

REDE DOCTUM DE ENSINO
FACULDADE DOCTUM DE CARATINGA
CURSO SUPERIOR DE ENGENHARIA ELÉTRICA

**AUTOMATIZAÇÃO DO SISTEMA DE IRRIGAÇÃO DE MILHO UTILIZANDO A
PLATAFORMA ARDUINO.**

JOÃO VIEIRA DIAS NETO

Trabalho de Conclusão de Curso

Caratinga/MG

2017

JOÃO VIEIRA DIAS NETO

**AUTOMATIZAÇÃO DO SISTEMA DE IRRIGAÇÃO DE MILHO UTILIZANDO A
PLATAFORMA ARDUINO.**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso Superior de Engenharia Elétrica da Faculdade DOCTUM de Caratinga como requisito parcial para obtenção do Grau de Bacharel em Engenharia Elétrica.
Professor Orientador: Vinicius Murilo Lima Rodrigues.

Caratinga/MG

2017

JOÃO VIEIRA DIAS NETO

**AUTOMATIZAÇÃO DO SISTEMA DE IRRIGAÇÃO DE MILHO UTILIZANDO A
PLATAFORMA ARDUINO.**

Trabalho de Conclusão de Curso apresentado à Banca Examinadora do Curso Superior de Engenharia Elétrica da Faculdade DOCTUM de Caratinga como requisito parcial para obtenção do Grau de Bacharel em Engenharia Elétrica.

**APROVADA PELA COMISSÃO EXAMINADORA
EM CARATINGA, XX DE DEZEMBRO DE 2017**

Prof. Joildo Fernandes Costa Júnior, (DOCTUM/MG)
Coordenador do Curso

Prof. Vinicius Murilo Lima Rodrigues, (DOCTUM/MG)
Orientador

Prof. Daniel Mageste Butters, (DOCTUM/MG)
Examinador

Prof. José Eugenio, (DOCTUM/MG)
Examinador

Deus, eu não poderia ter chegado até aqui sem ti.

AGRADECIMENTOS

A caminho desta conquista vivemos muitas experiências, ampliamos nossa visão e nosso conhecimento. Aprendemos a conviver, a não desistir de lutar, a tentar novamente, a superar e a surpreender. Hoje, a imaginação toma forma e histórias são revividas seja através de lágrimas ou sorrisos. O certo é que, ao olharmos para trás, precisamos reconhecer e agradecer a todos que contribuíram para que, hoje, pudéssemos chegar aonde chegamos.

Agradeço a Deus acima de tudo! A meus pais Valter Ferreira e Edilamar Silvério pelo amor e dedicação. Aos professores do Curso de Engenharia Elétrica, em especial meu orientador Vinicius Murilo e ao coordenador Joildo Fernandes pelos ensinamentos e dedicação, minha namorada Saliny, meus amigos, Ramon, Letícia, José Francisco, e todos que contribuíram nesta caminhada. Esta vitória tem o sabor das dificuldades superadas, do dever cumprido, das sólidas amizades e dos momentos inesquecíveis compartilhados.

“Mesmo desacreditado e ignorado por todos, não posso desistir, pois para mim, vencer é nunca desistir.”

(ALBERT EINSTEIN)

DIAS NETO, João Vieira. **AUTOMATIZAÇÃO DO SISTEMA DE IRRIGAÇÃO DE MILHO UTILIZANDO A PLATAFORMA ARDUINO**. Caratinga, 2017. Trabalho de Conclusão de Curso Superior de Engenharia Elétrica - Curso de Engenharia Elétrica. Faculdades Integradas de Caratinga, Rede DOCTUM, Caratinga, 2017.

RESUMO

Com o avanço tecnológico, houve um grande aumento pela busca de comodidade e praticidade em processos manuais. Um destes processos manuais é a irrigação de plantações de milho, onde o agricultor aciona a bomba de irrigação diariamente e controla o nível de água no solo com base em observações visuais. O presente trabalho apresenta um sistema de irrigação automatizado em uma plantação de milho utilizando plataforma Arduino com o objetivo de analisar a viabilidade de sua aplicação para atender um pequeno produtor de milho. O elemento fundamental deste projeto é a plataforma Arduino Mega, que possibilitou o desenvolvimento de toda lógica através de sua IDE de programação. A utilização do sensor de umidade do solo garante que a irrigação seja eficiente, não ultrapassando o nível necessário para o desenvolvimento da cultura do milho. Com a utilização de um *display*, é possível que o próprio agricultor monitore os níveis de umidade e em casos de necessidade acione a irrigação antes do horário determinado. Por fim, foram analisados os resultados obtidos com esta implementação, comprovando assim a sua viabilidade de instalação de acordo com os testes realizados.

Palavras-chave: Automação. Irrigação. Arduino.

DIAS NETO, João Vieira. AUTOMATIZAÇÃO DO SISTEMA DE IRRIGAÇÃO DE MILHO UTILIZANDO A PLATAFORMA ARDUINO. Caratinga, 2017. Trabalho de Conclusão de Curso Superior de Engenharia Elétrica - Curso de Engenharia Elétrica. Faculdades Integradas de Caratinga, Rede DOCTUM, Caratinga, 2017.

ABSTRACT

With the technological advance, there was a great increase for the search of convenience and practicality in manual processes. One of these manual processes is the irrigation of corn plantations where the farmer drives the irrigation pump on a daily basis and controls the water level in the soil based on visual observations. The present work presents an automated irrigation system in a corn plantation using Arduino platform with the objective of analyzing the viability of its application to attend a small producer of green corn. The fundamental element of this project is the Arduino Mega platform that enabled the development of all logic through its programming IDE. The use of the soil moisture sensor ensures that the irrigation is efficient, not exceeding the level necessary for the development of the maize crop. With the use of a display, it is possible for the farmer to monitor the humidity levels and, in case of need, to activate the irrigation before the set time. Finally, the results obtained with this implementation were analyzed, thus proving their feasibility of installation according to the tests performed.

Key-words: Automation. Irrigation. Arduino.

LISTA DE ILUSTRAÇÕES

Figura 1 – Sistema de Irrigação por Aspersão: sistema convencional com movimentação manual (à esquerda), sistema por pivô central (centro) e sistema a carretel enrolador (à direita).	15
Figura 2 – Sistema de Irrigação Localizada: gotejamento (à esquerda) e microaspersão (à direita).	15
Figura 3 – Sistema de Irrigação Superficial: sulcos (à esquerda) e inundação (à direita).	15
Figura 4 – Sistema de Irrigação Subsuperficial: elevação do lençol freático (à esquerda), gotejamento subterrâneo (centro) e mesas de subirrigação (à direita).	16
Figura 5 – Diagrama de blocos.	19
Figura 6 – Arduino Duemilanove.	20
Figura 7 – Arduino UNO.	20
Figura 8 – Arduino NANO.	21
Figura 9 – Arduino MEGA.	21
Figura 10 – IDE Arduino.	22
Figura 11 – Fluxograma do sistema de irrigação.	27
Figura 12 – Fluxograma da interface local.	28
Figura 13 – Simulação do projeto Fritzing.	29
Figura 14 – Protótipo do sistema de irrigação.	30
Figura 15 – Funcionamento do <i>display</i> do protótipo.	31
Figura 16 – Fluxograma de controle.	32
Figura 17 – Retorno do protótipo em dias quentes – anterior acionamento da bomba.	34
Figura 18 – Retorno do protótipo em dias quentes – durante o funcionamento da bomba.	35
Figura 19 – Fases de Desenvolvimento da Cultura do Milho	53

LISTA DE ABREVIATURAS E SIGLAS

LCD	Liquid Crystal Display
Kc	Coeficiente de Cultura
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
ETc	Evapotranspiração da Cultura
RTC	Real Time Clock
IDE	Integrated Development Environment
USB	Universal Serial Bus
ICSP	In Circuit Serial Programming
FTDI	Future Technology Devices International
PWM	Pulse Width Modulation
SENAI	Serviço Nacional de Aprendizagem Industrial
CC	Capacidade de Campo
UC	Unidade Crítica
PMP	Ponto de Murcha Permanente

LISTA DE SÍMBOLOS

m^2	Metro quadrodo
mm	Milímetro
m	Metro
KB	Kilobyte
cm	Centímetro.
MHZ	Megahertz
kPa	Quilopascal

SUMÁRIO

1 INTRODUÇÃO	12
2 DESENVOLVIMENTO	14
2.1 Métodos de Irrigação	14
2.2 Irrigação do Milho	16
2.3 Plataforma Arduino	18
2.3.1 Estrutura Arduino	18
2.3.2 Modelos de Arduinos	19
2.4 Sensores	22
3 METODOLOGIA	25
3.1 Objeto de estudo	25
3.2 Arquitetura da aplicação	25
3.2.1 Arduino Mega 2560	25
3.2.2 Sensor de umidade do solo	26
3.2.3 Motor elétrico	26
3.3 Projeto	27
3.4 Opções Do Controlador	28
3.5 Implementação	29
3.6 Modo de Operação do Controlador	31
4 ANALISE DOS RESULTADOS	33
4.1 DESAFIOS ENCONTRADOS	33
4.2 Resultados da implementação	33
4.2.1 Dias de clima quente – ensolarados	34
4.2.2 Dias de clima úmido – chuva	35
4.2.3 Resultados obtidos	36
5 CONCLUSÃO	37
5.1 TRABALHOS FUTUROS	37
6 REFERÊNCIAS BIBLIOGRÁFICAS	39
APÊNDICE A CÓDIGO FONTE	41
ANEXO A FASES DO MILHO	53

1 INTRODUÇÃO

Irrigação é uma ferramenta de grande importância, e uma técnica utilizada na agricultura que teve sua origem nas civilizações antigas, utilizada nas margens de grandes rios como rio Nilo no Egito, os rios Tigres e Eufrates na Mesopotâmia, entre outros, com principal objetivo de controlar o fornecimento de água para produção em quantidade suficiente e no momento certo, com base nas condições do solo e disponibilidade da água (ROTONDANO; MELO, 2017).

Com a evolução agrícola, a necessidade de automatizar os recursos produtivos, a competitividade no mercado, a necessidade de aumento de produtividade e redução de custos, levam a automação se fazer necessária não somente pela possibilidade de diminuição dos custos com mão de obra, mas principalmente por necessidades operacionais, tais como irrigação de grandes áreas no período noturno.

O milho é considerado uma hortaliça, em virtude do tempo de sua permanência no campo até o momento da colheita, seu tempo de produção é curto aproximadamente 90 dias no verão e de 100 dias no inverno. A agricultura é uma opção de grande valor econômico, principalmente para os produtores que utilizam mão de obra familiar, graças ao bom preço de mercado, à significativa demanda pelo produto natural e a crescente procura pelo produto por parte da indústria tornam o mercado de milho tão promissor e lucrativo (MARSCZAOKOSKI; CRUZ, 2013).

Devido ao uso dos equipamentos ligados à noite, tem-se um melhor proveito da água, já que neste período geralmente a velocidade do vento é menor, não há influência do sol e a umidade do ar é mais alta, portanto os desvios da água pelo vento são menores. Sendo assim, quase não se perde por evaporação e diminuição da umidade, havendo uma maior eficiência (YONEYA, 2017).

Este estudo delimitou-se na automatização de um sistema de irrigação já existente, onde o controle é realizado com tecnologia Arduino, ligado há um sensor de umidade no solo para que não seja acionado em dias de chuva. O acionamento será pré-programado para o período noturno para um melhor proveito da água.

Esta delimitação do tema vai ao encontro do seguinte problema de estudo: quais os benefícios da automatização da irrigação utilizando plataforma Arduino? Esta questão-problema veio de encontro com os objetivos deste trabalho que são os seguintes:

- a) Analisar a viabilidade da aplicação de um sistema automatizado na irrigação utilizando plataforma Arduino e desenvolver um sistema voltado para atender um pequeno produtor de milho;
- b) Realizar um estudo de automatização baseado em Arduino;

- c) Definir a plataforma Arduino que atenda as exigências do projeto;
- d) Definir a estrutura do sistema: sensores, contadores;
- e) Desenvolver a programação do Arduino;
- f) Desenvolver e testar o protótipo do sistema;
- g) Efetuar testes manipulando os sensores;

De acordo com os objetivos é sustentado a seguinte hipótese: é possível automatizar o processo de irrigação utilizando Arduino, com o intuito de implantar o monitoramento, eficiência e produtividade, sendo mais vantajoso para grandes áreas de plantio.

