

TRABALHO DE CONCLUSÃO DE CURSO

CURSO: AGRONOMIA

**SISTEMA IOT DE BAIXO CUSTO PARA AUTOMAÇÃO E
GESTÃO DE IRRIGAÇÃO**

LEONARDO NASCIMENTO LIMA



LEONARDO NASCIMENTO LIMA

**SISTEMA IOT DE BAIXO CUSTO PARA AUTOMAÇÃO E
GESTÃO DE IRRIGAÇÃO**

Trabalho de Conclusão de Curso apresentado ao Instituto de Ciências Agrárias da Universidade Federal de Minas Gerais, como requisito parcial, para a obtenção do título de Bacharel em Agronomia.

Orientador: Prof. Dr. Edson de Oliveira Vieira

Montes Claros


2021

LEONARDO NASCIMENTO LIMA, SISTEMA IOT DE BAIXO CUSTO
PARA AUTOMAÇÃO E GESTÃO DE IRRIGAÇÃO.

Aprovada pela banca examinadora constituída por:

Prof. Dra. Júlia Ferreira da Silva - ICA/UFMG

Prof. Dr. Sidney Pereira - ICA/UFMG



Prof.Dr. Edson de Oliveira Vieira - Orientador ICA/UFMG

Montes Claros, 15 de março de 2021

Sumário

1. INTRODUÇÃO.....	7
2. REFERENCIAL TEÓRICO.....	7
3. METODOLOGIA.....	8
3.1. Calibração do sensor de umidade de solo e ultrassônico.....	11
3.2. Custo de desenvolvimento.....	13
3.2. Protótipos anteriores.....	15
4. RESULTADOS.....	17
4.1. Protótipo com IoT.....	17
4.2. Teste em campo.....	18
5. CONCLUSÕES.....	21
6. REFERÊNCIAS.....	22
7. ANEXOS.....	23

RESUMO

A agricultura familiar, composta por pequenos produtores, apresenta importância social e econômica, além de ser uma garantia da segurança alimentar, mas mediante a crise hídrica, associada a baixa tecnologia para automação e gestão de recursos hídricos, estes agricultores tornam-se o público mais vulnerável a falta de água. Dentre as alternativas, os microcontroladores possuem grande potencial para solucionar estes problemas argumentados, logo, foi objetivado o desenvolvimento de um sistema de automação e gestão de recursos hídricos, aplicado a atividade de irrigação e de baixo custo. Foram utilizados os microcontroladores Arduino MEGA2650 e ESP 32, agregados com o sensor DHT11 para monitoramento de temperatura e umidade do ar, sensor de distância ultrassônico HC-SR04 e sensor de umidade de solo capacitivo V1.2, além dos módulos de SIM80L para envio de SMS e microSD para armazenamento dos dados. Como resultado foi obtido um protótipo funcional de baixo custo, testado em campo no Instituto de Ciências Agrárias e pertencente a Universidade Federal de Minas Gerais, capaz de gerenciar os recursos hídricos e realizar ações de modo automático, como o acionamento ou desligamento da irrigação, além de permitir o acesso aos dados remotamente via WEB, sem a necessidade de monitorar no local. Embora seja um equipamento de baixo custo, o protótipo conseguiu realizar as leituras com consistência, previsibilidade (poucas oscilações entre leituras seguidas) e com capacidade de ações automáticas no gerenciamento da irrigação, com a sua finalidade sendo alcançada nos testes em campo.

Palavras-chaves: Agricultura familiar-irrigação, microcontroladores-irrigação, IoT, recursos hídricos, uso racional-água, automação-irrigação.

1. INTRODUÇÃO

A crise hídrica, que de forma cíclica e sazonal, se estende em diversos estados brasileiros, causa perdas consideráveis nos setores de agricultura, principalmente a familiar, por queda na produção, na pecuária por baixa oferta de insumos e ingredientes para formulação de rações e, na alimentação humana, com alta nos preços.

Diante da importância social e econômica da agricultura, em especial a familiar, soluções devem ser providenciadas, com o intuito de reduzir os problemas gerados para a crise hídrica. Dentre as soluções, o monitoramento e gestão de recursos hídricos e o uso eficiente na atividade de irrigação, são uma alternativa para amenizar os impactos gerados pela falta de água no campo.

Contudo, para realizar o monitoramento e gestão são convencionalmente aplicadas as centrais de automação ou controladores que apresentam o inconveniente de um custo elevado, principalmente para os pequenos produtores e muitos oriundos da agricultura familiar, inviabilizando a aquisição de tais equipamentos.

Outro problema da automação no campo, com foco na irrigação, é o alto custo dos equipamentos, controladores e sensores, além de serem plataformas fechadas e forçam os produtores a ficarem dependentes do fabricante, de suas peças e de seu serviço especializado, excluindo os produtores com baixo aporte financeiro para investimento nesse tipo de tecnologia.

Por isso, teve-se como objetivo, desenvolver um sistema de baixo custo, utilizando microcontroladores acessíveis para gestão e monitoramento de recursos hídricos em sistemas de irrigação integrados com a “internet das coisas” (Internet of Things-IoT).

2. REFERENCIAL TEÓRICO

Os pequenos agricultores, muitos provenientes da agricultura familiar, são os responsáveis pela garantia da segurança alimentar e, desde o final da década de 90, por meio da terceira geração de políticas públicas para a agricultura familiar, e ganharam notoriedade pela produção diversificada de alimentos ao longo do território nacional (GRISA; SCHNEIDER, 2014).

Existe uma cadeia de laços comerciais com os pequenos produtores que varia do nível local ao nacional, com a produção destes agricultores atingindo cooperativas, associações, varejo tradicional, agroindústria alimentar e supermercados. A ampla gama de setores beneficiados diretamente pelos pequenos produtores, muitos da agricultura

familiar, demonstram mais uma vez a sua importância na produção de alimentos (MALUF, 2004).

Os pequenos agricultores acabam sendo os mais sensíveis, em cenários de crise em geral, uma vez que dispõem de menos recurso financeiro, estrutural e tecnológico para enfrentar as dificuldades eminentes. A causa dessa falta de recursos, deve-se, em sua maioria, à um histórico de abandono político e social desde os anos 60 (GRISA; SCHNEIDER, 2014).

Aliado aos problemas sociais e políticos dos últimos anos? com os pequenos produtores, também existe a crise hídrica que passou a ser uma realidade nas localidades onde não haviam sofrido um cenário de falta água, gerando um debate, principalmente com relação às atividades na agricultura que correspondem a vazão de 70% da água captada (EMBRAPA, 2015).

Dentre as atividades na agricultura que utilizam água, existe a irrigação, que varia de porte e metodologia e até mesmo em nível de agricultura familiar e com baixo custo, sendo que já existem recomendações para sistemas instalados em pequenas áreas (COELHO et al., 2012). Porém, esses usuários ainda carecem de um sistema de gerenciamento da água de baixo custo, de forma que possibilite o uso eficiente desse recurso.

