

Sistema de irrigação inteligente, gerenciado por um *Chatbot*, para gestão hídrica no contexto agrícola

Rafael Dutra Pereira, Leonardo Barreto Campos

Coordenação de Sistemas de Informação – Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA) – Vitória da Conquista – Bahia – Brasil

contatorafael Dutra@outlook.com, leonardobcampos@gmail.com

Abstract. *The present work addresses the development of an intelligent irrigation system aimed at optimizing water usage in agriculture. Water is an increasingly scarce resource, with alarming forecasts for the near future. Agriculture consumes around 70% of all potable water, and a significant portion of this volume is wasted due to inefficient irrigation methods. This project proposes the creation of a Wireless Sensor Network (WSN) that monitors soil moisture, the water reservoir level, and triggers automated irrigation, in addition to providing a communication interface with the end user via a chatbot.*

Keywords: *Smart Agriculture, Internet of Things; ESP32; Chatbot.*

Resumo. *O presente trabalho aborda o desenvolvimento de um sistema de irrigação inteligente visando otimizar o uso da água na agricultura. A água é um recurso cada vez mais escasso, com previsões alarmantes para o futuro próximo. A agricultura consome cerca de 70% de toda a água potável, e grande parte desse volume é desperdiçado devido a métodos ineficientes de irrigação. Este projeto propõe a criação de uma rede de sensores sem fio (RSSF) que monitora a umidade do solo, o nível do reservatório de água e aciona a irrigação de forma automatizada, além de disponibilizar uma interface de comunicação, via chatbot, com o usuário final.*

Palavras-Chave: *Agricultura Inteligente; Internet das Coisas; ESP32; Chatbot.*

1. Introdução

A água é um recurso natural finito e cada vez mais escasso. No último século, o consumo de água doce aumentou seis vezes e continua a avançar a uma taxa de 1% ao ano. Com isso, estima-se que, até 2030, mais de 40% da população do planeta estará vivendo em áreas de grave estresse hídrico, caso medidas não sejam tomadas (Koncagul et al, 2021). Dentre essas medidas, é necessário diminuir a utilização de água na agricultura, pois quase 70% de toda a água potável consumida no planeta é destinada para o campo, onde 40% é gasto exclusivamente no processo de irrigação, sendo que metade deste percentual é desperdiçado. (FAO, 2020).

De acordo com Pereira, Meireles e Santos (2017), a irrigação mal executada é um dos principais motivos para o desperdício de água na agricultura. Em primeiro lugar, esse desperdício pode ser minimizado através da escolha de um método de irrigação

eficiente. Conforme Carvalho e Araújo (2010), a irrigação por gotejamento é um método que possui uma eficiência na ordem de 90%. Além disso, de acordo com Guimarães e Lima (2021), a utilização desse método poderia possibilitar uma economia de 50% da água que é utilizada hoje.

Buscando ir além, este artigo apresentará uma rede de sensores e atuadores sem fio (RSSF) para automatizar o processo de irrigação por gotejamento em uma horta urbana. Além da automatização, a RSSF irá coletar dados da irrigação e do ambiente e disponibilizar para o usuário através de um interface de comunicação via *chatbot*. Esses dados serão utilizados para monitorar a irrigação e, posteriormente, na quarta seção, realizar a análise do impacto da implementação deste projeto.

As demais seções deste artigo estão divididas da seguinte forma: na segunda parte, serão apresentados resultados e propostas de trabalhos anteriores que estão relacionados com o contexto deste; na terceira parte, serão apresentados os métodos e as tecnologias utilizadas na construção do projeto; na quarta parte, são apresentados os resultados do projeto, assim como uma análise dos dados obtidos e; na última parte, são apresentadas as conclusões do autor.

2. Trabalhos Relacionados

Para construir a base fundamental para o desenvolvimento do projeto, o autor pesquisou por trabalhos correlatos que pudessem apresentar o estado da arte dos sistemas de irrigação automatizados. Para isso, foram coletados trabalhos publicados entre os anos de 2018 e 2023, onde um resumo dos mesmos serão apresentados nos parágrafos a seguir.

Em Louro (2022), é desenvolvido um sistema de irrigação automatizado e controlado por um aplicativo *android*. Para realizar a rega, o sistema verifica o nível de umidade do solo e aciona a irrigação quando o nível está abaixo do ideal. Além disso, o projeto também conta com outros dois sensores, um de chuva, levando em consideração as condições climáticas, e outro de UV (Radiação Ultravioleta), evitando que o sistema perca água por evaporação. Ademais, na interface *android* do sistema, é possível definir uma irrigação customizada, onde o usuário define o intervalo de irrigação e a duração da rega.