O desenvolvimento deste trabalho foi realizado através da metodologia de pesquisa exploratória por se tratar de um estudo de caso. O ambiente de pesquisa é uma plantação de milho da região que utilizava a irrigação de forma manual e sem supervisão. A escolha pelo Arduino se deu devido a sua facilidade e utilidade, resultando em um sistema com precisão e com baixo custo, sendo possível a sua utilização tanto para grandes agricultores como para pequenos que desejam automatizar sua irrigação.

A coleta de dados foi realizada através de observação participante já que o próprio autor realizou a simulação do sistema de irrigação e realizou as análises dos resultados obtidos.

Em seu desenvolvimento, primeiramente foi realizado uma pesquisa bibliográfica com o intuito da compreensão dos assuntos ligados ao tema em questão. Com um melhor entendimento do assunto a ser tratado, foi dado início ao estudo de caso onde foi definido os equipamentos necessários para a automatização, e por fim a análise dos resultados obtidos.

De acordo com o tema apresentado e os objetivos descritos anteriormente, a presente monografia teve sua estrutura definida em cinco capítulos.

O primeiro capítulo aborda a pesquisa bibliográfica utilizada como base para o desenvolvimento deste estudo como a tecnologia Arduino, sistemas de irrigação e sensores.

O segundo capítulo apresenta o estudo de caso descrevendo todo o processo de implementação do sistema de irrigação automatizado com plataforma Arduino.

O terceiro e quarto capítulos demonstram os resultados obtidos através da implementação do sistema de irrigação automatizado com plataforma Arduino em comparação ao cenário anterior, onde a irrigação era realizada manualmente.

O quinto e último capítulo apresenta conclusões alcançadas através desta pesquisa e as sugestões para futuros trabalhos.

2 DESENVOLVIMENTO

As seções apresentadas a seguir abordam os principais temas utilizados no desenvolvimento deste estudo que se mostrou necessário para alcançar os objetivos traçados.

2.1 MÉTODOS DE IRRIGAÇÃO

De acordo com Pereira (2014), pode-se definir irrigação como:

...a aplicação artificial, uniforme e oportuna de água, distribuída pontualmente na zona efetiva das raízes ou na área total, visando repor a água consumida pelas plantas, a perda por evaporação, transpiração e por infiltração profunda de forma a garantir condições ideais ao bom desenvolvimento das plantas. (PEREIRA, 2014, p.2)

A irrigação é uma prática que busca uma maior produtividade e qualidade dos produtos mesmo fora de época e principalmente quando a chuva é escassa. Porém, para implantar um sistema de irrigação é necessário a observação de certos aspectos referente a cultura e o ambiente da plantação, buscando sua eficiência através de um rendimento maior e economia da água utilizada. Os principais fatores são (PEREIRA, 2014):

- a) Conhecer a cultura: deve-se conhecer as exigências climáticas, a necessidade de diferentes exigências de água nas etapas de desenvolvimento da cultura, o crescimento das raízes e espaçamento;
- b) Solo: é necessário saber qual o tipo do solo, sua fertilidade e textura;
- c) Avaliação da água: é preciso verificar a qualidade da água utilizada através de amostra e verificar a sua vazão e o volume da água;
- d) Área a ser irrigada: é necessário verificar a sua topografia, drenagem e impedimentos físicos;
- e) Clima: é preciso verificar fatores como chuva, vento e temperatura.

Os métodos de irrigação são definidos de acordo com a maneira como a água é aplicada no solo. Os quatro principais métodos de irrigação são: aspersão, localizada, superfície e Subsuperfície ou subterrânea (TESTEZLAF, 2017):

- a) Aspersão: neste tipo de irrigação a água é lançada sobre as folhagens e acima do solo. Ela pode ser convencional ou mecanizado (pivô central e carretel).



Figura 1: Sistema de Irrigação por Aspersão: sistema convencional com movimentação manual (à esquerda), sistema por pivô central (centro) e sistema a carretel enrolador (à direita).

Fonte: (TESTEZLAF (2017, p. 13).

b) Localizada: neste caso a água é aplicada em uma área determinada próximo ao caule e debaixo da copa da planta umedecendo uma área restrita da mesma. Pode ser por gotejamento ou microaspersão.



Figura 2: Sistema de Irrigação Localizada: gotejamento (à esquerda) e microaspersão (à direita).

Fonte: (TESTEZLAF (2017, p. 14).

c) Superfície: a água é aplicada diretamente sobre a superfície do solo sendo absorvido pela gravidade. Este sistema pode ser por sulcos e por inundação.



Figura 3: Sistema de Irrigação Superficial: sulcos (à esquerda) e inundação (à direita).

Fonte: (TESTEZLAF (2017, p. 12).

d) Subsuperfície ou subterrânea: a água é aplicada abaixo da superfície da terra diretamente nas raízes.

A irrigação pode ser por gotejamento subterrâneo, elevação do lençol freático e sistemas de subirrigação de ambientes protegidos.



Figura 4: Sistema de Irrigação Subsuperficial: elevação do lençol freático (à esquerda), gotejamento subterrâneo (centro) e mesas de subirrigação (à direita).

Fonte: (TESTEZLAF (2017, p. 14).

O método de irrigação deve ser definido de acordo com a cultura em questão, sendo necessário compreender os fatores ligados ao seu desenvolvimento. Este trabalho se trata de uma plantação de milho, sendo necessário entender melhor sobre esta cultura.

2.2 IRRIGAÇÃO DO MILHO

O milho é o terceiro cereal mais importante para economia do Brasil, perdendo apenas para o trigo e o arroz. É uma planta típica de países quentes com mecanismos próprios para economia de água nestas condições. Devido a este fator e por ser uma planta com o coeficiente hídrico muito baixo, a irrigação se torna necessário (BIOSEMENTES, 2017).

De acordo com a Empresa Brasileira de Pesquisa Agropecuária - EMBRAPA (2017) o cultivo de milho exige muita água, porém é eficiente no seu uso, com grande acúmulo de matéria seca por unidade de água absorvida. O volume de água para irrigação deve ser estabelecida a ponto de molhar uma espessura em torno de 0,70m. Devido a conscientização sobre a escassez de recursos hídricos, o sistema de gotejamento foi adaptado para as plantações de milho. Este tipo de irrigação trouxe benefícios a plantação de milho (BIOSEMENTES, 2017):

- a) Aumento do rendimento de 30-40%;
- b) Redução de períodos de estresse hídricos reduzindo ataques de micotoxinas resultando na melhor qualidade dos grãos;
- c) Economia hídrica em torno de 30% devido a não ter perda pela evaporação e ser indiferente ao vento.

O consumo hídrico do milho pode variar entre 380 e 500 mm de acordo com as condições climáticas, onde a maior exigência de água é no período de pendramento e enchimento dos grãos para garantir maior produtividade (CRUZ et al., 2010). Desta forma, a irrigação deve ser programada de acordo com as necessidades hídricas em todas as fases fenológicas.

A programação da irrigação deve considerar alguns conceitos como o coeficiente de cultura (K_c) e a evapotranspiração da cultura (ET_c). O termo evapotranspiração se refere à evaporação da água através de solos parcialmente vegetados, onde há a transpiração da planta e a evaporação da água existente no solo (BERLATO E MOLION, 1981). O coeficiente de cultura é determinado através de observações, variando de acordo com a cultura e estágio de desenvolvimento com clima e as práticas adotadas (DOORENBOS; PRUITT, 1977). Estes parâmetros são necessários para estimar a necessidade hídrica de uma cultura.

A utilização de sensores auxilia na irrigação, verificando a umidade existente no solo evitando um estresse hídrico. Para que esta medição seja efetiva deve ser realizada em três pontos representativos na área de plantio em duas profundidades diferentes, em uma zona máxima radicular onde a área mediana da profundidade efetiva para cultura em seu máximo desenvolvimento, sendo o ponto A e o restante nas partes inferiores da zona radicular sendo o ponto B. Sendo assim no caso do milho, de acordo com Pereira Filho (2002), se for utilizar apenas de sensores para verificação da umidade, é necessário realizar irrigações com a zona radicular frequência após a semeadura durante 15 dias, entre 15 e 30 dias deve ser instalado o sensor a 10 cm no ponto A e a 20 cm de profundidade no ponto B. Após 30 dias deve-se colocar o sensor a 20 cm de profundidade no ponto A e a 40 cm de profundidade no ponto B. O ponto A será usado como critério para o momento da irrigação e as medições do ponto B servirão de apoio para controle da infiltração da água no solo.

Utilizando esta metodologia, há um maior controle referente à irrigação, onde a mesma será feita de acordo com a necessidade da cultura em questão, sem que haja desperdício através de irrigação desnecessária, o que resulta numa maior produtividade do milho. Conhecendo a cultura e escolhendo a forma de irrigação a ser trabalhada, é possível definir a melhor forma de automatizar este processo utilizando Arduino.

O solo é um material poroso que retém a água por capilaridade e adsorção. Se um solo é encharcado, a água drena para as camadas mais profundas, fora do alcance das raízes. Se não houver perdas por evapotranspiração, depois de um a dois dias para solos mais arenosos, estabelece-se um equilíbrio entre as forças capilares e de adsorção e força gravitacional: a drenagem cessa e o solo atinge a Capacidade de Campo (CC) (BERNARDO, 1997).

A medida que o milho absorve água do solo, a umidade do solo diminui, até atingir a Umidade Crítica (UC), ponto que compromete o pleno desenvolvimento da cultura, correspondente a 50 kPa. Se o milho continuar absorvendo água do solo, em um determinado momento pode-se chegar ao Ponto de Murcha Permanente (PMP), umidade na qual a planta não consegue mais retirar água do solo, que é de 1500 kPa. A quantidade de água do solo entre a CC e o PMP é

denominada água disponível no solo (BERNARDO, 1997). Quando a água está no solo significa que ela está retida por uma determinada tensão. Normalmente, defini-se que a água está na CC quando se encontra retida a uma tensão de 10 kPa, (BERNARDO et al., 2006).

2.3 PLATAFORMA ARDUINO

O Arduino surgiu na Itália em 2005, no *Interaction Design Institute*, onde o professor Massimo Banzi e o pesquisador visitante David Cuartielles da Universidade de Malmö, na Suécia procuravam uma solução mais barata para que os estudantes de design trabalhassem com tecnologia. Seu objetivo era criar um microcontrolador que pudesse ser utilizado pelos alunos e que seu preço fosse baixo. Cuartielles desenhou a placa e David Mellis, um aluno de Massimo, programou o *software*. Gianluca Martino, um engenheiro local contratado por Massimo produziu duzentas placas que recebeu o nome de Arduino, como referência a um bar local conhecido pelos alunos e professores do instituto (EVANS et al, 2013).

Segundo Banzi e Shiloh (2015), o Arduino pode ser definido como uma “plataforma de computação física aberta” para elaboração de instrumentos com a colaboração de computadores ou independentes. Por ser de fonte aberta, promove uma comunidade que faz com que o compartilhamento de conhecimento seja abundante, facilitando aos iniciantes com pouco conhecimento devido aos diversos usuários de diferentes níveis dispostos a ajudar.

O Arduino é uma placa de circuito impresso que utiliza um chip microcontrolador com entradas e saídas que combina *hardware* e *software* sendo de fácil entendimento e uso (NYSSEI, 2013).

2.3.1 Estrutura Arduino

O Arduino é uma plataforma de computação física que utiliza como base uma placa de entrada e saída com um microcontrolador. Este microcontrolador possui processador, memória e periféricos como se fosse um computador dentro de um chip. Assim pode-se compreender que o Arduino é como um computador que recebe dados de entrada através de sensores, processa os dados e gera a saída através de atuadores, interagindo com o ambiente (FONSECA; BEPPU, 2010).



Figura 5: Diagrama de blocos.

Fonte: FONSECA; BEPPU (2010, p. 2).

O Arduino é formado por duas partes principais: o *hardware* que é a parte física, ou seja, a placa e o *software* que é o ambiente de desenvolvimento integrado (IDE) do Arduino. Esta tecnologia com microprocessadores substituiu os fios antes utilizados para executar funções por um *software* que facilita no momento de fazer uma alteração (BANZI; SHILOH, 2015).