O uso eficiente da água é de singular relevância para que seja garantida uma seguridade hídrica, independente da região. No caso do semiárido, devido às oscilações que ocorrem nas precipitações pluviométricas, o uso eficiente da água é fundamental (FERREIRA et al., 2016).

A automação, no aspecto recursos hídricos, vem sendo utilizada desde os anos 50 (após a segunda guerra mundial), como meio de gerenciar a água nas mais diversas finalidades, como irrigação, abastecimento humano e animal, controle de doenças (através do saneamento básico) e produção de energia elétrica (SOUZA, 2006).

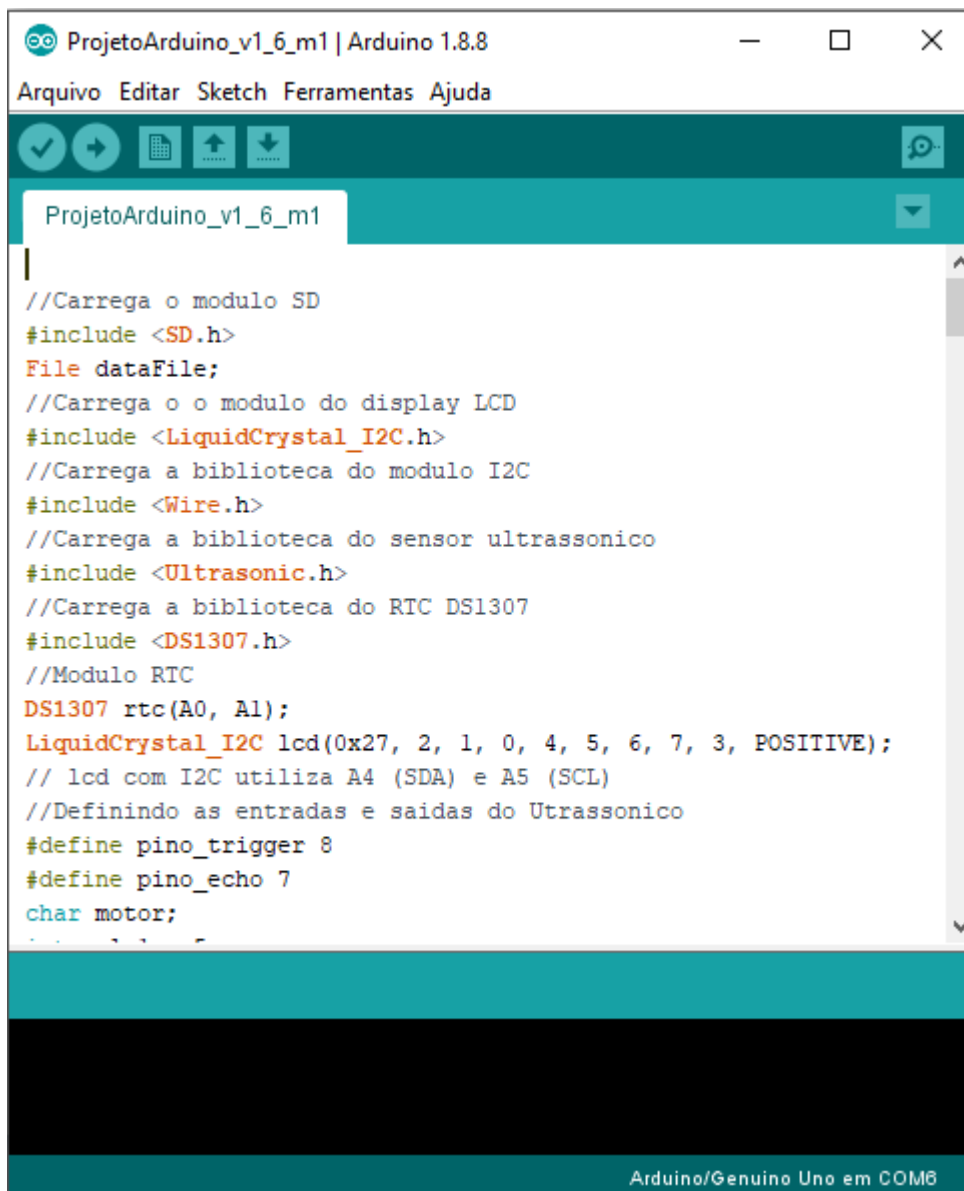
Um microcontrolador é um pequeno computador em que o usuário pode fazer uma programação para executar uma determinada tarefa, fazendo um processo de automação (ROBERTS, 2011).

3. METODOLOGIA

A metodologia consistiu na pesquisa e desenvolvimento de um protótipo para gestão, controle de recursos hídricos e para atividades de irrigação.

Para a programação utilizou-se a plataforma Arduino IDE, conforme a Figura 1, no desenvolvimento do código fonte que gerenciou todo o conjunto. O *software* é de desenvolvimento aberto e gratuito para qualquer usuário.

Figura 1 - Software Arduino IDE para desenvolvimento do código do projeto.



```
ProjetoArduino_v1_6_m1 | Arduino 1.8.8
Arquivo Editar Sketch Ferramentas Ajuda

ProjetoArduino_v1_6_m1

//Carrega o modulo SD
#include <SD.h>
File dataFile;
//Carrega o o modulo do display LCD
#include <LiquidCrystal_I2C.h>
//Carrega a biblioteca do modulo I2C
#include <Wire.h>
//Carrega a biblioteca do sensor ultrassonico
#include <Ultrasonic.h>
//Carrega a biblioteca do RTC DS1307
#include <DS1307.h>
//Modulo RTC
DS1307 rtc(A0, A1);
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
// lcd com I2C utiliza A4 (SDA) e A5 (SCL)
//Definindo as entradas e saidas do Utrassonico
#define pino_trigger 8
#define pino_echo 7
char motor;
```

Arduino/Genuino Uno em COM8

Fonte: Do autor, 2020.

Para a confecção do protótipo foram utilizados microcontroladores da família Arduino e ESP, modelo Mega 2560 e modelo ESP 32, respectivamente. Os sensores utilizados foram de umidade e temperatura de ambiente (DHT11), umidade de solo (Capacitivo V1.2 da *Fortek*) e de distância ultrassônica (HC-SR04). A comunicação entre o protótipo com a rede foi via SMS (modulo SIM800L) e via WEB, através de abertura de porta em roteador para acesso remoto.

O microcontrolador Arduino MEGA 2560, apresenta 54 portas digitais (sendo 15 PWM), 14 portas analógicas, tensão de operação de 5 volts, memória flash 256KB, SRAM de 8 KB, e velocidade de *clock* 16 MHz. Já o microcontrolador ESP 32 é muito mais evoluído, com um processador Dual Core 32 bits, com velocidade de *clock* de 160 MHz, *bluetooth* e *wifi* integrado, possui 32 pinos (sendo 16 PWM e com resolução de 12 bits), tensão de operação de 3,3 volts e memória flash de 512KB.