Em Alia, Natsution e Satriyo (2023), um sistema de irrigação automatizado é construído utilizando a irrigação por aspersão. No projeto, os autores utilizam um único microcontrolador, o *ESP32*, para receber os dados dos sensores, efetuar a comunicação com a base de dados e acionar a irrigação. A interface do sistema foi construída utilizando uma ferramenta de *low-code*, chamada de *Blynk*. Nela, o sistema oferece a possibilidade de realizar a irrigação através de três métodos diferentes: manual, semi automático e automático. O primeiro método permite que o usuário ligue e desligue a bomba através da interface; o segundo solicita que o usuário indique o horário do dia e a duração da irrigação; enquanto que o último solicita que o usuário defina o ponto de umidade ideal para a cultura.

Em Moura e Nobre (2022), foi desenvolvido um sistema de irrigação automatizado para o cultivo de alface. Para testar e validar o sistema, os autores implementaram-o numa área de cultivo com 25m². Apesar do sistema coletar quatro

variáveis (luminosidade, umidade do solo, temperatura do ar e umidade do ar), utiliza apenas a umidade do solo em consideração no processo de irrigação. Esse sistema foi desenvolvido com dois nós: o primeiro com um microcontrolador *ESP8266*, servindo para realizar o controle (acionar a bomba da irrigação e enviar os dados para o banco), e o segundo com o *Arduino Pro Mini*, utilizado para realizar a coleta dos dados dos sensores. Por fim, para visualizar os dados, os autores construíram uma aplicação mobile utilizando uma biblioteca JavaScript, o *React Native*.

Em Abreu (2023), é desenvolvido um sistema de irrigação automatizado para hortas verticais. O sistema é capaz de coletar a umidade e temperatura do ar, o nível do reservatório de água e a umidade do solo, registrando o histórico de dados em uma planilha do *Google Sheets*. A partir desses dados, o sistema decide se é necessário irrigar ou não. Além de ser possível consultar as variáveis através de um aplicativo *android* nativo, a autora desenvolveu um *chatbot*, utilizando o serviço de mensageria *Telegram*, que envia notificações para o usuário sempre que o nível de reservatório de água está abaixo do esperado.

Em Muxito et al (2018), tem-se a construção de um protótipo de um sistema de irrigação automatizado para um ambiente de produção de hortifrutigranjeiros. O projeto, ainda em desenvolvimento, pretende automatizar a irrigação realizada por pivôs em uma área de cinco mil metros quadrados. Para isso, o sistema contará com um central de controle, que usa um *Arduino Uno R3*, para receber as leituras de um sensor de umidade e acionar/desligar uma bomba de irrigação, conforme o nível da umidade. Além disso, o sistema conta com um *webservice*, exposto através de um microcontrolador *ESP8266*, para disponibilizar os dados coletados pelo projeto. Esses dados podem ser consultados através de um painel Web, onde também é possível configurar o sistema (definir o nível de umidade necessário para a irrigação ou acionar a irrigação manualmente).

Na Tabela 1, é possível visualizar uma comparação entre os trabalhos relacionados e o trabalho apresentado neste artigo.

Tabela 1. Comparação entre os trabalhos relacionados e o trabalho do autor.

Recursos	Louro, 2022	Alia, Natsution e Satriyo, 2023	Moura e Nobre, 2022	Abreu, 2023	Muxito et al, 2018	Autor
Sensores	Umidade do solo, UV e chuva	Umidade do solo, temperatura do ar, temperatura do solo, reservatório de água e pH do solo.	Umidade do solo, temperatura do ar, umidade do ar e luminosidade	Umidade do solo, temperatura do ar, umidade do ar e reservatório de água.	Umidade do solo	Umidade do solo e reservatório de água.

Microcontrolador	<i>ATmega2560</i>	<i>ESP32</i>	<i>ATmega328P e ESP8266</i>	<i>ESP8266</i>	<i>ATmega328</i>	<i>ESP32</i>
RSSF?	NÃO	NÃO	NÃO	NÃO	NÃO	SIM
Interface	<i>Android</i>	<i>Android</i>	<i>Android</i>	<i>Android, Web e chatbot no Telegram</i>	Web	<i>Chatbot no WhatsApp</i>
Espaço de aplicação	10m ²	-	25m ²	-	5000m ²	360m ²

3. Irrigação Inteligente

Nesta seção, serão apresentadas as etapas envolvidas na construção do sistema de irrigação inteligente. Na Figura 1, é possível visualizar a arquitetura geral do projeto. Primeiro, temos a rede de sensores sem fio, responsável por realizar o monitoramento do ambiente, por meio dos nós 1 e 2, controlar a irrigação com o nó 3 e enviar os dados para o *backend*, através do nó 4. Em seguida, temos o *backend*, que consiste em uma *Application Programming Interface (API) REST* feita em *Oracle® Rest Data Services (ORDS)*, sendo responsável por receber os dados do nó 4 e salvar no banco de dados, além de expor um *webhook* para receber as requisições da *Meta® Cloud API*. Por fim, temos a interação do usuário, através do *WhatsApp® Messenger*, com um *bot* criado na *Meta® Cloud*. Todas essas etapas serão exploradas nas próximas subseções.

