

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE
DO NORTE

ANDRESA RODRIGUES MARTINS

GABRIELA NERES DE OLIVEIRA E SILVA

**SISTEMA DE IRRIGAÇÃO AUTOMÁTICO PARA PLANTAS DE PEQUENO
PORTE**

JOÃO CÂMARA/RN

2017

ANDRESA RODRIGUES MARTINS

GABRIELA NERES DE OLIVEIRA E SILVA

**SISTEMA DE IRRIGAÇÃO AUTOMÁTICO PARA PLANTAS DE PEQUENO
PORTE**

Trabalho de Conclusão de Curso apresentado à coordenação do Curso Técnico Integrado em Informática do Instituto Federal de educação, ciência e tecnologia do Rio Grande do Norte – IFRN, em cumprimento às exigências legais como requisito parcial à obtenção do título de Técnico em Informática.

Orientador: Prof. Me. João Paulo Ferreira
Guimarães

JOÃO CÂMARA/RN

2017

ANDRESA RODRIGUES MARTINS
GABRIELA NERES DE OLIVEIRA E SILVA

**SISTEMA DE IRRIGAÇÃO AUTOMÁTICO PARA PLANTAS DE PEQUENO
PORTE**

Trabalho de Conclusão de Curso submetido à
Coordenação do Curso Técnico Integrado em
Informática, do Instituto Federal de Educação,
Ciência e Tecnologia do Rio Grande do Norte -
IFRN, em cumprimento às exigências legais
como requisito parcial à obtenção do título de
Técnico em Informática.

Trabalho de Conclusão de Curso apresentado e aprovado em ____/____/2017 pela seguinte banca
examinadora:

BANCA EXAMINADORA

Prof. João Paulo Ferreira Guimarães – Presidente

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – IFRN

Prof. Igor Augusto de Carvalho Alves – Examinador

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – IFRN

Prof. Heitor Bruno de Araújo Souza – Examinador

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – IFRN

JOÃO CÂMARA/RN

2017

Dedicamos este trabalho para nossa família e a todos que nos apoiaram, auxiliaram e acreditaram em nós em todos os momentos. Não somente a estes, mas também a todos que contribuíram de alguma forma em nossa formação acadêmica.

AGRADECIMENTOS

Agradecemos de imediato ao Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, que nos proporcionou experiências novas e proveitosas. Além disso, nos deu a possibilidade de conhecer profissionais que contribuíram imensamente em nossa formação.

A Deus, que nos concedeu a vida, a força e a coragem para seguirmos em nossa jornada.

Às nossas famílias, que nos apoiaram em nossas escolhas e etapas de nossas vidas. Sem eles não saberíamos a importância da educação e de seguir em frente em qualquer dificuldade. Obrigada pelo amor e paciência nas situações pelas quais passamos.

Aos nossos colegas e amigos da turma de Informática 2013.1, pelo companheirismo, amizade e momentos de descontração. Eles que nos alegraram muitas vezes em ocasiões em que nada parecia que ia funcionar. Obrigada por fazerem parte de nossa jornada e nossas vidas.

Ao nosso orientador Prof. Me. João Paulo Ferreira Guimarães, pela disponibilidade, confiança, paciência, dedicação e o auxílio em todas as vezes que precisamos. Obrigada por investir seu tempo e pela orientação e participação na elaboração deste trabalho.

A todos os professores, que nos acompanharam e têm nos acompanhado, garantindo nossa formação da melhor forma possível e nos incentivando a seguir nossos sonhos.

A todos, que de alguma maneira colaboraram para a realização desse trabalho.

RESUMO

Com a importância da agricultura e a busca pelo cultivo próprio de alimentos, torna-se necessário o controle da água utilizada na irrigação de plantações. Este controle permite o melhor aproveitamento da água, evitando desperdícios. Sendo assim, o projeto proposto é a elaboração de um sistema automático de irrigação construído a partir da plataforma Arduino. Além disso, o sistema também conta com o registro de informações que venham a ser relevantes para o usuário. Seguindo esta ideia, o protótipo constitui-se principalmente de sensores. As informações obtidas pelo sensor de umidade do solo possibilitam a ativação da bomba d'água, responsável pela irrigação. Este e outros sensores também fornecem informações que são registradas em um arquivo de texto e exibidas no display LCD. Com isso, utilizou-se o minicomputador Raspberry Pi para realizar esse registro por meio de um código Python. Então, com base nas pesquisas realizadas sobre os fatores importantes para as plantas e também sobre os componentes eletrônicos utilizados, construiu-se um protótipo que atendeu aos objetivos. Posteriormente, observou-se o funcionamento do protótipo em um ambiente de teste. Com isso, concluiu-se que o controle da umidade no sistema permitiu menos desperdício de água. Portanto, a proposta de irrigação automática e a função de monitoramento e registro das características do ambiente realizou-se com sucesso.

Palavras-chave: Arduino; Sensores; Irrigação; Raspberry Pi; Controle.

ABSTRACT

With the importance of agriculture and the demand for growing your own crops, it's necessary to control the water used in the irrigation. This control allows a better use of water, avoiding waste. Therefore, the proposed project is the elaboration of an automatic irrigation system built from the Arduino platform. Furthermore, the system also uses the registering of information that may be relevant to the user. Following this idea, the prototype consists mainly of sensors. The information obtained by the soil moisture sensor allows the activation of the water pump, responsible for irrigation. This and other sensors also provide information that is recorded in a text file and displayed on the LCD. Thereby, the Raspberry Pi was used to perform this entry using a Python code. So, based on the research carried out on the important factors for the plants and on the electronic components used, a prototype was built and met the objectives. Subsequently, the operation of the prototype was observed in a test environment. Thereby, it was concluded that the control of the humidity in the system allowed less water waste. Thus, the proposal of an automatic irrigation and the function of monitoring and recording the characteristics of the environment was successful.

Keywords: Arduino; Sensors; Irrigation; Raspberry Pi; Control.

LISTA DE TABELAS

Tabela 1: Descrição dos pinos do módulo sensor de umidade relativa e temperatura	26
Tabela 2: Relação de tensão obtida com a umidade relativa	28
Tabela 3: Relação de tensão obtida com a temperatura	29
Tabela 4: Descrição dos pinos do módulo display LCD	30
Tabela 5: Funções dos elementos do projeto	54
Tabela 6: Orçamento do projeto	58

LISTA DE ILUSTRAÇÕES

Figura 1: Irrigação localizada por gotejamento	16
Figura 2: Irrigação por aspersão	17
Figura 3: Arduino BlackBoard	20
Figura 4: Exemplos das conexões do Arduino UNO	22
Figura 5: IDE do Arduino e suas funcionalidades	24
Figura 6: Sensor de umidade do solo	25
Figura 7: Pinos do sensor de umidade relativa e temperatura	26
Figura 8: Módulo sensor de umidade relativa do ar e temperatura	28
Figura 9: Módulo display LCD	31
Figura 10: Funcionamento do relé	32
Figura 11: Módulo relé	33
Figura 12: Raspberry Pi com case	34
Figura 13: Representação do funcionamento da bomba de aquário	37
Figura 14: Bomba de aquário	37
Figura 15: Integração entre os componentes utilizados	38
Figura 16: Fluxograma do sistema de irrigação	41
Figura 17: Conexão do sensor de umidade do solo com o Arduino	42
Figura 18: Exemplificação das conexões entre Arduino e sensor de umidade do solo	42
Figura 19: Conexão do sensor de umidade relativa e temperatura com o Arduino	43
Figura 20: Exemplificação das conexões entre Arduino e sensor de umidade relativa e temperatura	44
Figura 21: Conexão do display LCD com o Arduino	45
Figura 22: Exemplificação das conexões entre Arduino e Display LCD	45
Figura 23: Conexão do módulo relé com o Arduino	46
Figura 24: Exemplificação das conexões entre Arduino e Display LCD	47
Figura 25: Conexão do Raspberry Pi com o Arduino	48
Figura 26: Fluxograma exemplificando parte da utilização dos dados do sensor de umidade do solo	49
Figura 27: Pseudocódigo exemplificando parte da utilização dos dados do sensor de umidade do solo	50
Figura 28: Fluxograma exemplificando os processos principais do código Python	52

Figura 29: Pseudocódigo exemplificando os processos principais do código Python	52
Figura 30: Protótipo montado em ambiente de teste	55
Figura 31: Sensor de umidade do solo em teste	56
Figura 32: Display LCD rolando caracteres informativos	57
Figura 33: Arquivo log.dat com as informações coletadas	58

