramento e controle de baixo custo para casa de vegetação. Em [Liao et al. \(2021\)](#) os autores trouxeram uma forma de analisar a necessidade de irrigação de forma inusitada. Foram usados diversos sensores de umidade de solo em profundidades de até 1 metro para medir o local de maior absorção d’água, o qual é definido dinamicamente pelo modelo. O trabalho teve sucesso em realizar a irrigação ideal para cada muda, sem a deixar secar ou encharcar. Em [Parada et al. \(2021\)](#) os autores têm o objetivo de minimizar a quantidade de litros utilizados para produzir um quilograma, assim realiza a comparação de três tipos diferentes de irrigação, duas delas com recirculação. Eles também fizeram uma análise das propriedades nutricionais dos frutos gerados. Os autores demonstram que a recirculação utiliza consideravelmente menos recursos hídricos e proporciona frutos mais nutritivos.

Em [Lin, Zhang e Xia \(2021\)](#) os autores mostram quatro modelos de otimização, todos construídos com embasamento físico, biológico e químico, para minimizar o consumo de energia, água e CO<sub>2</sub>, cada modelo otimiza o consumo de uma das variáveis, enquanto o último modelo otimiza as três variáveis simultaneamente com o objetivo de minimizar os custos de operação. Os autores nada dizem sobre uso de hardware ou IoT, já que foram usadas simulações. Os modelos cumprem com os seus objetivos. Em [Achour et al. \(2020\)](#) os autores mantiveram o foco exclusivo em otimizar o gasto de energia. Planejaram uma casa de vegetação que gera energia renovável e continua conectada a rede pública caso precise obter energia adicional. Os autores mostram o desenvolvimento de cada equação que é usada no modelo. O modelo analisa informações meteorológicas para prever a quantidade de energia que será gerada e quanto será necessário obter da rede. O trabalho obteve bons resultados no geral, com pequenos desvios em dias de pouca geração de energia.



Em [Mosharafian et al. \(2021\)](#) os autores usa cadeias de Markov para realizar a previsão da radiação solar, assim calculando a quantidade de iluminação artificial necessária para suprir o fotoperíodo da plantação e ajustando a intensidade da iluminação ao considerar a variação do custo da energia. O modelo se mostrou eficaz, ao ponto de reduzir em aproximadamente 50% no custo de energia elétrica ao decorrer do ano.

Em [Vanipriya et al. \(2021\)](#) mesmo o título sugerindo o uso de inteligência artificial os autores não deixaram claro qual ou como o método foi aplicado. Ele define as especificações de cada sensor e como eles foram utilizados e tem como objetivo colher em rápidas sucessões, o qual alcançou com sucesso. Em [Mahbub, Hossain e Gazi \(2020\)](#) os autores definem IoT em detalhes, descrevem camadas de comunicação e as tecnologia que hoje tornam possível a rápida expansão das IoTs e as suas aplicabilidades em diferentes áreas. Também descreveram quais os sensores e atuadores utilizados, mostraram um template da montagem e um pseudocódigo do programa, assim como capturas das telas usadas pelo usuário para monitorar o sistema remotamente. Os autores concluíram o objetivo de construir um simples dispositivo e de baixo custo para monitoramento e controle da casa de vegetação.

Em [Wang e Zhang \(2018\)](#) os autores usam o MATLAB para criar um modelo *fuzzy* com o objetivo de controlar as cortinas de uma casa de vegetação a fim de manter a temperatura desejada. Eles mostram o layout do controlador onde ocorre cada análise, mas pouco falam sobre o hardware. O modelo fuzzy criado controla corretamente as cortinas a fim de manter a temperatura adequada. Em [Liu et al. \(2020\)](#) os autores realizaram a comparação de diversas técnicas de ML para monitorar uma casa de vegetação vertical a fim de detectar anomalias. Os autores descrevem o funcionamento de cada técnica, as suas principais diferenças e como os dados obtidos foram usados. Eles mostraram que o uso de ML é de grande interesse para o monitoramento da casa de vegetação. Em [Fei, Xiao-Long e Yong \(2021\)](#) os autores desenvolveram um controlador ADRC (*Active Disturbance Rejection Control*) com *fuzzy* para obter a temperatura ideal e constante. Todas as etapas e cálculos são descritos e a tabela de *fuzzy* mostrada. Os autores tiveram sucesso em alcançar a estabilidade mais rápido e com menor temperatura de pico comparado com o controlador ADRC isolado. Em [Moon e Eek Son \(2021\)](#) os autores utilizaram a técnica de aprendizado por transferência para otimizar diversos ML já treinadas e fez a comparação com modelos treinados do zero. Ele mostrou que mesmo com uma quantidade limitada de dados é possível obter rapidamente um ML capaz de controlar uma casa de vegetação através de aprendizado por transferência.

Em [Filho \(2018\)](#) o autor tinha o objetivo de fazer um sistema de controle para uma casa de vegetação para gerar 3 tipos de mudas, ele definiu os controladores, sensores e atuadores usados e detalhou quais tecnologias e como o serviço web foi usado para monitoramento remoto. O objetivo alcançado foi validado ao comparar o crescimento das mudas em um ambiente externo e dois internos, um deles controlado, mostrando que o ambiente controlado trouxe melhores brotos em menor tempo.

Foram encontradas diversas publicações sobre casa de vegetação autônomas, mas houve poucos trabalhos onde o modelo foi aplicado em casos práticos ou que usam as técnicas modernas de ML. Também se percebeu a tendência de focar em otimizar apenas uma das variáveis em vez de diversas e ao ocorrer uso de ML o seu processamento ocorriam em nuvem. Este trabalho tem o objetivo de desenvolver um dispositivo capaz de monitorar, armazenar, processar e controlar diversas variáveis apenas usando o processamento local.

## 2 Fundamentação Teórica

### 2.1 Variáveis das Casas de vegetação

Apenas 3% da água disponível no mundo é potável e dessa porcentagem 67% se localiza nos glaciais, assim apenas 0,5% da água existente no mundo está disponível para uso. Dessa pequena quantidade, de 70% a 85% é utilizada para o agronegócio e ao usar métodos de irrigação tradicionais ocorrem desperdícios (AKHTER; SOFI, 2021). A coleta dessa variável será feita com sensores de umidade do solo, após a análise será tomada a decisão de realizar a irrigação usando sistema de bombas.

A radiação solar, temperatura e umidade do ar são variáveis que estão relacionadas, além disso há outros fatores que também as influenciam, como a geometria, o material, o posicionamento e a ventilação da casa de vegetação (ACHOUR; OUAMMI; ZEJLI, 2021).

A radiação solar é determinada diretamente pela incidência do sol e ocorre uma variação dependendo do material. É possível incrementar ou decrementar o índice de incidência com lâmpadas, sombras e cortinas (TERUEL, 2010; ACHOUR; OUAMMI; ZEJLI, 2021). A maior incidência reflete no aumento da temperatura e da evapotranspiração, aumentando a umidade do ar, enquanto o oposto também se aplica.

A temperatura afeta as funções metabólicas das plantas, o ideal para o seu controle seria manter ela dentro de uma margem entorno do seu valor ótimo para a plantação (TERUEL, 2010). O seu controle pode ser feito ao prever a variação e fazer o ajuste de antemão ou com ajustes pontuais. A leitura da temperatura será feita ao realizar a média dos sensores espalhados.

