

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA
CURSO DE ENGENHARIA DE COMPUTAÇÃO
PROJETO DE SISTEMAS EMBARCADOS**

**SISTEMA DE IRRIGAÇÃO PARA
HORTAS RESIDENCIAIS**

por

**Jonas Cesconetto
Paulo Henrique Trindade**

Itajaí (SC), outubro de 2019

**UNIVERSIDADE DO VALE DO ITAJAÍ
ESCOLA DO MAR, CIÊNCIA E TECNOLOGIA
CURSO DE ENGENHARIA DE COMPUTAÇÃO
PROJETO DE SISTEMAS EMBARCADOS**

**SISTEMA DE IRRIGAÇÃO PARA
HORTAS RESIDENCIAIS**

por

Jonas Cesconetto
Paulo Henrique Trindade

Relatório apresentado como requisito parcial da disciplina Projeto de Sistemas Embarcados do Curso de Engenharia de Computação para análise e aprovação.

Professores Responsáveis: Cesar Albenes Zeferino
Paulo Roberto Valim

Itajaí (SC), outubro de 2019

SUMÁRIO

1 INTRODUÇÃO.....	4
1.1 DEFINIÇÃO DO PROBLEMA.....	5
1.2 SOLUÇÕES EXISTENTES.....	6
1.3 SOLUÇÃO PROPOSTA	7
1.4 MERCADO.....	9
1.5 ANÁLISE DE VIABILIDADE	9
2 PROJETO	10
2.1 VISÃO GERAL.....	10
2.2 PREMISSAS.....	10
2.3 ANÁLISE DE REQUISITOS.....	11
2.4 ARQUITETURA DE HARDWARE - SISTEMA EMBARCADO.....	13
2.5 ARQUITETURA DE SOFTWARE - SISTEMA EMBARCADO.....	13
2.6 ARQUITETURA DE HARDWARE - SISTEMA WEB.....	15
2.7 ARQUITETURA DE SOFTWARE - SISTEMA WEB.....	15
2.8 LISTA DE MATERIAIS E ORÇAMENTO	16
2.9 PLANEJAMENTO	17
2.10 ANÁLISE DE TEMPO REAL	18
2.11 CRONOGRAMA	20
2.12 ANÁLISE DE RISCOS.....	20
3 DESENVOLVIMENTO	21
3.1 IMPLEMENTAÇÃO	21
3.2 VERIFICAÇÃO.....	25
3.3 RESULTADOS.....	26
4 CONSIDERAÇÕES FINAIS / CONCLUSÕES	32

1 INTRODUÇÃO

Kevin Ashton definiu o termo *Internet of Things* (Internet das Coisas) como a capacidade de todos dispositivos estarem conectados à Internet, essa capacidade auxilia em inúmeras tarefas de nosso dia a dia (Ashton, 2009). Neste contexto, onde todos os dispositivos de um ambiente estão conectados e trocando informações, questões como segurança, privacidade, confiabilidade e interoperabilidade dos dados são levadas em conta para conceber um ambiente inteligente.

O rápido desenvolvimento de dispositivos portáteis inteligentes oferece uma oportunidade para reconhecer o comportamento do usuário e analisar a relação entre o usuário e a demanda do dispositivo no ambiente. Acredita-se que em ambientes inteligentes os usuários não devem prestar muita atenção ao controle de seus aparelhos. Assim, um sistema doméstico inteligente "centrado no ser humano" deve aprender o comportamento do usuário e exigir automaticamente e minimizar o controle dos usuários para melhorar o conforto e a conveniência da vida do usuário (Silva, et al., 2019)

Por ambientes inteligentes entende-se que são ambientes que possuem dispositivos e sensores, os quais permitem compartilhar informações entre plataformas e aplicação com a intenção de alcançar a interoperabilidade. Um sensor é um dispositivo que responde a um estímulo físico/químico de maneira específica e que pode ser transformado em outra grandeza física para fins de medição e/ou monitoramento (Li, Yan, & Chang, 2018).

Casas inteligentes vêm se tornando uma tendência, uma vez que os dispositivos ganham conectividade e são mais acessíveis de se adquirir. O conceito cidades inteligentes (Smart Cities) operacionaliza a resolução de problemas, tais como o trânsito, poluição, segurança pública, dentre outros aplicando tecnologias da informação e comunicação. Dentro deste aspecto, enquadra-se os ambientes inteligentes, termo originado da computação ubíqua, que alinhado as tendências de IoT estão cada vez mais presentes no nosso cotidiano (Silva, et al., 2019).

A agricultura vertical tem como objetivo fornecer máxima produção com mínimos impactos ambientais, minimizando o uso da terra. Esse tipo de plantio tem recebido inúmeros investimentos e avanços tecnológicos. Pesquisas de mercado apontam que a agricultura vertical em 2025 terá um faturamento de 9,9 bilhões de dólares, nos quais é justificado o aumento no uso de sensores da Internet das Coisas (IoT) para produção. As

tecnologias recentes de Internet e comunicação são úteis no cenário agrícola para coletar informações de vários locais (Chaudhry & Mishra, 2019).

Para isso, este trabalho pretende contribuir com o desenvolvimento de um protótipo para o cultivo de hortas residenciais, onde as plantas podem variar de hortaliças, frutíferas, leguminosas, temperos e flores. A agricultura residencial traz consigo características terapeutas, harmonização de ambiente, além de possibilitar o consumo de produtos de alta qualidade, por meio de uma produção orgânica. Contudo, a redução dos espaços residenciais impõe a necessidade de verticalização do cultivo, alinhado a vida urbana dos dias atuais, inviabiliza a dedicação à atividade do cultivo. Este projeto tem como objetivo a implementação de sistema de irrigação automático para plantas em um ambiente residenciais.

1.1 DEFINIÇÃO DO PROBLEMA

Diante da necessidade de melhorar a gestão de recursos naturais, uma nova demanda pelo desenvolvimento de novas tecnologias que auxiliem e melhorem a qualidade do setor de agricultura e agronegócio. A agricultura é o setor econômico que mais utiliza recursos naturais, principalmente hídricos. A produção de soja por exemplo, em 2005, utilizou em torno de 50 bilhões de litros cúbicos (Pereira & Oliveira, 2018).

É comum nas grandes em pequenas cidades, pessoas trabalhem em locais distante de suas residências, obrigando por muitas vezes, trabalhadores se deslocarem grandes distâncias, para que possam chegar ao seu local de trabalho. Essa jornada de trabalho dificulta o cultivo de plantas em residência. Condições como temperatura e humidade do solo, podem variar muito nas diversas regiões do mundo, o que acaba comprometendo o cultivo residencial das plantas.

Dado que existe uma grande variedade espécies que possam ser cultivadas dentro de casa, cada uma com suas particularidades, cada uma com necessidades diferentes, condições específicas para que possam se permanecer saudáveis. Sendo assim, com a utilização de sensores de umidade do solo e temperatura é possível identificar o nível de necessidade de cada tipo de cultura, possibilitando assim, o mapeamento de calor e necessidade de irrigação em regiões específicas.

Pensando nesse contexto, apresentamos o projeto de baixo custo para irrigação e monitoramento em tempo real das condições do solo, o que viabiliza o cultivo residencial de plantas. Onde podemos aplicar tanto em um vaso, uma pequena horta vertical ou até mesmo um pequeno jardim de fachada, podendo ser aplicado à qualquer residência.

1.2 SOLUÇÕES EXISTENTES

No mercado existem diversas opções para irrigação, e tecnologias variadas, dentre eles estão:

Rain Bird F452

O ESP-RZX dispõe de funcionalidades de programação flexíveis que tornam o controlador ideal para uma ampla variedade de aplicações, incluindo sistemas de irrigação residenciais e comerciais de pequeno porte. O temporizador ESP-RZX foi concebido com a facilidade de utilização em mente. A programação por estações permite programar todas as válvulas de forma independente: acabou-se a explicação dos “programas” aos utilizadores finais, praticamente dispensando as visitas do técnico. O grande display LCD mostra toda a programação de cada estação em simultâneo. A interface de utilizador com gráficos simples é fácil de explicar e coloca todas as funcionalidades do controlador ao alcance dos seus dedos (Canal Agricola, s.d.).