SUMÁRIO

1. INTRODUÇÃO	12
1.1. JUSTIFICATIVA	12
1.2. OBJETIVOS	12
1.2.1. Objetivo geral	12
1.2.2. Objetivos específicos	13
1.3. ESTRUTURA DO TRABALHO	13
2. REFERENCIAL TEÓRICO	15
2.1. AGRICULTURA	15
2.2. ÁGUA	15
2.3. TEMPERATURA	17
2.4. UMIDADE RELATIVA DO AR	18
2.5. DESCRIÇÃO DOS COMPONENTES UTILIZADOS	19
2.5.1. Arduino	19
2.5.1.1. Microcontroladores	20
2.5.1.2. Arduino UNO	21
2.5.1.3. IDE do Arduino	23
2.5.2. Sensor de umidade do solo	24
2.5.3. Módulo sensor de umidade do ar e de temperatura	25
2.5.4. Display LCD	29
2.5.5. Módulo relé	31
2.5.6. Raspberry Pi	33
2.5.6.1. Sistema GNU/Linux	35
2.5.6.2. Python	35
2.5.7. Bomba SP-500	36
3. MATERIAIS E MÉTODOS	38
3.1. DESENVOLVIMENTO DO PROTÓTIPO	40
3.1.1. Apresentação do sistema de irrigação proposto	40
3.1.2. Sistema de irrigação	40
3.1.2.1. Montagem do sistema de irrigação	41
3.1.2.2. Desenvolvimento do software	48
3.1.2.2.1. Código do Arduino	48

3.1.2.2.2. <i>Código Python</i>	51
4. RESULTADOS OBTIDOS	54
5. CONSIDERAÇÕES FINAIS	59
5.1. TRABALHOS FUTUROS	60
REFERÊNCIAS	61
APÊNDICE A – Código do Arduino	66
APÊNDICE B – Código Python	70

1. INTRODUÇÃO

1.1. JUSTIFICATIVA

O jornal O Tempo (2014) ressalta que, o cultivo de plantas de pequeno porte, mais especificamente hortas caseiras, nos ambientes residenciais tem ganhado adeptos a cada dia. Porém, a preservação de hortas caseiras se torna difícil devido a rotina cada vez mais acelerada. Além disso, a maioria das plantas hortaliças que se compra em mercados brasileiros tem agrotóxicos ou é modificada geneticamente, conforme afirma o jornal El País (2015). Segundo o jornal ZH (2015), os alimentos orgânicos ainda são mais caros, já que são produzidos em menores quantidades. Esses alimentos orgânicos são mais saudáveis, pois são cultivados sem uso de substâncias químicas.

Com a busca pela aquisição de hortas caseiras, é importante o controle das necessidades das plantas a fim do melhor desenvolvimento delas. Com base nisso, a irrigação é uma das principais ações que devem ser monitoradas, principalmente se efetuada em excesso. Então o processo de irrigação precisa de mais controle sobre o uso da água, afinal sabe-se que o desperdício de água é um problema atual.

A agricultura familiar fornece saídas para a obtenção de alimentos para um pequeno grupo de pessoas. Nela, é preciso atenção e cuidado para com as plantas para um bom rendimento. Dessa forma, acredita-se que uma irrigação automática e programada possa ser a solução para um efetivo controle e melhor crescimento das plantas. Com o uso de equipamentos eletrônicos e plataformas possibilita-se elaborar essas propostas.

Em suma, a proposta do projeto é suprir as necessidades das plantas sem que seja fundamental a presença de alguém que as inspecione. O propósito é auxiliar as pessoas a prevenirem condições desfavoráveis à planta, conservarem e observarem o cultivo. Com a crescente busca por alimentos orgânicos (ZH, 2015), o projeto ajuda as pessoas a cultivarem sua horta. Dessa maneira, obtendo benefícios com uma alimentação mais saudável e por um custo acessível. A ideia é automatizar a irrigação da plantação de acordo com as condições adequadas para isso acontecer e informar algumas características do ambiente em que a planta está.

1.2. OBJETIVOS

1.2.1. Objetivo geral

Automatizar o processo de irrigação de plantas a partir do uso de um hardware microcontrolado, visando fornecer um produto bem monitorado e eficiente para que a planta cresça nas melhores condições possíveis.

1.2.2. Objetivos específicos

Para alcançar a meta colocam-se os seguintes objetivos:

- Elaborar um sistema de irrigação automático, com o acionamento de uma bomba por meio de um hardware microcontrolado.
- Elaborar um sistema de medição de temperatura, umidade do ar e do solo.
- Usar o display de LCD para exibir as informações relevantes do ambiente para o usuário.
- Utilizar o minicomputador Raspberry Pi a fim de possibilitar a coleta e armazenamento de dados de forma precisa e ampliar o conhecimento sobre o computador.
- Controlar a quantidade de água utilizada na irrigação em busca do maior aproveitamento.
- Testar e verificar o funcionamento do projeto.

1.3. ESTRUTURA DO TRABALHO

Este trabalho se divide nos seguintes capítulos:

- Capítulo 1: Destinado à introdução, contextualização do projeto e os objetivos a serem alcançados para realização da proposta.
- Capítulo 2: Apresenta o referencial teórico do trabalho, envolvendo uma contextualização sobre os fatores que influenciam no desenvolvimento das plantas e, também, informando as características dos componentes utilizados no protótipo.
- Capítulo 3: Destinado à apresentação dos materiais e métodos, com uma exemplificação dos elementos utilizados em suas funções e as metodologias utilizadas a fim de concluir os objetivos.
- Capítulo 4: Destinado ao desenvolvimento do protótipo, tanto na construção do hardware quanto na elaboração dos softwares utilizados.
- Capítulo 5: Destinado aos resultados obtidos com a execução do protótipo construído.

- Capítulo 6: Destinados às considerações finais do projeto, com conclusão e proposta de trabalhos futuros.

2. REFERENCIAL TEÓRICO

2.1. AGRICULTURA

Considerada herança da evolução das civilizações humanas, o cultivo do solo como atividade de subsistência representa grande parte da economia mundial. A agricultura é a “atividade que tem por objetivo a cultura do solo para produzir vegetais úteis ao homem e/ou para a criação de animais.” (HOUAISS; VILLAR, 2009). Um dos tipos de agricultura muito utilizado é a agricultura familiar.

A agricultura familiar tem se desenvolvido no cenário econômico brasileiro. Constituída por famílias de agricultores que produzem em parte para seu próprio consumo e comercializam a outra parte da produção como fonte de aquisição de renda. Esse tipo de agricultura necessita de uma atenção maior visto os desafios que o setor agrícola tem enfrentado. Um desses desafios são os recursos hídricos, que vem se tornando escassos devido à inúmeros fatores, dentre eles o desperdício.

Além disso, com o decorrer do tempo a exploração da atividade agrícola desencadeou a carência de técnicas de desenvolvimento mais eficientes. Dentre outros fatores, o uso de tecnologia tem auxiliado o aumento na qualidade dos produtos.

Embora a modernização das técnicas de produção agrícola tenha se ampliado ao decorrer dos anos ainda é essencial a busca por práticas que contribuam para um avanço no setor produtivo sem grandes impactos ambientais. A tecnologia é um dos principais aliados da agricultura nesse sentido. O monitoramento e a análise de dados contribuem tanto para a compreensão do problema, quanto no auxílio do desenvolvimento de práticas que o solucionem.

2.2. ÁGUA

A água é uma das substâncias mais importantes para a manutenção da vida no planeta Terra. Pode-se observar que a distribuição da vegetação no planeta está relacionada com a presença da água de alguma forma. “Hoje se sabe que a disponibilidade de água não só limita o crescimento vegetal, como também a ocupação humana e vegetal na Terra.” (PIMENTEL, 2004).

Além disso, as soluções aquosas de minerais, solutos, nutrientes e gases participam dos processos químicos dos seres vivos. E nos seres fotossintéticos, como as plantas, a água também tem importante participação na fotossíntese. Em uma das reações celulares, a água se torna fonte do oxigênio molecular da atmosfera e também do hidrogênio para reduzir o CO₂ para carboidrato. Dessa forma, apenas parte da água é incorporada à planta e outra parte é lançada

para a atmosfera sob a forma de vapor de água. Assim, “A elevada exigência de água, portanto, é intrínseca da planta, que, se não satisfeita, afeta o crescimento e a produção.” (PIMENTEL, 2004).

Então, com a necessidade de aprimorar a agricultura, a irrigação automática se apresenta como uma forma de manter o controle da água necessária para o desenvolvimento das plantas, visando a maior produtividade nas atividades agrícolas. Existem muitos tipos de irrigação, entre eles pode-se citar os seguintes: sistemas de irrigação localizada e sistemas de irrigação por aspersão.