O controle da umidade do ar, as vezes esquecido, é interessante por ajudar a evitar pragas, auxiliar a variação térmica e reduzir a necessidade de irrigação (TERUEL, 2010). A leitura se fará igual a temperatura, a média dos sensores espalhados. O seu controle será feito simultaneamente ao da temperatura ao usar sistemas de resfriamento, aquecimento, umidificação ou ventilação.

Além de auxiliar no controle de temperatura e umidade do ar, a ventilação também contribui para a renovação do CO<sub>2</sub> e O<sub>2</sub> no ambiente (TERUEL, 2010). O seu controle pode ser feito de forma passiva ou ativa com o uso de ventiladores ou injeção ativa desses gases.

Os níveis de CO<sub>2</sub> e O<sub>2</sub> são de grande importância para as plantas devido ao processo de respiração e fotossíntese (TERUEL, 2010), mas não será desenvolvido o controle direto e nem a monitoração dessas variáveis.

### 2.2 IoT

O controlador é responsável por processar, analisar e armazenar os dados. Normalmente a análise e o armazenamento é feito com serviços de terceiros, que possuem custos, mas neste trabalho será usado um *Raspberry Pi* para essas necessidades. O *Raspberry Pi* é capaz de rodar um sistema operacional completo com rápida conexão com a internet e baixo consumo de energia, portando ele será ótimo para a execução de um servidor para monitoramento remoto, realizar a execução dos modelos decisórios e armazenamento dos dados, entretanto por cautela será usado um serviço gratuito de terceiros para armazenamento de dados neste trabalho. Os trabalhos de Bone (2019), Vanipriya et al. (2021), Mahbub, Hossain e Gazi (2020), Filho (2018) foram os poucos que descreveram qual hardware foi usado e entre eles o *Raspberry Pi*, *Arduino* e *ESP* são mencionados.

Como comunicação com a internet foi utilizado conexão por Wi-Fi para uma boa performance do servidor. Não houve trabalhos que definiram a comunicação com o meio externo, mas em [Tao et al. \(2021\)](#) vemos que a comunicação GSM também é utilizada para tal.

Em [Tao et al. \(2021\)](#) é feita uma comparação entre as tecnologias usadas em diversos trabalhos e como era de se esperar Wi-Fi foi a mais utilizada, pela fácil implementação e integração com diversos dispositivos, assim ele foi o escolhido para este trabalho. Existem diversos protocolos de comunicação no mundo, cada um com as suas diferentes vantagens, ZigBee, Bluetooth, Sigfox e LoRa são algumas delas, mas para utilizar algumas delas é necessário pagar por serviço de terceiros ou adquirir módulo para tal, assim a decisão do uso de Wi-Fi se deu pela facilidade de implementação e integração com os controladores.

Como central de monitoramento será usado um *ESP 32* conectado por fios aos sensores para comunicação e fornecimento de energia e a rede Wi-Fi para comunicação com o servidor. Como central de atuadores também será usado um *ESP 32*, mas neste caso ele controla um módulo de relés e estará executando um servidor para ativação dos atuadores. O uso dos *ESP 32* foi pouco visto nos trabalhos referenciados, mas a capacidade do controlador e sua conectividade é de grande interesse para a área de IoT.

### 2.3 Machine Learning

Para a tomada de decisões foi usada uma técnica de ML que não foi mencionada nos trabalhos referenciados, o aprendizado por reforço (do Inglês: *Reinforcement Learning*(RL)). O método desta técnica se baseia no aprendizado humano, onde se entende o certo com recompensas e o errado com punições, assim o modelo irá reforçar as ações que possuem consequências desejadas e punir as que se desejam evitar. As entradas serão os grupos de sensores já mencionados e a saída serão as ações necessárias para o momento, a ativação ou não dos atuadores.

Com o objetivo de simplificar a implementação da ML se decidiu usar o algoritmo de *Q Learning Table*. Seu método é de construir uma tabela onde as linhas são as entradas e as colunas as saídas. O valor das células da tabela representa a melhor ação a ser tomada com aquelas entradas. O treinamento neste caso ocorre ao recalcular o valor da célula da tabela após receber o valor da recompensa, usando a equação de aprendizado por diferença temporal levemente modificada.

O objetivo da ML neste caso é de se tornar capaz de atingir os valores ótimos estipulados pelo usuário com maior eficiência, assim quando o tipo de plantação for trocado basta o usuário atualizar os valores ótimos que a ML saberá como atingi-los através da comparação dos estados e não do valores brutos.

A *OpenIA* utilizou esse método para treinar diversas inteligências artificiais(IA), um dos exemplos é onde a IA aprende a brincadeira de pique esconde. Um dos agentes da IA tem o objetivo de localizar o outro agente, que em contrapartida tem o objetivo de não ser encontrado([OPENIA, 2021](#)). Esse método de ML é interessante pois tende a encontrar melhorias não esperadas ao se comparar com outros métodos, já que as vezes uma ação que pareça estranha pode trazer recompensa em vez de punição.

### 3 Desenvolvimento

O desenvolvimento do projeto teve 5 etapas, que serão descritas nas próximas seções. Se iniciou com a construção de uma casa de vegetação doméstica com sensores e atuadores. Em seguida o desenvolvimento do firmware dos *ESP 32* para controle dos sensores e atuadores. Com a estrutura pronta para coletar dados e tomar ações se desenvolveu uma API para realizar o armazenamento dos dados e outro código para o treinamento da ML e envio das ações para a casa de vegetação.

#### 3.1 Casa de Vegetação

Foram utilizados tubos de PVC para realizar a montagem da estrutura.

Figura 1 – Estrutura do esqueleto



Fonte: Próprio autor

Figura 2 – Estrutura Com a lona



Fonte: Próprio autor

Na Figura 1 é mostrada a estrutura usada de cubo de aproximadamente  $1\text{ m}^3$ . A forma de cruz na parte de trás, de cima e dos lados foi usada para dar maior rigidez a estrutura. Em seguida a estrutura teve os sensores instalados e coberto com uma lona específica para casas de vegetação, mostrado na Figura 2.

Os sensores utilizados foram:

- DHT-22: Medição de temperatura e umidade relativa do ar
- GY-30: Medição da luminosidade do ambiente
- HD-38: Medição da umidade do solo
- FD-10: Medição do nível de água dos reservatórios

Os *datasheets* dos sensores foram lidos para ter o entendimento do funcionamento deles, mas foram usadas bibliotecas dedicadas ao sensores quando necessários fazer a decodificação do dado enviado, no caso do DHT-22 que usa comunicação *One Wire*. No caso do GT-30 foi utilizado o protocolo  $I^2C$  para comunicação, também auxiliado por

uma biblioteca. Os sensores HD-38 e FD-10 enviam um sinal analógico que precisa ser mapeado com os valores desejado para obter uma porcentagem da umidade do solo e do nível d'água.

Os atuadores utilizados foram:

- Placa Peltier TEC1 12706: Utilizado para esquentar e esfriar o ambiente
- Ventoinhas de 5V: Utilizado para resfriar as placas peltier
- Bomba Peristáltica Robocore: Utilizado para o sistema de irrigação
- Ventoinhas de 12V: Utilizado para o sistema de exaustão
- Umidificador comercial: Utilizado para umidificar o ambiente
- Barra de LED 18W: Utilizado para suprir a falta de luz

O controle de ativação dos atuadores é realizado por um *ESP 32* conectado a um módulo de relés, que executa uma API que tem com cliente a aplicação do ML, ambos serão descritos com maiores detalhes depois.