Rain Bird St8i

Mantenha contato com o seu quintal usando o novo controlador interno ST8i da Rain Bird. Ele substitui qualquer timer padrão, usando os mesmos fios da zona. Basta montá-lo, conectá-lo ao wifi da sua casa e fazer o download de um aplicativo Rain Bird gratuito que orienta você na programação (Mercado Livre, s.d.).

O temporizador WiFi para irrigação inteligente da Rain Bird coloca o controle do seu sistema de aspersão na palma da sua mão, economizando tempo, água e dinheiro. O aplicativo Rain Bird permite o controle de vários temporizadores localizados em qualquer lugar do mundo; instale o aplicativo em vários dispositivos ou compartilhe o controle com segurança com membros da família e paisagistas. Compatível com: Apple iOS 8 ou superior e Android 4.4 (KitKat) ou posterior. Conecte-se às funções de programação do temporizador via: dispositivo móvel através de uma rede Wi-Fi, mesmo fora de casa; diretamente do dispositivo móvel quando estiver ao alcance; ou através do painel de controle frontal do timer (Mercado Livre, s.d.).

É simples personalizar as programações de rega que podem ser ajustadas automaticamente o ano todo. Cada zona de rega pode ser definida com um nome, imagem, frequência, horário de início e tempo de execução diferentes para garantir plantas saudáveis e uma conta de água mais baixa (Mercado Livre, s.d.).

Kit Microaspersor De Irrigação Triangular

A Amanco leva a inovação para o mercado de jardinagem, e lança a linha para irrigação doméstica Amanco Jardim. São produtos utilizados no cuidado e manutenção de áreas verdes residenciais, jardins de condomínios, clubes, ginásios, tudo com a qualidade já reconhecida Amanco (Americanas, s.d.).

Vantagens: Praticidade para rega programada, promovendo economia de água, com Liga e Desliga automaticamente, versátil, possui produtos para as necessidades de irrigação domésticas e muito fácil de montar, basta rosquear na torneira, engatar a mangueira, programar a frequência e tempo de rega. Além de oferecer opções de frequências e regas, diárias, dia sim dia não, dois dias sim dois dias não ou semanal (Americanas, s.d.).

1.3 SOLUÇÃO PROPOSTA

Este projeto tem como finalidade oferecer um sistema automatizado de irrigação domésticas, que permita o cultivo em áreas fechadas, monitore as condições de temperatura ambiente e a humidade do solo, além informar o nível baixo de água no reservatório. As informações serão disponibilizadas em tempo real, e monitoradas por um sistema web.

Existem algumas soluções presentes no mercado que atendem diferentes situações. Podemos citar modelos com aplicações comerciais, como Kit de gotejamento, Timer's de irrigação, que em sua grande maioria necessitam de kits incluindo: programador, válvulas, aspersores, acessórios e tubos gotejadores. Alguns sistemas são listados a seguir:

- Irrigação por aspersão: trata-se de um sistema de irrigação automatizada usado em áreas mais amplas, como gramados em casas de campo. Exige um reservatório de água, além de canos, bombas de recalque e aspersores escamoteáveis, do tipo rotor ou spray. Para seguir a programação de rega à risca, também conta com um timer (cronômetro).

- Irrigação por microaspersão: esse sistema de irrigação automatizada é mais indicado para canteiros, jardineiras e jardins pequenos, implantados em áreas externas. Funciona por meio de pequenas hastes equipadas com microaspersores, os quais são dispostos na superfície do solo.
- Irrigação por gotejamento subterrâneo: a irrigação automatizada por gotejamento é especialmente indicada para jardins com espécies florais, pois os jatos d'água usados na aspersão ou mesmo na irrigação manual podem prejudicar a polinização. A tecnologia é resistente à intrusão de raízes, pois o emissor de gotas fica, em média, a 30 cm de profundidade, enterrado no solo. Outra vantagem é que a área umedecida se restringe ao entorno dos pontos de gotejamento, permitindo variar a oferta de água conforme o tipo de espécie.
- Irrigação por gotejamento vertical: como o nome sugere, a irrigação automatizada por gotejamento vertical é usada em paredes verdes — estejam elas em espaços gourmet, para o cultivo de ervas, ou em áreas externas, complementando o paisagismo. Funciona assim: um cano mestre, instalado nas laterais dos jardins verticais, abastece cada fileira por meio de ramificações embutidas. Um tubo gotejador autocompensante, acoplado às conexões, garante a água na quantidade ideal.

Quadro 1 – Comparativo da solução proposta com as soluções existentes

Nome	Funcionamento	Tipo de irrigação	Comunicação	Interface com usuário
Rain Bird	Temporizador	Programável	-	Física
Kit Irrigação AMANCO	Temporizador	Programável	-	Física
Rain Bird St8i	Temporizador	Programável	Wifi	App
Solução proposta	Sensor de umidade do solo	Determinado de acordo com a umidade do solo	Wifi	Web

1.4 MERCADO

O mercado principal engloba clientes que desejam realizar o cultivo de plantas em suas residências, mantendo a irrigação das plantas constante mesmo que seja a uma grande distância. A solução proposta visa permitir um controle efetivo das plantas. O projeto busca despertar atenção de pessoas amantes do cultivo de plantas, que não disponibilizam de tempo necessário para se dedicar a atividade, devido jornada de trabalho, viagens ou outra atividade que impossibilite o plantio.

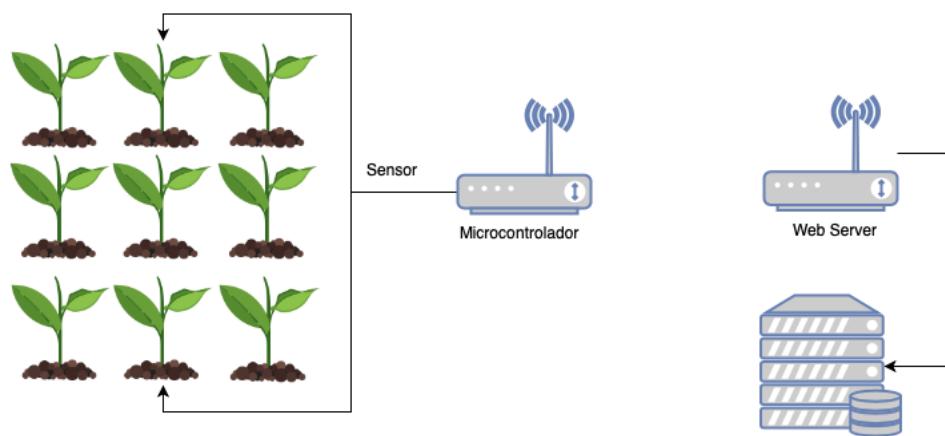
1.5 ANÁLISE DE VIABILIDADE

O projeto tem caráter exploratório com o objetivo de familiarização com o ambiente de desenvolvimento do kit disponibilizado em aula. A atividade consiste em aplicar os conceitos já estudados em semestres anteriores agregando um novo modelo de técnica de projeto.

2 PROJETO

2.1 VISÃO GERAL

Este projeto tem como princípio de funcionamento um sistema de irrigação autônomo, com reservatório de água, que permite instalação em locais diversos, sem a preocupação com a parte hidráulica das residências. O sistema também é controlado por sistema web, que informa o nível de água no reservatório e mantém as condições de temperatura e umidade do solo estabelecidas para cada tipo cultivo.