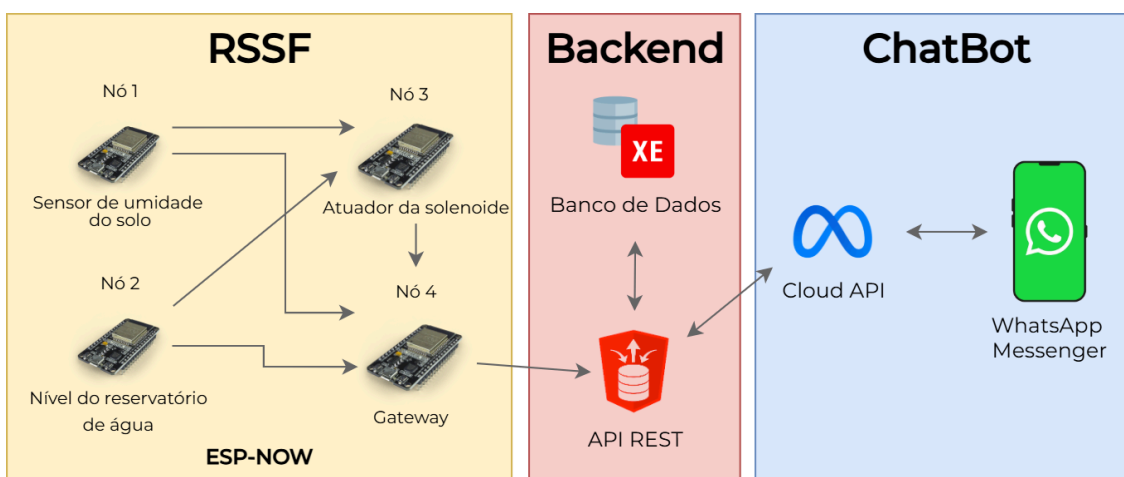


Figura 1. Arquitetura do sistema de irrigação inteligente.

3.1 Desenvolvimento do *backend*

O *backend* consiste em um banco de dados e uma *API REST*. Para o armazenamento das informações, foi utilizado a versão gratuita do banco de dados da *Oracle®*, o *Oracle®*

Express Edition (XE). Decidiu-se por utilizar o *XE* porque o local de implantação do sistema já possuía uma infraestrutura e um *software* que utilizam o banco. Então, ao utilizar a mesma infraestrutura, além de facilitar o desenvolvimento, também seria possível, por exemplo, realizar uma integração com esse *software*, apesar disso não estar incluído no escopo deste projeto.

O núcleo do modelo de dados é composto por duas tabelas. A primeira tabela servirá para armazenar as variáveis do ambiente que são coletadas pelos nós (nível do reservatório de água e a umidade do solo), registrando o valor e o horário de coleta. Por sua vez, a segunda tabela armazena os horários de início e término de cada irrigação. Para alimentar essas duas tabelas, foi utilizado o *Oracle® Rest Data Services (ORDS)* para criar uma *API REST*.

O *ORDS* é uma ferramenta disponível gratuitamente em qualquer edição de banco de dados da *Oracle®*. Essa ferramenta interpreta chamadas *HTTP (Hypertext Transfer Protocol)* e permite que o desenvolvedor consiga lidar com essas chamadas através da linguagem procedural da *Oracle®*, o *PL/SQL*. A partir disso, foi criada uma *API* com três *endpoints*, que podem ser visualizados, de forma sumarizada, na Figura 2. Os dois primeiros *endpoints* foram criados para receber as informações da RSSF. O primeiro *endpoint* (*irrigacao/solenoide* na Figura 2) espera receber uma requisição do terceiro nó sempre que a irrigação for iniciada ou interrompida. Já o segundo *endpoint* (*irrigacao/variavel* na Figura 2) foi projetado para receber as informações dos nós 1 e 2 e espera receber, no corpo da requisição, o tipo de variável (nível do reservatório ou umidade do solo) e o valor coletado. Por fim, o último *endpoint* (*webhook/message* na Figura 2) foi projetado para atuar como um *webhook*, esperando receber as requisições do *chatbot*, enviadas pela *Meta® Cloud*, que será apresentada em uma subseção seguinte.