Nas Figuras 3 e 4 são mostrados o projeto final da casa de vegetação, o aspecto externo e interno, com os sensores e atuadores indicados na Figura 4

Figura 3 – Casa de vegetação finalizada exterior



Fonte: Próprio autor

Figura 4 – Casa de vegetação finalizada interior



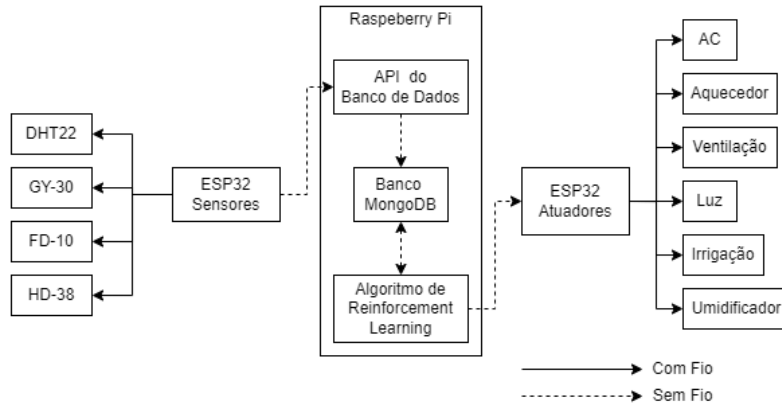
Fonte: Próprio autor

### 3.2 Fluxos de programação

Um *ESP 32* está conectado aos sensores e ele funciona como cliente para uma API que está sendo executada por um contêiner *Docker* no *Raspberry Pi 4*, o qual salva o JSON do request "GET" na base de MongoDB. Esse mesmo *Raspberry Pi 4* também executa o algoritmo de RL em um *Docker* que realiza as tomadas de decisões e é cliente para a API do servidor que se encontra no outro *ESP 32* para controlar o módulo de relés

para ativação dos atuadores. O fluxo de informações e de decisões dentro do projeto pode ser melhor entendida pelo fluxograma na Figura 5.

Figura 5 – Fluxograma de funcionamento geral do projeto

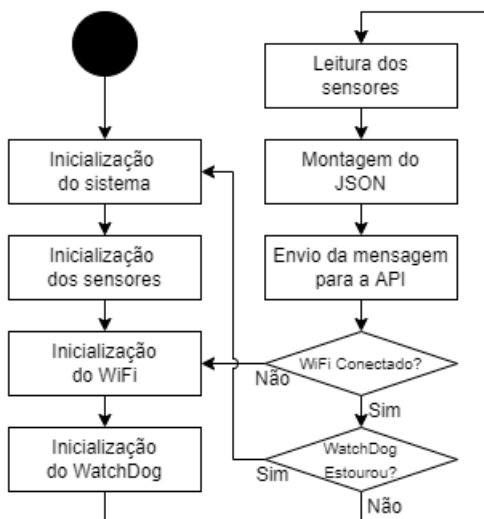


Fonte: Próprio autor

### 3.2.1 Sensores

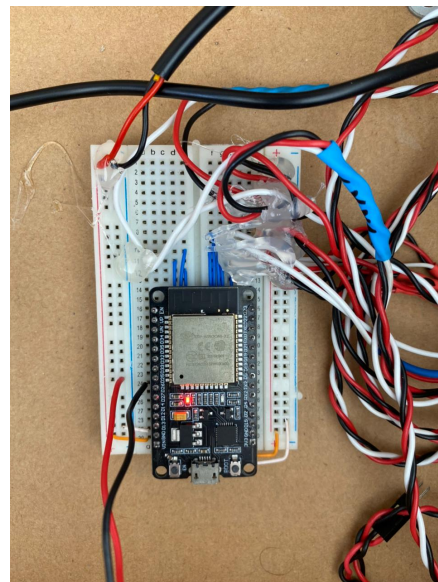
Como mencionado as variáveis da casa de vegetação são monitoradas por um *ESP32* que é cliente de uma API para realizar o armazenamento na base. Este controlador é a entrada das variáveis e é essencial que possua uma boa estabilidade e acurácia na leitura para o correto treinamento, assim se usou de *watchdog* e *Wi-fi reconnect* para garantir o constante envio das medições. Na Figura 6 tem o fluxograma do firmware e na Figura 7 o *ESP32* instalado na casa de vegetação.

Figura 6 – Fluxograma dos Sensores



Fonte: Próprio autor

Figura 7 – *ESP32* dos sensores

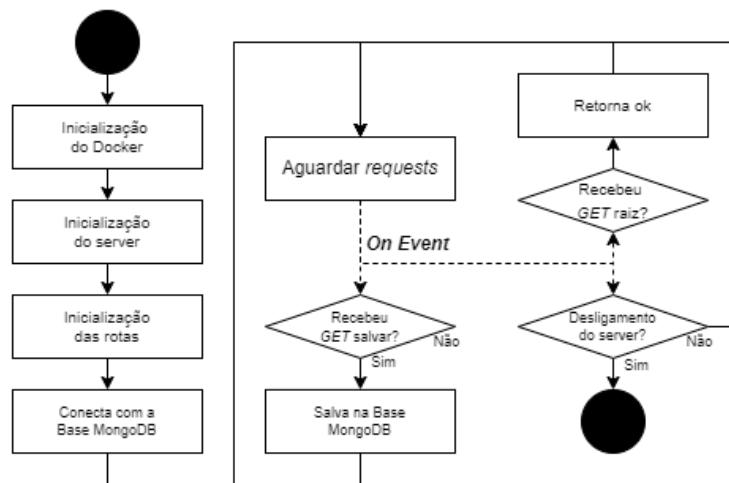


Fonte: Próprio autor

### 3.2.2 API do Banco de Dados

Foi usado uma API em *Python* utilizando o framework *FastAPI* para rápido desenvolvimento e fácil manutenção. *Docker* foi utilizado para criar uma imagem do script e o executar em um contêiner dentro do *Raspberry Pi 4*. A API fica no aguardo da *request* do *ESP32* dos sensores para realizar o armazenamento dentro da base MongoDB. Como comentado anteriormente, idealmente seria usada uma imagem *Docker* do MongoDB no *Raspberry pi 4* para evitar custo com terceiros, mas para garantir a segurança das informações neste trabalho foi usado um serviço gratuito.

Figura 8 – Fluxograma da API



Fonte: Próprio autor

### 3.2.3 Algoritmo de Reinforcement Learning - Q Learning Table

A compreensão da biblioteca *Gym* da OPENIA (2021) foi necessária para desenvolver seguindo o padrão existente, onde normalmente é aplicada para treinamento em cima de jogos, que possuem suas variáveis e ações bem definidas. O algoritmo de RL *Q Learning Table* trabalha com um *environment* que possuiu as suas variáveis de entrada mais conhecidas como *states*, por isso foi necessário definir o *observation space*, que são as informações que podem ser obtidas do *environment* e quais os seus possíveis valores, a definição do *observation space* se encontra na Figura 9.