2.2 PREMISSAS

Este projeto propõe automação residencial no cultivo de plantas no âmbito doméstico. Levando em conta a necessidade da planta e da rotina das pessoas. Sendo assim, será desenvolvido um sistema embarcado para coletar e enviar as informações para um servidor web que será implementado.

Diante da solução proposta as seguintes premissas são assumidas para o sistema embarcado:

- A umidade do solo e temperatura do ambiente é coletada a cada dez minutos;
- O nível baixo de água do reservatório deve ser informado imediatamente ao usuário.

Para o servidor Web:

- As informações coletadas serão armazenadas em um banco de dados;
- O usuário terá acesso as informações do sistema em tempo real;

- O usuário será alertado do nível baixo do reservatório imediatamente.

2.3 ANÁLISE DE REQUISITOS

Requisitos funcionais do SISTEMA EMBARCADO

- RF01: Sistema deve coletar informações da umidade do solo;
- RF02: Sistema deve coletar informações da temperatura ambiente;
- RF03: Sistema deve verificar nível baixo de água em seu reservatório;
- RF03: Sistema deve informar o nível baixo de água em seu reservatório;
- RF04: Sistema deve acionar a irrigação respeitando os níveis mínimo e máximo de umidade do solo;

Requisitos não funcionais do SISTEMA EMBARCADO

- RNF01: o sistema será prototipado no microcontrolador ESP32;
- RNF02: o código da aplicação será escrito em linguagem C;
- RNF03: será utilizado o ambiente de desenvolvimento Arduino IDE;
- RNF04: o protótipo deverá custar no máximo R\$ 100,00;
- RNF05: o protótipo deverá coletar as informações com um tempo de resposta máximo de 30 segundos;
- RNF06: o protótipo deverá ter uma vazão de pelo menos 10 ml/minuto (conforme a aplicação).

Regras de negócio do SISTEMA EMBARCADO

- RN01: As informações de umidade do solo, temperatura ambiente e nível de água do reservatório deverão ser coletas e enviadas com intervalo de 10 minutos;
- RN02: O alerta de nível baixo do reservatório deverá ser feito por um aviso luminoso (LED), com o nível critico do reservatório.
- RN03: O motor deverá ser ligado em série com um sensor de nível crítico, garantindo a proteção do motor.

- RN04: A irrigação deverá ser feita pelo método de gotejamento;

Requisitos funcionais do SISTEMA WEB

- RF01: Sistema deve verificar recebimento dos dados coletados dos sensores e armazena-los;
- RF02: Sistema deve informar a umidade do solo;
- RF03: Sistema deve informar a temperatura ambiente;
- RF04: Sistema deve alertar nível baixo de água em seu reservatório;
- RF05: O sistema deve permitir o usuário consultar informações passadas.

Requisitos não funcionais do SISTEMA WEB

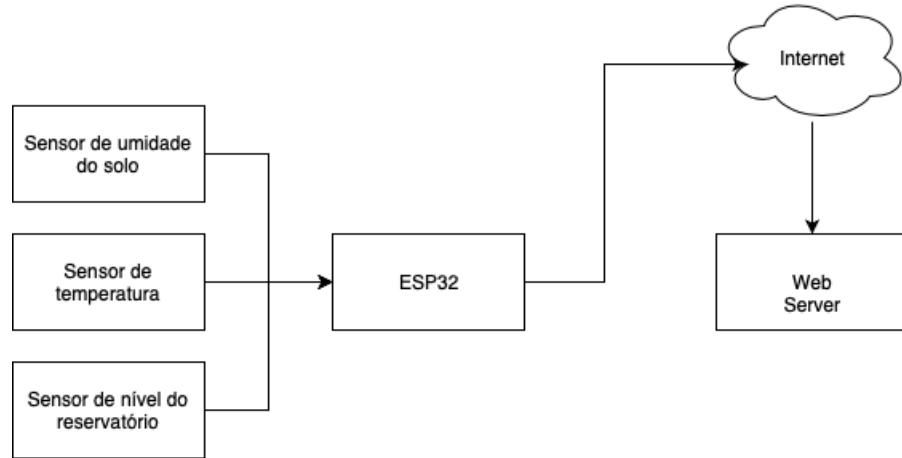
- RNF01: o sistema será prototipado para um desktop.

Regras de negócio do SISTEMA WEB

- RN01: As informações de umidade do solo, temperatura ambiente e nível de água do reservatório deverão ser armazenadas em banco de dados;
- RN02: O usuário poderá acessar histórico das leituras realizadas;

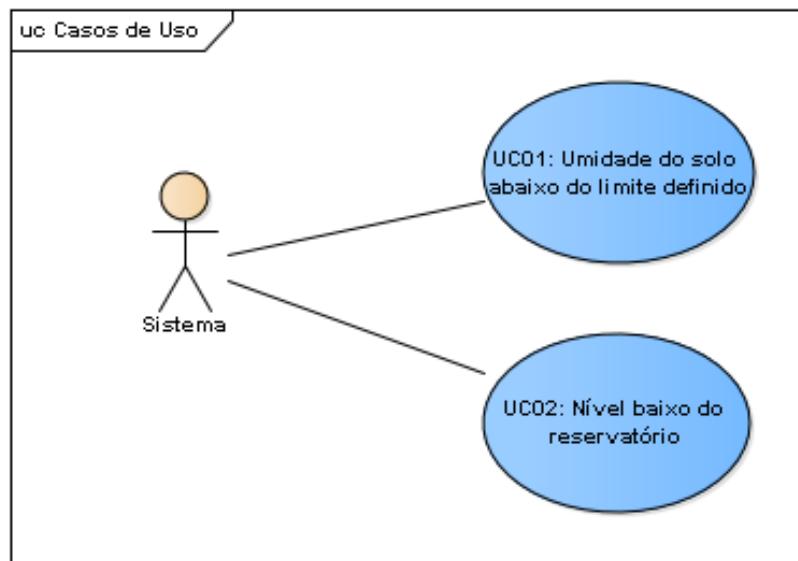
2.4 ARQUITETURA DE HARDWARE - SISTEMA EMBARCADO