Devido ao fato de ser um projeto livre que pode ser utilizado e adaptado por qualquer pessoa, surgiram várias placas com base no projeto inicial do Arduino. A equipe de desenvolvimento restringiu apenas sobre o nome, no qual as novas placas criadas não podem utilizar o nome Arduino (MCROBERTS, 2011).

2.3.2 Modelos de Arduinos

As diversas versões atuais do Arduino utilizam como base o microcontrolador Atmel AVR de 8 bits. O primeiro modelo foi baseado no microcontrolador ATmega8 onde posteriormente passou a utilizar o ATmega168. Os modelos mais recentes utilizam o microcontrolador ATmega328 com memória flash de 32 KB (EVANS et al, 2013). Os modelos mais comuns de Arduinos são Arduino Duemilanove, Arduino Uno, Arduino Nano e Arduino Mega 2560.

O Arduino Duemilanove é a última de um conjunto de placas USB Arduino, seu nome italiano significa 2009. É uma placa de microcontrolador fundamentada no ATmega168 ou ATmega328, que dispõe de um oscilador de cristal de 16 MHz, 14 pinos de entrada/saída, 6 entradas analógicas, uma conexão USB, uma tomada de energia, um cabeçalho ICSP e um botão de reinicialização. Possui ainda um microcontrolador adicional FTDI responsável pela conversão usb/serial (ARDUINO, 2017).

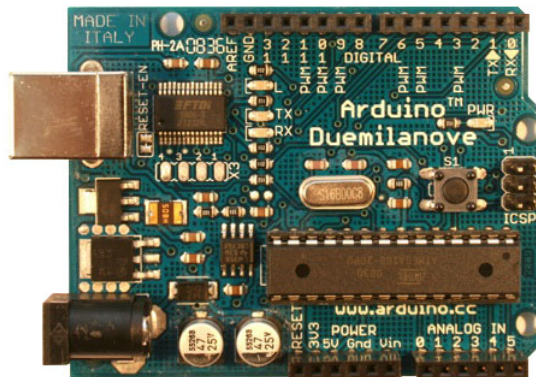


Figura 6: Arduino Duemilanove.

Fonte: ARDUINO (2017).

O Arduino Uno é a placa mais utilizada de toda família Arduino e consequentemente a que possui maior documentação disponível. Seu nome italiano significa “um” para marcar o lançamento do *Arduino software* 1.0 se tornando referência. É a primeira placa USB de um conjunto de placas Arduino. Utiliza o microcontrolador ATmega328p, um cristal de quartzo de 16 MHz, 14 pinos de entrada/saída, 6 entradas analógicas, uma conexão USB, uma tomada de energia, um cabeçalho ICSP e um botão de reinicialização. Possui também um microcontrolador adicional ATmega8U2, responsável pela conversão de USB/serial (ARDUINO, 2017).

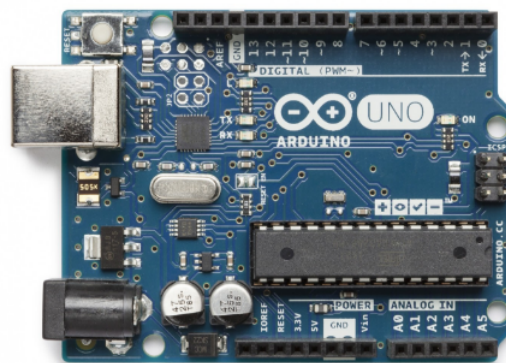


Figura 7: Arduino UNO.

Fonte: ARDUINO (2017).

O Arduino Nano é uma placa compacta, porém idêntica ao Arduino UNO com um quadro menor baseado no ATmega328, não possui uma tomada de energia e utiliza cabo mini-usb (ARDUINO, 2017).

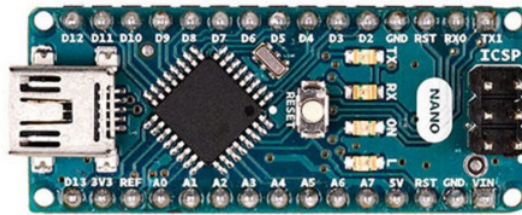


Figura 8: Arduino NANO.

Fonte: ARDUINO (2017).

O Arduino Mega 2560 é uma placa de microcontrolador com base no ATmega2560 indicada para projetos mais complexos que exigem um maior número de pinos de entrada e saída e de maior capacidade de memória de programa. Possui 54 pinos de entrada/saída, onde 15 pinos podem ser usados como saída PWM, 16 entradas analógicas, 4 portas seriais, um cristal de 16 MHz, uma conexão USB, uma tomada de energia, um cabeçalho ICSP e um botão de reinicialização (ARDUINO, 2017).

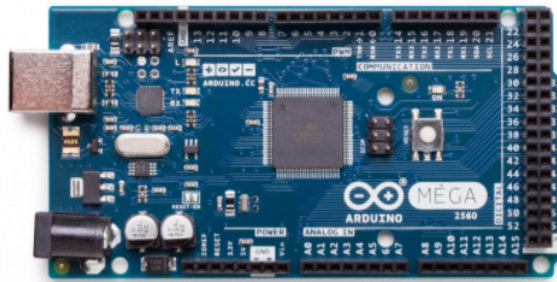


Figura 9: Arduino MEGA.

Fonte: ARDUINO (2017).

A IDE Arduino é o ambiente de desenvolvimento utilizado por programadores que possui um editor de texto, um console, uma barra de ferramentas com botões de suas funcionalidades e menus. É composto pelo editor, compilador, vinculador e depurador que junto ao *hardware* é responsável pela inserção de uma programa facilitando sua comunicação (RODRIGUES et al, 2012).

A linguagem de programação utilizada essencialmente é C/C++, mantendo várias características da linguagem. Hernando Barragan desenvolveu o Wiring que é uma plataforma para desenvolvimento própria para aplicações eletrônicas (MCROBERTS, 2011; ARDUINO, 2017).

A instalação da IDE é muito simples e seu layout pode ter algumas diferenças dependendo o sistema operacional utilizado. O arquivo de instalação está disponível no site do projeto www.arduino.cc/en/Main/Software, sendo necessário apenas o download e a instalação (MCROBERTS, 2011).

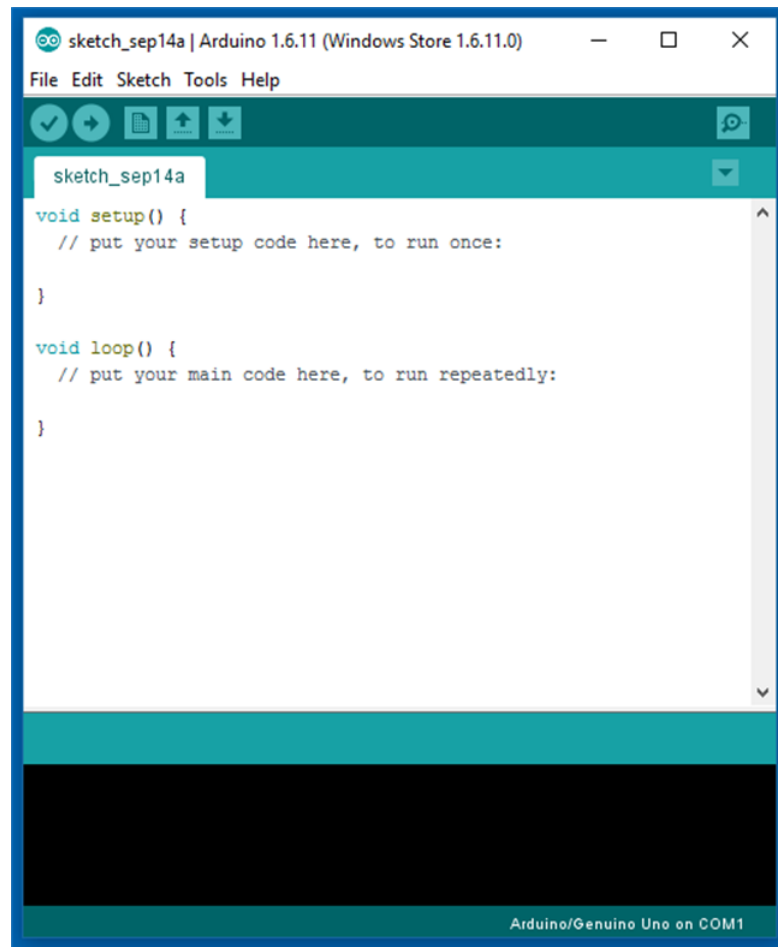


Figura 10: IDE Arduino.

Fonte: ARDUINO (2017).

De acordo com a figura 10, a IDE é dividido em três partes principais: o Toolbar com os botões de sua funcionalidade, o Sketch no centro onde é escrito o programa e a janela de mensagens no final.

O primeiro botão do Toolbar é o de verificar, ele faz a análise sintática do código escrito no sketch, conferindo se há erros de sintaxe. O segundo botão é o de Carregar que compila o código e carrega para o Arduino. O terceiro botão é o Novo que abre um sketch em branco pronto para receber um código. O quarto botão é o de Abrir com a função de localizar um projeto salvo e abrir. O quinto botão é o de Salvar, que grava o código escrito no sketch. O sexto e último botão é o de Monitor Serial que mostra os dados depurados que foram para o Arduino.

Para a construção de um sistema automatizado com Arduino, é necessário outros dispositivos para interagir com o ambiente. Alguns destes dispositivos são os sensores.

2.4 SENSORES

De acordo com Karvinen K. e Karvinen T. (2014) pode-se definir o seguinte:

Os sensores são componentes elétricos que funcionam como dispositivos de entrada ... Um Sensor é uma entrada física para um circuito. Os sensores são transdutores. O processo de converter a energia detectada em outra forma é chamado de transdução. Por exemplo, um sensor de faz a transdução de luminosidade para resistência. (KARVINNEN, K; KARVINEN, T., p. 16 e p. 24).

O sensor é o meio pelo qual o *hardware* interage com o ambiente. Um sistema quando é dito inteligente está repleto de sensores. Os sensores podem ser classificados em (SENAI, 2016):

- a) Autoalimentados: produzem uma saída sem a necessidade de alimentação externa;
- b) Com alimentação externa: necessita de alimentação de energia para produzirem um sinal de saída.

Os sensores também são classificados de acordo com o sinal de saída, que são de dois tipos (SENAI, 2016):

- a) Contínuo: os sinais de saída são proporcionais a uma faixa de medição de uma variável analógica determinada;
- b) Discreto: proporciona condições digitais de saída a ser acionadas.

Em sistemas de irrigação automatizados são utilizados quatro sensores de umidade de solo: blocos de resistência elétrica, tensiômetro, condutividade térmica e irrigás.

O bloco de resistência elétrica é um bloco de gesso poroso com eletrodos inseridos, onde a solubilização em água dos seus eletrólitos componentes causa a passagem de corrente elétrica entre estes eletrodos que é uma função não linear da tensão da água no solo. Estes blocos exigem uma calibração individual periodicamente devido a sua deterioração. É um sensor de baixo custo de fácil fabricação e com uma ampla faixa de resposta no tempo (NEVES JUNIOR, 2017).

O tensiômetro é um sensor formado por capsulas porosas que contem água em sua cavidade. Estas cavidades são seus dispositivos de medição de tensão que é mantido em contato com o solo ou inserido nele. Este sensor tem a vantagem de ser fácil de construir e não precisar de calibração, porém precisa de manutenções frequentes, por causa do acúmulo de ar na cavidade. Isso faz com que este sensor não seja indicado para um sistema de irrigação automatizado não assistido (NEVES JUNIOR, 2017).

O sensor de condutividade térmica é constituído por uma fonte de calor, com dissipação térmica ajustada e estável, comumente uma resistência elétrica centralizada e um sensor para acompanhar a diferença de temperatura entre dois pontos, ao longo do raio de capsulas porosas cilíndricas. Estas capsulas precisam ser calibradas individualmente e a relação entre a diferença da temperatura e a tensão da água não é linear, aumentando conforme o solo seca (NEVES JUNIOR, 2017).

O sensor irrigás possui capsulas porosas de tensão crítica de água apropriada a casa

cultivo. Quando o solo fica seco acima da tensão crítica, alguns poros se esvaziam o que faz o sensor permeável a passagem de gás (NEVES JUNIOR, 2017).

