



Desenvolvimento de sistema para monitoramento e controle da irrigação aplicável a cultivos sem solo em ambiente protegido

Palavras-Chave: Internet das Coisas, Irrigação, Aplicação web

Renan da Silva Guedes [FEAGRI/UNICAMP]

Prof. Dr. Antonio Pires de Camargo (orientador) [FEAGRI/UNICAMP]

INTRODUÇÃO:

Com o aumento da necessidade da automação para reduzir custos com mão-de-obra e otimizar processos, a incorporação de sistemas eletrônicos para aquisição de dados e controle de processos em ambientes diversos é crescente. Avanços no tema Internet das Coisas (IoT) tem favorecido o uso de tecnologias wireless em aplicações relacionadas a agricultura inteligente (*smart agriculture*) (BORRERO; ZABALO, 2020; SADOWSKI; SPACHOS, 2020). Sensores e atuadores podem interagir como objetos IoT e ser integrados utilizando sistemas para monitoramento e controle de processos (BORGIA, 2014).

Na agricultura, sistemas de cultivo sem solo em ambiente protegido costumam ser empregados para a produção de hortaliças, plantas ornamentais e outras culturas de alto valor. Estes sistemas de cultivo podem proporcionar elevada eficiência no uso de água e nutrientes (MASSA et al., 2020), além de reduzir efeitos de sazonalidade de produção, proporcionando uma oferta mais equilibrada ao longo dos meses (ABRAHÃO et al., 2014).

O manejo apropriado da irrigação nesses sistemas é necessário para o adequado desenvolvimento das culturas, sendo que tecnologias IoT podem contribuir nestas aplicações. Esta pesquisa teve como objetivo desenvolver um sistema de baixo custo para aquisição de dados, controle e supervisão de processos aplicável ao controle da irrigação em cultivos em ambiente protegido. O maior enfoque foi direcionado para o desenvolvimento da aplicação web empregada para supervisão dos processos.

DESENVOLVIMENTO:

O sistema é um protótipo em desenvolvimento para atender demandas de uma estufa agrícola que é parte do Laboratório de Hidráulica e Irrigação (LHI/FEAGRI/UNICAMP), podendo ser

adaptado para outras situações. A estufa apresenta cobertura plástica em polietileno, tela antiafídica nas laterais e dimensões 6,40 x 18,00 x 3,00 m (Largura x Comprimento x Altura). A estufa dispõe de infraestrutura básica para a condução de experimentos com tomates, comportando 6 fileiras de cultivo de 15 m de comprimento.

A estufa contém um Raspberry PI que desempenha a função de computador servidor. Esta unidade está conectada a internet pela rede cabeada e também opera como ponto de acesso Wi-Fi (*access point*, rede WLAN) para conexão de outros dispositivos (módulos ESP32, celulares, tablets, etc.). Por motivos de segurança, a rede WLAN configurada no Raspberry PI permite apenas a comunicação local, sem conectividade com a internet. A rede WLAN permite envio e recepção de dados apenas entre os dispositivos conectados a ela, sendo que o acesso remoto via internet é restrito ao Raspberry PI.

O protocolo MQTT (*Message Queuing Telemetry Transport*) foi utilizado para comunicação de dispositivos na rede WLAN. Eclipse Mosquitto é uma aplicação do broker MQTT, que foi instalada no Raspberry PI para gerenciamento de clientes e tópicos. Para a comunicação MQTT com o navegador (*browser*) habilitou-se o protocolo WebSockets no broker MQTT. Nas páginas web, utilizou-se a biblioteca Paho JavaScript Client que utiliza o protocolo WebSockets para conexão com o broker MQTT. Simulações de conexão e transmissão de dados na rede MQTT foram realizadas utilizando o aplicativo MQTT-Explorer e via algoritmos em JavaScript/NodeJS que geram dados aleatórios simulando funcionalidades de uma rede de módulos.