Diagrama de blocos

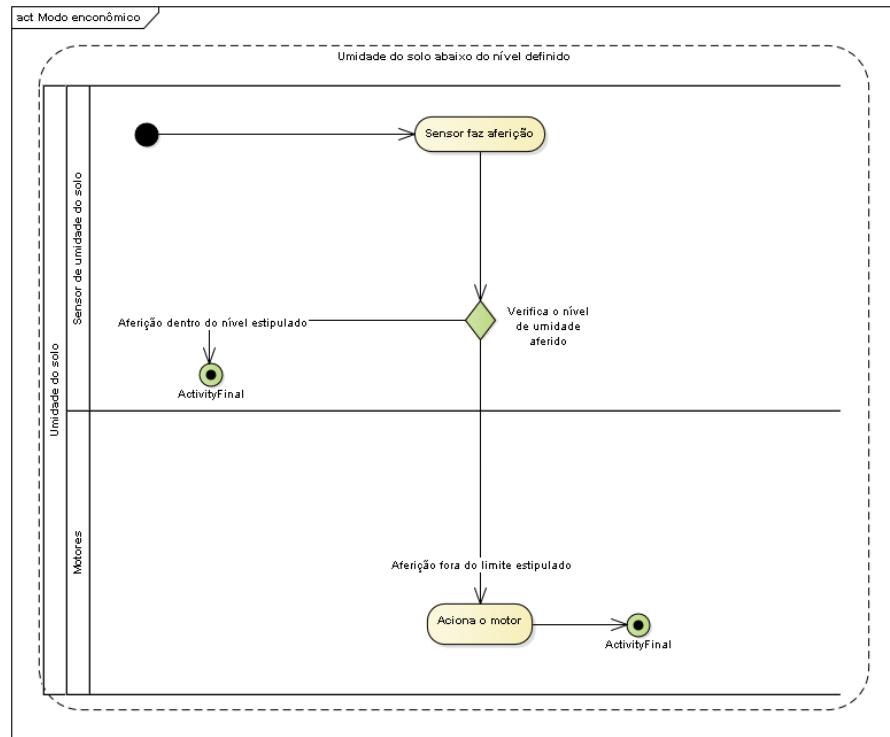


2.5 ARQUITETURA DE SOFTWARE - SISTEMA EMBARCADO

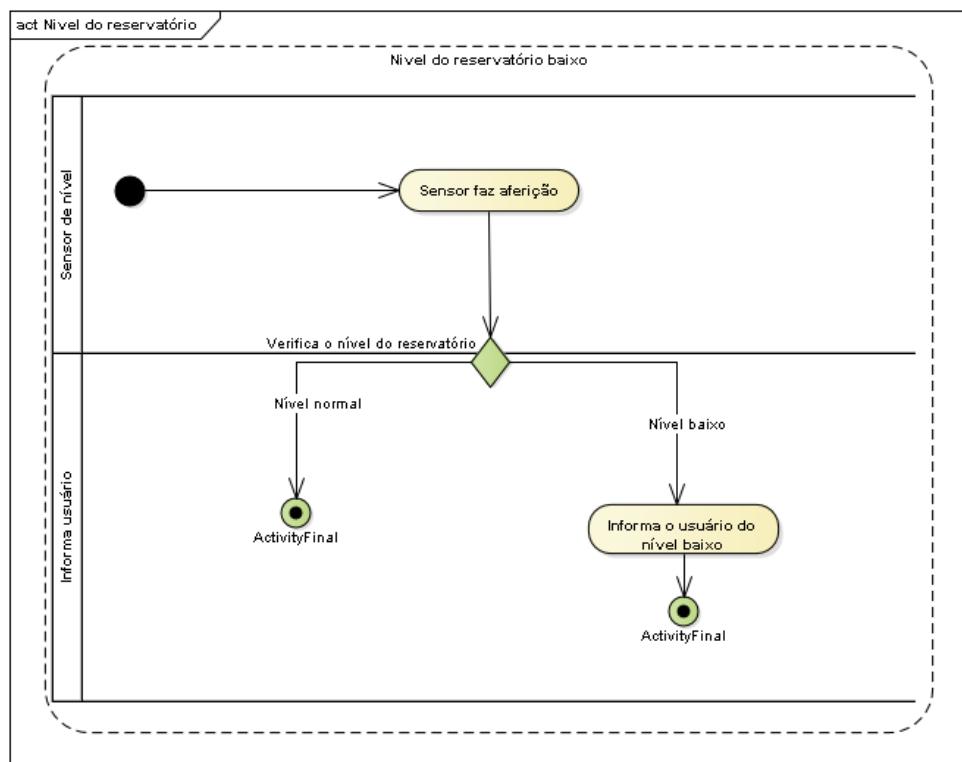
Casos de Uso



Modelo de Negócio Umidade do Solo

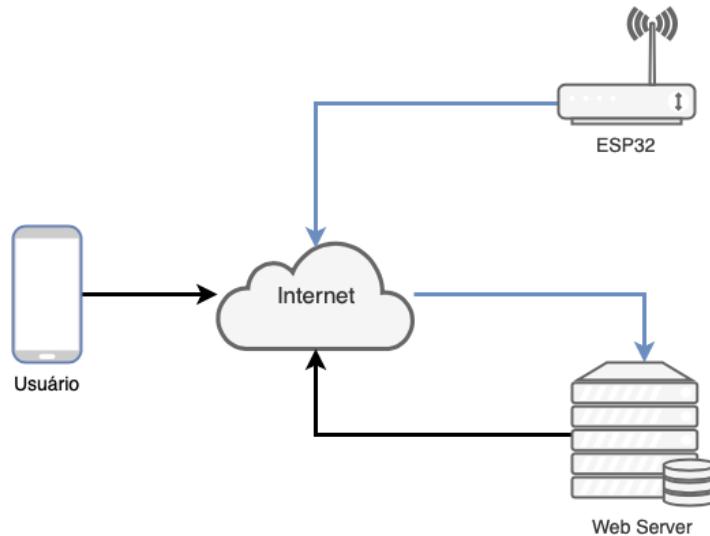


Modelo de Negócio Nível do Reservatório



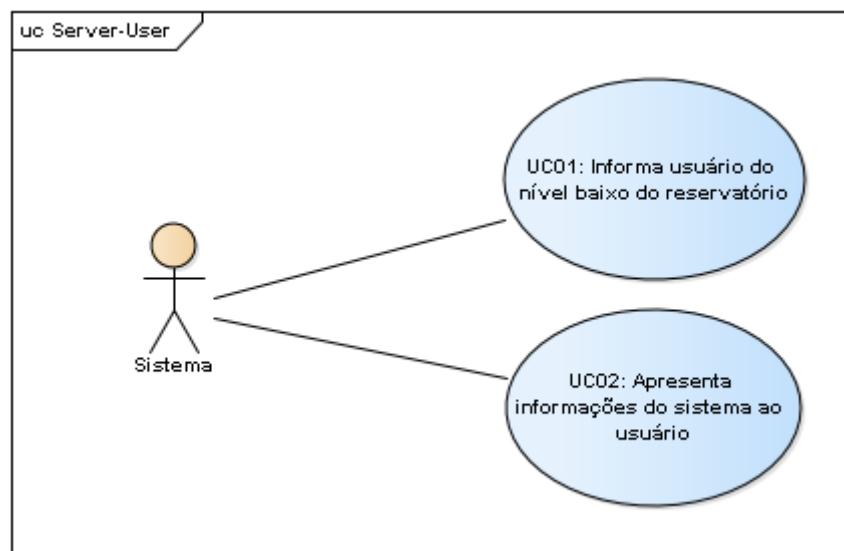
2.6 ARQUITETURA DE HARDWARE - SISTEMA WEB

Diagrama de blocos

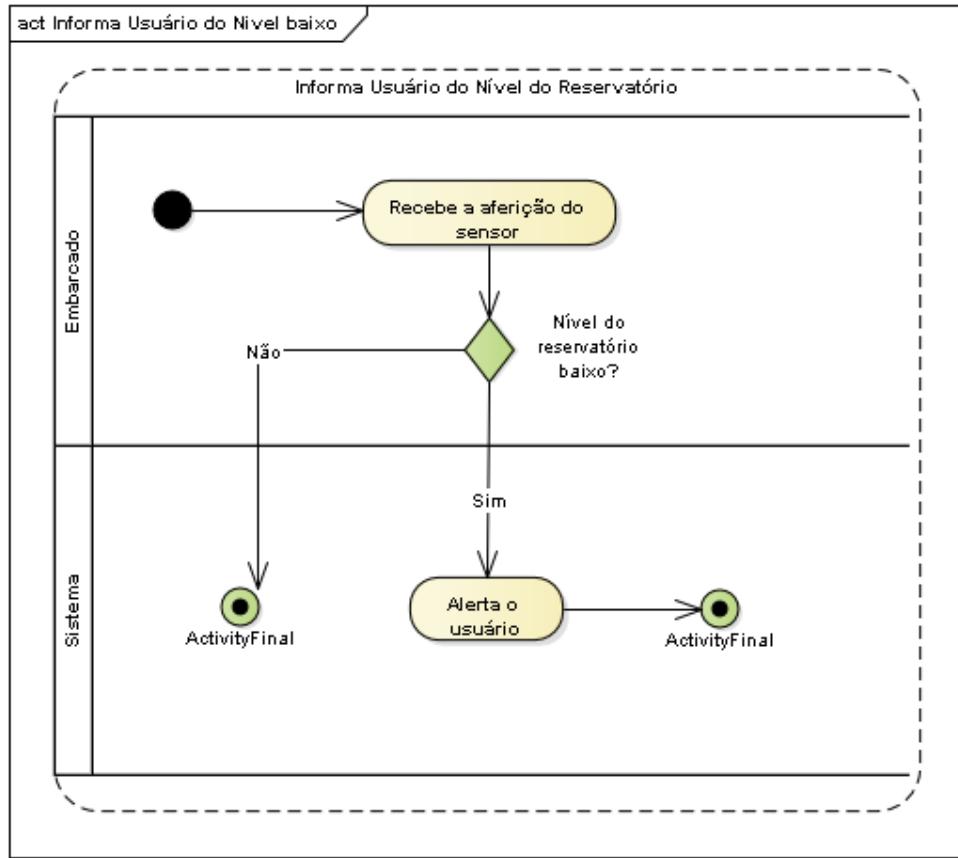


2.7 ARQUITETURA DE SOFTWARE - SISTEMA WEB

Casos de Uso



Modelo de Negócio



2.8 LISTA DE MATERIAIS E ORÇAMENTO

Logo abaixo se encontram a lista dos materiais solicitados, bem com o orçamento do dos demais materiais necessários para concepção do projeto.