Para automatizar um sistema de irrigação é necessário sensores para identificar as variáveis físicas de um ambiente como a umidade e temperatura. São os sensores os responsáveis pelas informações obtidas e o envio para o sistema.

3 METODOLOGIA

3.1 OBJETO DE ESTUDO

O presente trabalho foi desenvolvido utilizando a metodologia de estudo de caso de caráter exploratório através de observação participante em pequena plantação de milho localizada na região de Caratinga.

A plantação de milho em questão fica localizada na zona rural da cidade de Caratinga com uma área medindo $3.910,52m^2$, plantação em questão está em atividade desde 1997 e possui um sistema de irrigação por aspersão de acionamento manual.

Ao observar que a irrigação poderia ser otimizada sem a necessidade de um acionamento manual, utilizando um protótipo que automatize esta atividade, controlando hora, duração, e condições do solo como a umidade, este estudo se mostrou necessário para a questão apresentada.

3.2 ARQUITETURA DA APLICAÇÃO

Para o desenvolvimento do trabalho em questão foram necessários os seguintes equipamentos:

- a) Sensor de umidade do solo higrômetro;
- b) Plataforma Arduino Mega 2560;
- c) Ds3231 At24c32 Arduino I2c Rtc *Real Time Clock*;
- d) *display* Lcd Tela Azul 20x4 2004 Arduino;
- e) Motor elétrico monofásico *Weg*;
- f) Módulo Relé 5V 1 Canal

A seguir será feito a descrição dos principais módulos utilizados.

3.2.1 Arduino Mega 2560

De acordo com o objetivo do trabalho que é automatizar um sistema de irrigação, foi definido a utilização de plataforma Arduino devido ao seu baixo custo e sua simplicidade em implementação. O modelo escolhido foi o Arduino Mega 2560 Atmega2560 por causa do seu maior número de pinos de entrada e saída que é ideal para projetos mais complexos.

Devido ao acionamento da irrigação ser realizada em um horário pré-definido, é necessário que este protótipo possua um meio de receber os dados em tempo real para seu perfeito

funcionamento. Para que isso fosse possível, foi utilizado um módulo de relógio de tempo real RTC DS3231 que tem a capacidade de fornecer informações como hora, minuto, segundo, dia, data, mês e ano. Este módulo possui uma bateria para evitar perda de dados em caso de queda de energia, mantendo as informações do relógio sempre atualizadas.

Outro componente utilizado para melhor desenvolvimento deste projeto junto a plataforma arduino escolhida foi um *display* LCD 20x4. Desta forma foi possível mostrar a data e hora atual e o estado do sensor de umidade para um melhor controle. Este módulo LCD pode exibir letras, números e símbolos exibindo 4 linhas de até 20 caracteres.

3.2.2 Sensor de umidade do solo

Foi utilizado para este estudo um sensor de umidade higrômetro que identifica variações da umidade do solo, onde o limite do solo seco e úmido é ajustado através de um potenciômetro este componente regula a saída do sensor, onde o solo seco a saída do sensor fica em estado alto e quando esta úmido a saída fica em estado baixo.

Por apresentar uma determinação imediata referente a umidade do solo, os sensores são mais indicados para determinar o início e a duração da irrigação.

De acordo com o retorno apresentado pelo sensor, que através da programação do arduino mostrará no *display* a porcentagem de umidade do solo em tempo real, será possível fazer o monitoramento do solo onde a irrigação ocorrerá obedecendo a parâmetros como o horário de início da irrigação e a umidade atual encontrada no solo. Caso o solo já se encontre úmido o suficiente, a irrigação não ocorrerá, evitando o encharcamento do solo, o que poderia ocorrer sem o uso de sensores.

Para a calibração do sensor capacitivo, procedeu-se com a construção de um modelo, envolvendo a umidade do solo e a capacitância. Para isso, fez-se a secagem de 4 amostras deformada e homogênea do solo retirado no mesmo local onde o sistema de irrigação seria implantado. As amostras secas foram pesadas e adicionadas diferentes volumes de água, na proporção de 10, 30, 40 e 90% de umidade, no sentido de determinar a capacitância do sensor em relação à porcentagem da água de cada amostra. Foram coletados os valores de capacitância em um sensor capacitor acoplado na plataforma Arduino, associou-se o potencial de irrigação do milho com a capacitância, tornando possível a programação no Arduino para que o mesmo libere o sinal digital de acionamento do relé sempre que o setpoint for atingido.

3.2.3 Motor elétrico

A automatização da irrigação de uma pequena plantação de milho foi realizada tendo como base um sistema de irrigação por aspersão manual já existente. Sendo assim, a parte estrutural da irrigação não foi alterada, mantendo o motor já utilizado. O motor usado é um motor

elétrico *Weg* monofásico de uso rural com capacitor permanente. Esse tipo de motor transforma energia elétrica em mecânica. Esse motor é de indução com corrente alternada, que funciona com uma velocidade constante. É um motor simples robusto e de baixo custo, o que faz com que ele seja o mais utilizado.

3.3 PROJETO

Após definido os *hardwares* que foram utilizados, foi determinado a estrutura do programa que foi utilizado para controlar a irrigação. A imagem a seguir apresenta um fluxograma de execução do sistema de irrigação.

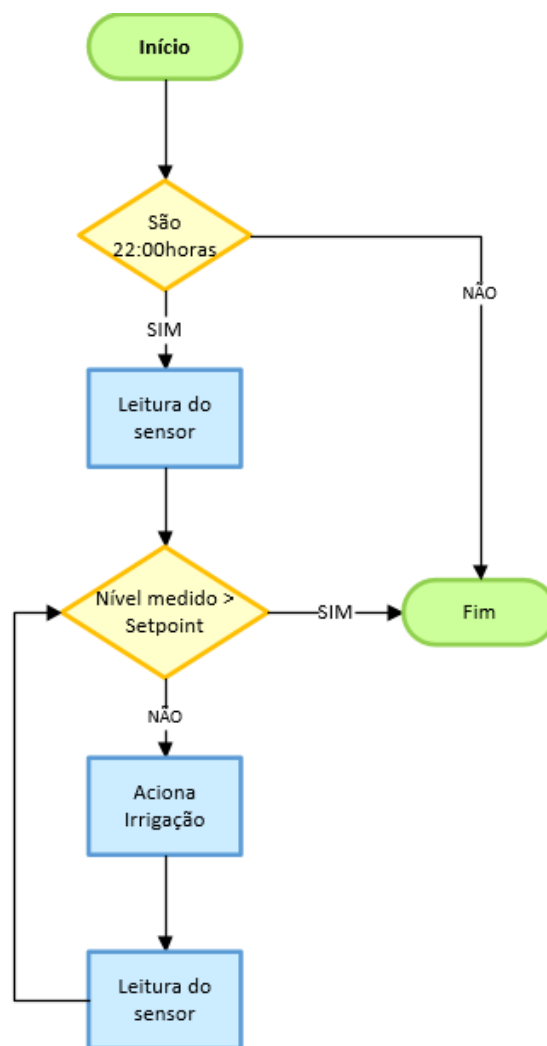


Figura 11: Fluxograma do sistema de irrigação.

Fonte: próprio autor.

De acordo com o fluxograma apresentado, a partir do horário de 22:00 é realizada a análise da umidade do solo através da leitura do sensor. É utilizado um nível de referência que no caso seria de 40% de umidade. Sendo assim, caso o sensor identifique que o nível de umidade

é inferior a 40% a bomba é acionada para irrigação até atingir a umidade mínima determinada. Quando o nível estiver igual ou superior os 40% (como em caso de chuvas), a bomba não será acionada.

3.4 OPÇÕES DO CONTROLADOR

Além do modo Manual e Automático, que são alteráveis pela interface local, o controlador disponibiliza a possibilidade de alteração de qualquer parâmetro. A lógica para a alteração dos parâmetros, na interface local, pode ser compreendida analisando-se a Figura 12. A rotina se mantém no “Controle do Processo” até que o botão Menu/Enter seja pressionado. Quando este evento ocorre o processo de irrigação é desativado e na tela do LCD são mostradas as opções. A primeira opção a ser alterada é apresentada ao usuário, ao pressionar o botão localizado a direita ou a esquerda fazem a navegação para a opção desejada, em seguida pressionando Menu/Enter os valores são gravados e enviado ao controlador, retornando ao processo que agora terá novos parâmetros de controle.

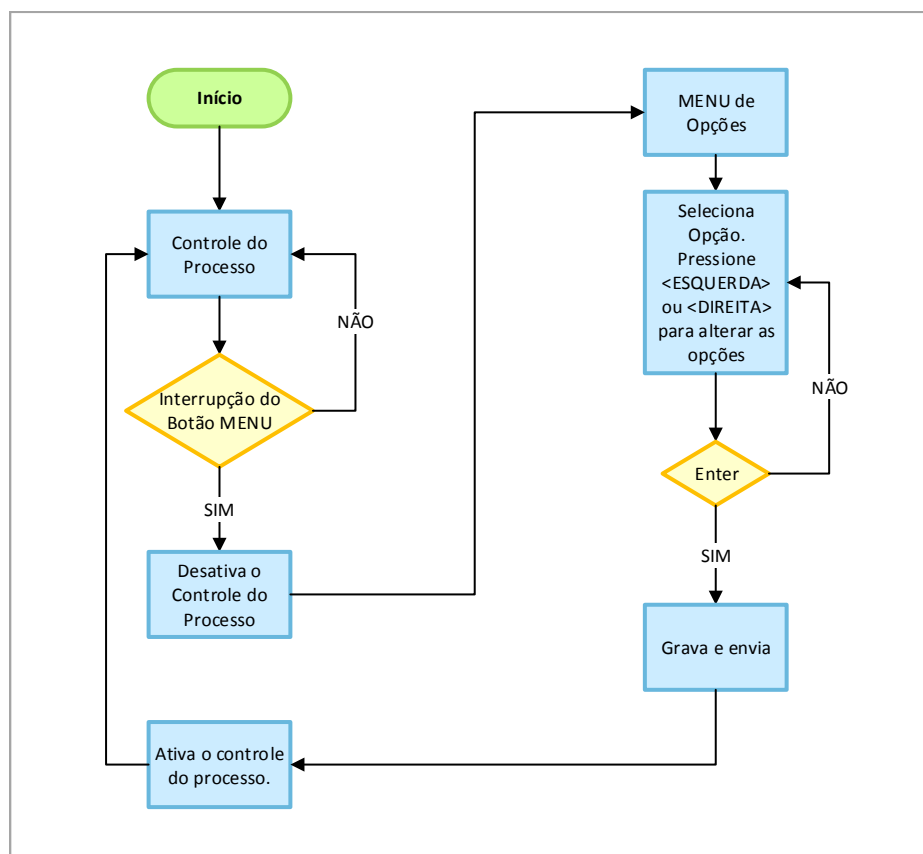


Figura 12: Fluxograma da interface local.

Fonte: próprio autor.

Abaixo segue a lista de todos os parâmetros que podem ser alterados via Menu:

a) Definir horário;

- b) Definir umidade (setpoint);
- c) Delay de acionamento;
- d) Modo de operação(tempo/umidade);
- e) Acerta data e hora;
- f) Zera programação.

3.5 IMPLEMENTAÇÃO

O projeto desenvolvido teve o objetivo de comprovar a viabilidade da implementação de um sistema automatizado para irrigação utilizando a plataforma Arduino e que atenda a um pequeno produtor de milho. Com este projeto espera-se que a irrigação automatizada resulte em um menor desperdício de água, já que haverá o controle da umidade do solo e um horário fixo noturno evitando a perda por evaporação.

Antes da montagem do projeto, foi criado um protótipo das ligações de circuitos através do *software* FRITZING 0.9.2, que tem as opções de utilizar o código da programação e desenhar o sistema da placa do projeto.