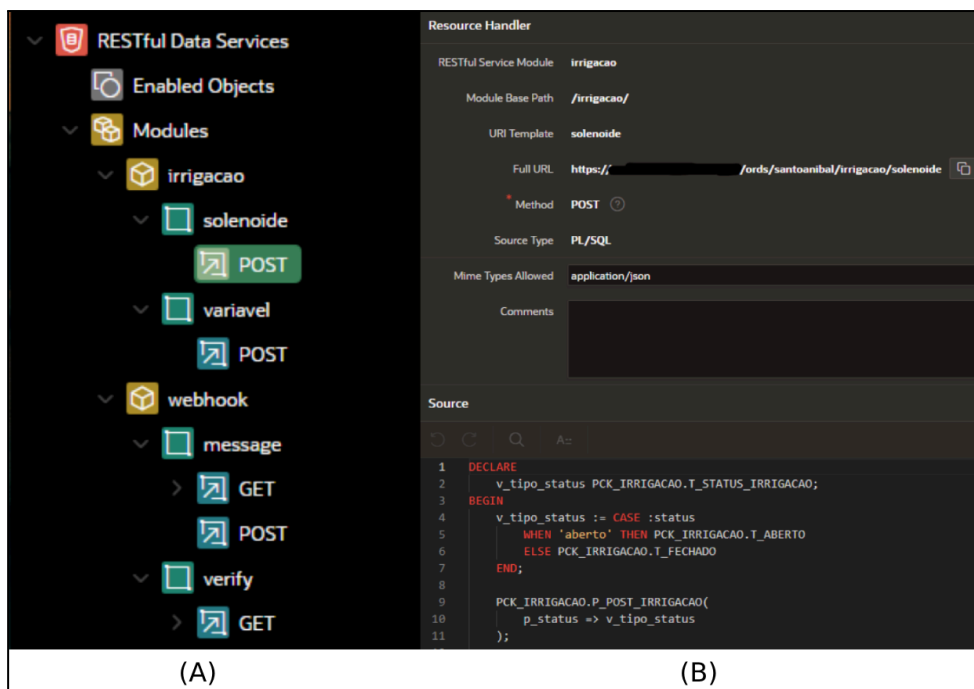


Figura 2. (A) Endpoints disponíveis. (B) Definição do endpoint *irrigacao/solenoide*.

Com o *backend* finalizado, é possível partir para a construção das demais partes do projeto: a RSSF e o *chatbot*.

3.2 Desenvolvimento da RSSF

Uma rede de sensores e atuadores sem fio consiste em dispositivos interdependentes (nós) que se comunicam entre si através de radiofrequência. Por conta disso, é necessário que os nós concordem, entre si, em relação a qual protocolo será utilizado para enviar e interpretar as ondas de rádio, onde cada protocolo possui suas vantagens e desvantagens, e sua escolha irá depender dos requisitos do projeto. No caso deste sistema, o protocolo utilizado na comunicação dos nós foi o *ESP-NOW*, que já vem integrado ao microcontrolador utilizado em cada nó, o *ESP32*.

O *ESP-NOW* é um protocolo desenvolvido pela *Espressif*¹, que permite uma comunicação de média distância, alta velocidade e baixo consumo de energia. Assim, tendo em vista a simplicidade da utilização do protocolo e, considerando que, no caso deste projeto, os nós não ultrapassam a distância de 300 metros entre si, optou-se por utilizar o *ESP-NOW*.

Definido o protocolo e o microcontrolador, é importante esclarecer o fluxo de comunicação que está ilustrado na Figura 1.

O primeiro nó é responsável por coletar o percentual de umidade do solo a cada 5 minutos. Em seguida, via *ESP-NOW*, o dado é enviado para outro nó: o terceiro, que é responsável por iniciar ou desligar a irrigação. Para isso, o terceiro nó deve verificar se a umidade (dado recebido) está abaixo do mínimo ou acima do máximo.

No caso de estar abaixo do mínimo, esse terceiro nó deve solicitar que o segundo nó realize uma leitura do nível do reservatório de água. Com isso, o segundo nó realiza a leitura e comunica o valor ao terceiro nó. Dessa forma, se o nível da umidade estiver abaixo do ideal e houver água suficiente no reservatório, o terceiro nó deve, então, acionar uma válvula solenoide para iniciar o processo de irrigação. No caso do nível de umidade estar acima do máximo, o terceiro nó deve fechar a válvula solenoide e interromper a irrigação. Durante esse fluxo, cada leitura ou acionamento deve ser enviado ao *backend*. Na Figura 3 é possível visualizar a árvore de possibilidades descrita.

¹ A *Espressif*® (<https://www.espressif.com>) é uma empresa que desenvolve dispositivos para serem utilizados em projetos de Internet das Coisas, sendo responsável pela criação do próprio *ESP32*.