Quadro 2 – Lista de materiais

Item	Qtd	Disponível	Responsável(is)
Kit de desenvolvimento ESP32	01	Sim	Professores
Sensor de Umidade do Solo Higrômetro	01	Não	Professores
Bomba D'água Submersiva 3-6V 100L/H	02	Não	Professores
Sensor de Umidade e Temperatura DHT11	01	Não	Professores
Sensor de Nível de água	02	Não	Professores

Quadro 3 – Orçamento (apenas para itens não disponíveis)

Item	Custo unitário	Qtd	Custo parcial	Fornecedor de referência
Sensor de Umidade do Solo Higrômetro	R\$ 10,90	01	R\$ 10,90	www.filipeflop.com
Bomba D'água Submersiva 3-6V 100L/H	R\$ 12,90	02	R\$ 25,80	www.filipeflop.com
Sensor de Umidade e Temperatura DHT11	R\$ 13,90	01	R\$ 13,90	www.filipeflop.com
Sensor de Nível de Água	R\$ 14,90	2	R\$ 29,80	www.filipeflop.com

2.9 PLANEJAMENTO

Protótipo do sistema embarcado será implementado com linguagem de programação C, utilizando ambiente de desenvolvimentos Arduino IDE, os testes de verificação estão descritos no quadro logo abaixo.

Quadro 4 – Plano de verificação para sistema embarcado

Requisito	Procedimento de verificação/teste	Resultado esperado
RF01	Verificar na prática leitura de umidade do solo, com tipos de solo e níveis de umidade diferente Verificar dados coletados do sensor humidade do solo	Se espera diferentes leituras de resistividade conforme níveis estabelecidos
RF02	Verificar correta leitura do sensor aferindo com termômetro manual Verificar dados coletados do sensor temperatura ambiente	Se espera valor de leitura bem aproximado ao termômetro manual
RF03	Verificar acionamento sensor nível de água com teste de continuidade em um multímetro Verificar funcionamento sensor de nível de água	Se espera sinal sonoro com mudança de estado do sensor
RNF01	O sistema será prototipado no microcontrolador ESP32	Se espera correto funcionamento
RNF05	Tempo de resposta será avaliado em teste prático acompanhando comportamento do sistema	Se espera tempo de resposta de máximo 30s após mudança de estados dos sensores
RNF06	Será calculando vazão de um litro de água em 3 níveis distintos	Se espera encontrar coeficiente ideal relacionados limites de vazão da bomba

Quadro 5 – Plano de verificação para web server

Requisito	Procedimento de verificação/teste	Resultado esperado
RF01	Verificar se dados foram recebidos no WebServer	Conectar com página web verificar pacote de dados recebidos
RF02	Verificar dados de umidade do solo recebidos comparar com coletados antes do envio	Se espera dados recebidos seja idêntico a os dados enviados
RF03	Verificar dados de temperatura ambiente recebidos comparar com coletados antes do envio	Se espera dados recebidos seja idêntico a os dados enviados
RF04	Verificar estado logico do sensor de nível recebido e comparar com enviado	Se espera dados recebidos seja idêntico a os dados enviados
RF05	Verificar se dados foram recebidos no WebServer	Se espera dados recebidos seja idêntico a os dados enviados
RF06	O sistema deve permitir o usuário consultar informações passadas.	Se espera que dados estejam acessíveis via página Web
RNF01	O sistema será prototipado para um dispositivo com acesso à web;	Se espera possibilidade de leitura por qualquer dispositivo com acesso web

Verificação dos componentes será realizada inicialmente de forma individual de seus componentes, seguido de teste prático a cada um destes, aferindo características de leituras e vazão, condições de coleta de dados, informações estas que demandará baixo consumo de dados agilizando envio, recebimento e a leitura dessas informações.

2.10 ANÁLISE DE TEMPO REAL

Na fase inicial do projeto, caracterizada pela descrição do projeto, a maioria do código fonte ainda não está disponível, tornando impossível medir, traçar ou realizar análises de código estático. Assim, é possível dividir a análise temporal em três fases:

- Fase inicial – responsável por determinar os requisitos de tempo, definir o hardware apropriado (microcontrolador, sensores, atuadores, dentre outros), iniciar a implementação.

- Fase de integração – finalizar a implementação, integrar os componentes em um ambiente operacional, depurar e otimizar a temporização, medir os tempos e relacionar os resultados com os respectivos requisitos, validar os modelos.
- Fase final - medir e supervisionar os tempos, realizar testes de temporização (pode ser feito em paralelo com testes funcionais), usar abordagens baseadas em modelo para localizar casos não previstos e realizar a verificação formal.

Diante disso apresentamos a tabela abaixo apresenta os itens que identificamos com requisitos de tempo real:

Quadro 5 – Requisitos de tempo real do sistema.

Itens	Classificação da tarefa	Técnica de Estimação do WCET	Critérios e Parâmetros
Premissas			
O nível baixo de água do reservatório deve ser informado imediatamente ao usuário.	Hard real-time	Análise estática	Tempo de execução
O usuário terá acesso as informações do sistema em tempo real	Soft real-time	Análise estática	Tempo de execução
Requisitos Funcionais (Embarcado)			
Sistema deve informar o nível baixo de água em seu reservatório	Soft real-time	Análise estática	Tempo de execução
Sistema deve acionar a irrigação respeitando os níveis mínimo e máximo de umidade do solo	Hard real-time	Análise estática	Tempo de execução
Requisitos Não Funcionais (Embarcado)			
O protótipo deverá coletar as informações com um tempo de resposta máximo de 30 segundos	Soft real-time	Análise estática	Tempo de execução Tempo de execução por instrução
Requisitos Funcionais (Web Server)			
O sistema deve permitir o usuário consultar informações passadas	Soft real-time	Análise estática	Tempo de execução Tempo de requisição no banco (TTL)

2.11 CRONOGRAMA

Quadro 7 – Cronograma de execução

Atividade	Sem. 1	Sem. 2	Sem. 3	Sem. 4	Sem. 5	Sem. 6	Sem. 7	Sem. 8	Entregável
Projeto	X								
Implementação		X	X						
Verificação			X	X	X				
Avaliação						X			
Documentação		X						X	

2.12 ANÁLISE DE RISCOS

Quadro 8 – Análise de riscos

Risco	Probabilidade	Impacto	Gatilho	Plano de contingência
Sensor de umidade do solo com precisão errada	Baixa	Alto	Acionamento do motor de forma inesperada.	Troca do sensor
Sensor de temperatura com precisão errada	Baixa	Baixo	Temperatura fora do padrão de amostragem.	Troca do sensor
Sensor de nível com precisão errada	Baixa	Baixa	Usuário sendo informado do nível baixo do reservatório, e reservatório cheio.	Troca do sensor
Motor trabalhando a seco	Baixa	Alto	Motor queimar	Troca do motor