Figura 9 – Observation Space

```
self.observation_space = spaces.Dict({
    'Temp': spaces.Discrete(3, start=-1),
    'RH': spaces.Discrete(3, start=-1),
    'Soil_Moisture': spaces.Discrete(2, start=-1),
    'LUX_Index': spaces.Discrete(2, start=-1)
})
```

Fonte: Próprio autor

Para ser consistente com o arquivo *JSON* enviado para a base MongoDB foi definido um dicionário com as 4 variáveis a serem controladas e os seus possíveis estados.

- Temp:
  - -1: Temperatura abaixo do valor desejado
  - 0: Temperatura dentro do valor desejado
  - 1: Temperatura acima do valor desejado
- RH:
  - -1: Umidade relativa do ar abaixo do valor desejado
  - 0: Umidade relativa do ar dentro do valor desejado
  - 1: Umidade relativa do ar acima do valor desejado
- Soil Moisture:
  - -1: Umidade do solo abaixo do valor desejado
  - 0: Umidade do solo dentro do valor desejado
- LUX Index:
  - -1: Luminosidade abaixo do valor desejado
  - 0: Luminosidade dentro do valor desejado

Essas são as definições de possíveis *states*, que nada condizem com a informação analógica enviada pelo sensor e armazenada na base, assim para converter do valor lido se usou uma função que compara o valor na base com o valor desejado e define o valor do estado dentro do escopo discreto definido.

Após a leitura de um estado é necessário tomar uma ação, no algoritmo *Q Learning Table* o valor do estado é inserido como a linha de uma tabela e se escolhe a melhor ação para aquele estado, ou seja, o valor com maior nível de ativação. Neste caso é usado todos os valores das ações e se o nível de ativação da ação for atingido ocorre a ativação do atuador. O dicionário possui 6 chaves e o seu escopo varia de 0 a 1, considerando o nível de ativação sendo de 0.8, sua definição se encontra na Figura 10.

Figura 10 – *Action Space*

```
self.action_space = spaces.Dict({
    'heater': spaces.Box(low=0, high=1, shape=(1,)),
    'AC': spaces.Box(low=0, high=1, shape=(1,)),
    'ventilation': spaces.Box(low=0, high=1, shape=(1,)),
    'irrigation': spaces.Box(low=0, high=1, shape=(1,)),
    'light': spaces.Box(low=0, high=1, shape=(1,)),
    'humidifier': spaces.Box(low=0, high=1, shape=(1,)),
})
```

Fonte: Próprio autor

- *heater*: Sistema de aquecimento - 2 placas peltier com 2 ventoinhas 5V para resfriamento e manter eficiência.
- *AC*: Sistema de resfriamento - 2 placas peltier com 2 ventoinhas 5V para resfriamento e manter eficiência.



- *ventilation*: Sistema de exaustão - 4 ventoinhas 12V, ventilação funciona por pressão negativa ao extrair o ar interno.
- *irrigation*: Sistema de irrigação - Bomba peristáltica
- *light*: Sistema de iluminação - Barra de LED 18W
- *humidifier*: Sistema de umidificação - Umidificador comercial

Em seguida é definido a *Q Table* com o auxílio da biblioteca *Pandas* para construir um *dataframe* com múltiplos índices, assim com a combinação de todos os possíveis estados individuais se obtém 36 possíveis estados, onde cada estado poderá tomar diversas possíveis ações. Na Figura 11 tem o exemplo da *Q Table* inicializada no começo do treinamento, onde seu índice, como já mencionado, é a combinação dos estados e sua ordem é: Temp, RH, Soil Moisture, LUX Index

Figura 11 – *Q Table* no início do treinamento

	AC	heater	humidifier	irrigation	light	ventilation
(-1, -1, -1, -1)	0	0	0	0	0	0
(-1, -1, -1, 0)	0	0	0	0	0	0
(-1, -1, 0, -1)	0	0	0	0	0	0
(-1, -1, 0, 0)	0	0	0	0	0	0
(-1, 0, -1, -1)	0	0	0	0	0	0
(-1, 0, -1, 0)	0	0	0	0	0	0
(-1, 0, 0, -1)	0	0	0	0	0	0
(-1, 0, 0, 0)	0	0	0	0	0	0
(-1, 1, -1, -1)	0	0	0	0	0	0
(-1, 1, -1, 0)	0	0	0	0	0	0
(-1, 1, 0, -1)	0	0	0	0	0	0
(-1, 1, 0, 0)	0	0	0	0	0	0
(0, -1, -1, -1)	0	0	0	0	0	0
(0, -1, -1, 0)	0	0	0	0	0	0
(0, -1, 0, -1)	0	0	0	0	0	0
(0, -1, 0, 0)	0	0	0	0	0	0
(0, 0, -1, -1)	0	0	0	0	0	0
(0, 0, -1, 0)	0	0	0	0	0	0
(0, 0, 0, -1)	0	0	0	0	0	0
(0, 0, 0, 0)	0	0	0	0	0	0
(0, 1, -1, -1)	0	0	0	0	0	0
(0, 1, -1, 0)	0	0	0	0	0	0
(0, 1, 0, -1)	0	0	0	0	0	0
(0, 1, 0, 0)	0	0	0	0	0	0
(1, -1, -1, -1)	0	0	0	0	0	0
(1, -1, -1, 0)	0	0	0	0	0	0
(1, -1, 0, -1)	0	0	0	0	0	0
(1, -1, 0, 0)	0	0	0	0	0	0
(1, 0, -1, -1)	0	0	0	0	0	0
(1, 0, -1, 0)	0	0	0	0	0	0
(1, 0, 0, -1)	0	0	0	0	0	0
(1, 0, 0, 0)	0	0	0	0	0	0
(1, 1, -1, -1)	0	0	0	0	0	0
(1, 1, -1, 0)	0	0	0	0	0	0
(1, 1, 0, -1)	0	0	0	0	0	0
(1, 1, 0, 0)	0	0	0	0	0	0

Fonte: Próprio autor

Após todas as configurações do ambiente e inicialização da *Q Table* se inicia o treinamento da ML, primeiro é necessário dar um *reset* no *environment*, obtendo a média das últimas medições ou recuperando as informações da última ação tomada. Em seguida se realiza um *step* dentro do *environment*, que seria realizar as *actions* definidas a partir dos *states*, avaliar o efeito e calcular o *reward*.

O *step* consiste em mandar a ação à casa de vegetação, aguarda 1 minuto para a ação ter alguma consequência no ambiente, avaliar o efeito da ação, calcular o *reward* e recalculando a ativação das ações. A escolha do tempo de espera foi feita de forma arbitrária, mas considerando que o modelo em questão não penaliza as ações realizadas, assim se uma ação foi tomada de forma correta ela constantemente seria repetida e premiada, enquanto uma ação errada não seria premiada, assim não reforçando a sua repetição.