O sensor DHT11 permite a medição de temperatura de 0°C a 50°C (precisão $\pm 2^\circ\text{C}$) e umidade do ar entre 20% a 90% (precisão $\pm 5\%$ UR). O sensor de umidade de solo capacitivo versão 1.2 possui dois eletrodos protegidos por uma camada de plástico (evitando o contato direto com o solo), ele gera um sinal de tensão pela capacitância do solo (0V a 5V, com resolução de 10 bits para saídas analógicas). Já o sensor de distância HC-SR04 permite monitorar objetos ou líquidos de 2 cm a 400 cm (precisão de $\pm 3\text{mm}$).

O módulo SIM800L foi responsável por enviar SMS quando solicitado pelo usuário, ele possui capacidade de comunicação com redes de telefonia móvel 2G e 3G, através de um chip de celular previamente inserido e com número válido. Já o DS1307 é o módulo responsável pela data e hora que foram anexadas em cada leitura, basicamente é um relógio digital.

O regulador de tensão LM7805, foi utilizado com a finalidade de manter um fornecimento de tensão em parâmetros próximos a 5 volts de saída, para os módulos agregados aos microcontroladores, como o *Display* LCD16x2, módulo MicroSD, SIM800L e Relé 5V com dois canais. A tensão de entrada ou alimentação, foi de 12 volts através de uma fonte de computador genérica.

Foi erguida uma base metálica fixa para instalação do sistema e acomodação dos sensores, visando também manutenção e ajustes nas primeiras horas de teste do protótipo.

Na irrigação, foram utilizadas mangueiras de gotejamento para simulação do sistema em uma área de 1,5 m², sendo a cultura de teste escolhida rabanete, por possuir ciclo rápido e por ser de fácil manuseio. A área de teste fica atrás do laboratório de hidráulica do ICA/UFMG.

Para o monitoramento dos dados utilizou-se um *display* LCD16x2 presente no protótipo, mas os dados também podem ser acessados no site ou no cartão micro SD, conforme mostrado na Figura 2.

Figura 2 - Central do protótipo com o display LDC16x2 para visualização dos parâmetros no local.



Fonte: Do autor, 2021.

3.1. Calibração do sensor de umidade de solo e ultrassônico

O sensor de umidade de solo foi o mais crítico e que necessitou de calibração uma vez que as características do solo mudam de local para local. Portanto, o mesmo necessita ser calibrado com o solo em que será inserido para que as leituras de umidade de solo sejam verídicas.

O sensor utilizado foi capacitivo versão 1.2 da *Fortek*, sendo um sensor de saída analógica ou digital (dependendo do pino escolhida)

O procedimento de calibração consistiu na retirada de uma amostra de solo da área de teste (FIGURA 3), sendo submetida a secagem em estufa a 105°C por 24 horas. Posteriormente foram separadas 100g de solo seco, o sensor foi inserido e a leitura gerada foi anotada. A cada leitura foram adicionados 10mL de água, contidos em um copo do tipo becker e os valores da leitura novamente anotados, até que o solo e o sensor saturassem.

Figura 3 - Calibração do sensor de umidade de solo capacitivo.



Fonte: Do autor, 2021.

Considerando que 1 mL H₂O equivalente a 1g e que o recipiente (copo) contém 100g de solo, tem-se:

$$U_{p10\%} = \left(\frac{Mu - Ms}{Ms} \right) = \left(\frac{110g - 100g}{100g} \right) = 0,1 \text{ gH}_2\text{O} / \text{g de solo.}$$

$$U_{p20\%} = \left(\frac{Mu - Ms}{Ms} \right) = \left(\frac{120g - 100g}{100g} \right) = 0,2 \text{ gH}_2\text{O} / \text{g de solo.}$$

$$U_{p30\%} = \left(\frac{Mu - Ms}{Ms} \right) = \left(\frac{130g - 100g}{100g} \right) = 0,3 \text{ gH}_2\text{O} / \text{g de solo.}$$

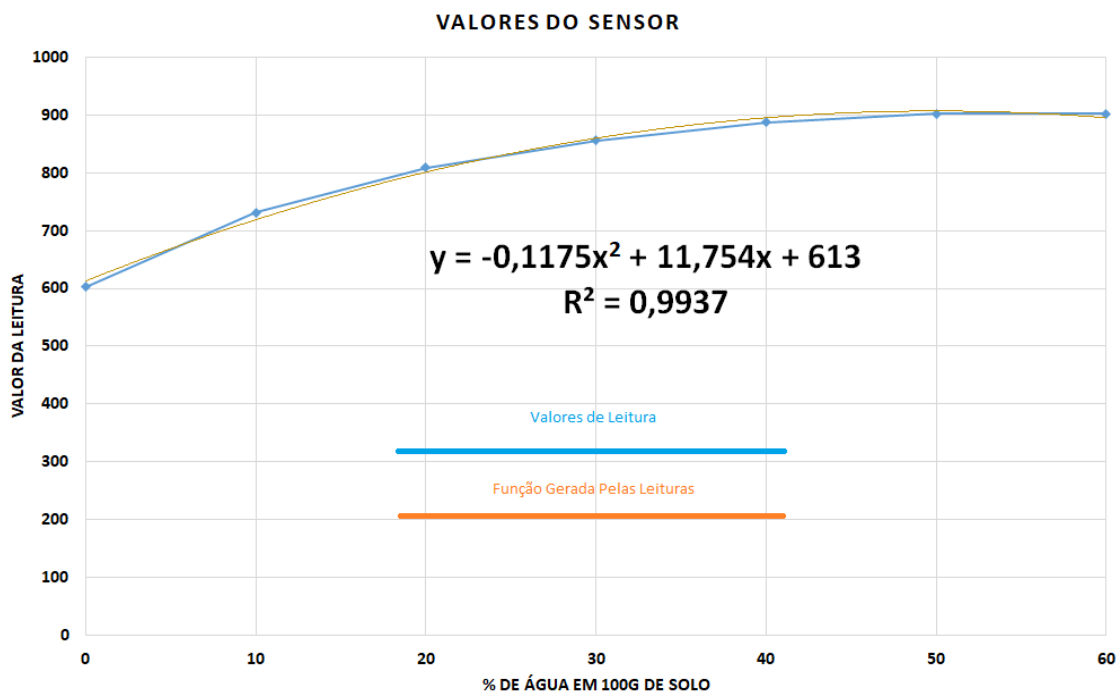
$$U_{p40\%} = \left(\frac{Mu - Ms}{Ms} \right) = \left(\frac{140g - 100g}{100g} \right) = 0,4 \text{ gH}_2\text{O} / \text{g de solo.}$$

$$U_{p50\%} = \left(\frac{Mu - Ms}{Ms} \right) = \left(\frac{150g - 100g}{100g} \right) = 0,5 \text{ gH}_2\text{O} / \text{g de solo.}$$

Em que M_u representa a massa úmida, M_s representa a massa seca e U_p representa a quantidade de água no solo de amostra.