O Sistema Gerenciador de Banco de Dados (SGBD) MariaDB foi instalado no Raspberry PI. Implementou-se um banco de dados com três tabelas (Figura 1) para registro de leituras de sensores (*records*), atividade de atuadores relacionados à irrigação (*actuatorsActivity*) e registro de eventos referentes a conexão de módulos ESP32 na rede IoT (*logmodules*).

logModules		records		actuatorsActivity	
id	int(11)	id	int(11)	id	int(11)
idModule	int(11)	idSensor	int(11)	idActuator	int(11)
nmeModule	varchar(30)	dt	timestamp	start	timestamp
dtOnConnect	timestamp	value	float	duration	float
				volume	float

Figura 1. Estrutura de tabelas do banco de dados

A aquisição de sinais de sensores e o controle de atuadores é realizada utilizando módulos compostos por microcontroladores ESP32 e periféricos (*shields*). Os microcontroladores ESP32 dispõem de recursos integrados para conexão Wi-Fi, facilitando a transmissão de dados através de rede WLAN utilizando o protocolo MQTT. Os módulos podem ser distribuídos no ambiente de cultivo conforme for mais conveniente para facilitar a conexão com sensores e atuadores. O firmware dos módulos ESP32 é implementado na linguagem de programação da plataforma Arduino, baseada

em C/C++. Esta plataforma apresenta inúmeras bibliotecas e vasta documentação que facilitou o desenvolvimento do projeto.

O Node.js foi instalado no Raspberry PI, sendo que este é um ambiente de execução de JavaScript que permite a execução de instruções no lado servidor (*server-side*). Tradicionalmente, JavaScript é uma linguagem utilizada para programação de páginas web, sendo interpretada somente por navegadores de internet no lado cliente (*client-side*). A principal função do Node.js foi possibilitar o desenvolvimento de aplicações *server-side* desenvolvidas em JavaScript. O uso de JavaScript para acessar banco de dados e outras aplicações *server-side* torna o Node.js bastante interessante por permitir desenvolver aplicações *client-side* e *server-side* com a mesma linguagem de programação.

As páginas web foram desenvolvidas utilizando basicamente HTML, CSS e jQuery, sendo instalado o servidor web Apache no Raspberry PI. Para facilitar o desenvolvimento das interfaces gráficas utilizou-se ainda Bootstrap 5 e AdminLTE, sendo que ambos caracterizam recursos de acesso livre. Para a geração de gráficos utilizou-se a biblioteca Flot. As operações com o banco de dados utilizam requisições assíncronas (AJAX) implementadas utilizando jQuery. No início do projeto foram realizados testes com os frameworks ReactJS e React Native, mas desafios diversos de integração de funcionalidades não puderam ser superados, inviabilizando o uso desses frameworks.

Desenvolveu-se também um algoritmo em JavaScript para processar as requisições AJAX e realizar consultas e outras rotinas no banco de dados (*mysqlBackground.js*). Esse algoritmo utilizou Node.js e os seguintes pacotes: cors, mysql, express, bodyParser, jsonexport, fs e path. O retorno de requisições sempre foi realizado utilizando o formato json.