O recálculo dos valores de ativação é baseado na seguinte equação:

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[R_{t+1} + \gamma Q(S_{t+1}, A_t) - Q(S_t, A_t)] \quad (1)$$

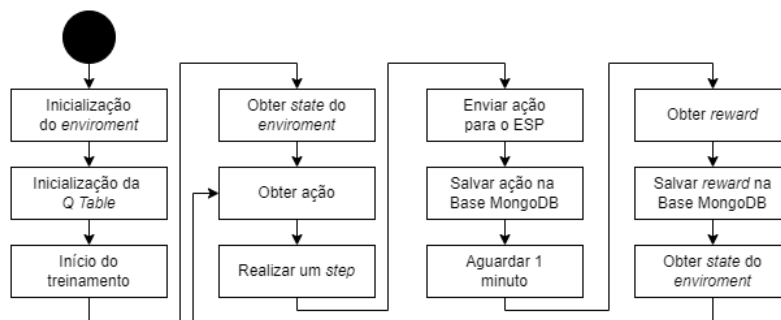
Onde  $Q(S_t, A_t)$  é o *state* atual,  $Q(S_{t+1}, A_t)$  é o *state* futuro,  $R_{t+1}$  é o *reward* da *action*,  $\alpha$  é a taxa de aprendizagem e  $\gamma$  é quanto o estado futuro irá influenciar no novo nível de ativação do estado atual, quanto maior esse valor melhor o modelo será para ações de longo prazo, enquanto menores valores o modelo será melhor para ações imediatistas.

Um dos pontos mais delicados no RL, especialmente no *Q Table*, é como será feito o cálculo do *reward* para o *step*. Se mal planejado o modelo pode reforçar decisões erradas para alguns *states*. Há exemplos de modelos onde o *reward* é mal estimado, reforçando algumas ações mas não o objetivo final, assim deixando o modelo ineficiente por entrar em loop infinitos ou por tomar ações que não o fariam alcançar o objetivo final, em ambos os casos o modelo se torna ineficiente para a solução do problema.

Uma das características atrativas do RL, já mencionada, é que ao final o modelo poderia ter descoberto uma ação com uma consequência logicamente não óbvia ao humano, assim podendo ter otimizações melhores do que o usuário esperava. No caso deste trabalho isto não irá ocorrer pois a definição de quando deve ocorrer um *reward* teve seu escopo muito limitado devido ao fator do ambiente externo, que pode alterar os valores das variáveis internas da casa de vegetação. Assim o *reward* só será dado quando o *environment* estiver fora do *state* desejado (0,0,0,0) e ocorrer uma consequência, se houve alguma ação, que poderia fazer o *environment* alcançar o *state* desejado, assim tentando evitar que o sistema de *reward* possa ser afetado por alterações climáticas drásticas.

Assim o fluxo do algoritmo de *Reinforcement Learning* para este trabalho é mostrado na Figura 12

Figura 12 – Fluxograma do RF



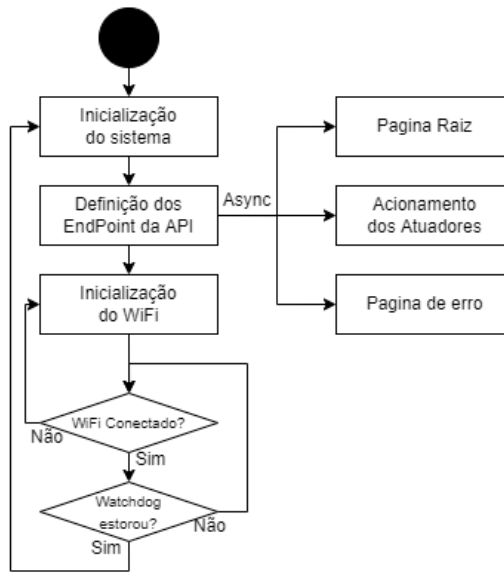
Fonte: Próprio autor

### 3.2.4 Atuadores

Por fim a ativação dos atuadores é feita por um servidor rodando dentro do segundo *ESP 32*, se usou um biblioteca para definir os *Endpoints* assíncronos, assim eles serão executado quando houver a *request GET* enquanto o *loop* principal mantém a conexão com a internet e realiza o reinício do *watchdog timer* <sup>1</sup>.

<sup>1</sup> Timer para reinício do sistema caso ocorra travamento do sistema

Figura 13 – Fluxograma dos Atuadores



Fonte: Próprio autor

### 3.3 Dificuldades no desenvolvimento

Com o hardware a dificuldade foi de encontrar os componentes desejados, testar eles individualmente e como um todo, ocasionalmente ocorrendo erros de difícil depuração sem as ferramentas corretas e encontrar fontes capazes de fornecer energia suficiente para toda a casa de vegetação. Como a montagem da casa de vegetação foi feita usando módulo prontos e não os sensores propriamente ditos a quantidade de energia que é usada pela casa de vegetação é grande, não contribuindo para a redução do consumo de recursos artificiais, se acredita que um projeto de PCB desenvolvido especificamente para este trabalho poderia trazer melhor eficiência.

No desenvolvimento do software a maior dificuldade foi definir um *action space* e um *observation space* que poderiam ser usados para a *Q Table*. A própria *Q Table* também trouxe problemas para o desenvolvimento, inicialmente se considerou usar uma matriz com profundidade 4, mas o uso do *Pandas* com múltiplos índices tornou a solução do problema e o desenvolvimento do software simples e rápido. Outro problema foi a incerteza entre a utilização do algoritmo *Q Table Learning* ou *Deep Q Learning*, que possui os mesmo conceitos para o treinamento, mas se utiliza de *deep learning* e rede neurais artificiais (RNA) para a tomada de decisão e realiza o ajuste dos pesos sinápticos da RNA com base no *reward* usando o algoritmo de *back propagation*. Outro motivo ao se decidir em utilizar *Q Learning Table* foi a dificuldade de implementação que o uso de *tensorflow* ou *pytorch*, *frameworks* de *deep learning*, trazem.

Na inicialização todos os códigos o upload do firmware dos *ESPs 32* não trouxe desafios, entretanto a configuração do *Docker* para executar a API do MongoDB e o algoritmo RL trouxeram diversos desafios. Ao se tentar ler a sua documentação mais dúvidas surgiram, assim sendo necessário realizar diversos testes com configurações diferentes até atingir o funcionamento desejado. Mesmo assim o seu uso foi essencial por garantir a constante execução dos códigos caso ocorresse o reinício do *Raspeberry* e garantir o funcionamento desejado, por ser uma instância isolada do OS em que é executado

## 4 Resultados e Discussões

Após uma semana de execução o modelo não conseguiu convergir e isso pode ser causado por diversos motivos. Como mencionado a limitação do escopo do *reward* pode ser um dos principais motivos desse problema. Sem a liberdade de receber pontuações por qualquer ação tomada o modelo precisa de muito tempo para convergir, se acredita que seria necessário pelo menos 1 ano para o modelo convergir e ter uma boa acurácia nas decisões, ao analisar a *Q Table* da Figura 14 é fácil perceber os estados vazios e que provavelmente apenas ocorreriam em diferentes estações do ano.

Há também a falta de punição por ações tomadas erradas, algo que se evitou implementar por causa do ambiente externo, um exemplo seria no caso do acionamento do sistema de refrigeração, se uma súbita onda de calor aparecesse o sistema de *reward* iria punir o modelo que teria tomado a decisão correta, mas como o sistema de refrigeração não é potente o suficiente e perderia para a onda de calor o modelo seria punido. Mesmo assim se espera que após um longo período de treinamento esses casos ocasionais não afetariam o modelo no final, mas ao considerar o pouco tempo que se teria para treinar o modelo, neste caso, se evitou realizar punições.