Após a adição total de 50 mL de água não houve variações na leitura de umidade de solo, representando a saturação do solo e consequentemente do sensor. As leituras geradas estão dispostas no Gráfico 1.

Gráfico 1 - Valores de leitura gerados na calibração do sensor.



Fonte: Do autor, 2021.

Com os valores de leitura, gerou-se a função no Excel para calibrar a umidade do solo, com o sensor retornando um sinal entre 603 a 903 unidades (entrada 10 bits do microcontrolador que vai de 0 a 1023, referenciando um sinal de saída de 0 a 5 volts), sendo posteriormente convertidos através de fórmula, em um valor percentual de umidade de solo.

A calibração do sensor ultrassônico consistiu apenas em informar os limites de altura do nível de água, que variou de 0 a 100 cm. A medição foi realizada com uma trena comum no reservatório de água, com a própria biblioteca do sensor fazendo os cálculos para determinação do nível de água via efeito *Doupler*, sendo um pulso de som enviado e com o tempo de seu retorno determinado a distância do objeto ou líquido.

3.2. Custo de desenvolvimento

O teto máximo de custo estipulado para a confecção do protótipo foi de R\$600,00 (seiscentos reais) já incluindo materiais adquiridos anteriormente ou reaproveitado de outros projetos, (TABELA 1).

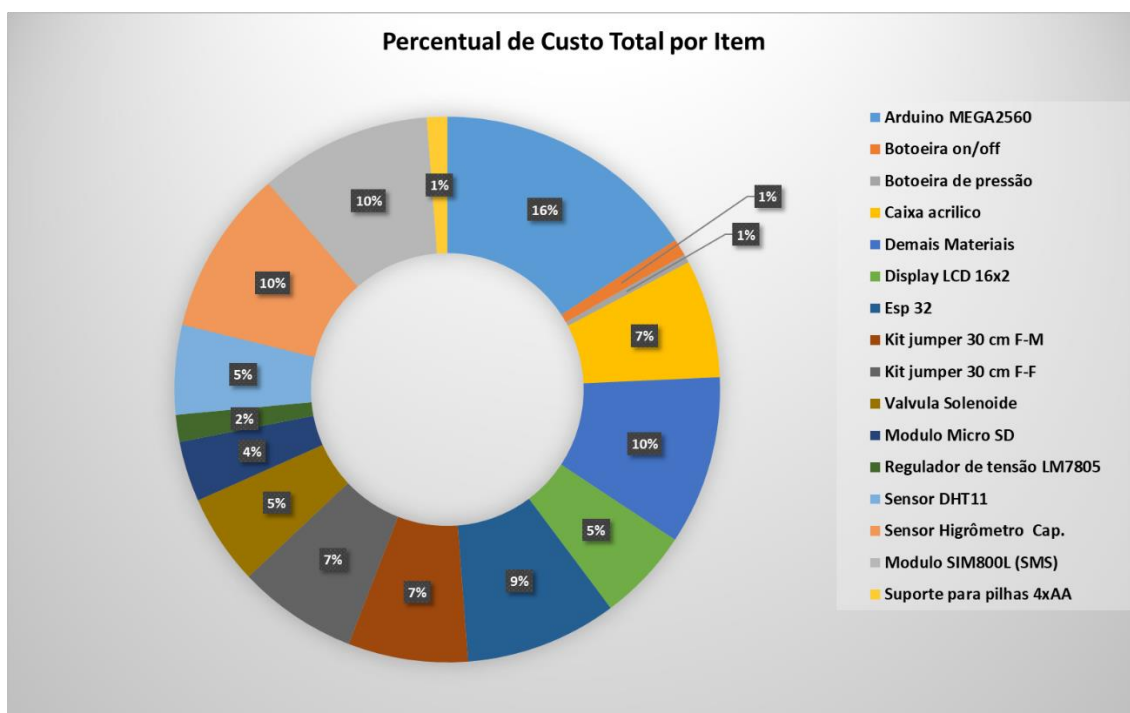
Tabela 1 - Lista de materiais necessários para a confecção do protótipo utilizado e custo individual.

Item	Quantidade	Valor (Reais)	Total (Reais)
Arduino MEGA2560	1	78,9	78,9
Botoeira on/off	2	2,5	5
Botoeira de pressão	1	2,5	2,5
Caixa acrílico	2	35	35
Demais Materiais	1	50	50
Display LCD 16x2	1	27,49	27,49
Esp 32	1	45	45
Kit jumper 30 cm F-M	1	35,5	35,5
Kit jumper 30 cm F-F	1	35,5	35,5
Válvula Solenoide	1	59,75	59,75
Modulo Micro SD	1	17,89	17,89
Regulador de tensão LM7805	4	2,00	8,00
Sensor DHT11	1	26,5	26,5
Sensor Higrômetro Cap.	2	24,5	49
Modulo SIM800L	1	50,9	50,9
Suporte para pilhas 4xAA	2	3	6
Valor total			532,53

Fonte: Do autor, 2021.

Os itens com maior peso para a execução do projeto foram os microcontroladores Arduino, o modulo SIM800L e o ESP32, contudo os preços podem alterar no decorrer do tempo, sendo que se o custo reduzir, será cada vez mais vantajoso. Com o Gráfico 2 torna-se mais fácil visualizar o percentual do custo de cada item que no conjunto formou o protótipo.

Gráfico 2 - Percentual do custo por item para confecção do protótipo.



Fonte: Do autor, 2021.

Novamente, percebe-se que os microcontroladores Arduino, o módulo SIM800L (para envio de SMS) e o ESP32 (para envio de dados via WEB) são os mais caros para a montagem de protótipo, contudo não podem ser dispensados, pois são essenciais.

3.3. Protótipos anteriores

Foi desenvolvido e testado um protótipo anterior no laboratório de hidráulica do ICA – UFMG, desde 2018, capaz de coletar os dados, processar e enviar via SMS para um celular com número cadastrado previamente no código desenvolvido. As mensagens eram enviadas a cada minuto, contudo o intervalo de tempo de envio pode ser modificado para a necessidade do usuário. Na Figura 4, pode-se observar o envio de dados via SMS para o usuário.

Figura 4 - Recebimento dos dados via SMS do protótipo em campo em 2018.



Fonte: Do autor, 2018.

Essa versão anterior conseguia monitorar a umidade e temperatura ambiente, nível e volume do reservatório, além da presença ou não de chuva no local. Os testes foram realizados no ICA/UFMG no reservatório destinado a atividade irrigação.

Na Figura 5, pode-se observar a estrutura externa que contém os principais sensores para o monitoramento dos dados no reservatório. A central necessita de um local protegido, como a casa de máquinas que possui estrutura como telhado e paredes, protegendo todo o conjunto em caso de chuva e intempéries, já os sensores não necessitam dessa estrutura, pois foram adaptados para ambiente externo.