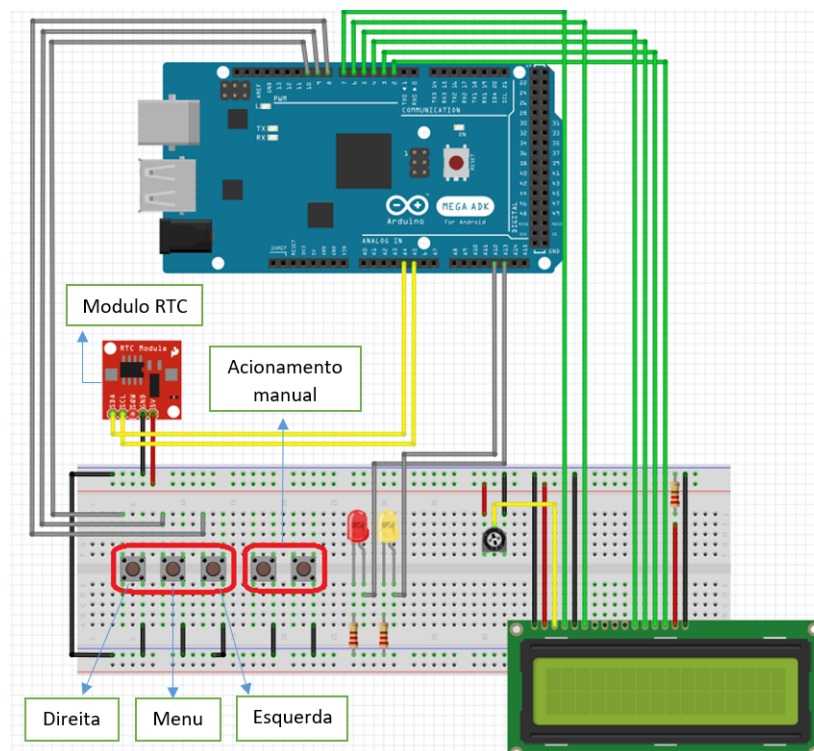


Figura 13: Simulação do projeto Fritzing.

Fonte: próprio autor.

De acordo com a figura 13 anterior foi possível identificar os módulos utilizados no protótipo. As microchaves utilizadas para interação do usuário tem as seguintes finalidades:

a) Temos um conjunto de 3 microchaves para configuração manual do sistema de irrigação, onde a microchave central acessa o menu, e as localizadas a direita e esquerda fazem a navegação para seleção das opções desejadas;

b) Temos um conjunto de duas microchaves responsáveis pelo acionamento manual, onde a localizada a direita desliga e a esquerda liga.

Temos dois leds utilizados para indicar quando a umidade está baixa (led amarelo) e quando a bomba está acionada (led vermelho). Foi utilizado dois resistores, um para cada led, para limitar a corrente, oferecendo uma tensão nominal ideal para os leds.

Com base no desenho foi realizado o protótipo físico para implementação. A ideia do protótipo físico é ver o comportamento do sensor antes de implementar no ambiente onde se deseja automatizar a irrigação.

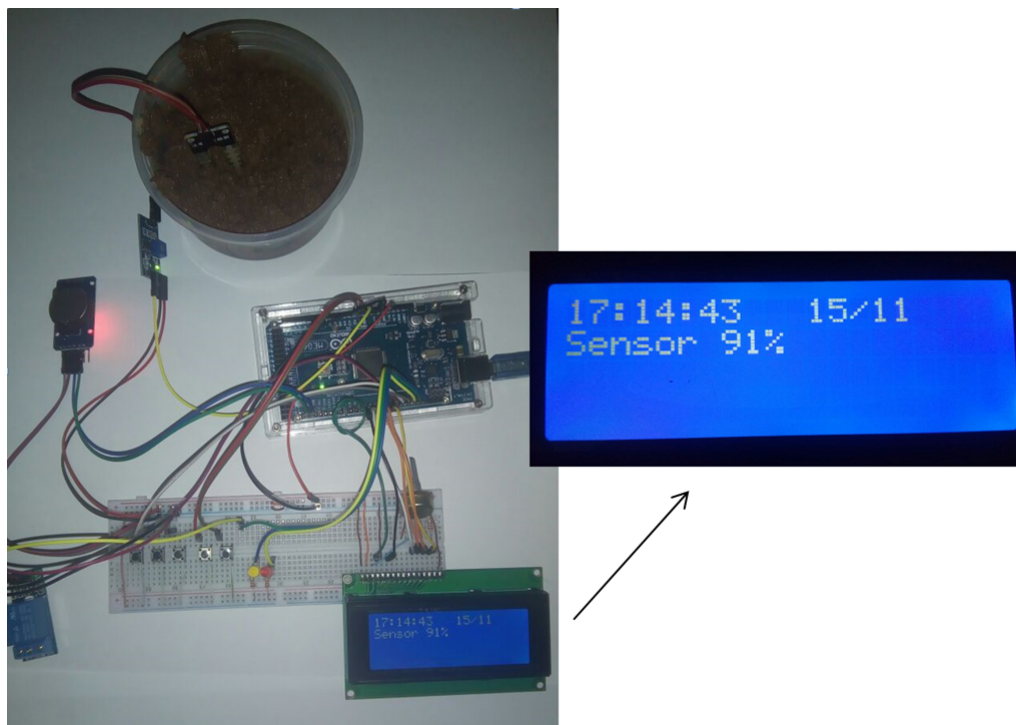


Figura 14: Protótipo do sistema de irrigação.

Fonte: próprio autor.

Através do protótipo foi possível observar o funcionamento do sensor e sobre o acionamento da bomba para irrigação. A utilização do *display* auxilia no controle da irrigação onde é apresentado o estado atual do solo e se a bomba foi acionada.



Figura 15: Funcionamento do *display* do protótipo.

Fonte: próprio autor.

Na figura 15 anterior é apresentada a forma como o *display* apresenta as mensagens referente ao estado atual do solo, de acordo sua umidade e sobre o sistema de irrigação quanto a bomba estar acionada ou não.

Com o protótipo pronto e funcional, foram realizados vários testes para coleta de resultados apresentados na seção a seguir.

3.6 MODO DE OPERAÇÃO DO CONTROLADOR

O sistema dará ao usuário duas opções de operação: a automática e a manual. Essas opções são escolhidas via local. O fluxograma da Figura 16 ilustra o funcionamento do sistema para ambos os modos de operação.

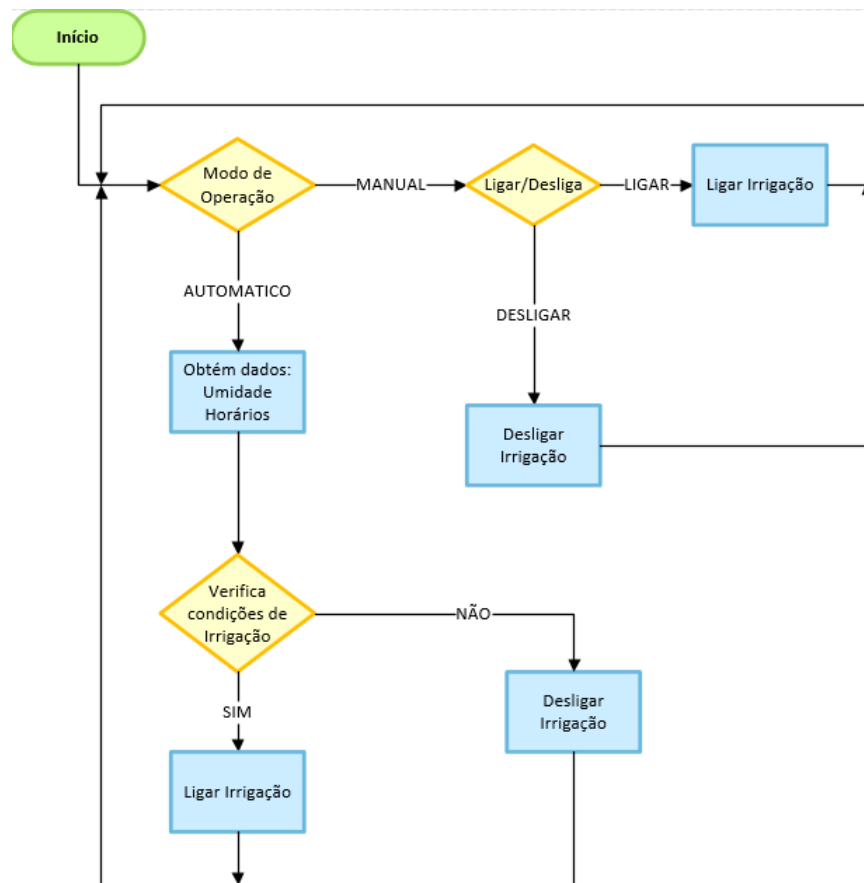


Figura 16: Fluxograma de controle.

Fonte: próprio autor.

Logo no início verifica-se o modo de operação, losango “Modo de Operação”, caso o controlador esteja configurado para funcionar em modo manual o código verifica se o comando para ligar a irrigação, representado pelo losango Liga/Desliga, foi iniciado, no caso afirmativo inicia-se a irrigação retornando na posição inicial, no caso negativo desliga-se a irrigação direcionando o código para a posição inicial. Caso o modo de controle seja o automático, uma leitura nas entradas analógicas será realizada, recebendo assim dados da umidade do sistema, em seguida verifica-se qual é o horário definido para a irrigação. Com todos esses dados atualizados, as condições de irrigação serão analisadas, conforme indicado, iniciando ou não a irrigação e logo em seguida voltando ao início para realizar novamente todos os testes.

4 ANALISE DOS RESULTADOS

Os resultados foram obtidos através de testes realizados com protótipo em diferentes estados, onde foi considerado diferentes climas como dias de sol e dias de chuva. Os testes foram repetidos durante 30 dias coletando inicialmente a terra da plantação em profundidade de 10 cm da superfície de acordo com o exigido para que o sensor colete a real umidade do solo, para a plantação que estava em sua fase inicial, onde foi semeada no início dos testes. A seção a seguir aborda os desafios encontrados para o desenvolvimento do projeto.

4.1 DESAFIOS ENCONTRADOS

Foram encontrados vários desafios referente ao desenvolvimento do projeto, em sua maioria não eram questões técnicas ou de conhecimento teórico, mas relacionado ao próprio desenvolvimento do trabalho que necessitou de conhecimentos além da área de estudo deste trabalho.

Uma das dificuldades encontradas foi em adquirir os componentes eletrônicos necessários para o projeto, e pouca informação sobre a compatibilidade entre eles. Foi necessário a verificação de suas especificações antes de determinar quais seriam usados.

Outro desafio, foi em relação ao código fonte utilizado para programar a plataforma Arduino, onde foi necessário grande pesquisa sobre a linguagem de programação C/C++. Algumas das funções desenvolvidas tiveram correções até apresentar o resultado esperado. Foi necessário um conhecimento de lógica de programação para que o sistema conseguisse atender aos requisitos.

Um desafio encontrado relacionado ao projeto é o conhecimento de técnicas de irrigação e plantio principalmente do milho, onde foi necessário uma pesquisa aprofundada para que o protótipo funcione de forma eficaz, aumentando a produtividade.

4.2 RESULTADOS DA IMPLEMENTAÇÃO

Para análise dos resultados, obtidos foi verificado o solo em dois horários diferentes por diversos dias. O que se observou foi o comportamento do sensor referente ao solo nestes horários e a resposta obtida através da informação recebida.

4.2.1 Dias de clima quente – ensolarados

Para verificar o funcionamento do protótipo em climas quentes, analisado a medição da umidade realizada pelo sensor. De acordo com o estado do solo e o retorno apresentado pelo sensor foi verificado que o mesmo está coerente. Os dados estão sendo interpretados corretamente e o protótipo está sendo executado conforme o programado.