Da *Q Table* da Figura 14 é possível tirar outras informações, uma delas é a não convergência do modelo. Como na coluna de estados, a primeira coluna, que segue a sequência de “Temperatura”, “RH”, “Umidade do Solo” e “Índice de Luminosidade”, é possível observar na única linha que tem o estado de “Umidade do Solo” igual a “-1” (Linhas 5), indicando que faltou irrigação, que o nível de ativação da irrigação não atinge o valor estipulado de 0.8 para a ativação do atuador. Entretanto o nível de ativação dos outros estados preenchidos que possuem o estado de “Umidade do Solo” igual a “0” não possuem um nível de ativação de 0,8, mostrando que o modelo começou a convergir.

Figura 14 – *Q Table* durante o treinamento

	AC	heater	humidifier	irrigation	light	ventilation
(-1, -1, -1, -1)	0	0	0	0	0	0
(-1, -1, -1, 0)	0	0	0	0	0	0
(-1, -1, 0, -1)	0.198433	0.563149	0.234889	0.0467207	0.251827	0.316076
(-1, -1, 0, 0)	0.54428	0.321311	0.363952	0.529955	0.439529	0.353867
(-1, 0, -1, -1)	0.0159965	0.0363522	0.0267335	0.0161726	0.0307814	0.00270559
(-1, 0, -1, 0)	0	0	0	0	0	0
(-1, 0, 0, -1)	0.634278	0.703465	0.375753	0.321428	0.231621	0.710006
(-1, 0, 0, 0)	0	0	0	0	0	0
(-1, 1, -1, -1)	0	0	0	0	0	0
(-1, 1, -1, 0)	0	0	0	0	0	0
(-1, 1, 0, -1)	0.343366	0.314258	0.308253	0.74225	0.556728	0.771863
(-1, 1, 0, 0)	0	0	0	0	0	0
(0, -1, -1, -1)	0	0	0	0	0	0
(0, -1, -1, 0)	0	0	0	0	0	0
(0, -1, 0, -1)	0.0864256	0.47291	0.368261	0.775684	0.537266	0.093969
(0, -1, 0, 0)	0	0	0	0	0	0
(0, 0, -1, -1)	0	0	0	0	0	0
(0, 0, -1, 0)	0	0	0	0	0	0
(0, 0, 0, -1)	0.564768	0.163523	0.700866	0.190662	0.424677	0.521817
(0, 0, 0, 0)	0.584093	0.320537	0.178962	0.281445	0.454851	0.681674
(0, 1, -1, -1)	0	0	0	0	0	0
(0, 1, -1, 0)	0	0	0	0	0	0
(0, 1, 0, -1)	0.703747	0.0948268	0.347075	0.254037	0.378032	0.708119
(0, 1, 0, 0)	0	0	0	0	0	0
(1, -1, -1, -1)	0	0	0	0	0	0
(1, -1, -1, 0)	0	0	0	0	0	0
(1, -1, 0, -1)	0.6139	0.423752	0.661342	0.466631	0.636978	0.304719
(1, -1, 0, 0)	0.571125	0.378178	0.406676	0.652315	0.614269	0.177638
(1, 0, -1, -1)	0	0	0	0	0	0
(1, 0, -1, 0)	0	0	0	0	0	0
(1, 0, 0, -1)	0.401772	0.340091	0.108359	0.416234	0.239331	0.202289
(1, 0, 0, 0)	0.043767	0.00367473	0.0447699	0.018646	0.0173681	0.0198325
(1, 1, -1, -1)	0	0	0	0	0	0
(1, 1, -1, 0)	0	0	0	0	0	0
(1, 1, 0, -1)	0.00305976	0.0309052	0.0265406	0.0509881	0.0378788	0.00602946
(1, 1, 0, 0)	0	0	0	0	0	0

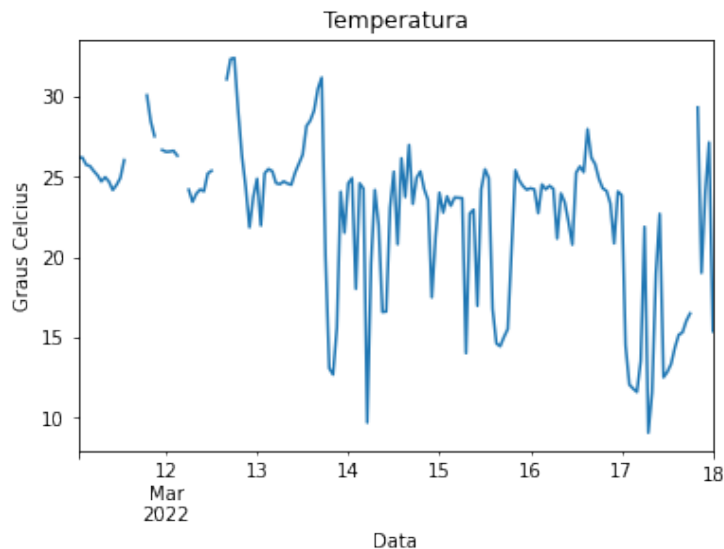
Fonte: Próprio autor

Outro indício do início da convergência é visto ao comparar “AC” a “heater” que quando o ambiente estava com o estado “-1” no estado “Temperatura”, baixa temperatura, o nível de ativação do “heater” são maiores que os do sistema de refrigeração, o mesmo também ocorre no estado oposto. Em contrapartida os níveis de ativação da “light” não condizem com o estado, é bem provável que esses valores sejam causados pelo escopo do *reward*, o valor para realizar a pontuação da “light” acabou sendo muito alto, assim nunca reforçando a ação de ligar a luz quando a pouca luminosidade.

Durante os primeiros dias de execução também foi necessário realizar diversas intervenções e atualizações dos códigos para tornar a captura de dados e a tomada da ação o mais estável possível, assim dificultando mais a convergência do modelo. No final a API do MongoDB, o algoritmo de ML e a API dos atuadores não apresentaram mais erros, porém a leitura dos dados continuou apresentando problemas ocasionalmente.

O problema na leitura dos sensores de não retornar valores de temperatura e RH ou retornar valores errados afeta o *state* ao definir que a temperatura está abaixo do alvo sendo que pode estar no valor certo ou acima, assim alterando níveis de ativações das ações para *states* errados. Pode ser visto no gráfico da Figura 15 onde há descontinuidade no gráfico e valores muito baixos de temperatura que não ocorreram com a temperatura local durante o tempo de teste. Esse evento ocorria após vários horas de funcionamento correto e mais uma vez sem as ferramentas corretas para medir se o sinal do sensor está sendo enviado corretamente ou se o problema se encontra no *ESP 32* a depuração se torna complicada, dificultando o ajuste.