Figura 5 - Estrutura externa de fixação dos sensores de nível e pluviômetro, na versão anterior.



Fonte: Do autor, 2018.

Este protótipo de 2018, serviu de base para o desenvolvimento dos protótipos posteriores, com cada versão desenvolvida, sendo agregados novas funcionalidades. Também foram implementados sensores mais precisos para as atividades de irrigação, em especial o higrômetro capacitivo, por ser mais preciso e confiável.

4. RESULTADOS

Contemplando os aspectos da agricultura familiar, sua importância para a segurança alimentar e do comércio a ela agregado, foi desenvolvido um protótipo, com base nos anteriores de 2018 e com novas funcionalidades, em especial o envio de dados via IoT, além da capacidade de armazenar os dados localmente como meio de *backup*.

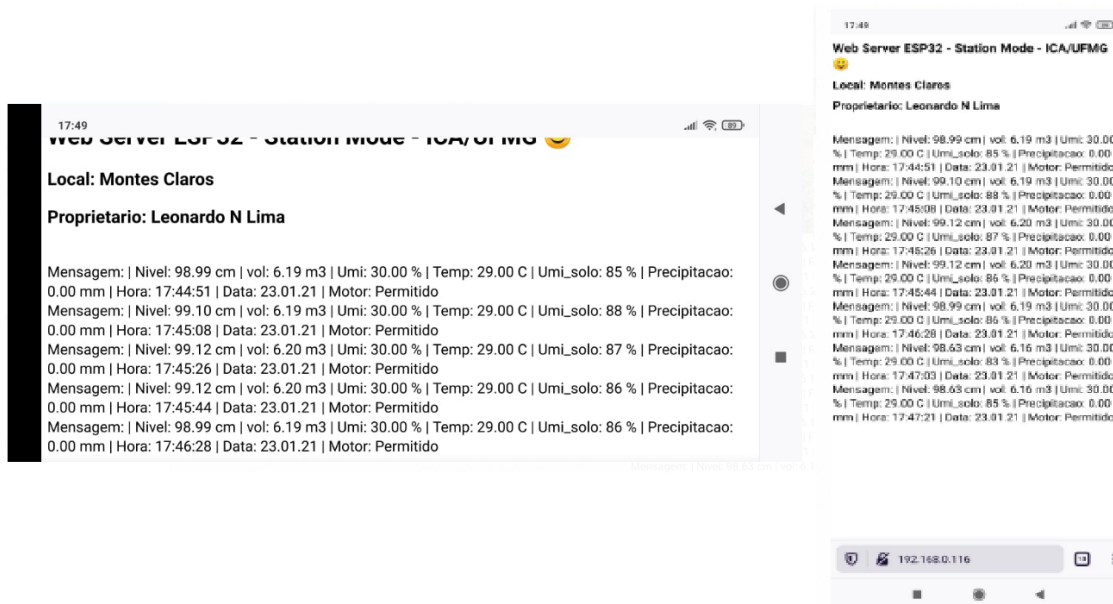
Mesmo a automação, em especial de recursos hídricos, não sendo algo recente, conforme argumentado por SOUZA (2006), o sistema desenvolvido (protótipo) de baixo custo também permitiu a automação e monitoramento de recursos hídricos, além da irrigação.

4.1. Protótipo com IoT

O protótipo final consegue enviar os dados para um servidor web (FIGURA 6), onde os dados podem ser acessados por qualquer dispositivo vinculado na web, como

por exemplo, computadores de mesa, notebooks, tablets e smartphones, através da abertura de acesso remoto nas configurações do roteador instalado no local.

Figura 6 - Página web com os dados em tempo real abertos em navegador androide.



Fonte: Do autor, 2021.

Os parâmetros monitorados foram o nível e volume do reservatório, umidade e temperatura do ar e umidade do solo, além do *datalogger* com registro da hora e data de cada leitura de parâmetro, com intervalos de oito em oito segundos. De acordo com os valores registrados de umidade de solo, a irrigação é acionada ou desligada automaticamente.

4.2. Teste de campo

A área de teste foi montada no dia 08/02/2021 (FIGURA 7), com plantio de rabanetes na área com a finalidade de criar um potencial de evapotranspiração e assim gerar valores de leituras mais reais para o protótipo, forçando o acionamento do sistema de irrigação, através de uma válvula solenoide localizada na saída de água do reservatório de teste, monitorando a gestão da água no reservatório e armazenamento dos dados, podendo também ser monitorado via internet, SMS ou através de cartão micro SD.

Figura 7 - Início do experimento de teste do protótipo em campo.



Fonte: Do autor, 2021.

Os dados captados pelo sensor de umidade de solo foram enviados para a central, via cabo, onde as informações foram processadas para a tomada de decisão acerca do acionamento ou não da irrigação, na Figura 8, também pode-se observar a central, a haste com o sensor de distância ultrassônica e o reservatório de água.

Figura 8 - Reservatório e local de instalação da central do protótipo.



Fonte: Do autor, 2021.

No dia 23/02/2021, após quinze dias de teste em campo, o experimento foi encerrado e os dados analisados para verificar a estabilidade das leituras realizadas pelo protótipo. Os rabanetes apresentaram, visualmente, bom desenvolvimento, podendo ser visualizado na Figura 9.

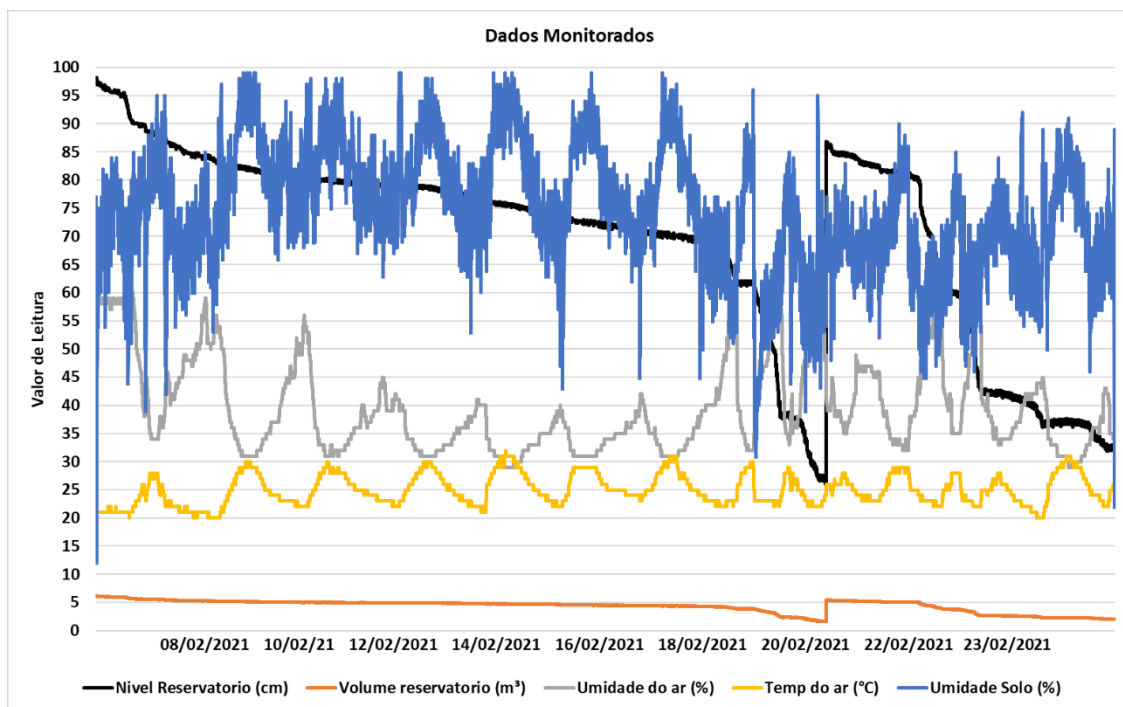
Figura 9 - Finalização do experimento de teste do protótipo no décimo quinto dia, fonte autor.