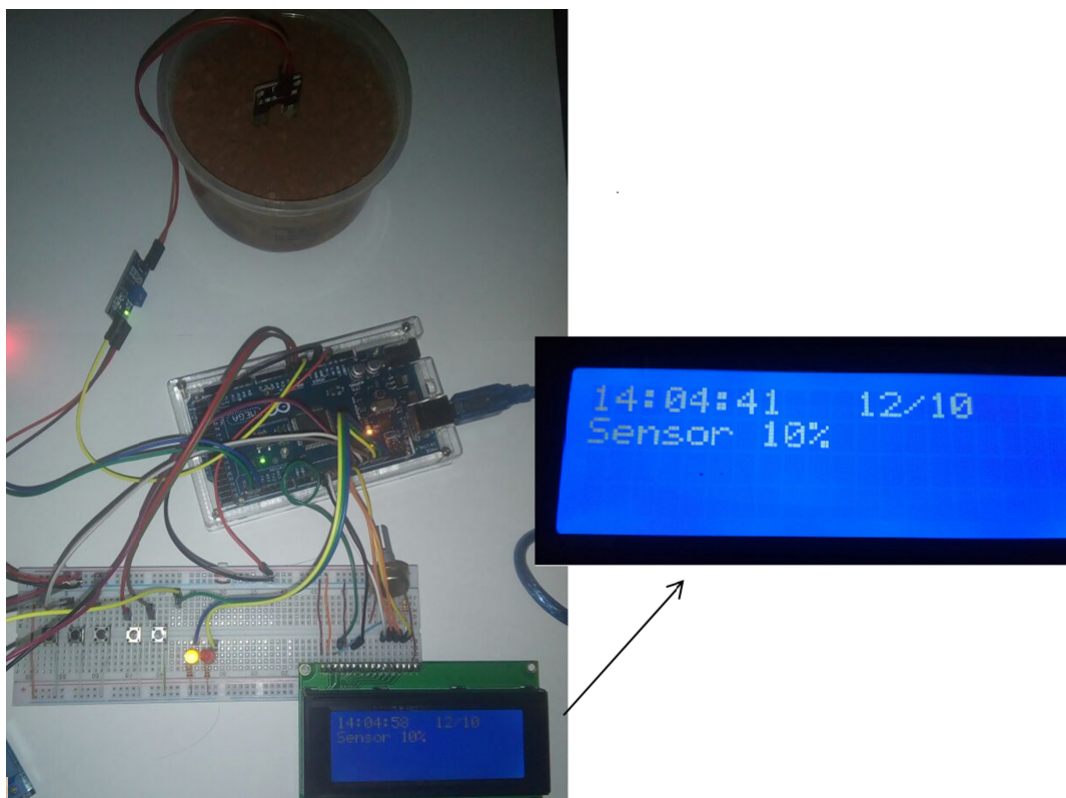


Figura 17: Retorno do protótipo em dias quentes – anterior acionamento da bomba.

Fonte: próprio autor.

O retorno da medição do sensor em dias de estiagem e de temperatura alta, mostrou a baixa umidade do solo. Uma característica da nossa região são estes períodos de secas com temperaturas elevadas. A figura 15 mostra um dia de um período de seca onde a umidade estava abaixo do preciso para uma plantação de milho.

Uma resposta esperada do protótipo é que o acionamento da bomba seja realizado apenas no horário definido, para que não haja perda por evaporação. Foi verificado que este parâmetro está sendo seguido sem falhas pelo protótipo, onde mesmo em dias quentes em que a umidade fica relativamente baixa durante o dia, o acionamento está sendo realizado apenas a noite no horário predefinido.

Foi verificado que o acionamento da bomba está sendo realizado corretamente no horário definido e que a irrigação está obedecendo os parâmetro definidos, onde atingindo o nível de umidade necessário para a cultura, o sistema desliga a bomba.

Os led's indicadores da bomba acionada e da umidade baixa também apresentaram seu funcionamento de acordo com o esperado. Na figura a seguir mostra o funcionamento correto destes led's. Pode-se verificar também, que durante a irrigação os sensores conseguem atualizar o valor da umidade perfeitamente. Um fator importante durante a irrigação é em relação ao tempo de absorção da terra referente a água. Devido a preparação realizada no solo anterior ao período, a absorção da água pelo solo não sofreu uma diferença significativa, não sendo necessário ajustes na programação com relação a isso.

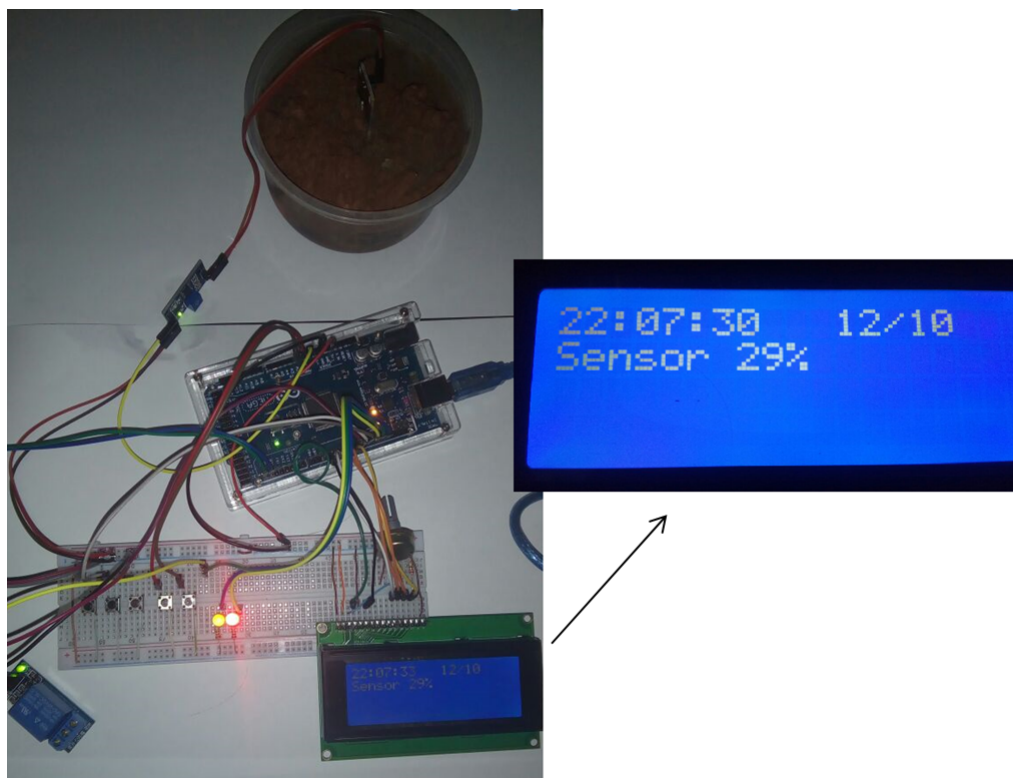


Figura 18: Retorno do protótipo em dias quentes – durante o funcionamento da bomba.

Fonte: próprio autor.

Na figura 16 mostra o protótipo durante período de irrigação que foi acionada as 22h. Durante a irrigação, o valor da umidade aumentando gradativamente de acordo com a absorção de água realizada pelo solo.

4.2.2 Dias de clima úmido – chuva

Em dias chuvosos, há um grande aumento do nível da umidade do solo. Foi possível observar que após períodos de chuva, o sensor retornou que o valor da umidade era alto como esperado. Assim pode-se observar que o comportamento do sensor está agindo como esperado também em dias chuvosos.

Com a umidade alta, ao chegar a hora determinada para início da irrigação, a bomba não foi acionada. O sensor se tornou eficiente em períodos de chuvas que duraram mais de um dia,

pois em dias de estiagem, o solo ainda se apresentou úmido não sendo necessário a irrigação e evitando assim seu encharcamento.

4.2.3 Resultados obtidos

O principal objetivo proposto para este trabalho foi alcançado. Mostrou-se que é viável a implementação e implantação de um sistema automatizado para irrigação de uma plantação de milho, não sendo necessário que o agricultor acione manualmente a irrigação. Porém será necessário que o mesmo observe as condições do sistema e seus retornos frequentemente, para verificar se seu funcionamento permanece eficiente.

A ideia de utilizar a plataforma Arduino como base para desenvolver o projeto, é o desenvolvimento de um sistema com certa complexidade e com baixo custo. Procurou-se também utilizar outros componentes de baixo custo, desta forma, espera-se que a economia de água e energia obtida através do sistema compense o valor do protótipo.

5 CONCLUSÃO

O sistema automatizado de irrigação desenvolvido neste estudo se apresentou eficiente cumprindo assim os objetivos estabelecidos no início do projeto. Utilizando a plataforma Arduino foi possível desenvolver um projeto bem elaborado com acessível.

A opção de alterar a forma de irrigação manualmente através de microchaves, sem a necessidade da alteração do código do sistema, também foi de grande auxílio, assim o agricultor pode realizar a irrigação fora do horário predefinido, em alguma situação que ele jogue necessário.

Com os testes realizados utilizando o protótipo desenvolvido, ficou visível que existe a possibilidade de desenvolver um produto comercializável, utilizando as mesmas tecnologias deste projeto. Com a utilização de mais sensores é possível obter uma média da umidade de toda área de plantio. Desta forma, criando uma central de controle com uma caixa de proteção para que a placa não fique exposta ao tempo, este sistema pode ser implementado em estufas e jardins.

Com relação a plataforma Arduino, mesmo com a dificuldade em desenvolver a programação, é visível que é possível a implementação de qualquer aplicação mesmo sem ter grande conhecimento em computação.

A automatização permite que se faça mais e melhor, além de diminuir a participação do homem em tarefas cansativas e repetitivas, livrando-o de serviços manuais e poupando para tarefas que exigem mais capacidade. Com o sistema automatizado garantimos níveis adequados de umidade do solo propiciando condições de elevar a produtividade. O sistema desenvolvido garante utilização da quantidade de água necessária para uma irrigação ideal evitando o desperdício.

Concluído este trabalho, pode-se notar a importância da engenharia elétrica quando voltada para automatização de processos manuais, tornando a vida de usuários, como no caso deste projeto, do agricultor, mais fácil e cômoda.

5.1 TRABALHOS FUTUROS

Devido este trabalho se tratar de um estudo de caso que apresenta observações realizadas pelo autor durante seu desenvolvimento e implementação, é possível ampliar este estudo através de diferentes pontos de vista de acordo com as opiniões dos envolvidos neste tipo de implementação, como agricultores. Uma possibilidade é a utilização de um aplicativo, que possibilite o monitoramento e o controle do processo através do celular, como o Elipse Mobile.

Uma melhoria possível neste trabalho é o uso de placas para comunicação *wireless* para não haver a necessidade de fios passando por toda plantação até os sensores. Apesar da complexa configuração deste tipo de equipamento, a redução referente ao custo de cabos que

seriam utilizados e a maior mão de obra em sua fixação e proteção, reduziria os custos para este tipo de projeto.

Outro ponto que pode ser estudado é a utilização do sistema de irrigação a longo prazo e se o mesmo permanece respondendo ao testes do início da implementação. A implementação de um sistema de irrigação utilizando tecnologia *wireless*, sem a necessidade de cabos em comparação ao protótipo atual também apresentaria um novo ponto de vista.

Por fim a implementação deste sistema de irrigação em um ambiente diferente do qual ele foi projetado, como um jardim ou uma estufa que tem particularidades diferentes a cultura estudada e analisar os desafios encontrados nesta aplicação.