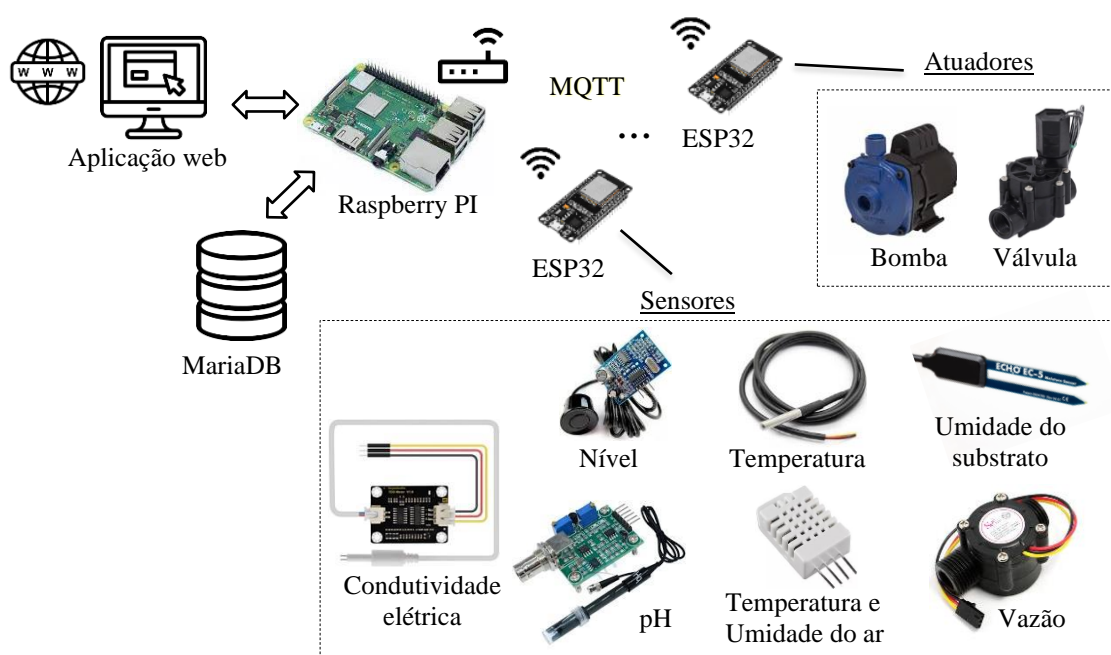


Figura 2. Componentes básicos e principais blocos do sistema

A *homepage* estruturada apresenta elementos do HTML, de modo que foi feita a divisão entre cabeçalho, barras laterais de navegação, corpo do documento e rodapé. No cabeçalho foram incluídas duas opções: *Home* e *contato*. Com a utilização do menu lateral foram ancoradas opções como: *Principal*, *Gráficos*, *Banco de dados* e *Configurações*. Tanto os menus quanto o conteúdo das páginas web podem ser reconfigurados conforme as necessidades da aplicação.

A Figura 3 ilustra a interface gráfica do usuário (GUI) desenvolvida. Na página principal é possível monitorar variáveis e controlar atuadores diretamente via MQTT.

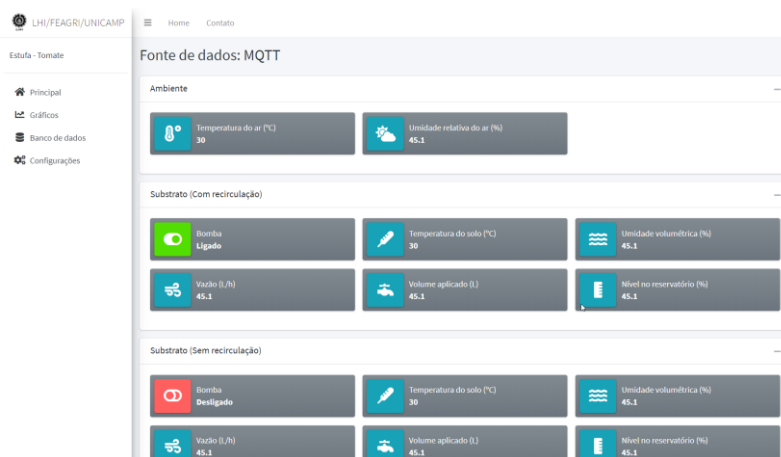


Figura 3. Interface gráfica da aplicação web desenvolvida

Na página dos Gráficos, implementou-se a atualização dos dados do sistema de cultivo a cada atualização de dados via MQTT, sendo que os gráficos contêm duas séries de dados. A primeira corresponde ao valor de um determinado atributo medido na estufa (temperatura do solo, umidade volumétrica, vazão, condutividade elétrica, etc) e atualizado em função do tempo. Os pontos registrados são conectados por meio de uma reta para facilitar a visualização da variabilidade dos dados. A segunda série consiste numa reta horizontal que representa o valor médio do histórico de valores registrados até o instante atual. Esse recurso utiliza como artifício a inserção de itens em *arrays* combinada a uma função que calcula a média baseada nos valores armazenados e no comprimento do *array*. A Figura 4 apresenta um exemplo de gráfico de temperatura do ar. A tabela à direita com as colunas *Data/Hora* e *T (°C)* expõe os últimos dados coletados. Os dados mostrados são valores aleatórios utilizados nas simulações para teste do sistema.