3 DESENVOLVIMENTO

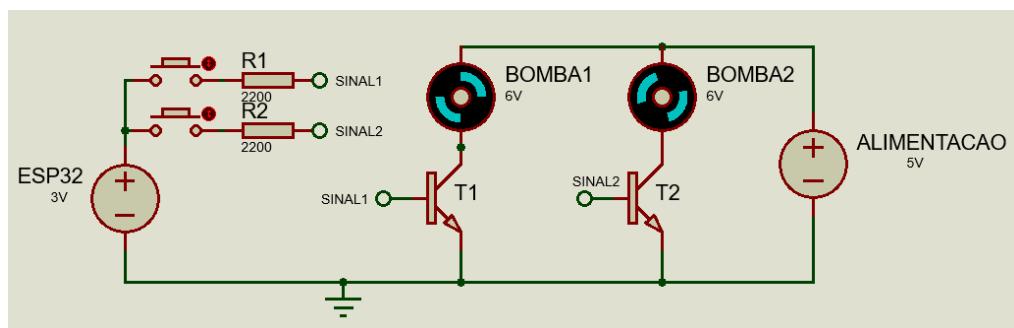
A seção seguinte é destinada a implementação do projeto. Com os elementos para concepção do sistema embarcado e do Web Server.

3.1 IMPLEMENTAÇÃO SISTEMA EMBARCADO

Para implementar o sistema foi utilizado o ambiente de desenvolvimento IDE Arduino, utilizando linguagem C. Para elaboração do protótipo foi necessário uma Esp32, dois sensores de umidade de solo, um sensor de nível de água (tipo boia), duas bombas d'água submersivas, uma protoboard de 800 furos, cabos, jumpers, transistores TIP41, resistores, capacitores e um cabo USB.

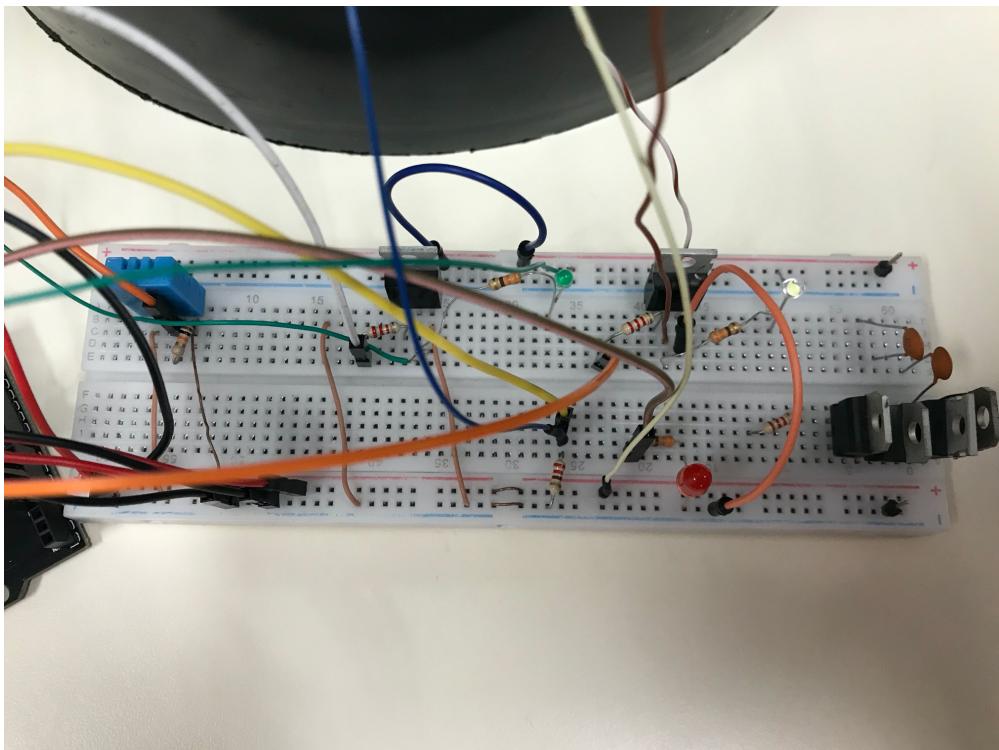
A primeira etapa foi testar os sensor e elementos associados ao projeto. Os testes foram realizados separadamente, cada sensor com seu código e princípio de funcionamento operando independentemente. A finalidade de testá-los separado consiste em definir limiares e parâmetros de funcionamento exclusivo de cada sensor. Na sequência, foi integrado os sensores em um único circuito e programado um único código para controlar todo o sistema.

Devido a insuficiência de corrente para o acionamento das bombas. Foi necessário adequar o funcionamento do sistema. A solução foi projetar um circuito auxiliar para o acionamento das bombas. O esquemático logo abaixo representa este circuito. O ESP32 é a fonte de tensão com 3v, os botões representam comando de acionamento ou sensores de umidade do solo com saídas SIGNAL1 e SIGNAL2 e a alimentação auxiliar de 5v de fonte externa.



Rotina do sistema embarcado é apresentada logo abaixo:

1. Temperatura ambiente é monitorada de forma ininterrupta;
2. Sensor de umidade do solo verifica condições do solo;
3. Sinalizado umidade do solo abaixo do limiar estabelecido, a bomba de água é acionada por 3 segundo fazendo irrigação no vaso, e aguardando por mais 3s para absorção da água pelo sol;
4. O nível do reservatório foi configurado com maior prioridade. As bombas de água devem trabalhar submersas. O nível do reservatório caso seja considerado crítico, impede o acionamento das bombas, preservando dessa forma a integridade das bombas de água;



Com as definições de funcionamento estabelecidas e com o circuito montado, foi possível passar para fase de testes do circuito.

3.2 IMPLEMENTAÇÃO DO WEB SERVER

Para implementar o sistema foi utilizado um Desktop, utilizando a IDE Visual Studio, disponibilizada gratuitamente, uma conta no “CloudMQTT” (para instanciar o broker da aplicação), um banco de dados “Postgres” e suíte de bibliotecas para Node.js.

A primeira etapa foi construir uma rota para acessar as informações escritas no tópico MQTT, com as configurações do websocket.

```
function showPage(message) {
    let messages = message.payloadString.split(' - ');
    document.getElementById("temperatura").innerHTML = "<span>Temperatura do Ambiente: " + messages[0] + " °C</span><br/>";
    document.getElementById("umidade1").innerHTML = "<span>Umidade relativa do solo (sensor 1): " + messages[1] + "%</span><br/>";
    document.getElementById("umidade2").innerHTML = "<span>Umidade relativa do solo (sensor 2): " + messages[2] + "%</span><br/>";

    let cor;
    if(messages[3] == 'OK'){
        cor = 'green'
    }else{
        cor = 'red'
    }

    document.getElementById("nivelReservatorio").innerHTML = "<span>Status do Reservatório: <span style='color: " + cor + "'>" + messages[3] + "</span></span><br/>";

    let today = new Date();
    let date = today.getDate() + '/' + (today.getMonth() + 1) + '/' + today.getFullYear();
    let time = today.getHours() + ":" + String(today.getMinutes()).padStart(2, '0') + ":" + String(today.getSeconds()).padStart(2, '0');

    document.getElementById("time").innerHTML = "<span>Última atualização: " + date + " " + time + "</span>";
    document.getElementById("wait").style.display = "none"
}
```