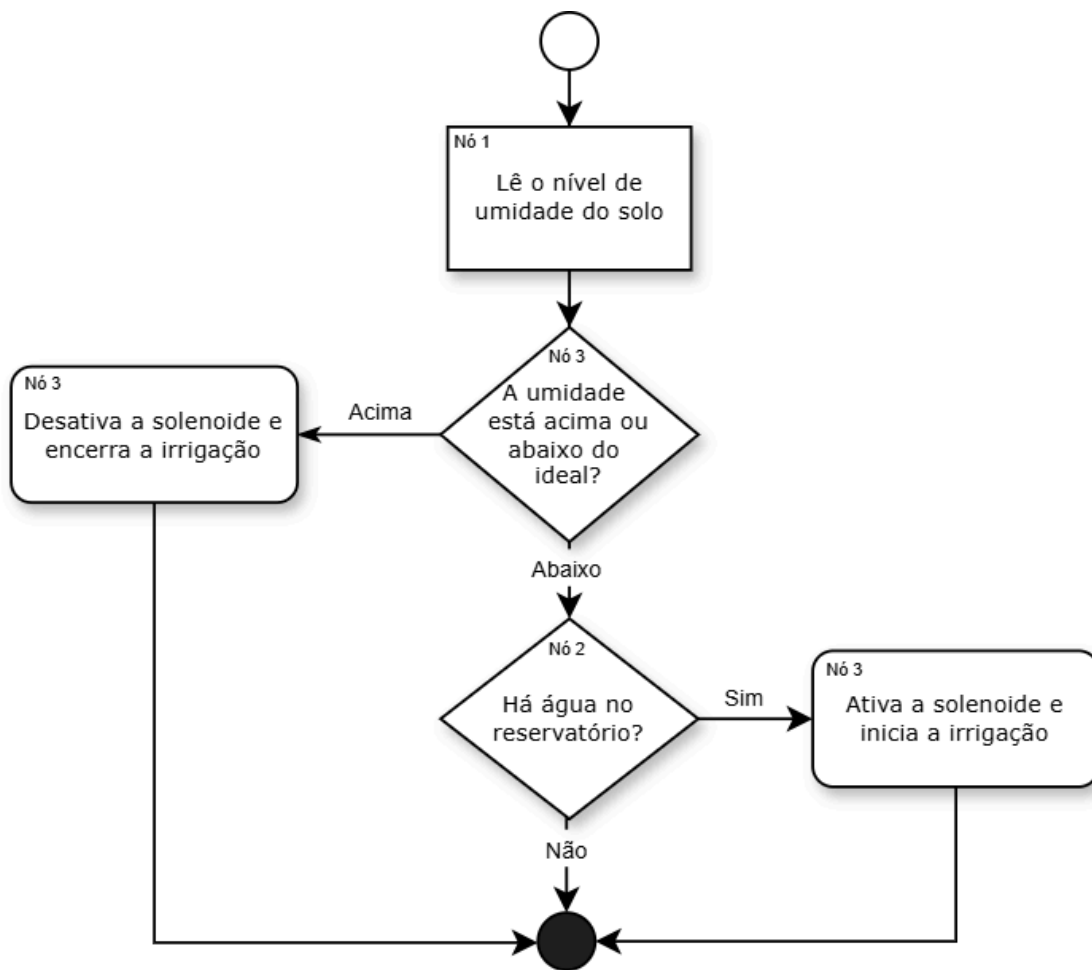


Figura 3. Fluxograma do processo de irrigação.

Para a construção dos nós, foram utilizados seis componentes principais, que podem ser visualizados na Figura 4: um microcontrolador *ESP32*, uma protoboard para facilitar as conexões, uma antena para aumentar o alcance da transmissão de dados, uma fonte de alimentação e *leds* indicadores. Com exceção do quarto nó (*gateway*), todos os demais nós possuem componentes extras que são necessários para executar tarefas específicas.



Figura 4. Componentes utilizados nos nós da rede.

O primeiro nó possui um sensor de umidade do solo. Com isso, a cada cinco minutos, esse nó realiza a leitura do nível de umidade do solo e transmite o dado. No caso de um problema na transmissão, o nó tenta realizar, em sequência, algumas outras tentativas. No caso do envio bem sucedido, o sistema entra em modo de sono profundo, economizando energia.

O segundo possui um sensor ultrassônico que é utilizado para medir o nível do reservatório de água com base na distância que a sonda leva para sair do sensor, atingir a superfície da água e retornar.

O terceiro nó possui um relé responsável por fechar ou abrir o circuito elétrico entre a válvula solenoide e a fonte de energia elétrica. Ao ser energizada, a válvula permite que a irrigação ocorra.

Por fim, com a RSSF e o *backend* construídos, o próximo passo é a construção do *chatbot* para realizar a consulta aos dados.

3.3 Desenvolvimento do *chatbot*

O *chatbot* foi desenvolvido utilizando o serviço de mensageria *WhatsApp*®, por ser o mais popular e, conseqüentemente, facilitar a adaptação por parte dos usuários. Para isso, foi necessário implementar um *webhook*, no *backend*, que fosse capaz de receber as mensagens dos usuários, processá-las e responder com os dados necessários.