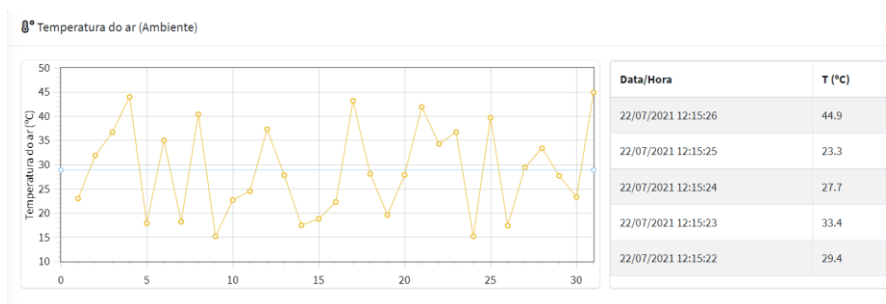


Figura 4. Exemplo de gráfico para monitoramento da temperatura do ar

Na página do Banco de dados (Figura 5) é possível recuperar dados das tabelas do banco, filtrando resultados por intervalo de datas. Também é possível efetuar download dos resultados em um arquivo no formato CSV. Do lado servidor, a aplicação *mysqlBackground.js* desenvolvida em NodeJs, é responsável por processar as requisições geradas na aplicação web e retornar resultados no formato json.

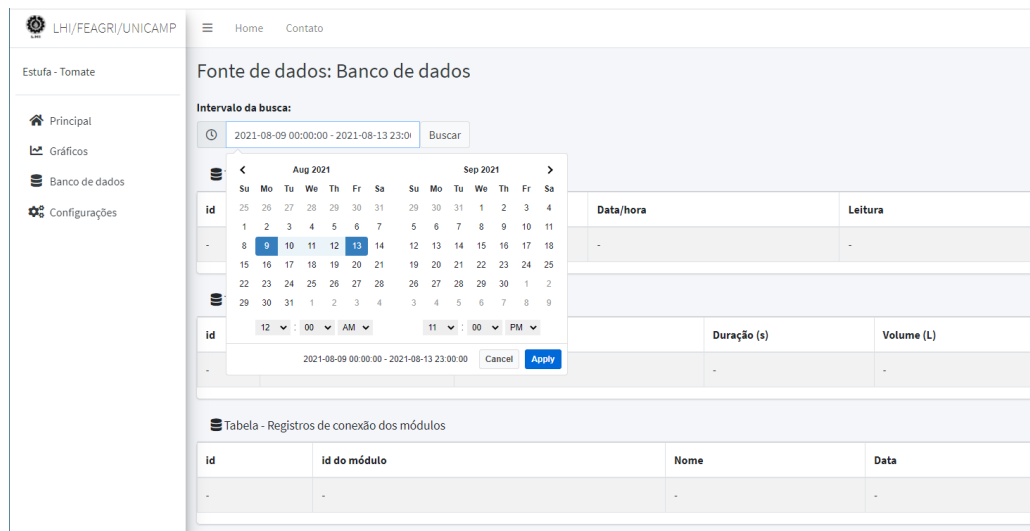


Figura 5. Página desenvolvida para consulta ao banco de dados

CONCLUSÕES:

Desenvolveu-se uma aplicação web que foi integrada a um protótipo de solução IoT para monitoramento e controle da irrigação de cultivos em ambiente protegido. As tecnologias empregadas apresentam baixo custo, são versáteis e escaláveis apresentando potencial para uso em atividades didáticas e pesquisas.

As soluções de hardware e software empregados nessa pesquisa servem de base para aprimoramentos futuros e para uso em outros problemas envolvendo aplicações agrícolas.

BIBLIOGRAFIA:

ABRAHÃO, B.; SILVA, A. R.; PAGIUCA, L. G. Cultivo protegido: Em busca de mais eficiência produtiva. Disponível em: https://www.cepea.esalq.usp.br/hfbrasil/edicoes/132/mat_capa.pdf. Acesso em: 4 ago. 2021.

BORGIA, E. The Internet of Things vision: Key features, applications and open issues. *Computer Communications*, v. 54, p. 1–31, 2014.

BORRERO, J. D.; ZABALO, A. An Autonomous Wireless Device for Real-Time Monitoring of Water Needs. *Sensors*, v. 20, n. 7, p. 1-16, 2020.

MASSA, D.; MAGÁN, J. J.; MONTESANO, F.F.; TZORTZAKIS, N. Minimizing water and nutrient losses from soilless cropping in southern Europe. *Agricultural Water Management*, v. 241, p. 1-14, 2020.

SADOWSKI, S.; SPACHOS, P. Wireless technologies for smart agricultural monitoring using internet of things devices with energy harvesting capabilities. *Computers and Electronics in Agriculture*, v. 172, p. 1-8, 2020.