Figura 15 – Histórico de temperatura

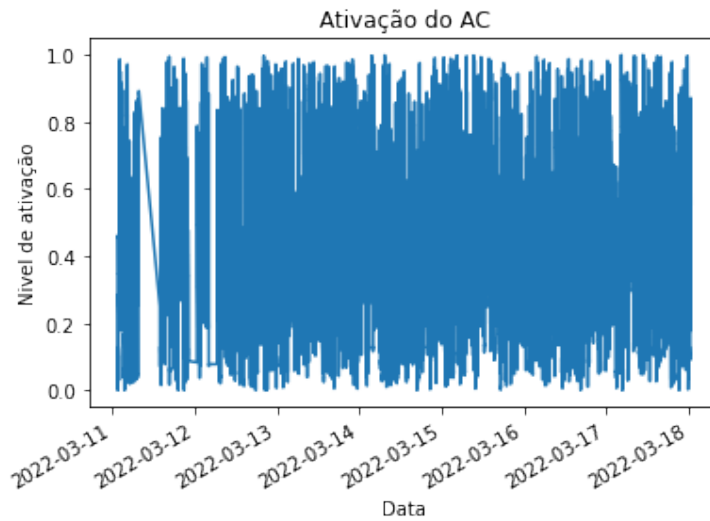


Fonte: Próprio autor

A solução para o problema na tomada de ação é bem nítido no gráfico da Figura 16, que possui uma descontinuidade na metade do primeiro dia e algumas intermitências no começo do segundo, que foi quando os códigos foram ajustados. mas em seguida ele mantém a sua continuidade. Infelizmente por falta da convergência analisar as ações tomadas durante o treinamento não traria outras conclusões devido a sua alta quantidade de níveis de ativação errados para os estados e a quantidade de ações randômicas tomadas durante o treinamento, que após 3 dias foi reduzida de 40% para 10%, a fim de fazer o

modelo reforçar mais as ações salvas na Q Table.

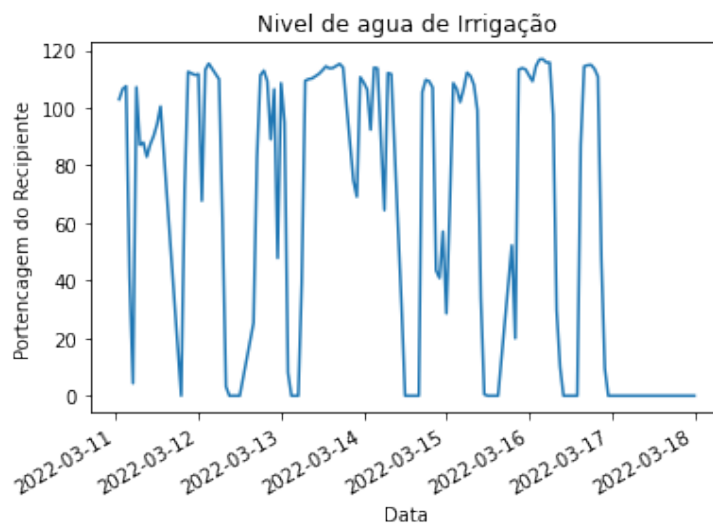
Figura 16 – Histórico de acionamentos do “AC”



Fonte: Próprio autor

Um problema relacionado a preparação da estufa é a potência da bomba peristáltica que usa muita água, talvez um ajuste na duração da ativação ou fazer um ciclo de pulsos na irrigação ou utilizar uma bomba mais fraca solucionariam esse problema. O problema apenas ocorria quando uma ação randômica ativava o sistema de irrigação, a planta acabava sendo muito irrigada e isso esgotava o reservatório de água muito rápido, sendo necessário fazer a única iteração do usuário com a casa de vegetação de encher o reservatório. No gráfico da Figura 17 é possível ver a quantidade de vezes que o reservatório foi esvaziado e por quanto tempo, devido ao fator humano, o sistema de irrigação não estava funcional 100% do tempo, também afetando a convergência do modelo.

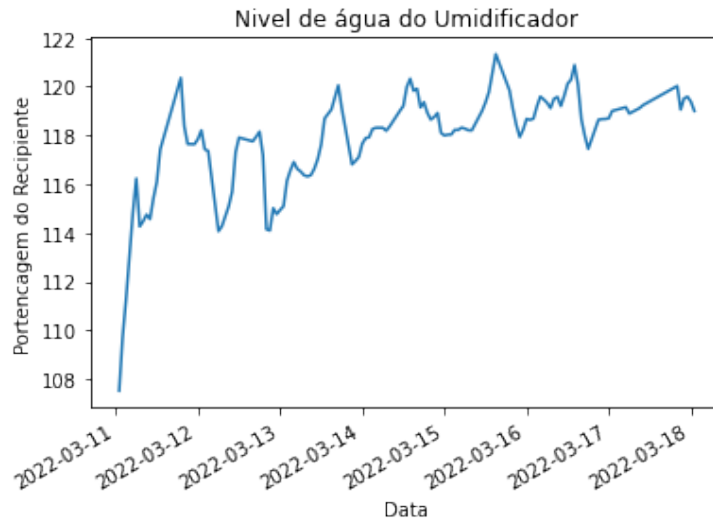
Figura 17 – Histórico do nível de água de irrigação



Fonte: Próprio autor

Outro problema descoberto após o término do treinamento foi que o umidificador comercial não estava funcionando, isso é perceptível pelo gráfico da Figura 18 onde o nível de água não foi alterado drasticamente e porque durante o período de treinamento o reservatório do umidificador não precisou ser reabastecido.

Figura 18 – Histórico do nível de água do Umidificador



Fonte: Próprio autor

Algumas informações gerais sobre o funcionamento da casa de vegetação, na zona de fuso-horário UTC é a data de início em 2022-03-11 01:21:42, a data final em 2022-03-18 00:41:16 e o tempo total de execução de 7 Dias, 3 minutos, 42 segundos

Durante a plotagem dos gráficos, como já mostrado, houve alguns dados faltantes ou incorretos. Ao analisar os dados foram removidas 157 linhas por leitura de luminosidade incorreta e removidas 23.880 linhas por leitura de temperatura ou RH incorreta. Dessa forma de um total de 95.760 foram removidas 24.037 leituras erradas, restando um total de 71.723, percebesse que ocorreu uma perda de 25% das leituras armazenadas, assim fazendo esse valor aumentar ao considerar as leituras que tiveram problemas de alcançar a API nos primeiros dias antes dos ajustes.

Com o pouco tempo de execução a interpretação dos dados das ações seria difícil, assim estes são as quantidades de vezes que os atuadores foram acionados:

- Quantidade de ativações do “AC”: 1437
- Quantidade de ativações do “heater”: 1506
- Quantidade de ativações do “humidifier”: 1442
- Quantidade de ativações do “irrigation”: 1294
- Quantidade de ativações do “light”: 1437
- Quantidade de ativações do “ventilation”: 1430

## 5 Conclusão e Trabalhos Futuros

Neste trabalho se compreendeu a dificuldade de desenvolvimento de hardware com interação com o mundo, a necessidade de uma especificação de hardware bem detalhada e a importância de testar o hardware por longos períodos para conseguir depurar os erros em tempo de execução. A falta dos equipamentos certos, como osciloscópio, para o desenvolvimento correto foi impactante e em futuros trabalhos o uso de um laboratório bem equipado será de suma importância.

Se entendeu a importância do cultivo com eficiência para o momento crucial que o mundo se encontra e que o futuro do agronegócio será a integração com tecnologias, sendo casas de vegetação automatizadas ou outros dispositivos que ainda serão desenvolvidos. A dificuldade para cultivar frutos com eficiência também foi percebida, pois se tratando de um ser vivo há muitos detalhes para fazer o fruto crescer rápido e saudável.