Para realizar a integração com o *WhatsApp*®, é necessário utilizar a *Cloud Meta*® *API*, que é uma *API* desenvolvida para que terceiros consigam interagir com as ferramentas e serviços da *Meta*®. É importante considerar que a utilização do serviço é paga em casos específicos. Na versão gratuita, é possível utilizar até 1000 janelas de conversas por mês. Uma janela de conversa é a troca de mensagens entre um usuário e o *chatbot*. A partir do momento em que o *chatbot* recebe uma mensagem de um usuário, uma janela de conversa é aberta entre os dois e dura um período de 24 horas. Nesse período, não há cobrança pela troca de mensagens entre o *chatbot* e o usuário final. No plano gratuito, não é possível que o *chatbot* inicie a janela de conversa. Então, caso fosse necessário que o *chatbot* enviasse, espontaneamente, notificações sobre o estado do sistema automatizado, por exemplo, seria necessário utilizar a versão paga do serviço. Além disso, caso o número de usuários fosse grande (ultrapassando as 1000 janelas gratuitas), também seria necessário pagar pelo serviço. Entretanto, para o escopo deste projeto, a versão gratuita atende aos requisitos.

Após criar uma conta na *Meta*® *Cloud* e realizar a configuração do *webhook* para receber todas as mensagens enviadas ao número que atuará como o *bot*, é necessário preparar o *backend* para receber as requisições e respondê-las. No caso deste projeto, sempre que o *chatbot* recebe uma nova mensagem, ele deve responder com uma lista de opções para que o usuário consiga indicar qual informação deseja obter. Ao selecionar uma opção, os dados devem ser resgatados nas tabelas do banco e, após o processamento, o *chatbot* responde com as informações solicitadas.

4. Resultados e discussão

Nesta seção, serão apresentados os resultados da construção do projeto, além da discussão sobre o impacto causado e dos problemas encontrados.

Na figura 5, é possível visualizar o primeiro e o terceiro nó em atividade. Durante a construção do primeiro nó, o autor encontrou muitos relatos acerca da baixa durabilidade do sensor de umidade utilizado. Esse sensor, para identificar o nível de umidade, atua de forma resistiva, ou seja, ele verifica a resistência elétrica do solo. Assim, quanto maior a umidade, menor a resistência elétrica e vice versa. Entretanto, caso esse sensor fique ativado por um longo período, as hastes, dentro de poucos meses, podem ficar corroídas, o que pode acabar atrapalhando a leitura ao longo do tempo. Para ampliar a vida útil do sensor, inicialmente o microcontrolador foi programado para ativá-lo somente no momento em que era necessário realizar uma leitura. Assim, a cada cinco minutos, o sensor ficava ativado durante alguns segundos e, logo que a leitura era finalizada, a alimentação do sensor era cortada. Entretanto, ao final do projeto, o sensor foi trocado por um outro com uma maior durabilidade, ficando permanentemente ativado.

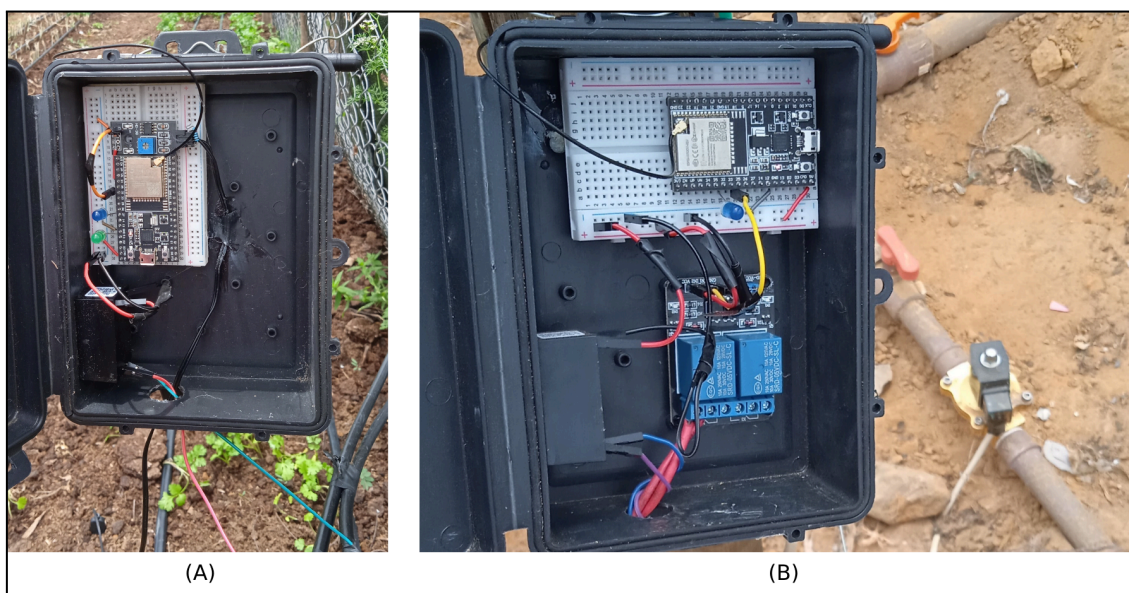


Figura 5. (A) Nó responsável por monitorar a umidade do solo. (B) Nó responsável por controlar a válvula solenoide.