Os sistemas de irrigação localizada, como por gotejamento e micro aspersão, funcionam aplicando baixa taxa de água a uma alta frequência e em certas localidades da cultura. Isso controla-se dependendo da intenção do agricultor. A água pode ser aplicada na forma de gotas ou com dispositivos de mini aspersores.

Já nos sistemas de irrigação por aspersão, a água é espalhada por meio de aspersores ligados a tubulações, podendo ser fixos ou móveis. É o sistema de irrigação que simula uma chuva, a água se apresenta em pequenas gotas.

Abaixo observa-se, nas figuras 1 e 2, os sistemas de irrigação definidos no trabalho.

Figura 1: Irrigação localizada por gotejamento



Fonte: Cultura Mix (2013)

Figura 2: Irrigação por aspersão



Fonte: Revista Globo Rural (2013)

Com isso, tem-se que agricultura irrigada, se bem planejada, possui grande potencial produtivo, principalmente para a produção de alimentos. Porém, “A agricultura irrigada é conhecida como a maior usuária de água doce no mundo.” (PIMENTEL, 2004). Portanto, é imprescindível a busca da economia da água utilizada no ato de irrigar plantas, principalmente se a irrigação é automática.

2.3. TEMPERATURA

Todos os seres vivos, embora de forma diferente, sofrem influência da temperatura, que é uma das principais responsáveis pela distribuição da biodiversidade por todo o planeta e também participa de processos de seleção natural. Segundo Mollo (2009), cada espécie tem uma faixa de temperatura que é aceitável. Acima ou abaixo da faixa pode ocasionar a morte, pois a temperatura fora do ideal atrapalha as funções vitais. Entretanto, mesmo com a faixa, ainda existe uma temperatura ótima para o melhor desenvolvimento.

Assim, é perceptível que a temperatura também tem influência muito evidente no desenvolvimento das plantas, em todos os processos de crescimento. Conforme Mollo (2009), a temperatura apresenta influência nos processos de germinação e até na quantidade de água

absorvida pela planta e, conseqüentemente, afetando os processos metabólicos das células dos vegetais, nos quais a água tem um papel muito importante.

Como Mollo (2009) e Ribeiro et al (2012) destacam, com a temperatura inadequada, o desenvolvimento das plantas encontra-se comprometido. Nas temperaturas altas as sementes e plantas podem, em casos extremos, morrer prematuramente. Na fase de germinação, por exemplo, temperaturas elevadas inibem as atividades enzimáticas, o que diminui a velocidade da germinação. E no solo, que também é afetado pela temperatura, as raízes das plantas diminuem, pela diminuição do potencial do solo. Já com temperaturas baixas as plantas podem encontrar-se em estado dormente, como em sementes ou bolbos, à espera da mudança de temperatura.

Portanto, o controle e o conhecimento da temperatura do ambiente em que as plantas se encontram é de suma importância para controlar também seu desenvolvimento. Afinal, a temperatura ideal fornece melhor produtividade por um menor custo, logo é uma informação necessária para o agricultor.

2.4. UMIDADE RELATIVA DO AR

Entende-se a umidade relativa como a concentração de vapor água na atmosfera. A água, como descrita anteriormente, possui grande importância no processo de absorção de nutrientes pela planta. Segundo Suhogusoff, V. G. et al (2008), com a maior responsabilidade pela incorporação de nutrientes, tem-se as raízes. Porém, as folhas também possuem a capacidade de absorção. Dessa maneira, a umidade relativa do ar também influencia na assimilação de nutrientes.

Conforme afirma Costa (2001), a umidade relativa também influencia na transpiração das plantas. As plantas ao reterem água pelas raízes precisam liberá-la através das folhas após a água exercer seu papel de transportar minerais. Porém, quando a umidade do ar se encontra muito baixa ou muito alta, isso pode prejudicar a planta.

Como afirma o site afe.com.br, quando a umidade está muito alta prejudica a transpiração e propicia o surgimento de doença fúngicas, pois a água tem dificuldade de evaporar quando o ambiente está muito úmido. Por outro lado, quando a umidade está muito baixa, a taxa de transpiração aumenta, podendo evaporar mais água que o necessário, às vezes prejudicando a polinização e fecundação.

Com a importância da transpiração nos mais variados processos metabólicos dos vegetais, é necessário o conhecimento da umidade relativa do ar do ambiente. Por isso, o projeto

apresenta também uma forma de informar a umidade relativa. É de grande importância o conhecimento das áreas de cultivo antes da prática da agricultura.

2.5. DESCRIÇÃO DOS COMPONENTES UTILIZADOS

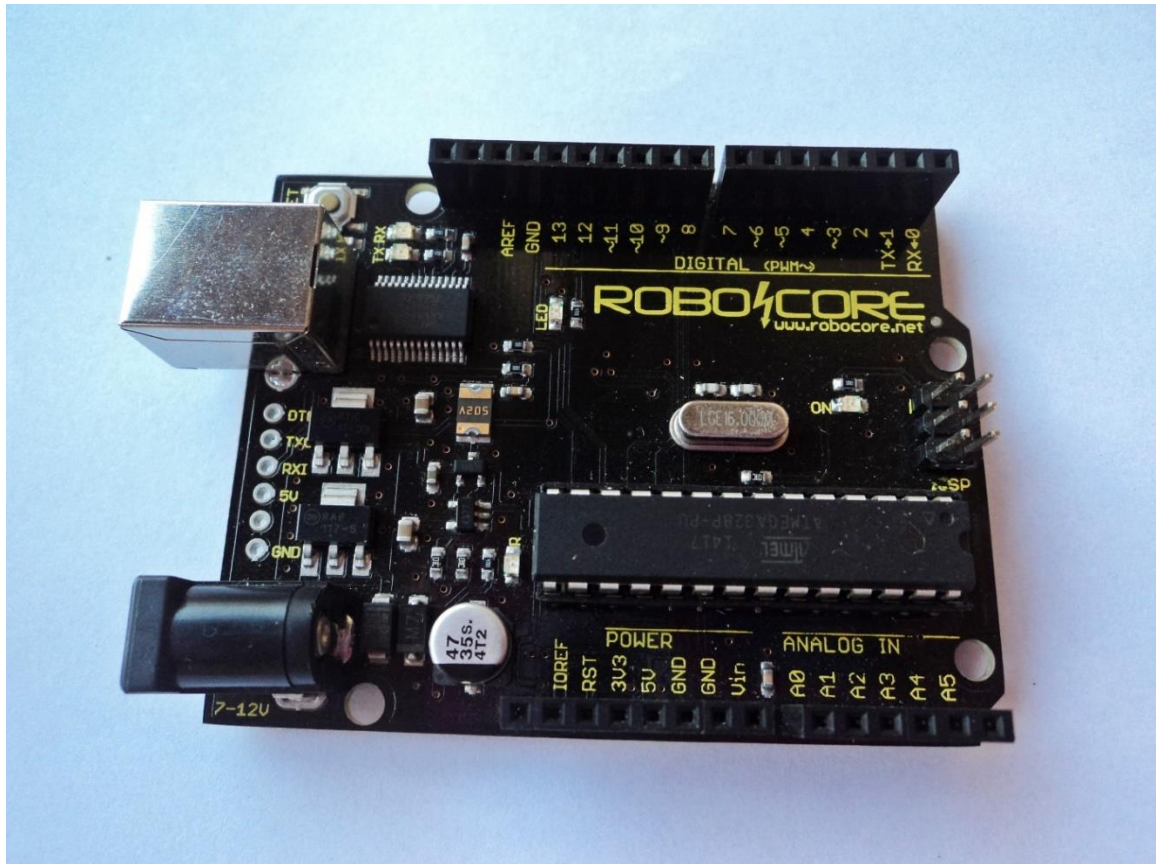
2.5.1. Arduino

Segundo o site Arduino.cc, o Arduino é uma plataforma de hardware microcontrolado e software open-source e de baixo custo. O Arduino pode ter várias entradas analógicas ou digitais, que retornam várias saídas digitais ou analógicas.

“O Arduino é uma plataforma de prototipagem eletrônica open-source que se baseia em hardware e software flexíveis e fáceis de usar. É destinado a artistas, designers, hobbistas e qualquer pessoa interessada em criar objetos ou ambientes interativos. ” (ARDUINO.CC, 2016).