Na sequência foi necessário estabelecer uma rota para persistir as informações em banco de dados.

```
app.get('/insere', (req, res) => {
    pool.connect(function (err, client, done) {
        var d = new Date();

        var datestring = d.getFullYear() + "-" + (d.getMonth() + 1) + "-" +
d.getDate() + " " +
            d.getHours() + ":" + d.getMinutes() + ":" + d.getSeconds();

        let info_01 = req.query.temperatura;
        let info_02 = req.query.umidade1
        let info_03 = req.query.umidade2
        let info_04 = req.query.nivelReservatorio;

        //console.log(req.query);

        ///insere?temperatura=10&umidade1=10&umidade2=10&nivelReservatorio=OK
        let query = "insert into public.\"controleIrrigacao\""
        (temperatura_ambiente, umidade_sensor_1, umidade_sensor_2, nivel_reservatorio,
        data) values (" + info_01 + "," + info_02 + "," + info_03 + "," + info_04 + "','" +
        datestring + "')";

        if (err) {
            console.log("Erro ao conectar ao banco de dados." + err);
        }
        client.query(query, function (err, result) {
            done();
            if (err) {
                console.log(err);
            }
            res.status(200).send("Inserido");
        })
    })
});
```

Estabelecida a rota de persistência, foi necessária a rota de relatório, rota responsável pela visualização do histórico das leituras dos sensores.

```
app.get('/relatorio', (req, res) => {
    pool.connect(function (err, client, done) {
        if (err) {
            console.log("Can not connect to the DB" + err);
        }
        client.query('select * from public.\"controleIrrigacao\"', function (err,
        result) {
            done();
            if (err) {
                console.log(err);
                res.status(400).send(err);
            }
        })
    })
});
```

```

        }
        res.status(200).send(result.rows);
    })
});
});
```

Com as rotas estabelecidas, conectando o ESP32 ao WiFi pode-se estabelecer uma conexão para troca de mensagens entre os sistemas.

3.3 VERIFICAÇÃO

Os requisitos funcionais apresentados na seção de Projeto foram verificados mediante a testes de componentes e funcionamento do protótipo. O resultado de cada requisito é apresentado na tabela abaixo:

Plano de verificação para sistema embarcado

Requisito	Procedimento de verificação/teste	Resultado esperado	Resultado
RF01	Verificar na prática leitura de umidade do solo, com tipos de solo e níveis de umidade diferente Verificar dados coletados do sensor umidade do solo	Se espera diferentes leituras de resistividade conforme níveis estabelecidos	Requisito atende ao resultado esperado.
RF02	Verificar correta leitura do sensor aferindo com termômetro manual Verificar dados coletados do sensor temperatura ambiente	Se espera valor de leitura bem aproximado ao termômetro manual	Requisito atende ao resultado esperado.
RF03	Verificar acionamento sensor nível de água com teste de continuidade em um multímetro Verificar funcionamento sensor de nível de água	Se espera sinal sonoro com mudança de estado do sensor	Requisito atende ao resultado esperado.
RNF01	o sistema será prototipado no microcontrolador ESP32 Verificar dados enviados WebServer	Se espera correto funcionamento	Requisito atende ao resultado esperado.
RNF05	Tempo de resposta será avaliado em teste prático acompanhando comportamento do sistema	Se espera tempo de resposta de máximo 30s após mudança de estados dos sensores	Requisito atende ao resultado esperado.
RNF06	Será calculando vazão de um litro de água em 3 níveis distintos	Se espera encontrar coeficiente ideal relacionados limites de vazão da bomba	Requisito atende ao resultado esperado.

Plano de verificação para web server

Requisito	Procedimento de verificação/teste	Resultado esperado	Resultado
RF01	Verificar se dados foram recebidos no WebServer	Conectar com página web verificar pacote de dados recebidos	Requisito atende ao resultado esperado.

RF02	Verificar dados de umidade do solo recebidos comparar com coletados antes do envio	Se espera dados recebidos seja idêntico a os dados enviados	Requisito atende ao resultado esperado.
RF03	Verificar dados de temperatura ambiente recebidos comparar com coletados antes do envio	Se espera dados recebidos seja idêntico a os dados enviados	Requisito atende ao resultado esperado.
RF04	Verificar estado logico do sensor de nível recebido e comparar com enviado	Se espera dados recebidos seja idêntico a os dados enviados	Requisito atende ao resultado esperado.
RF05	Verificar se dados foram recebidos no WebServer	Se espera dados recebidos seja idêntico a os dados enviados	Requisito atende ao resultado esperado.
RF06	O sistema deve permitir o usuário consultar informações passadas.	Se espera que dados estejam acessíveis via página Web	Requisito atende ao resultado esperado.
RNF01	O sistema será prototipado para um dispositivo com acesso à web;	Se espera possibilidade de leitura por qualquer dispositivo com acesso web	Requisito atende ao resultado esperado.

3.4 RESULTADOS

Sistema Embarcado

Os resultados obtidos para o sistema embarcado são apresentados na sequência:

Em uma situação de operação normal e envio de dados, obtemos a seguinte resposta da porta serial da IDE do Arduino:

Leituras do Sensores

Umidade sensor 1: 56%

Umidade sensor 2: 43%

Temperatura Ambiente: 25.50

Nível do Reservatório: OK

Em uma situação onde o sensor de umidade 1 está abaixo de 40%, limiar que indica a necessidade de irrigação da planta e envio de dados, obtemos a seguinte resposta da porta serial da IDE do Arduino:

Leituras do Sensores

Umidade sensor 1: 34%
 Umidade sensor 2: 43%
 Temperatura Ambiente: 25.50
 Nível do Reservatório: OK

Planta 1 Precisa de Água
 ---> MOTOR '1' LIGADO <---
 ---> MOTOR '1' DESLIGADO <---

O motor 1 é ativado por 3 segundo, e a espera, entre desativado, e voltar a rotina de execução do código, também é 3 segundo, tempo para a terra absorver a água adicionada.

Em uma situação onde o sensor de umidade 2 está abaixo de 40%, limiar que indica a necessidade de irrigação da planta e envio de dados, obtemos a seguinte resposta da porta serial da IDE do Arduino:

Leituras do Sensores

Umidade sensor 1: 67%
 Umidade sensor 2: 28%
 Temperatura Ambiente: 25.50
 Nível do Reservatório: OK

Planta 1 Precisa de Água
 ---> MOTOR '2' LIGADO <---
 ---> MOTOR '2' DESLIGADO <---

O motor 2 é ativado e desativado, com o mesmo período de tempo do motor 1.

Em uma situação onde o sensor de nível identifica um nível baixo do reservatório o envio de dados dos sensores não é interrompido. Porém o acionamento dos motores não é permitido, uma vez que os motores exigem estar debaixo d'água para seu correto funcionamento. Para essa situação obtemos a seguinte resposta da porta serial da IDE do Arduino:

Leituras do Sensores

Umidade sensor 1: 89%

Umidade sensor 2: 39%

Temperatura Ambiente: 25.50

Nível do reservatório: BAIXO

Sistema WebServer

Os resultados obtidos para o Web Serve são apresentados na sequência:

O sistema Web é uma forma diferente de apresentar as informações contidas no sistema embarcado. Sendo assim, o sistema é encarregado de receber as informações providas dos sensores e direcionar para uma camada de apresentação mais prática para o usuário final. Ou seja, o usuário será capaz de acessar de qualquer lugar as informações coletadas. Outra funcionalidade adicionada ao sistema Web é persistência dos dados em um banco de dados.

A imagem apresentada abaixo ilustra uma situação normal de operação do sistema. Informando as leituras dos sensores, bem como o estado “OK” do reservatório de água.

Sistema de Irrigação para Hortas Domiciliares

Temperatura do Ambiente: 24.80 °C

Umidade relativa do solo (sensor 1): 45 %

Umidade relativa do solo (sensor 2): 58 %

Status do Reservatório: OK

Última atualização: 9/12/2019 16:13:22

A imagem apresentada abaixo ilustra uma situação crítica do sistema. Informando as leituras dos sensores, bem como bem o estado “BAIXO” do reservatório de água.