Outro problema encontrado com a RSSF foi a transmissão dos dados para o *backend*. Inicialmente, todos os nós iriam, de forma independente, fazer as requisições diretamente ao *backend*, sem a necessidade de um *gateway*. Entretanto, devido a distância entre os nós e o roteador conectado à Internet (cerca de 300 metros), essa comunicação estava ficando comprometida, pois o sinal WiFi estava fraco. Assim, a solução foi adicionar um outro nó, o *gateway*, próximo ao roteador, que recebe todas as requisições dos outros nós, se conecta à Internet e envia os dados ao *backend*. Com isso, os dados passaram a ser enviados corretamente, permitindo a consulta através do *chatbot*.

Na Figura 6, é possível visualizar a interação com o *chatbot* para a consulta dos dados. Como o *chatbot* foi implementado com o *WhatsApp*®, não houve dificuldades de adaptação por parte dos agricultores, pois já tinham o hábito de utilizar o aplicativo de mensageria. Inicialmente, o *chatbot* contava com uma opção para consultar se a irrigação estava ativada ou desativada e outra opção para verificar o nível de umidade do solo. Posteriormente, por solicitações dos agricultores, foi adicionada uma opção para consultar o histórico de irrigações.

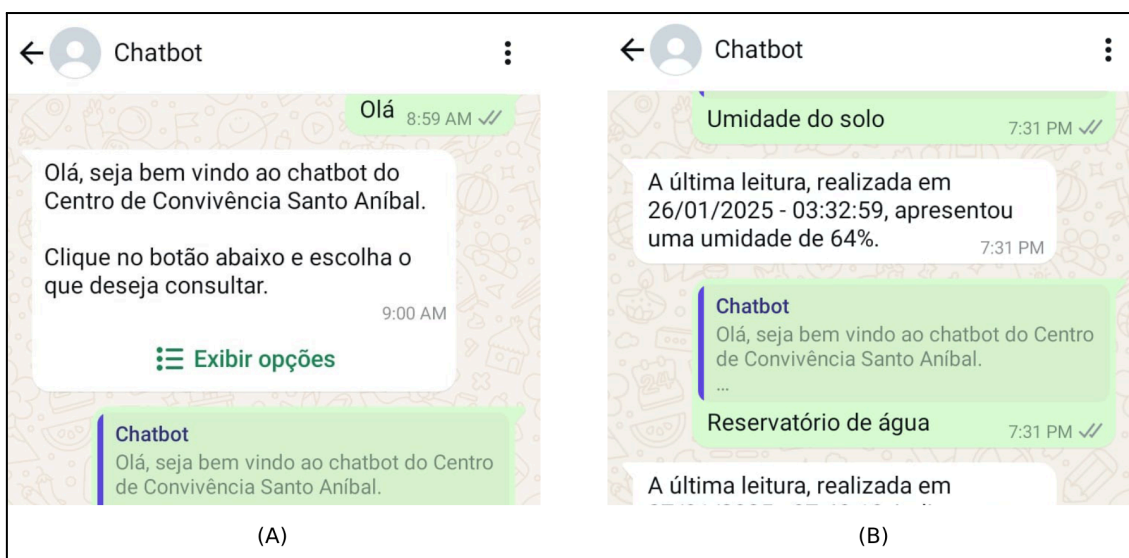


Figura 6. (A) Iniciando *chatbot*. (B) *Chatbot* respondendo com as informações solicitadas.

Antes da implantação do sistema, o processo de irrigação era iniciado de forma manual, gastando 5 mil litros de água por dia. Como os agricultores não realizavam um sensoriamento do solo, tinham o hábito de irrigar baseando-se na umidade da superfície do solo. Caso entendessem que a superfície estava seca, iniciavam a irrigação e deixavam que toda a água do reservatório fosse consumida. Entretanto, esse método é ineficiente, porque o solo do local não é capaz de reter toda a água consumida, então ocorre o escoamento de uma grande quantidade e, conseqüentemente, um desperdício. Além disso, mesmo que a superfície esteja aparentemente seca, é possível que a parte baixa do solo ainda esteja úmida.