O Arduino é constituído por dois componentes principais: hardware e software. O hardware é composto pela placa na qual os projetos são elaborados. Essa placa é um circuito microcontrolado construída com o intuito de facilitar a comunicação com o computador. No projeto proposto será utilizado um genérico do Arduino UNO, BlackBoard, fabricado no Brasil, pela empresa RoboCore, <https://www.robocore.net/>. A seguir, a figura 3 apresenta a placa utilizada:

Figura 3: Arduino BlackBoard



Fonte: Elaborado pelos autores deste trabalho (2016)

Para o desenvolvimento do programa para o Arduino, que recebe o nome de sketch, é utilizada uma linguagem de programação fundamentada nas linguagens C e C++. Para a criação de sketches utiliza-se uma IDE, um software que permite a programação. Após a criação do sketch, o programa criado é transmitido para o Arduino por meio do upload realizado pela IDE usando a comunicação USB.

Então, depois que o software é gravado no microcontrolador do Arduino ele será executado de forma independente até que o software seja alterado. Mas, é necessário que o Arduino esteja ligado a uma fonte de energia.

2.5.1.1. Microcontroladores

Com os avanços tecnológicos que ocorrem com o passar do tempo, novos componentes eletrônicos são criados com o intuito de auxiliar em projetos. Os microcontroladores se encaixam nesses avanços, pois é amplamente aplicável nas mais variadas funcionalidades.

“O microcontrolador é o agrupamento de vários componentes em um sistema microprocessado. Basicamente o microcontrolador é microprocessador com memória RAM e de programa, temporizadores e circuitos de clock embutidos. ” (LIMA, 2009).

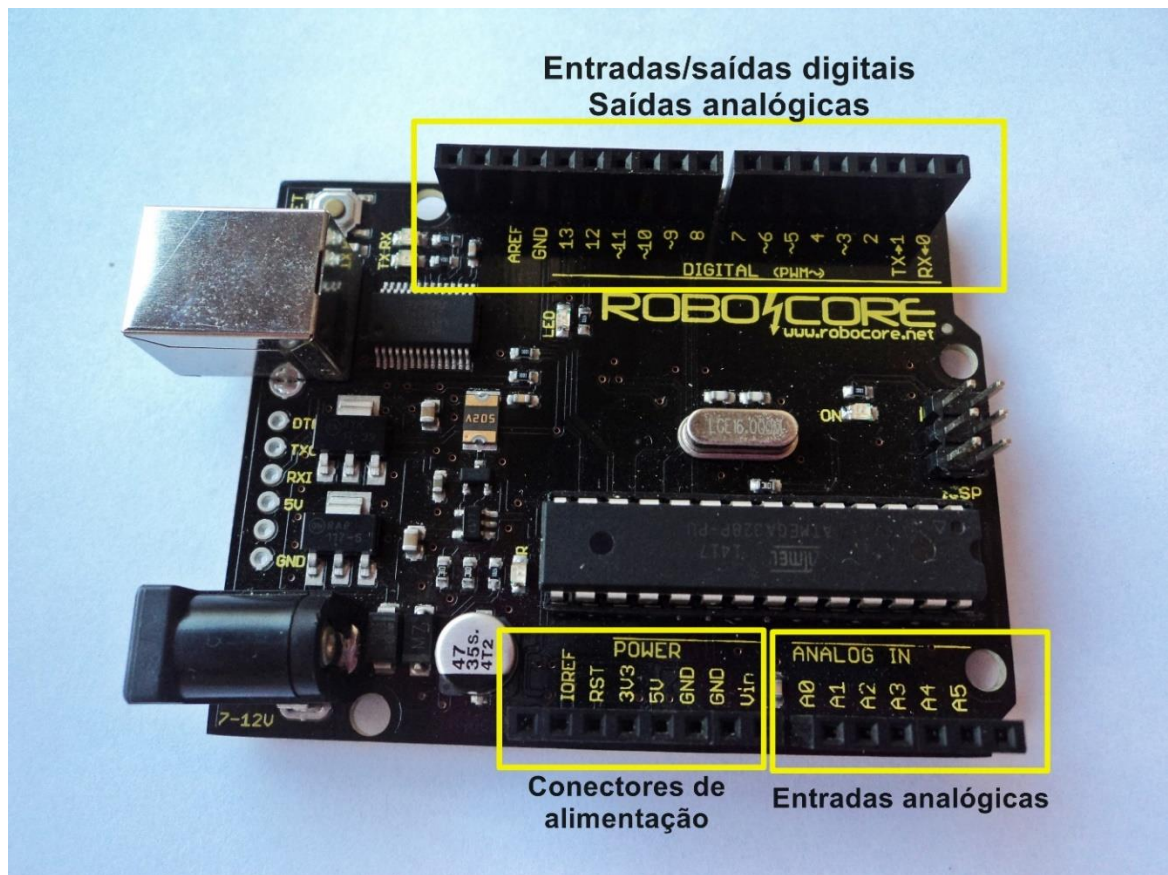
Dada a variedade de componentes e funcionalidades compactados em um microcontrolador sua utilidade oferece uma variabilidade de funções e aplicações que um projeto requisite. Em suma, o microcontrolador é um componente capaz de realizar tarefas com base em um software.

No projeto proposto utiliza-se a plataforma Arduino, uma placa baseada em um microcontrolador da família AVR. Segundo Lima (2009), a unidade processamento contido no microcontrolador AVR é eficiente, sua arquitetura fundamentada na programação C gera códigos reduzidos.

2.5.1.2. Arduino UNO

De acordo com o site Arduino.cc, o Arduino UNO é uma placa de microcontrolador microcontrolada baseada no ATmega328, um dispositivo de 8 bits da família AVR. Possui 14 pinos digitais de entrada e saída (os quais 6 podem ser usados como saídas analógicas), 6 pinos analógicos de entrada, um clock de 16 MHz, uma conexão USB, um conector de alimentação, um conector ICSP e um botão de reset. A figura 4 abaixo ilustra alguns pinos da placa.

Figura 4: Exemplos das conexões do Arduino UNO



Fonte: Elaborado pelos autores deste trabalho (2017)

Além disso, com os componentes presentes na placa, torna-se mais simples a conexão com o computador, a programação do microcontrolador e até o modo de iniciar o funcionamento. O Arduino UNO é muito utilizado pela sua simplicidade, baixo custo e a possibilidade de realização de diversos projetos. Por isso, é bastante indicado para quem é iniciante.

A placa pode ser alimentada tanto por uma fonte externa quanto pela conexão USB. Quando a conexão USB provém de um computador, por exemplo, a placa não realiza a regulação de tensão recebida. Isso porque a tensão vinda do computador já se encontra em uma faixa que o Arduino pode utilizar. Já quando a alimentação é por fonte externa a tensão deve estar entre os limites 6V – 20V. Entretanto, recomenda-se a faixa de tensão 7V – 12V para evitar tensões instáveis ou danificação da placa. A tensão é então regulada para o funcionamento do Arduino. Além disso, a placa tem a capacidade de comutar as tensões utilizadas. Se a fonte externa e USB estiverem conectadas, a placa utiliza a fonte externa para alimentação.

Abaixo segue as características do Arduino UNO:

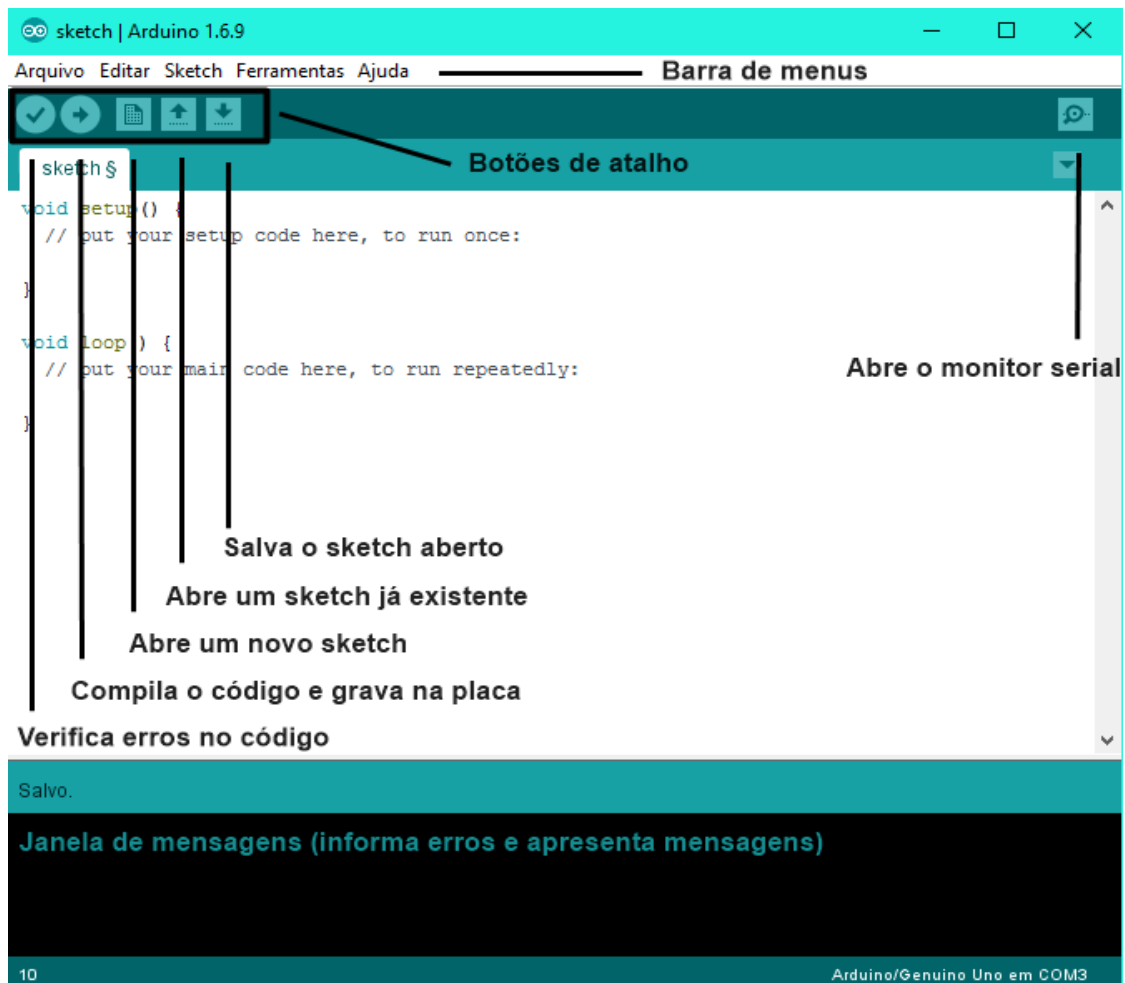
- Microcontrolador: ATmega328
- Tensão de operação: 5V
- Tensão de entrada (recomendada): 7-12V
- Tensão de entrada (limite): 6-20V
- Entradas/Saídas digitais: 14 (6 pinos PWM)
- Pinos Analógicos (Entrada): 6
- Memória Flash: 32 KB (ATmega328), 0.5kb usado pelo bootloader
- SRAM: 2 KB (ATmega328)
- Velocidade do Clock: 16 MHz

2.5.1.3. IDE do Arduino

A IDE do Arduino possui uma linguagem fundamentada nas linguagens de programação C e C++. Ela é encontrada fácil e gratuitamente no site oficial do Arduino. Com ela são desenvolvidos os sketches que ao receberem upload para a placa passarão pelo compilador avr-gcc. Assim, o sistema será compreendido pelo microcontrolador da placa.

A figura 5 ilustra a IDE e suas funcionalidades:

Figura 5: IDE do Arduino e suas funcionalidades



Fonte: Elaborado pelos autores deste trabalho (2016)

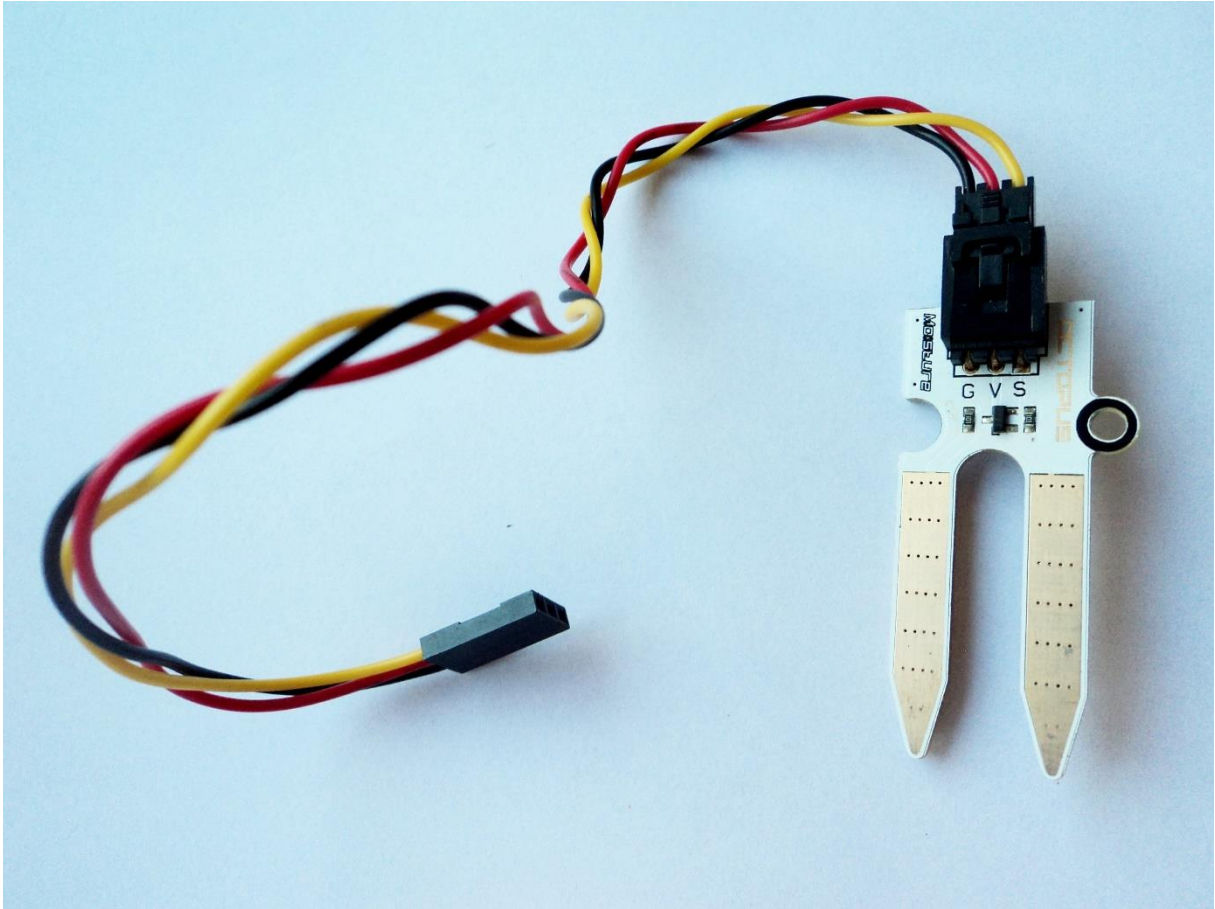
2.5.2. Sensor de umidade do solo

O sensor de umidade do solo verifica os níveis de umidade do solo, sendo considerado seco ou úmido. O sensor de umidade pode ser usado tanto na terra quanto na água. O sensor utilizado (figura 6) é da Octopus, <http://www.electronics.com/estore/>, e tem natureza analógica. Quanto maior for a umidade do solo, maior é o valor transmitido para o Arduino. O sensor é formado por dois eletrodos que ao entrarem em contato com algum material passaram uma corrente elétrica caso este não seja isolante ou o contrário caso seja isolante. O solo seco dificulta a passagem da corrente, já o solo com água facilita essa passagem.

Então, para o conhecimento do nível de umidade é realizada uma comparação de acordo com a passagem ou não da corrente elétrica. Ao enviar sinais analógicos, o software no Arduino verifica se deve ou não irrigar as plantas.

O sensor possui três conectores. O fio preto é o GND, o fio vermelho é o VCC e o fio amarelo transmite os sinais analógicos.

Figura 6: Sensor de umidade do solo

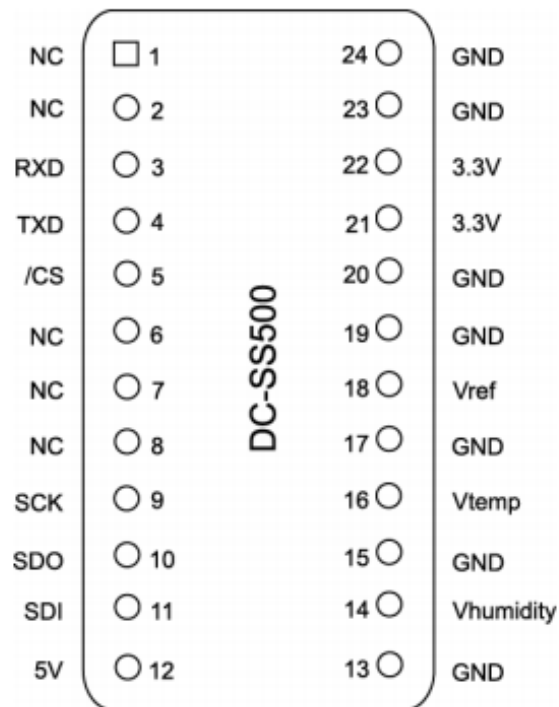


Fonte: Elaborado pelos autores deste trabalho (2016)

2.5.3. Módulo sensor de umidade do ar e de temperatura

O módulo DC-SS500 sensor de umidade relativa do ar e de temperatura é um componente eletrônico formado pelo microcontrolador PIC16F690 para processamento da informação. Ele mede a umidade relativa por meio do sensor HS1101, que é formado por uma célula capacitiva. Este fornece uma saída de tensão que corresponda a faixa 1 – 99% de umidade. A interface do componente pode ser por comunicação serial com taxa de transmissão 9600 bps, SPI ou analógica. Com isso, ele conta com 24 pinos, distribuídos como a figura 7 ilustra.