Fonte: Do autor, 2021.

Os dados monitorados foram coletados do cartão memória micro SD e analisados, estando dispostos no Gráfico 3. Ao todo foram mais de 10 mil leituras, com intervalos regulares de oito em oito segundos, que monitoraram o nível, o volume, a temperatura, a umidade do ar e a umidade do solo.

Gráfico 3 - Dados monitorados e coletados do teste em campo.



Fonte: Do autor, 2021.

O nível e o volume de água no reservatório apresentaram boas leituras e com poucas oscilações, em especial para o volume que manteve uma linha bem uniforme ao longo do gráfico de monitoramento. Também se percebe a rápida resposta ao enchimento

do reservatório, onde contém o aumento do nível e do volume no dia 20/02/2021. Caso houvesse uma redução abrupta do nível e conseqüentemente o volume, poderia indicar duas situações, a primeira um vazamento ou rompimento e a segunda um erro ou falha no sensor.

A umidade e temperatura do ar (sendo parâmetros auxiliares) por variarem ao longo do dia e noite, ou em situações de chuva, geram leituras formando um padrão semelhante a ondas. Em alguns momentos do gráfico nota-se uma redução da temperatura e aumento da umidade relativa do ar, podendo também ser um indicador de chuva no local. Caso houvesse alguma leitura abrupta na umidade do ar e na temperatura, este poderia ser um indicador de falha no sensor.

Já a umidade de solo apresentou boas leituras, mas foram identificadas algumas oscilações, contudo pelo sensor de umidade de solo realizar uma leitura a cada oito segundos, a próxima leitura corrige a oscilação. Também existe no código um contador que impede o acionamento da irrigação por uma leitura incorreta, com a irrigação sendo acionada apenas após três leituras seguidas de umidade abaixo de 60% no solo.

Deste modo, as leituras realizadas pelo protótipo em campo foram de boa integridade, significando que tiveram previsibilidade e pouca oscilação entre uma leitura e outra, além da capacidade de interromper a irrigação quando o nível de água ficou abaixo dos 25 cm de profundidade no solo, negando o acionamento de irrigação para que a água pudesse ser utilizada para outros fins, como o consumo humano e animal, que nesse cenário, possui prioridade sobre a atividade de irrigação.

Por fim, com os resultados obtidos, o protótipo desenvolvido conseguiu fazer uma gestão dos recursos hídricos e da irrigação, com valores sólidos (com pouca oscilação entre leituras) e consistentes, sendo uma alternativa de baixo custo viável para o campo, principalmente para atender pequenas e médias propriedades ou produtores pertencentes a agricultura familiar.

5. CONCLUSÕES

Com o uso da automação e plataformas de baixo custo como Arduino e ESP foi possível desenvolver um protótipo para irrigação utilizando IoT para a automação e monitoramento a distância, destinado a atividade de irrigação no campo, visado também, uso eficiente de recursos hídricos e energia, utilizando apenas o necessário para a atividade de irrigação destinada.

O protótipo desenvolvido, custou menos de R\$ 600,00 (seiscentos reais), sendo uma ferramenta acessível para os produtores rurais, principalmente os pequenos que buscam alternativas de automação no campo, pois seu baixo custo e ampla finalidade o torna interessante para a gestão de recursos hídricos e para a atividade irrigação, tendo poder de tomada de decisões em caso de falta de água, priorizando atividades.

Com relação as leituras de variáveis, os dados apresentaram boa consistência (com pouca oscilação entre leituras) e foram acessíveis através de navegador WEB, SMS e no local com cartão micro SD ou no display LCD da estação. Ao todo foram geradas mais de dez mil leituras ao longo dos 15 dias de experimento, que mostraram no Gráfico 3 uma boa consistência na série de dados.

Por fim, mesmo com todos esses sensores e módulos, além do extenso código desenvolvido para funcionamento, o protótipo ainda permite expansões que podem adicionar finalidades específicas para cada cenário em que ele for destinado, mostrando a grande versatilidade desta tecnologia e que se bem aproveitada pode ser um meio para redução dos problemas sociais e econômicos gerados pela falta de recursos hídricos nas propriedades rurais.

6. REFERÊNCIAS

ALEXANDER, C.K. **Fundamentos de circuitos elétricos**. Porto Alegre: AMGH: Capítulo I, p.26-40, 2013.

COELHO, E. F. *et al.* **Sistemas e Manejo de Irrigação de Baixo Custo para Agricultura Familiar**. Cruz das Almas: Embrapa Mandioca e Fruticultura, 2014.

EMBRAPA. Produção de Hortaliças & Crise Hídrica. **Hortaliças em revista**. Brasília, v.4, n.15, p. 2, 2015.

FERREIRA, E. P. *et al.* Uso eficiente da água de chuva armazenada em cisterna para a produção de hortaliças no Semiárido pernambucano. **Revista Verde de Agroecologia e Desenvolvimento Sustentável**. v.11, n.2, p.1-7, 2016.

GRISA, C.; SCHNEIDER, S. Três gerações de políticas públicas para a agricultura familiar e formas de interação entre a sociedade e o estado no Brasil. **Revista de economia e sociologia rural**. Brasília, v.52, p. 7-8, 2014.

MALUF, R. S. **Mercados agroalimentares e a agricultura familiar no Brasil: agregação de valor, cadeias integradas e circuitos regionais**. Ensaio FEE, Porto Alegre, v. 25, n. 1, p. 299-322, abr. 2004.

ROBERTS, M. **Arduíno Básico**. São Paulo: NOVATEC: Capítulo I, p. 22, 2011.

SOUZA, M. **PROPOSTA DE UM SISTEMA DE GESTÃO EMPREGANDO INSTRUMENTAÇÃO INTELIGENTE E REDE DE CAMPO NA AUTOMAÇÃO**

DO PROCESSO DE TRATAMENTO DE ÁGUA. Universidade de São Paulo -USP, SÃO PAULO, p.21-25, 2006. Disponível em < <https://www.teses.usp.br/teses/disponiveis/3/3143/tde-13122006-170823/publico/MarceloDeSouza.pdf> >. Acessado em 8 de março de 2021.