Ficou claro que *machine learning* e *deep learning* não são semelhantes e que há muitos métodos e formas de realizar a tomada de decisão de forma inteligente sem fazer uso de redes neurais artificiais ou outros métodos que necessitam de equipamentos com alta capacidade computacional para realizar o processamento. O uso do *Raspberry Pi 4* como central computacional deixou isso bem claro, mesmo se tratando da última versão do aparelho disponível no momento de desenvolvimento, ele nada se compara com serviços de terceiros em poder computacional

Para trabalhos futuros uma casa de vegetação feita em laboratório seria um grande diferencial, uma casa de vegetação que toma medidas do ambiente externo e usa isso ao seu favor também faria a tomada de decisões melhor. Usar uma ou mais casas de vegetação para controle e comparação de velocidade de treinamento entre ambiente interno e externo podem ser benéficos.

Utilizar *Deep Learning* e usar as entradas analógicas como entradas podem fazer um modelo mais preciso, utilizar saídas analógicas também para permitir um controle fino das variáveis em vez de saídas booleanas. Integrar mais técnicas de *Machine learning* para previsão meteorológica como mostrado no trabalhos referenciados para fornecer mais dados ao modelo, assim melhorando mais a tomada de decisão.

Uma provável melhoria poderia ser usar apenas o *Raspeberry Pi 4* com tudo conectado a ele, assim existiriam menos pontos de falha no projeto todo, entretanto o uso de *ESP 32* foi de grande importância para separar e simplificar o desenvolvimento.

### Considerações finais

Mesmo o modelo não tendo convergido e a casa de vegetação ter alguns problemas muito foi aprendido sobre manuseio com hardware, uso de ML, desenvolvimento de servidores e deploy usando contêiner *docker*, possibilitando no futuro um outro projeto mais preciso e trazendo novos conhecimentos.



## Referências

ACHOUR, Y.; OUAMMI, A.; ZEJLI, D. Technological progresses in modern sustainable greenhouses cultivation as the path towards precision agriculture. *Renewable and Sustainable Energy Reviews*, v. 147, p. 111251, 2021. ISSN 1364-0321. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1364032121005384>>. Citado na página 10.

ACHOUR, Y. et al. Supervisory model predictive control for optimal operation of a greenhouse indoor environment coping with food-energy-water nexus. *IEEE Access*, v. 8, p. 211562–211575, 2020. Disponível em: <<https://ieeexplore.ieee.org/document/9253647>>. Citado 2 vezes nas páginas 7 e 8.

AKHTER, R.; SOFI, S. A. Precision agriculture using iot data analytics and machine learning. *Journal of King Saud University - Computer and Information Sciences*, 2021. ISSN 1319-1578. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1319157821001282>>. Citado 2 vezes nas páginas 7 e 10.

BONE, O. dos Santos e A. C. D. Redes de sensores aplicadas À agricultura. *Revista Científica FAESA*, v. 15, n. 2 Especial, p. 99–120, 2019. ISSN 2316-7327. Disponível em: <<http://revista.faesa.br/revista/index.php/Faesa/article/view/268>>. Citado 3 vezes nas páginas 7, 8 e 10.

FEI, X.; XIAO-LONG, W.; YONG, X. Development of energy saving and rapid temperature control technology for intelligent greenhouses. *IEEE Access*, v. 9, p. 29677–29685, 2021. Disponível em: <<https://ieeexplore.ieee.org/document/9353543>>. Citado 2 vezes nas páginas 7 e 9.

FILHO, B. d. S. F. *Agricultura de precisão em casas de vegetação: controle e gestão de cultivo em produção de mudas*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Araranguá, SC, Brazil, 11 2018. Disponível em: <<https://repositorio.ufsc.br/handle/123456789/191129>>. Citado 3 vezes nas páginas 7, 9 e 10.

IDDIO, E. et al. Energy efficient operation and modeling for greenhouses: A literature review. *Renewable and Sustainable Energy Reviews*, v. 117, p. 109480, 2020. ISSN 1364-0321. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1364032119306884>>. Citado na página 7.

LIAO, R. et al. Development of smart irrigation systems based on real-time soil moisture data in a greenhouse: Proof of concept. *Agricultural Water Management*, v. 245, p. 106632, 2021. ISSN 0378-3774. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S037837742032179X>>. Citado 2 vezes nas páginas 7 e 8.

LIN, D.; ZHANG, L.; XIA, X. Model predictive control of a venlo-type greenhouse system considering electrical energy, water and carbon dioxide consumption. *Applied Energy*, v. 298, p. 117163, 2021. ISSN 0306-2619. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0306261921005973>>. Citado 2 vezes nas páginas 7 e 8.

LIU, Y. et al. Anomaly detection based on machine learning in iot-based vertical plant wall for indoor climate control. *Building and Environment*, v. 183, p. 107212, 2020. ISSN 0360-1323. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360132320305837>>. Citado 2 vezes nas páginas 7 e 9.

MAHBUB, M.; HOSSAIN, M. M.; GAZI, M. S. A. Iot-cognizant cloud-assisted energy efficient embedded system for indoor intelligent lighting, air quality monitoring, and ventilation. *Internet of Things*, v. 11, p. 100266, 2020. ISSN 2542-6605. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2542660520301001>>. Citado 2 vezes nas páginas 9 e 10.

MOON, T.; Eek Son, J. Knowledge transfer for adapting pre-trained deep neural models to predict different greenhouse environments based on a low quantity of data. *Computers and Electronics in Agriculture*, v. 185, p. 106136, 2021. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S016816992100154X>>. Citado 2 vezes nas páginas 7 e 9.

MOSHARAFIAN, S. et al. Optimal lighting control in greenhouse by incorporating sunlight prediction. *Computers and Electronics in Agriculture*, v. 188, p. 106300, 2021. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168169921003173>>. Citado 2 vezes nas páginas 7 e 9.

OPENIA. *Site oficial do OPEN IA sobre aprendizado por reforço com multi agentes*. 2021. Disponível em: <<https://openai.com/blog/emergent-tool-use>>. Acesso em: 15 setembro 2021. Citado 2 vezes nas páginas 11 e 15.

PARADA, F. et al. Optimizing irrigation in urban agriculture for tomato crops in rooftop greenhouses. *Science of The Total Environment*, v. 794, p. 148689, 2021. ISSN 0048-9697. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S004896972103761X>>. Citado 2 vezes nas páginas 7 e 8.

TAO, W. et al. Review of the internet of things communication technologies in smart agriculture and challenges. *Computers and Electronics in Agriculture*, p. 106352, 2021. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168169921003690>>. Citado 2 vezes nas páginas 7 e 11.

TERUEL, B. J. Controle automatizado de casas de vegetação: variáveis climáticas e fertigação. *Revista Brasileira de Engenharia Agrícola e Ambiental [online]*, v. 14, n. 3, p. 237–245, 2010. ISSN 1807-1929. Citado na página 10.

VANIPRIYA, C. et al. Artificial intelligence enabled plant emotion xpresser in the development hydroponics system. *Materials Today: Proceedings*, v. 45, p. 5034–5040, 2021. ISSN 2214-7853. Second International Conference on Aspects of Materials Science and Engineering (ICAMSE 2021). Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2214785321006039>>. Citado 2 vezes nas páginas 9 e 10.

WANG, L.; ZHANG, H. An adaptive fuzzy hierarchical control for maintaining solar greenhouse temperature. *Computers and Electronics in Agriculture*, v. 155, p. 251–256, 2018. ISSN 0168-1699. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0168169918309141>>. Citado 2 vezes nas páginas 7 e 9.