Sistema de Irrigação para Hortas Domiciliares

Temperatura do Ambiente: 24.10 °C

Umidade relativa do solo (sensor 1): 33 %

Umidade relativa do solo (sensor 2): 67 %

Status do Reservatório: BAIXO

Última atualização: 9/12/2019 16:12:20

A imagem abaixo ilustra o dump do banco de dados. No qual as informações armazenadas como temperatura ambiente, sensor de umidade 1, sensor de umidade 2 e nível do reservatório.

```
[{"temperatura_ambiente":10,"umidade_sensor_1":10,"umidade_sensor_2":10,"nivel_reservatorio":"OK","d":{"temperatura_ambiente":20,"umidade_sensor_1":20,"umidade_sensor_2":10,"nivel_reservatorio":"OK","d":{"temperatura_ambiente":10,"umidade_sensor_1":10,"umidade_sensor_2":10,"nivel_reservatorio":"OK","d":{"temperatura_ambiente":25,"umidade_sensor_1":55,"umidade_sensor_2":89,"nivel_reservatorio":"BAIXO"}}, {"temperatura_ambiente":25,"umidade_sensor_1":55,"umidade_sensor_2":89,"nivel_reservatorio":"BAIXO"}}, {"temperatura_ambiente":18,"umidade_sensor_1":55,"umidade_sensor_2":89,"nivel_reservatorio":"BAIXO"}}, {"temperatura_ambiente":24,"umidade_sensor_1":73,"umidade_sensor_2":77,"nivel_reservatorio":"OK"}, {"d":{"temperatura_ambiente":25,"umidade_sensor_1":76,"umidade_sensor_2":82,"nivel_reservatorio":"OK"}, {"d":{"temperatura_ambiente":25,"umidade_sensor_1":75,"umidade_sensor_2":80,"nivel_reservatorio":"OK"}, {"d}}
```

Análise de Tempo Real

Itens	Classificação da tarefa	Técnica de Estimação do WCET	Critérios e Parâmetros	Resultado
Premissas				
O nível baixo de água do reservatório deve ser informado imediatamente ao usuário.	Hard real-time	Análise estática	Tempo de execução	Resultado atende aos requisitos.
O usuário terá acesso as informações do sistema em tempo real	Soft real-time	Análise estática	Tempo de execução	Resultado atende aos requisitos parcialmente.
Requisitos Funcionais (Embarcado)				
Sistema deve informar o nível baixo de água em seu reservatório	Soft real-time	Análise estática	Tempo de execução	Resultado atende aos requisitos.
Sistema deve acionar a irrigação respeitando os níveis mínimo e máximo de umidade do solo	Hard real-time	Análise estática	Tempo de execução	Resultado atende aos requisitos.
Requisitos Não Funcionais (Embarcado)				
O protótipo deverá coletar as informações com um tempo de resposta máximo de 30 segundos	Soft real-time	Análise estática	Tempo de execução Tempo de execução por instrução	Resultado atende aos requisitos.
Requisitos Funcionais (Web Server)				
O sistema deve permitir o usuário consultar informações passadas	Soft real-time	Análise estática	Tempo de execução Tempo de requisição no banco (TTL)	Resultado atende aos requisitos.

A aplicação permite extrair os tempo de execução de cada tarefa crítica utilizando uma função específica do ESP32 (`ESP.getCycleCount();`) que realiza a contagem de ciclos de processador. Para essa aplicação foi definida como prioridade o tempo de atualização de

sensores, o tempo de acionamento dos motores (1 e 2) e o tempo te envio de dados. A imagem abaixo apresenta a contagem de ciclos de processador.

Tempo de Execução

Tempo atualiza sensores: 15854853
Tempo acionamento Motor 1: 1439992482
Tempo acionamento Motor 2: 1439992472
Tempo envio de dados: 511723

Para extrair o tempo em milissegundos é necessário fazer uma divisão pela frequência de operação do ESP32, neste caso 80MHz. A imagem abaixo ilustra o resultado obtido convertido de ciclos de processador em milissegundos.

Tempo de Execução

Tempo atualiza sensores: 198 ns
Tempo acionamento Motor 1: 17,99 ms
Tempo acionamento Motor 2: 17,99 ms
Tempo envio de dados: 6,3 ns

4 CONSIDERAÇÕES FINAIS / CONCLUSÕES

Este projeto apresentou uma proposta de monitoramento de plantas no âmbito residencial. Os resultados obtidos comprovam as premissas iniciais e validam a substituição futura de “timers”, que são utilizados em sistemas de monitoramento convencionais por sensores de umidade, os quais garantem uma melhor qualidade da planta, ou seja, neste trabalho qualidade na irrigação da planta. O trabalho ainda apresentou alguns desafios. Primeiramente a multidisciplinaridade, utilizando recursos e conhecimento de outro profissionais em uma outra área de atuação, neste caso, a biologia. Na área da engenharia, uma adaptação do circuito, foi necessária para suprir a alimentação do ESP32, não fornecia a quantidade de corrente necessária para acionar os motores.

Como possíveis desdobramentos desse projeto podemos elencar a alteração dos sensores de prateleiras por sensores mais precisos e eficientes. Como o de temperatura ambiente e principalmente os sensores de umidade, visto que oscilavam demais em pequenos intervalos de tempo. Um outro desdobramento seria a fabricação de um protótipo comercial, com a finalidade de obter investimentos para uma futura linha de produção.

REFERÊNCIAS

- Americanas. (s.d.). Kit 1 - Timer + 1 - kit microaspersor Amanco. Acesso em Outubro de 2019, disponível em Americanas: https://www.americanas.com.br/produto/28767072/kit-1-timer-1-kit-microaspersor-amanco?WT.srch=1&acc=e789ea56094489dff798f86ff51c7a9&epar=bp_pl_00_go_pla_cas_aeconst_geral_gmv&gclid=Cj0KCQjw3JXtBRC8ARIsAEBHg4mUglhG2LqhsWBjvJSCGqN409as8pe7ljVqeZEZ6Gp2gFg0jX
- Ashton, K. (2009). That 'internet of things' thing. *RFID journal*, 22(7), 97-114.
- Canal Agricola. (s.d.). Temporizador para Irrigação Ambiente Interno 8 Estações ESP RZX Rain Bird (F452). Fonte: Canal Agricola: https://www.canalagricola.com.br/temporizador-irrigacao-interno-8-estacoes-rainbird?gclid=Cj0KCQjw3JXtBRC8ARIsAEBHg4INNC76WxGYxoFxluaysOLYMnb6DLOLuMRLnn7AhiUuRw8F0E1bJDgaAu77EALw_wcB
- Chaudhry, A. R., & Mishra, V. P. (2019). A Comparative Analysis of Vertical Agriculture Systems in Residential Apartments. *2019 Advances in Science and Engineering Technology International Conferences (ASET)*. Dubai: IEEE.
- Li, J., Yan, Q., & Chang, V. (2018). Internet of things: Security and privacy in a connected world. *Future Generation Computer Systems*, 931-932.
- Mercado Livre. (s.d.). Controlador Rain Bird St8i 8 Estações Indoor C/ Wifi 120v. Acesso em Outubro de 2019, disponível em Mercado Livre: https://produto.mercadolivre.com.br/MLB-1005588998-controlador-rain-bird-st8i-8-estacoes-indoor-c-wifi-120v-JM?quantity=1#position=16&type=item&tracking_id=78e888f9-a260-426e-9f18-97d56cc2eb48
- Pereira, H. A., & Oliveira, R. F. (2018). Sistema de monitoramento de umidade de solo via web. São Paulo: Curso Superior em Tecnologia em Análise e Desenvolvimento de Sistemas Instituto Federal de São Paulo – Campus Hortolândia (IFSP).
- Silva, L. A., Leithardt, V. R., Rolim, C. O., Gonzalez, G. V., Geyer, C. F., & Silva, J. S. (2019). Priser: Managing notification in multiples devices with data privacy support. *Sensor*, 19(14), 1-18.