6 REFERÊNCIAS BIBLIOGRÁFICAS

- Agência Embrapa de Informação Tecnológica. **Milho Verde**. Disponível em: <<http://www.agencia.cnptia.embrapa.br/gestor/milho/arvore/CONT000fy779fnk02wx5ok0pvo4k3c1v9rbg.html>>. Acesso em 02 abr.2017.
- ALBUQUERQUE, P. U. B., THOMAZINI, D. **Sensores Industriais: fundamentos e aplicações**. 1ª Ed. São Paulo: Editora Érica, 2005.
- BANZI, Massimo; SHILOH, Michael. **Primeiro Passos com o Arduino: A plataforma de prototipagem eletrônica open source**. 2. Ed. São Paulo: Novatec, 2015.
- BERLATO, M. A; MOLION, L. B. **Evaporação e evapotranspiração**. Porto Alegre: Instituto de Pesquisas Agronomicas, 1981. (Boletim Técnico 7).
- BERNARDO, S. **Impacto Ambiental da Irrigação no Brasil: recursos hídricos e desenvolvimento sustentável da agricultura**. Viçosa:MMA, SRH, ABEAS, UFV, 1997. 252 p.
- BERNARDO, S., SOARES, A. A.; MANTOVANI, E. C. **Manual de Irrigação**. 8.ed. Viçosa: UFV, 2006, 625p.
- CAVALCANTE, Michelle M. et al. A Plataforma Arduino para fins didáticos: Estudo de caso com recolhimento de dados a partir do PLX-DAQ. In: **XXXIV Congresso da Sociedade Brasileira de Computação – CSBC 2014**, Brasília. Brasília, Universidade de Brasília, 2014.
- CRUZ, Jose Carlos et al. **Cultivo do milho**. Sete Lagoas: Embrapa Milho e Sorgo, 2017. (Embrapa Milho e Sorgo. Sistema de Produção,1). Disponível em: . Acesso em: 22 ago. 2017.
- DOORENBOS, J.; PRUITT, W. O. **Guidelines for predicting crop water requirements**. 2. ed. Rome: FAO, 1977. (FAO. Irrigation and Drainage Paper).
- EVANS, Martin et al. **Arduino em ação**. 1. Ed. São Paulo: Novatec, 2013.
- KARVINEN, Kimmo; KARVINEN, Tero. **Primeiro Passos com Sensores: Perceba o mundo usando eletrônica, Arduinp e Raspberry Pi**. 1. Ed. São Paulo: Novatec, 2014.
- LIMA, Charles Borges de; VILLAÇA, Marco V. M. **AVR e Arduino; técnicas de projeto**. 2. Ed. Florianópolis: 2012.
- MARSCZAOKOSKI, Fábio G.; CRUZ, Raphaela P. da; SILVA, Wesley de A. E. **Sistema Microcontrolado de Irrigação Aplicado a Morangueiros: um estudo de caso**. Curitiba: Universidade Tecnológica Federal do Paraná, 2013. 119 P. (Trabalho de Conclusão de Curso).
- MCROBERTS, Michael. **Arduino Básico**. 2. Ed. São Paulo: Novatec, 2015.
- NATALE, Ferdinando. **Automação Industrial**. Editora Érica. São Paulo, 2002
- NEVES JUNIOR, Giovanni. **Principais métodos de determinação da umidade do solo**. Dispo-

nível em: <<https://giovanijr.wordpress.com/agricultura-irrigada/principais-metodos-de-determinacao-da-umidade-do-solo/>>. Acesso em: 21 AGO. 2017.

NUSSEY, John. **Arduino for Dummies**. 1. Ed. 2013

PEREIRA, João Batista Alves. **Manual prático de irrigação**. Niterói, 2014.

PEREIRA FILHO, Israel Alexandre et al. **Milho Verde**. Disponível em: <<http://www.agencia.cnptia.embrapa.br/gestor/milho/arvore/CONT000fy779fnk02wx5ok0pvo4k3c1v9rbg.html>>. Acesso em: 02 ABR. 2017.

ROTONDANO, A. K. F.; MELO, B. **HISTÓRICO DA IRRIGAÇÃO**. Disponível em: <<http://www.fruticultura.iciag.ufu.br/irriga2.html>>. Acesso em 02 abr.2017.

SENAI. **Fundamentos de automação**. São Paulo: SENAI-SP editora, 2015.

SOUZA, Rodrigo Barbosa de; **Uma Arquitetura para Sistemas Supervisórios Industriais e sua Aplicação em Processos de Elevação Artificial de Petróleo** Dissertação de Mestrado, Universidade Federal do Rio Grande do Norte – UFRN, 2005.

TESTEZLAF, Roberto. **Irrigação: métodos, sistemas e aplicações**. Faculdade de Engenharia Agrícola UNICAMP. Campinas: 2017.

WILCHER, Don. **Learn Electronics with Arduino**. 1. Ed. 2012.

YONEYA, F. **Irigar à noite é mais econômico**. Disponível em: <<http://www.saopaulo.sp.gov.br/spnoticias/na-imprensa/irrigar-a-noite-e-mais-economico/>>. Acesso em 04 abr.2017.

APÊNDICE A CÓDIGO FONTE

Código fonte elaborado no ambiente de desenvolvimento (software) do Arduino.

```

/*AUTOMATIZACAO DO SISTEMA DE IRRIGACAO DE MILHO*/

//INCLUSAO DAS BIBLIOTECAS NECESSARIAS PARA A EXECUCAO DO CODIGO.

#include <string.h>
#include <Wire.h>
#include <DS1307.h>
#include <avr/wdt.h>
#include <EEPROM.h>
#include <LiquidCrystal.h>

char dateTime[20];
int RTCValues[7];
char acertahora[8];
int programas[21];
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

//(VERDADEIRO OU FALSO.
#define analogS A0
#define relay1 12
#define led2 13
#define relay2 11
#define acionar 15
#define desligar 16
#define up 8
#define sete 9
#define down 10

//DECLARANDO VARIAVEIS.
int i;
int p;
byte z;
byte h;
byte db;
boolean m;
boolean o;
int d;
int d1;
byte menu;
byte program;
int humidade;
int reading;
byte estouro;
int deslbomba;
int deslbomba2;

int estadorelay1 = 1;
int estadorelay2 = 1;

```

```
int leitura1 = 0;
int leitura2 = 0;

// VARIÁVEIS DE INTERRUPTO.
int C11;
int C12;

// INICIALIZAR COMUNICAÇÃO SÉRIE 9600 BPS.
void setup() {

    Serial.begin(9600);
    DS1307.begin();
    lcd.begin(20, 4);

    //DEFINE PINOS DOS BOTÕES COMO ENTRADA.
    pinMode(acionar, INPUT);
    pinMode(desligar, INPUT);
    pinMode(relay1, OUTPUT);
    pinMode(relay2, OUTPUT);
    pinMode(led2, OUTPUT);
    pinMode(up, INPUT);
    digitalWrite(up, HIGH);
    pinMode(down, INPUT);
    digitalWrite(down, HIGH);
    pinMode(sete, INPUT);
    digitalWrite(sete, HIGH);
    digitalWrite(relay1, HIGH);
    humidade = EEPROM.read(50);
    o = EEPROM.read(51);
    for (int x = 0; x < 10; x++) {
        programas[x] = EEPROM.read(x);
    }
    *((char*)&deslbomba) = EEPROM.read(52);
    *((char*)&deslbomba + 1) = EEPROM.read(53);
}

// COMANDO PARA ACERTAR A DATA E A HORA.
void acertartc()
{
    wdt_disable();
    delay(1000);

    while (i > 0 && i < 8)
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        if (i == 1) {
            lcd.print(" Entre com Ano ");
        }
        if (i == 2) {
            lcd.print(" Entre com Mes ");
        }
        if (i == 3) {
            lcd.print(" Entre com Dia ");
        }
        if (i == 4) {
```

```
    lcd.print(" Entre cm Semana");
}
if (i == 5) {
    lcd.print(" Entre com Hora ");
}
if (i == 6) {
    lcd.print(" Entre com Minut");
}
if (i == 7) {
    lcd.print(" Entre com Segun");
}
if (digitalRead(up) == LOW) {

    acertahora[i]++;
    estouro = 0;
}
if (digitalRead(down) == LOW && acertahora[i] >= 0) {

    acertahora[i]--;
    estouro = 0;
}
if (digitalRead(sete) == LOW) {
    i ++;
    estouro = 0;
}
if (acertahora[1] < 1 ) {
    acertahora[1] = 13;
}
if (acertahora[2] > 12) {
    acertahora[2] = 1;
}
if (acertahora[2] == 0) {
    acertahora[2] = 12;
}
if (acertahora[3] > 31) {
    acertahora[3] = 1;
}
if (acertahora[3] == 0) {
    acertahora[3] = 31;
}
if (acertahora[4] > 7) {
    acertahora[4] = 1;
}
if (acertahora[4] == 0) {
    acertahora[4] = 7;
}
if (acertahora[5] > 23) {
    acertahora[5] = 0;
}
if (acertahora[5] == -1) {
    acertahora[5] = 23;
}
if (acertahora[6] > 59) {
    acertahora[6] = 0;
}
if (acertahora[6] == -1) {
    acertahora[6] = 59;
```

```
    }
    if (acertahora[7] > 59) {
        acertahora[7] = 0;
    }
    if (acertahora[7] == -1) {
        acertahora[7] = 59;
    }
    lcd.setCursor(9, 1);
    lcd.print(acertahora[i], DEC);
    if (estouro == 20) {
        i = 0 ;
        estouro = 0;
    }
    delay(200);
}

DS1307.setDate(acertahora[1], acertahora[2], acertahora[3],
acertahora[4], acertahora[5], acertahora[6], acertahora[7]);
// AQUI ENVIAMOS OS DADOS DEFINIDOS PARA O REAL TIME CLOK

    lcd.clear();

}
// COMANDO PARA ACESAR OPCOES DO MENU.
void menus()
{
    wdt_disable();
    delay(1000);
    lcd.clear();
    while (menu > 0)
    {
        lcd.setCursor(0, 0);
        if (menu == 1) {
            lcd.print(" Definir Horarios ");
            if (digitalRead(sete) == LOW) {
                p = 1;
                menu = 0;
                programa();
            }
        }
        if (menu == 2) {
            lcd.print(" Definir Umidade ");
            if (digitalRead(sete) == LOW) {
                h = 1;
                menu = 0;
                humidade();
            }
        }
        if (menu == 3) {
            lcd.print(" Zerar programas ");
            if (digitalRead(sete) == LOW) {
                z = 1;
                menu = 0;
                zera();
            }
        }
        if (menu == 4) {
```

```

    lcd.print("Modo de Operacao");
    if (digitalRead(sete) == LOW) {
        m = 1;
        menu = 0;
        operacao();
    }
}
if (menu == 5) {
    lcd.print("Acerta Data e a Hora");
    if (digitalRead(sete) == LOW) {
        i = 1;
        menu = 0;
        acertartc();
    }
}
if (menu == 6) {
    lcd.print("Delay de acionamento ");
    if (digitalRead(sete) == LOW) {
        db = 1;
        menu = 0;
        offbomba();
    }
}
if (digitalRead(up) == LOW && menu < 6) {
    menu++;
    estouro = 0;
}
if (digitalRead(down) == LOW && menu > 0) {
    menu--;
    estouro = 0;
}
if (estouro == 20) {
    menu = 0 ;
    estouro = 0;
}
delay(400);
}
lcd.clear();
}

```

//COMANDO QUE RECEBE O HORARIO DE ACIONAMENTO.

```

void programa()
{
    wdt_disable();
    delay(1000);
    lcd.clear();
    while ( p > 0 && p < 7)
    {
        lcd.clear();
        lcd.setCursor(0, 0);
        if (p == 1 || p == 2) {
            lcd.print(" Programa 1 ");
            lcd.setCursor(2, 1);
            lcd.print(programas[1], DEC);
            lcd.print(':');
            lcd.print(programas[2]);
            lcd.print(" horas");
        }
    }
}