Figura 7: Pinos do sensor de umidade relativa e temperatura



Fonte: SURE Electronics (2004)

A tabela 1 descreve a função de cada pino:

Tabela 1: Descrição dos pinos do módulo sensor de umidade relativa e temperatura

RÓTULO	PINO	FUNÇÃO	RÓTULO	PINO	FUNÇÃO
NC	1	Sem conexão	GND	13	Terra
NC	2	Sem conexão	Vhumidity	14	Saída de voltagem da umidade
RXD	3	Receptor de dados UART	GND	15	Terra
TXD	4	Transmissor de dados UART	Vtemp	16	Saída de voltagem da temperatura
/CS	5	SPI disponível	GND	17	Terra

NC	6	Sem conexão	Vref	18	Referência p/ saída de voltagem
NC	7	Sem conexão	GND	19	Terra
NC	8	Sem conexão	GND	20	Terra
SCK	9	SPI clock	3.3V	21	Entrada de 3.3V em DC
SDO	10	Saída de dados SPI	3.3V	22	Entrada de 3.3V em DC
SDI	11	Entrada de dados SPI	GND	23	Terra
5V	12	Entrada de 5V em DC	GND	24	Terra

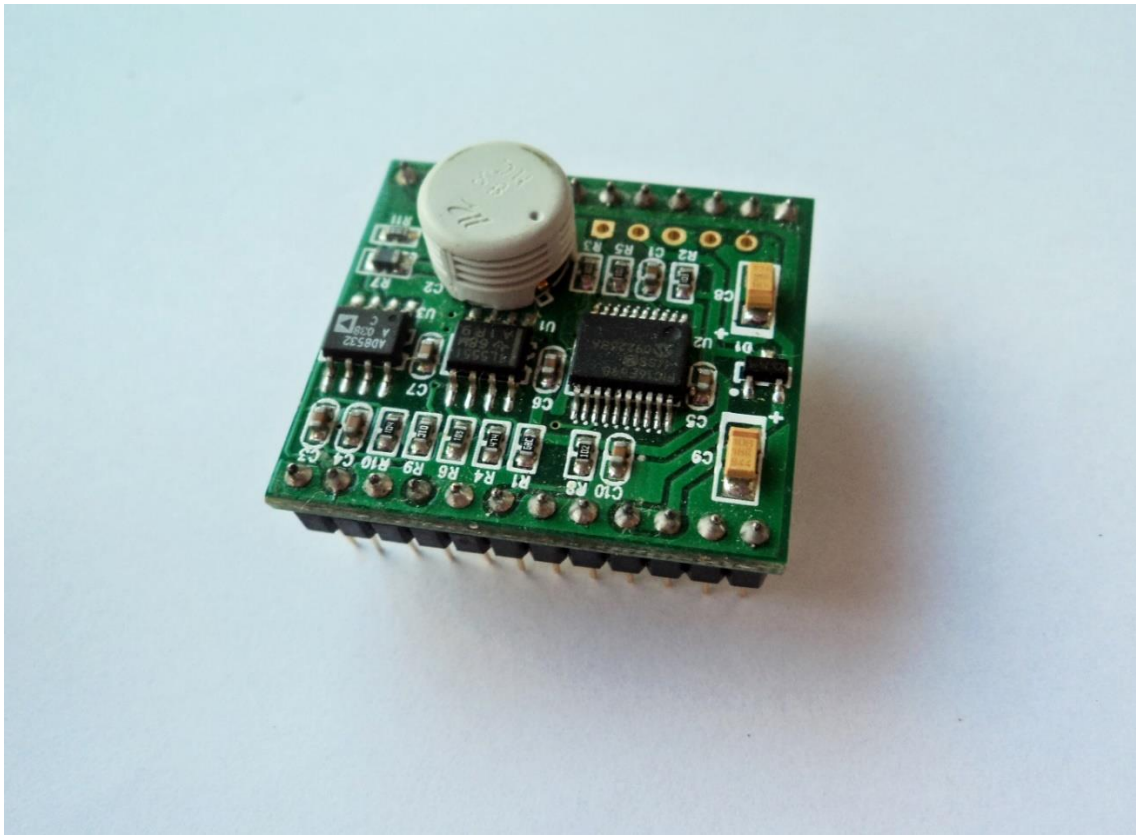
Fonte: SURE Electronics, traduzido pelos autores deste trabalho (2016)

Com a possibilidade de fornecer diferentes tipos de saída de dados, o módulo pode fornecer tanto dados mais básicos quanto dados mais detalhados e seguros. Abaixo segue algumas características gerais do módulo:

- Tensão de operação: 3.3V – 5V
- Rápido tempo de resposta
- Faixa extensa da medição de temperatura e umidade relativa.
- Saída digital e analógica
 - Analógico: saída com voltagem da temperatura e umidade relativa do ar
 - Digital: UART half duplex e SPI modo slave
- No formato DIP (Dual In-Line Package)

A figura 8 apresenta como é o sensor utilizado:

Figura 8: Módulo sensor de umidade relativa do ar e temperatura



Fonte: Elaborado pelos autores deste trabalho (2016)

No projeto, o sensor de umidade do ar e temperatura transmite dados analógicos para o Arduino e tais informações serão tratadas e convertidas para a numeração convencional de temperatura em graus Celsius e umidade relativa. Para conversão de uma tensão recebida para a numeração habitual de temperatura e umidade relativa, tem-se como base as tabelas 2 e 3 abaixo:

Tabela 2: Relação de tensão obtida com a umidade relativa

UMIDADE (% RH)	VOLTAGEM DA UMIDADE ($V_{ref} = 3.07V$)	VOLTAGEM DA UMIDADE ($V_{ref} = 4.95V$)
10	0.33 V	0.52 V
20	0.64 V	1.04 V
30	0.92 V	1.48 V
40	1.24 V	1.95 V
50	1.54 V	2.44 V

60	1.84 V	2.92 V
70	2.1 V	3.4 V
80	2.44 V	3.91 V
90	2.78 V	4.4 V

Fonte: SURE Electronics, traduzido pelos autores deste trabalho (2016)

Tabela 3: Relação de tensão obtida com a temperatura

TEMPERATURA (°C)	VOLTAGEM DA TEMPERATURA (Vref = 3.07V)	VOLTAGEM DA TEMPERATURA (Vref = 4.95V)
-15	0.54 V	0.89 V
-10	0.64 V	1.06 V
-5	0.75 V	1.24 V
0	0.88 V	1.45 V
5	1.02 V	1.67 V
10	1.17 V	1.91 V
15	1.32 V	2.15 V
20	1.47 V	2.40 V
25	1.62 V	2.64 V
30	1.76 V	2.87 V
35	1.90 V	3.10 V
40	2.03 V	3.30 V
45	2.15 V	3.50 V
50	2.26 V	3.67 V
55	2.36 V	3.83 V
60	2.45 V	3.98 V

Fonte: SURE Electronics, traduzido pelos autores deste trabalho (2016)

2.5.4. Display LCD

O display LCD é componente eletrônico utilizado para a transmissão de informações visuais para as pessoas. Muito utilizado em diversos tipos de projetos, ele pode ser usado com uma variedade de microcontroladores e placas. Os módulos podem ser encontrados em diversos tamanhos, o módulo utilizado no projeto possui o tamanho de 16x2, sendo 16 colunas e 2 linhas, que é um módulo comum para caracteres.

Esse display possui *backlight*, que é um recurso com luz de fundo que auxilia a leitura da informação em locais escuros. A alimentação do *backlight* normalmente é encontrada nos últimos pinos do display, no display escolhido ela se encontra nos pinos 15 e 16. Para o controle do contraste do display utilizou-se um potenciômetro de 10K no pino 3 do LCD.