Com a adição do sensoriamento, o sistema passou a irrigar somente quando o solo estava abaixo do valor de umidade ideal. Inicialmente, esse valor foi definido com base em testes com uma amostra do solo completamente encharcada e outra completamente seca. Em seguida, ao instalar o sistema em campo, foi necessário realizar novos ajustes nos níveis de umidade, algo que foi simples de realizar, uma vez que o sistema já estava preparado para ser configurado através de radiofrequência, sem a necessidade de remover os equipamentos do local de cultivo.

Após a definição das configurações, passou-se para a etapa de avaliação do sistema. Entretanto, nesse período, houveram diversas chuvas na região em que o sistema foi aplicado, o que resultou na coleta de poucos dados para a avaliação, pois a irrigação acabou não sendo acionada por um bom período. Ainda assim, o sistema foi avaliado durante um mês e a irrigação passou a ser acionada em uma frequência menor

do que o processo manual, demonstrando uma propensão a diminuir o consumo de água.

5. Considerações finais

Neste projeto, foi desenvolvido um sistema de irrigação automático, que é monitorado através de um *chatbot*. O sistema foi construído para atuar através de uma rede de sensores sem fio, sem a necessidade de cabeamento para transferência dos dados. O código-fonte de cada nó da rede foi disponibilizado no *GitHub*² para contribuições futuras. Além disso, também foi desenvolvido um *backend* para realizar o armazenamento dos dados coletados e disponibilizá-los, através do *WhatsApp*®, para os agricultores. Por fim, o resultado do projeto foi satisfatório, resultando em um monitoramento mais preciso do cultivo, menos carga de trabalho para os agricultores e, claro, uma economia de água no processo de irrigação. Como mecanismo de segurança, o antigo sistema de rega manual foi mantido, garantindo uma alternativa de *backup* em caso de falha do sistema automatizado.

Entretanto, alguns pontos poderiam ser melhorados no projeto, o que pode ser utilizado como ideias para futuros projetos. Em primeiro lugar, apesar do sistema não utilizar cabos para a transferência dos dados, os nós ainda são alimentados através de fiação elétrica. Isso poderia ser melhorado através da utilização de placas solares, tornando o sistema mais autossustentável e independente. Em segundo lugar, o *chatbot* desenvolvido não é capaz de realizar o processamento de linguagem natural. Então, para o agricultor, não há a sensação de estar conversando com uma outra pessoa. Caso isso seja implementado, o processo de utilização do *chatbot* seria de mais fácil adaptação. Em terceiro lugar, os nós foram desenvolvidos utilizando componentes separados que estão integrados através de uma *protoboard*. No cenário de produção, isso não é o ideal, uma vez que a *protoboard* deve ser utilizada somente para a prototipação. Sendo assim, uma outra evolução seria integrar os componentes de um nó em uma única placa de circuito impresso. Isso diminuiria o espaço ocupado pelo nó, facilitaria a escalabilidade e evitaria problemas de mau contato. Em quarto lugar, para aplicações de longo alcance, seria necessário a utilização de um outro protocolo, como o *LoRa*, uma vez que o *ESP-NOW* se mostrou ineficiente entre distâncias maiores do que 300m.

Referências

- Alia, S., Natsution, I. Satriyo, P. (2023). IoT-enable smart agriculture using multiple sensors for sprinkle irrigation systems. In IOP Conference Series: Earth and Environmental Science. Vol. 1290, No. 1.
- Abreu, A. (2023). Sistema De Irrigação Automatizado Para Hortas Verticais. Universidade Federal Do Ceará.
- Carvalho, E. Araujo, L. (2010). Irrigação inteligente. In Anuário da produção de iniciação científica discente, Anhanguera Educacional.
- FAO. (2020). The State of Food and Agriculture 2020. Overcoming water challenges in agriculture. Rome.

² <https://github.com/rafael-dutr4/smart-irrigation-wsn>

- Guimarães, D. Lima, S. (2021). A agricultura e o consumo de água.
- Koncağül, E., Tran, Michael. Connor R. (2021). Relatório mundial das Nações Unidas sobre desenvolvimento dos recursos hídricos 2021: o valor da água; fatos e dados.
- Louro, P. (2022). Desenvolvimento de um sistema de irrigação automatizado e controlado por aplicativo para um quintal produtivo. Centro Federal de Educação Tecnológica de Minas Gerais.
- Moura, M. Nobre, T. (2022). Sistema de Automação e Controle Inteligente no Processo de Irrigação Usando IoT no Cultivo da Alface. Instituto Federal da Paraíba.
- Muxito, E. M., Silva, A., M., Duarte C. (2018). IoT na Agricultura – Automação de Pivôs e Canais de Irrigação com Arduino e Webservice. III Congresso Internacional Adventista de Tecnologia (CIAT). Centro Universitário Adventista de São Paulo.
- Pereira, C. Da C., Meireles, G. C. De S., Santos, R. B. (2017). Sistema de irrigação automatizado utilizando arduino uno. In Mostra Nacional De Robótica.