7. ANEXOS

O código fonte utilizado para o desenvolvimento do protótipo, no microcontrolador Arduíno MEGA.

```
//Carrega o modulo SD
#include <SD.h>
#include <dht.h>
File dataFile;
//Carrega o o modulo do display LCD
#include <LiquidCrystal_I2C.h>
//Carrega a biblioteca do modulo I2C
#include <Wire.h>
#include <SoftwareSerial.h>
//Carrega a biblioteca do sensor ultrassonico
#include <Ultrasonic.h>
//Carrega a biblioteca do RTC DS1307
#include <DS1307.h>
//Modulo RTC
DS1307 rtc(A0, A1);
LiquidCrystal_I2C lcd(0x27, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
// lcd com I2C utiliza A4 (SDA) e A5 (SCL)
#define dht_dpín A2
//Definindo as entradas e saidas do Utrassonico
#define pino_trigger 8
#define pino_echo 7
//SIM800 TX is connected to Arduino D14
#define SIM800_TX_PIN 14
//SIM800 RX is connected to Arduino D15
#define SIM800_RX_PIN 15
SoftwareSerial serialSIM800(SIM800_TX_PIN, SIM800_RX_PIN);
char motor;

int rele1 = 22;
int rele2 = 23;
int ledON = 24;
int ledOFF = 25;
int ledSTD = 26;
int chave = 3;
int recive = 2;
float vol = 0;
bool error;
// Constantes:
const int REED = 6;
//Sensor de solo Variaveis
const int pinoSensor = A3; //PINO UTILIZADO PELO SENSOR Solo
int valorLido; //VARIABEL QUE ARMAZENA O PERCENTUAL DE UMIDADE DO SOLO
int SoloSeco = 601; //VALOR MEDIDO COM O SOLO SECO
int SoloMolhado = 903; //VALOR MEDIDO COM O SOLO MOLHADO
int FaixaSolo = 0;
int FaixaSensor = 0;
int UmiSolo = 0;
// Variaveis:
int val = 0;
int old_val = 0;
int REEDCOUNT = 0;
int cont = 0;
int cont2 = 0;
int cont3 = 0;
int ContAC = 0;
```



```

Ultrasonic ultrasonic(pino_trigger, pino_echo);
dht DHT; //Inicializa o sensor
void setup()
{
  Serial.begin(115200);
  serialSIM800.begin(9600);
  //Aciona o relógio
  rtc.halt(false);

  //As linhas abaixo setam a data e hora do módulo
  //e podem ser comentada após a primeira utilização
  // rtc.setDOW(THURSDAY); //Define o dia da semana
  //rtc.setTime(12, 33, 0); //Define o horário
  // rtc.setDate(8, 2, 2021); //Define o dia, mês e ano

  //Definições do pino SQW/Out
  rtc.setSQWRate(SQW_RATE_1);
  rtc.enableSQW(true);

  pinMode(chave, INPUT);
  pinMode(rele1, OUTPUT);
  pinMode(rele2, OUTPUT);
  pinMode(ledON, OUTPUT);
  pinMode(ledSTD, OUTPUT);
  pinMode(ledOFF, OUTPUT);
  pinMode(10, OUTPUT);
  // inicializa o pino do switch como entrada
  pinMode (REED, INPUT_PULLUP);

  lcd.begin(16, 2);
  lcd.setBacklight(HIGH);
  lcd.setCursor(2, 0);
  lcd.print("!CARREGANDO!");
  delay (2000);
  lcd.clear();
  digitalWrite(53, HIGH);
  if (!SD.begin(53))
  {
    lcd.setCursor(3, 0);
    lcd.print("!SD FALHOU!");
    delay(2000);
    lcd.clear();
    return;
  }
  lcd.setCursor(4, 1);
  lcd.print("!SD OK!");
  delay (2000);
  lcd.clear();
  byte grau[8] = { B00001100,
                  B00010010,
                  B00010010,
                  B00001100,
                  B00000000,
                  B00000000,
                  B00000000,
                  B00000000,
                  };
}
void loop()
{

```

```

//Set SMS format to ASCII
serialSIM800.write("AT+CMGF=1\r\n");
delay(100);
lcd.clear();
digitalWrite(ledSTD, LOW);
while (digitalRead(chave) == LOW)
{
  DHT.read11(dht_dpin); //LÃª as informaÃ§Ãµes do sensor
  float h = DHT.humidity;
  float t = DHT.temperature;
  String motor;
  float nivel;
  float cmMsec;
  long microsec = ultrasonic.timing();
  cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
  nivel = 110 - cmMsec;
  FaixaSolo = SoloMolhado-SoloSeco;
  UmiSolo=analogRead(A3);
  FaixaSensor = SoloMolhado-UmiSolo;
  valorLido = 100-((FaixaSensor*100)/FaixaSolo);
  // ler o estado do switch pelo pino de entrada:
  val = digitalRead(REED); //Read the status of the Reed switch

  if ((val == LOW) && (old_val == HIGH)) {

    delay(10);

    REEDCOUNT = REEDCOUNT + 1;

    old_val = val;

    lcd.setCursor(8, 1);
    lcd.print(REEDCOUNT * 0.25);
    lcd.setCursor(14, 1);
    lcd.print("mm");
    delay(2000);
    lcd.clear();
  }

  else {
    old_val = val;
    lcd.setCursor(0, 1);
    lcd.print("SEM CHUVA");
    delay(1500);
    lcd.clear();
  }
  vol = (nivel / 100) * 6.25; // area de 6.25m³
  lcd.setCursor(0, 0);
  lcd.print("Umidade: ");
  lcd.setCursor(9, 0);
  lcd.print(h);
  lcd.setCursor(15, 0);
  lcd.print("%t");
  lcd.setCursor(0, 1);
  lcd.print("Temp: ");
  lcd.setCursor(8, 1);
  lcd.print(t);
  lcd.setCursor(14, 1);
  lcd.print((char)223);
  lcd.println("C");
}