Além disso, esses módulos possuem um controlador que permite a comunicação entre eles e outros componentes. Então, os displays possuem pinos para alimentação, para receber dados e também os pinos para controle do display. Com isso, é evidente a existência de protocolos para a efetivação da comunicação entre módulos e placas.

Abaixo segue a tabela 4 que descreve a pinagem do display LCD.

Tabela 4: Descrição dos pinos do módulo display LCD

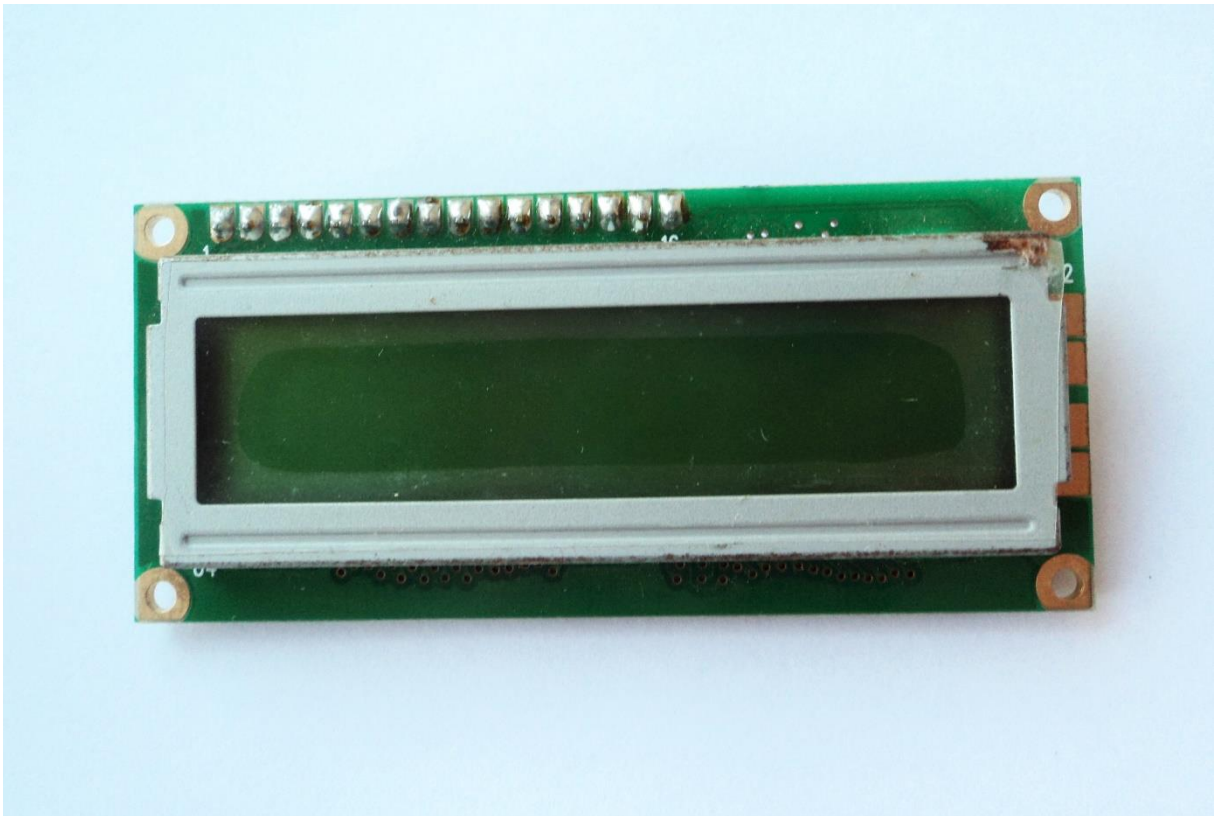
PINO	FUNÇÃO	DESCRIÇÃO
1	Alimentação	Terra
2	Alimentação	VCC 5V
3	V0	Ajuste de contraste
4	Escolha RS	1 – Dado; 0 – Instrução
5	Escolha R/W	1 – Leitura; 0 – Escrita
6	E Chip Select	1 ou (1 → 0) – Habilita; 0 – Desabilita
7	B0 LSB	Barramento de dados
8	B1	Barramento de dados
9	B2	Barramento de dados
10	B3	Barramento de dados
11	B4	Barramento de dados
12	B5	Barramento de dados
13	B6	Barramento de dados
14	B7 MSB	Barramento de dados

15	A (quando existir)	Ânodo para <i>backlight</i>
16	K (quando existir)	Cátodo para <i>backlight</i>

Fonte: Barbacena e Fleury (1996)

A figura 9 ilustra o módulo display LCD utilizado na construção do protótipo:

Figura 9: Módulo display LCD



Fonte: Elaborado pelos autores deste trabalho (2016)

2.5.5. Módulo relé

Segundo Braga (2012), o relé é um comutador eletromecânico, ou seja, ele servirá como uma chave eletromecânica que fecha e abre circuitos ao receber um sinal. Logo, é muito utilizado em projetos com Arduino, pois permite o acionamento ou não de equipamentos, realizando assim diversas tarefas. A bobina do relé, ao ter uma corrente a percorrendo, gera um campo elétrico que atrai contatos elétricos, que abrem ou fecham outro circuito.

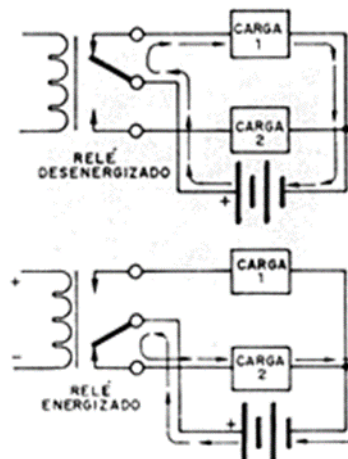
Através de uma corrente de controle aplicada à bobina de um relé, podemos abrir, fechar ou comutar os contatos de uma determinada forma, controlando assim as correntes que circulam por circuitos externos. Quando a corrente deixa de circular

pela bobina do relé o campo magnético criado desaparece, e com isso a armadura volta a sua posição inicial pela ação da mola.

Os relés se dizem energizados quando estão sendo percorridos por uma corrente em sua bobina capaz de ativar seus contatos, e se dizem desenergizados quando não há corrente circulando por sua bobina. (BRAGA, 2012)

Na figura 10 abaixo é possível observar o funcionamento de um relé.

Figura 10: Funcionamento do relé



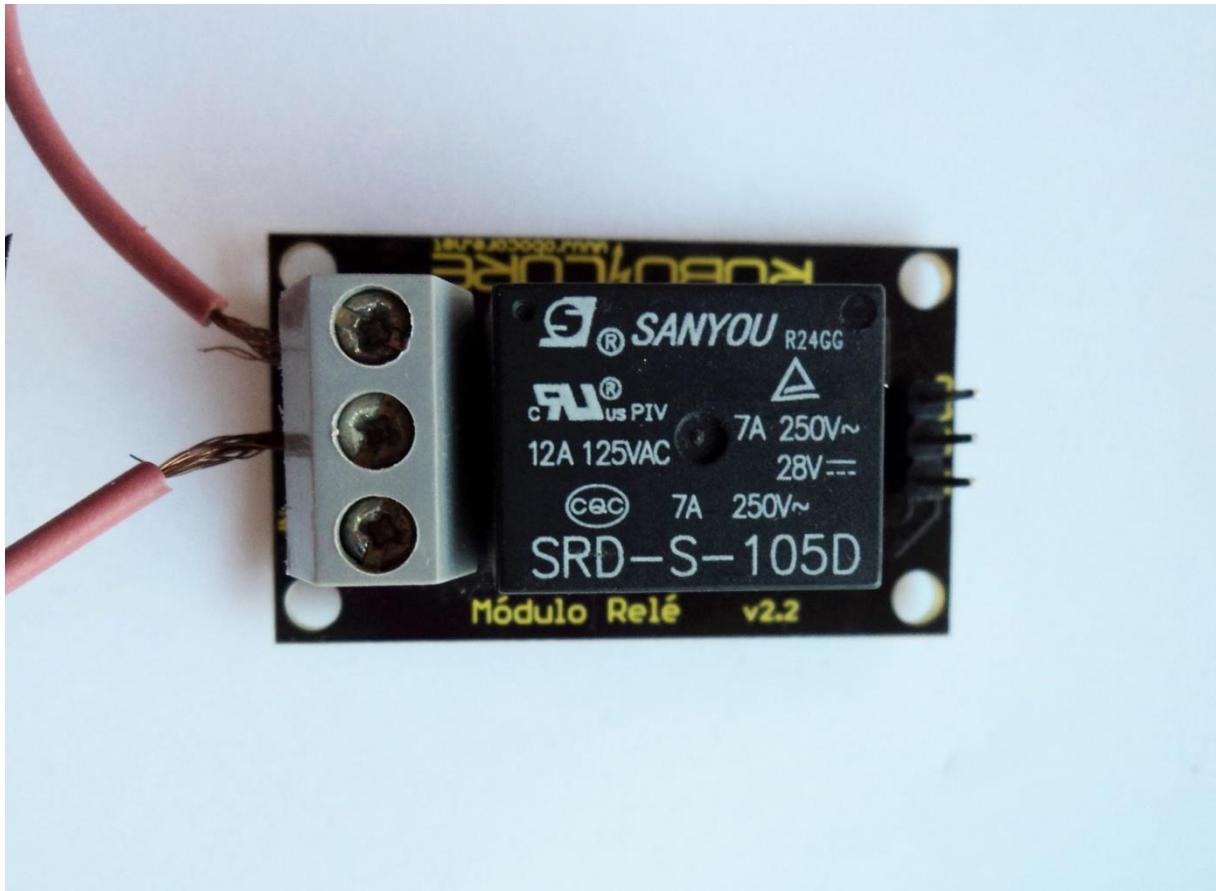
Fonte: Braga (2012)