```

```
}
if (p == 3 || p == 4) {
    lcd.print("    Programa 2    ");
    lcd.setCursor(2, 1);
    lcd.print(programas[3], DEC);
    lcd.print(':');
    lcd.print(programas[4]);
    lcd.print(" horas");
}
if (p == 5 || p == 6) {
    lcd.print("    Programa 3    ");
    lcd.setCursor(2, 1);
    lcd.print(programas[5], DEC);
    lcd.print(':');
    lcd.print(programas[6]);
    lcd.print(" horas");
}
if (digitalRead(up) == LOW ) {
    programas[p]++;
    estouro = 0;
}
if (digitalRead(down) == LOW && programas[p] >= 0 ) {
    programas[p]--;
    estouro = 0;
}
if (digitalRead(sete) == LOW) {
    p ++;
    estouro = 0;
}
if (estouro == 30) {
    p = 0 ;
    estouro = 0;
}
if (programas[1] > 23) {
    programas[1] = 0;
}
if (programas[1] < 0) {
    programas[1] = 23;
}
if (programas[3] > 23) {
    programas[3] = 0;
}
if (programas[3] < 0) {
    programas[3] = 23;
}
if (programas[5] > 23) {
    programas[5] = 0;
}
if (programas[5] < 0) {
    programas[5] = 23;
}
if (programas[2] > 59) {
    programas[2] = 0;
}
if (programas[2] < 0) {
    programas[2] = 59;
}
```

```
    if (programas[4] > 59) {
        programas[4] = 0;
    }
    if (programas[4] < 0) {
        programas[4] = 59;
    }
    if (programas[6] > 59) {
        programas[6] = 0;
    }
    if (programas[6] < 0) {
        programas[6] = 59;
    }

    delay(400);
}
for (int x = 0; x < 10; x++)
{
    EEPROM.write(x, programas[x]);
}
for (int x = 0; x < 16; x++) {
    delay(90);
    lcd.setCursor(x, 1);
    lcd.write(B11111111);
}
lcd.clear();
p = 0;
}
// COMANDO PARA ANALISAR A UMIDADE.
void humidade()
{
    wdt_disable();
    delay(1000);
    lcd.clear();
    while (h > 0)
    {
        lcd.setCursor(0, 3); lcd.print(" Definir Umidade ");
        if (digitalRead(up) == LOW && humidade < 100 ) {
            humidade ++;
            estouro = 0;
        }
        if (digitalRead(down) == LOW && humidade > 0) {
            humidade --;
            estouro = 0;
        }
        if (digitalRead(sete) == LOW) {
            h = 0;
            estouro = 0;
        }
        lcd.setCursor(5, 1); lcd.print(humidade); lcd.print("%    ");
        if (estouro == 20) {
            h = 0 ;
            estouro = 0;
        }
        delay(150);
    }
    EEPROM.write(50, humidade);
    for (int x = 0; x < 16; x++) {
```

```
    delay(90);
    lcd.setCursor(x, 1);
    lcd.write(B11111111);
}
lcd.clear();
}

// COMANDO PARA O DELAY DE ACIONAMENTO.
void offbomba()
{
    {
        wdt_disable();
        delay(1000);
        lcd.clear();
        while (db > 0)
        {
            lcd.setCursor(0, 0); lcd.print("Delay de acionamento");
            if (digitalRead(up) == LOW) {
                deslbomba ++;
                estouro = 0;
            }
            if (digitalRead(down) == LOW && deslbomba > 0) {
                deslbomba --;
                estouro = 0;
            }
            if (digitalRead(sete) == LOW) {
                db = 0;
                estouro = 0;
            }
            lcd.setCursor(5, 1); lcd.print(deslbomba); lcd.print(" Segundos");
            delay(150);
            if (estouro == 30) {
                db = 0 ;
                estouro = 0;
            }
        }
        EEPROM.write(52, *((char*)&deslbomba ));
        EEPROM.write(53, *((char*)&deslbomba + 1));
    }
    for (int x = 0; x < 16; x++) {
        delay(90);
        lcd.setCursor(x, 1);
        lcd.write(B11111111);
    }
    lcd.clear();
}

// COMANDO QUE ZERA AS PROGRAMACOES.
void zera()
{
    lcd.setCursor(4, 0);
    lcd.clear();
    lcd.print("Zerando");

    for (int x = 0; x < 10; x++)
    {
        EEPROM.write(x, 0);
    }
}
```

```
}
for (int x = 0; x < 10; x++) {
    programas[x] = EEPROM.read(x);
}
for (int x = 0; x < 16; x++) {
    delay(90);
    lcd.setCursor(x, 1);
    lcd.write(B11111111);
}
lcd.clear();
}

// COMANDO QUE DEFINE O MODO DE OPERACAO.
void operacao()
{
    wdt_disable();
    delay(500);
    lcd.clear();
    while (m > 0)
    {
        lcd.setCursor(0, 0); lcd.print("Modo de Controle");
        lcd.setCursor(0, 1); lcd.print("<- Umidade Tempo ->");
        if (digitalRead(up) == LOW) {
            o = 0;
            digitalWrite(relay2, LOW);
            m = 0 ;
        }
        if (digitalRead(down) == LOW) {
            o = 1;
            m = 0 ;
        }
        if (estouro == 30) {
            m = 0 ;
            estouro = 0;
        }
    }
    EEPROM.write(51, o);
    for (int x = 0; x < 16; x++) {
        delay(90);
        lcd.setCursor(x, 1);
        lcd.write(B11111111);
    }
    lcd.clear();
}

// COMANDO QUE MOSTRA DATA E A HORA NO DISPLAY.
void mostrahora()
{
    DS1307.getDate(RTCValues);
    sprintf(dateTime, "%02d:%02d:%02d    %02d/%02d", RTCValues[4],
        RTCValues[5], RTCValues[6], RTCValues[2], RTCValues[1]);
    delay(50);
    lcd.setCursor(0, 0); lcd.print(dateTime);
    delay(50);
    C11 = RTCValues[6];
}
```

```
void teste()
{
    if (programas[1] == RTCValues[4] && programas[2] == RTCValues[5]
        && RTCValues[6] == 0 && reading < humedad) {
        digitalWrite(relay2, HIGH);
    }
    if (programas[3] == RTCValues[4] && programas[4] == RTCValues[5]
        && RTCValues[6] == 0 && reading < humedad) {
        digitalWrite(relay2, HIGH);
    }
    if (programas[5] == RTCValues[4] && programas[6] == RTCValues[5]
        && RTCValues[6] == 0 && reading < humedad) {
        digitalWrite(relay2, HIGH);
    }
    if (o == 1 && reading < humedad) {
        digitalWrite(relay2, HIGH); // Caso queira que a bomba nao desligue imediatamente
        coloque aqui seu delay.
    } else {
        if (o == 1 && (deslbomba2 / 2) >= deslbomba) {
            digitalWrite(relay2, LOW);
        }
    }
}

if (RTCValues[4] == 0 && RTCValues[5] == 0 && RTCValues[6] == 0) {
    digitalWrite(relay2, LOW);
}

}

// COMANDO QUE INFORMA O ESTADO DO PROGRAMA.
void informa()
{
    Serial.println(d1);
    if (d1 <= 10) {
        lcd.setCursor(0, 1); lcd.print("Set point. ");
        lcd.print(humedad); lcd.print("%");
    }
    if (d1 > 50 && d1 <= 100) {
        lcd.setCursor(0, 1); lcd.print("Sensor ");
        lcd.print(reading); lcd.print("%");
    }
    if (d1 > 150 && d1 <= 200) {
        lcd.setCursor(0, 1); if (digitalRead(relay1) == LOW)
        {
            lcd.print("> Bomba ligada <");
        }
        else {
            lcd.print("Bomba Desligada ");
        }
    }
    if (d1 > 250 && d1 <= 300) {
        lcd.setCursor(0, 1);
        if (reading < humedad)
        {
            lcd.print("Umidade Baixa!!!");
        } else {
```

```
        lcd.print(" Umidade Normal ");
    }
}
if (d1 > 350 && d1 <= 370) {
    lcd.setCursor(0, 1);
    lcd.print(" Irrigador Automatico");
}
if (d1 >= 400) {
    d1 = 0;
}
}

void Nivel()
{
    int nivelMax = analogRead(1);
    int nivelMini = analogRead(2);
    if (nivelMax < 10) {
        digitalWrite(relay1 , LOW);
    }
    if (nivelMini < 10 && nivelMax > 100) {
        digitalWrite(relay1 , HIGH);
    }
}

void loop()
{
    //VERIFICA O ACIONAMENTO DO BOTAO ACIONAR.
    leitura1 = digitalRead(acionar);
    if (leitura1 != 0)
    {
        while(digitalRead(acionar) != 0)
        {
            delay(100);
        }
        //INVERTE O ESTADO DA PORTA.
        estadorelay1 = !estadorelay1;
        //Comandos para o rele 1
        digitalWrite(relay1 , relay1);
    }

    //VERIFICA O ACIONAMENTO DO BOTAO DESLIGAR.
    leitura2 = digitalRead(desligar);
    if (leitura2 != 0)
    {
        while(digitalRead(desligar) != 0)
        {
            delay(100);
        }
        //INVERTE O ESTADO DA PORTA.
        estadorelay2 = !estadorelay2;
        //Comandos para o rele 2
        digitalWrite(desligar , relay2);
    }

    //EXECUTA AS OPCOES SELECIONADAS.
    if(C11 != C12){d1++; C11 = C12; interrup();}
```

```
if (digitalRead(sete) == LOW) {
    menu = 1;
    menus();
}
reading = map(analogRead(analogS), 0, 1023, 190, 0); //REGULAR O SENSOR DE UMIDADE
Serial.print("C11 "); Serial.println(C11);
Serial.print("C12 "); Serial.println(C12);
mostrahora();
informa();
teste();
Nivel();

if (digitalRead(relay1) == LOW && deslbomba2 >= deslbomba) {
    digitalWrite(relay1, HIGH);
    deslbomba2 = 0;
}
if (reading < humedad) {
    digitalWrite(led2, HIGH);
    digitalWrite(relay1, LOW);
} else {
    digitalWrite(led2, LOW);
    digitalWrite(relay1, HIGH);
}
}
```

ANEXO A FASES DO MILHO

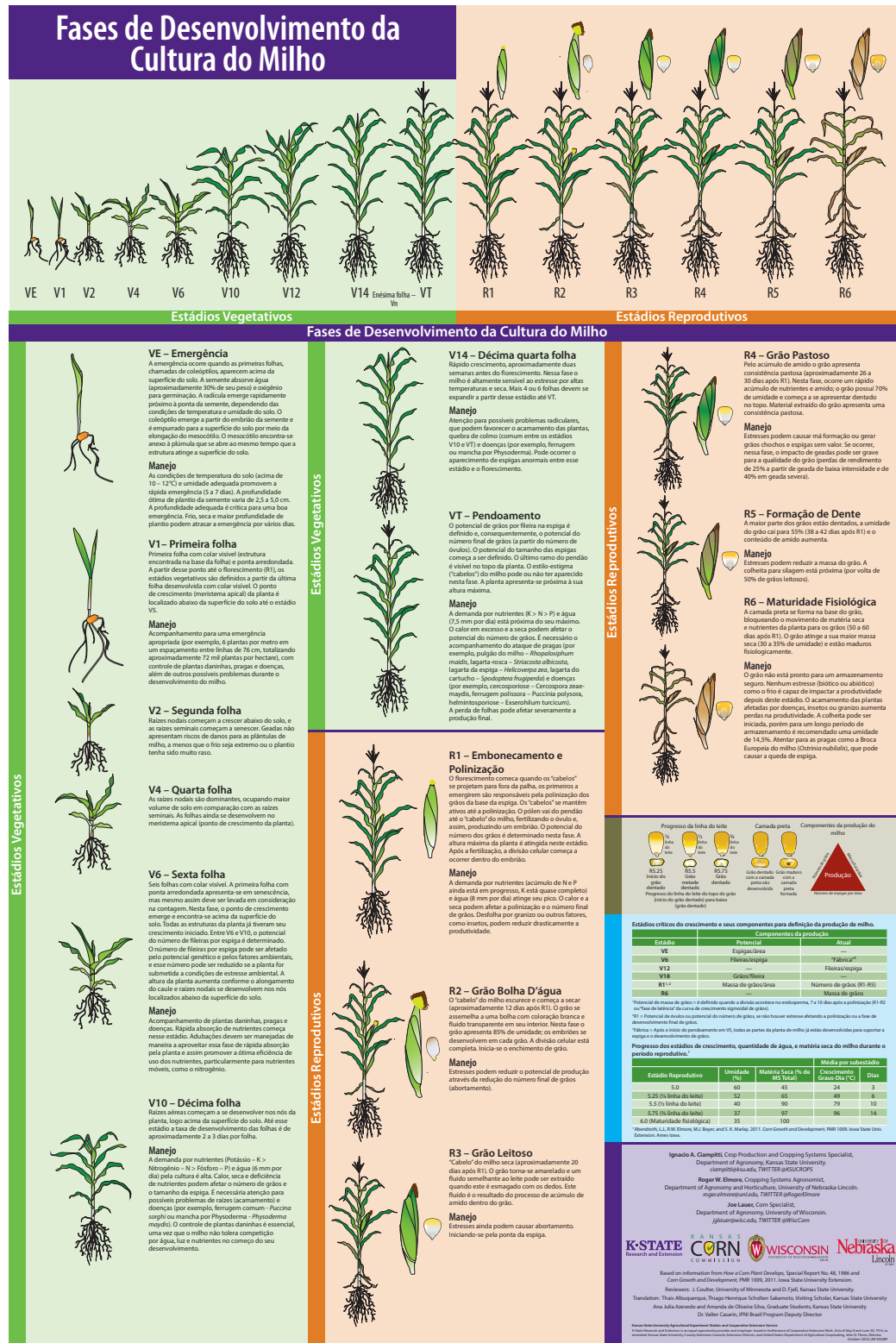


Figura 19: Fases de Desenvolvimento da Cultura do Milho .

Fonte: <http://brasil.ipni.net>