```

```

delay(1500);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("SOLO: ");
lcd.setCursor(8, 0);
lcd.print(analogRead(A3));
lcd.setCursor(0, 1);
lcd.print(valorLido);
lcd.setCursor(4, 1);
lcd.print("%");
delay(1500);
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Nivel cm: ");
lcd.setCursor(10, 0);
lcd.print(nivel);
lcd.setCursor(0, 1);
lcd.print("Vol. m3:");
lcd.setCursor(9, 1);
lcd.print(vol);
delay(1500);
lcd.clear();
lcd.setCursor(1, 0);
lcd.print("HORA:");
lcd.setCursor(7, 0);
lcd.print(rtc.getTimeStr());
lcd.setCursor(1, 1);
lcd.print("DATA:");
lcd.setCursor(7, 1);
lcd.print(rtc.getDateStr(FORMAT_SHORT));
delay(1500);
lcd.clear();
if (nivel <= 20)
{
  digitalWrite(ledON, LOW);
  digitalWrite(ledOFF, HIGH);
  if ( rele1 != HIGH) {
    digitalWrite(rele1, HIGH);
    motor = "Negado";
  }
}
else
{
  digitalWrite(ledON, HIGH);
  digitalWrite(ledOFF, LOW);
  if ( rele1 != LOW) {
    digitalWrite(rele1, LOW);
  }
  motor = "Permitido";
}

if ( valorLido <= 60)
{
  ContAC++;
  If (ContAC > 3) {
    if (rele2 != HIGH) {
      digitalWrite(rele2, HIGH);
      ContAC=0;
    }
  }
}
}

```

```

}
else
{
  if (rele2 != LOW) {
    digitalWrite(rele2, LOW);
  }
}

if (cont >= 10){
  // Abre o arquivo arquivo.txt do cartao SD
  File dataFile = SD.open("arquivo.txt", FILE_WRITE);
  //Inserindo os dados no carto sd
  if (dataFile)
  {
    dataFile.print(" ");
    dataFile.print(nivel);
    dataFile.print(" ");
    dataFile.print(vol);
    dataFile.print(" ");
    dataFile.print(motor);
    dataFile.print(" ");
    dataFile.print(h);
    dataFile.print(" ");
    dataFile.print(t);
    dataFile.print(" ");
    dataFile.print(valorLido);
    dataFile.print(" ");
    dataFile.print(REEDCOUNT * 0.25);
    dataFile.print(" ");
    dataFile.print(rtc.getTimeStr());
    dataFile.print(" ");
    dataFile.print(rtc.getDateStr(FORMAT_SHORT));
    dataFile.println("");
    dataFile.close();
  }
  else
  {
    // Mensagem de erro caso ocorra algum problema
    // na abertura do arquivo
    lcd.clear();
    lcd.setCursor(3, 0);
    lcd.println("Erro SD!");
    delay(2000);
    lcd.clear();
    dataFile.close();
  }
  cont =0;
}
//convertendo para char
String enviados;
char dados[15];
char dados2[165];
double variable = nivel;
enviados = "| Nivel: ";
dtostrf(variable, 4, 2, dados);
enviados += dados;
variable = vol;
enviados += " cm | vol:";
dtostrf(variable, 7, 2, dados);
enviados += dados;

```

```

enviados += " m3 | Umi: ";
enviados += h;
enviados += " % |";
enviados += " Temp: ";
enviados += t;
enviados += " C |";
enviados += " Umi_solo: ";
enviados += valorLido;
enviados += " % |";
enviados += " Precipitacao: ";
enviados += REEDCOUNT * 0.25;
enviados += " mm |";
enviados += " Hora: ";
enviados += rtc.getTimeStr();
enviados += " | Data: ";
enviados += rtc.getDateStr(FORMAT_SHORT);
enviados += " | Motor: ";
enviados += motor;
enviados += " ";
enviados.toCharArray(dados2, 165);
delay(100);
Serial.write(dados2);
//Send new SMS command and message number
if (cont2 >= 10){
serialSIM800.write("AT+CMGS=\"01538998471997\"\\r\\n");
delay(100);
//Send SMS content
serialSIM800.write(dados2);
//Send Ctrl+Z / ESC to denote SMS message is complete
serialSIM800.write((char)26);
delay(100);
//Serial.println("SMS Sent!");
serialSIM800.write("AT + CPMS = SM");
delay(100);
cont2 = 0;
}
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("!!LENDO SENSOR!!");
delay(1000);
lcd.clear();
if(cont3 >= 225)
{
  if (rele2 != HIGH) {
    digitalWrite(rele2, HIGH);
  }
  cont3=0;
  delay(7000);
}
cont++;
cont2++;
cont3++;
}
while (digitalRead(chave) == HIGH)
{
  digitalWrite(ledON, LOW);
  digitalWrite(ledOFF, LOW);
  digitalWrite(ledSTD, HIGH);
  lcd.setCursor(1, 0);
  lcd.print("!Controle");
}

```

```

    lcd.setCursor(11, 0);
    lcd.print("OFF!");
    digitalWrite(rele1, LOW);
    digitalWrite(rele2, LOW);
    delay(2000);
    lcd.clear();
    delay(100);
    lcd.setCursor(1, 0);
    lcd.print("HORA:");
    lcd.setCursor(7, 0);
    lcd.print(rtc.getTimeStr());
    lcd.setCursor(1, 1);
    lcd.print("DATA:");
    lcd.setCursor(7, 1);
    lcd.print(rtc.getDateStr(FORMAT_SHORT));
    delay(12500);
    lcd.clear();
}
}

```

O segundo código abaixo foi utilizado para receber os dados do Arduíno MEGA e enviar via WEB para acesso local e remoto no microcontrolador ESP32, através da abertura de porta externa no roteador.

```

#include <WiFi.h>
#include <WebServer.h>
#include <HardwareSerial.h>
HardwareSerial serialmega(2);

// SSID & Password
const char* ssid = "wireless.hidraulica"; // Enter your SSID here
const char* password = "ica2018hidra"; //Enter your Password here
WebServer server(80); // Object of WebServer(HTTP port, 80 is default)
String texto="";
String texto2="";
const int led = 2;

void setup() {
  pinMode(led,OUTPUT);
  digitalWrite(led, 0);
  Serial.begin(115200);
  serialmega.begin(115200, SERIAL_8N1, 16, 17); // Pino RX e TX
  Serial.println("Try Connecting to ");
  Serial.println(ssid);

  // Connect to your wi-fi modem
  WiFi.begin(ssid, password);

  // Check wi-fi is connected to wi-fi network
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected successfully");
}

```

```

Serial.print("Got IP: ");
Serial.println(WiFi.localIP()); //Show ESP32 IP on serial

server.on("/", handle_root);

server.begin();
Serial.println("HTTP server started");
delay(100);
}

void loop() {
  server.handleClient();
  if (serialmega.available() > 0) {
    texto = serialmega.readString();
    Serial.println(texto);
    digitalWrite(led,HIGH);
  }
  digitalWrite(led,LOW);
  texto2 = "Mensagem: ";
  texto2 += texto;
}
// HTML & CSS contents which display on web server
String HTML = "<!DOCTYPE html>\n
<html>\n
<body>\n
<h1> Web Server ESP32 - Station Mode - ICA/UFGM &#128522;</h1>\n
<h2> Local: Montes Claros </h2>\n
<h2> Proprietario: Leonardo N Lima </h2>\n
<br>\n
</body>\n
</html>";
// Handle root url (/)
void handle_root() {
  server.send(200, "text/html", HTML+=texto2+="<br>");
  //server.send(200, "text/html", page+=texto );
}

```