Portanto, com o relé é possível o controle de circuitos. No projeto o relé controla o circuito da bomba d'água, ativando-a quando recebe o sinal vindo do Arduino e desativando-a quando ocorre o oposto. Além disso, uma das características do relé é a possibilidade de controlar circuitos de altas correntes com circuitos de baixas correntes. Também há um isolamento entre o circuito que controla e o circuito controlado, promovendo segurança à aplicação.

Abaixo se encontram algumas características, de acordo com a Sanyou Relays, do módulo utilizado:

- Corrente dos contatos de 7A em 250V AC ou 28V DC;
- Tensão de alimentação de 5V;
- Controle AC/DC;
- Contato normal aberto (NA) e normal fechado (NF).

Figura 11: Módulo relé



Fonte: Elaborado pelos autores deste trabalho (2016)

2.5.6. Raspberry Pi

O Raspberry Pi é um minicomputador do tamanho de um cartão de crédito que foi desenvolvido pela instituição de caridade The Raspberry Pi Foundation, do Reino Unido. Possui todos os componentes que constitui um computador: memória RAM, processador, entrada de energia, slots para memória secundária, interfaces como USB e HDMI. Além disso, ele pode suportar diversas distribuições baseadas em Linux e ainda pode-se conectar mouse e teclado ao minicomputador. E, a partir de um sistema operacional, é possível utilizá-lo de diversas formas, como editar textos e usar a internet.

O objetivo da criação do Raspberry Pi foi criar um pequeno computador mais barato e de software livre para escolas. Para, então, possibilitar a educação na área da computação para jovens de diferentes idades. Logo, ao ser acessível, o Raspberry Pi permite a realização desse propósito pela exploração mais prática da computação.

O minicomputador também possibilita a conexão com componentes, como sensores, pelo conector GPIO (General Purpose Input/Output) de 26 pinos. O modelo de Raspberry Pi

utilizado no projeto é o Rev 2 Model B fabricado no Reino Unido. Segue abaixo algumas especificações desse modelo:

- 512MB de RAM
- Pontos de fixação
- Processador de 700Mhz
- 2 x Portas USB
- Saída de vídeo HDMI
- Saída de vídeo RCA
- Saída de áudio Jack
- Porta Ethernet 10/100Mb
- 5V Micro USB para alimentação
- Slot para cartão de memória
- 26 pinos e slot de expansão Header

A figura 12 ilustra o minicomputador utilizado.

Figura 12: Raspberry Pi com case



Fonte: Elaborado pelos autores deste trabalho (2016)

2.5.6.1. Sistema GNU/Linux

Linux é o núcleo do sistema operacional, que consiste em vários softwares que tornam possível a comunicação e interação entre o usuário e a máquina. Linux não é o sistema operacional completo. Ele é o kernel, que é basicamente o coração do sistema operacional, a parte essencial. O kernel estabelece a relação entre o hardware e outros programas que o usuário utiliza.

Segundo o site debian.org, “a parte mais importante de um sistema operacional é o kernel. Em um sistema GNU/Linux o Linux é o componente do kernel. O resto do sistema consiste de outros programas, muitos dos quais foram escritos pelo ou para o projeto GNU.”

Dessa forma, o sistema operacional completo é chamado, na teoria, de GNU/Linux. Afinal, é uma integração do kernel com o hardware do computador e os programas executados. O Linux foi feito baseado no sistema Unix. Além disso, por ser OpenSource, não possui um proprietário, o desenvolvimento é feito a partir da colaboração de voluntários. E existem muitas distribuições desse sistema, como exemplo tem-se: Debian, Ubuntu e Fedora.

2.5.6.2. Python

“Python é uma linguagem de altíssimo nível (em inglês, Very High Level Language) orientada a objeto, de tipagem dinâmica e forte, interpretada e interativa.” (BORGES, 2010).

A linguagem de programação Python é uma linguagem com sintaxe das mais fundamentais, simples e clara. É livre e de código aberto, com uma licença similar a GPL, a licença Python Software Foundation License.

Além disso, Python é utilizado como linguagem para o desenvolvimento de sistemas web, de desktop ou até móvel. E também é utilizado como linguagem de script, permitindo a automação e realização de tarefas. Como exemplos de softwares que utilizam Python como linguagem de script, tem-se: BrOffice.org, PostgreSQL, Blender e GIMP.

Segundo Borges (2010),

A linguagem inclui diversas estruturas de alto nível (listas, dicionários, data/hora, complexos e outras) e uma vasta coleção de módulos prontos para uso, além de frameworks de terceiros que podem ser adicionados. Também possui recursos encontrados em outras linguagens modernas, tais como: geradores, introspecção, persistência, metaclasses e unidades de teste. Multiparadigma, a linguagem suporta programação modular e funcional, além da orientação a objetos. Mesmo os tipos básicos no Python são objetos. A linguagem é interpretada através de bytecode pela máquina virtual Python, tornando o código portátil. Com isso é possível compilar

aplicações em uma plataforma e rodar em outros sistemas ou executar direto do código fonte.

É importante salientar ainda que Python consegue integrar-se com outras linguagens de programação, como com C, Java, Fortran. Possui igualmente a vantagem de ser multiplataforma, já que é uma linguagem interpretada só a presença de um interpretador torna possível a execução de um sistema Python.

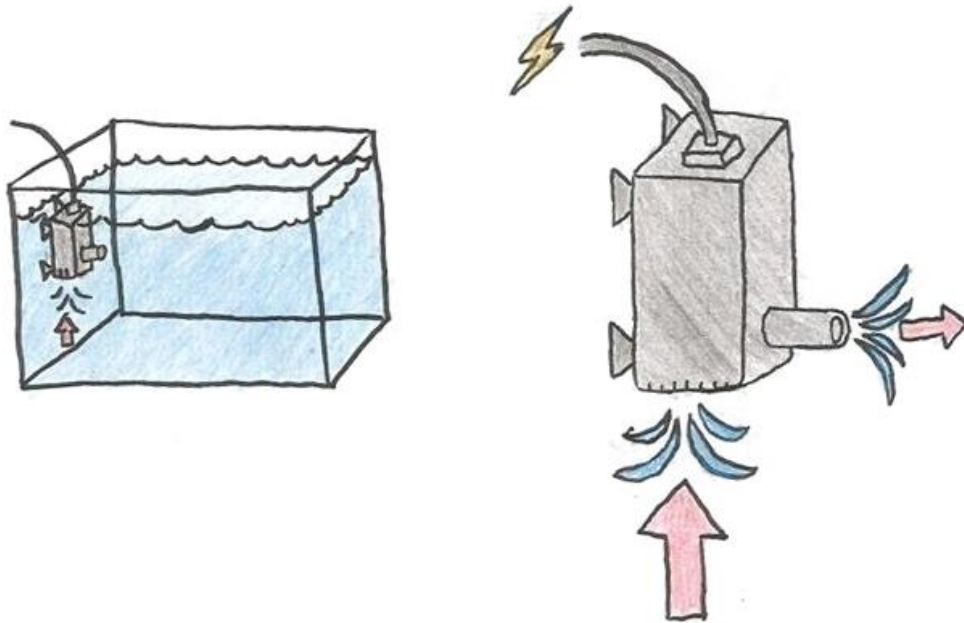
Como é utilizado Linux no Raspberry Pi do projeto, a linguagem Python foi escolhida, pois, em sistemas Linux, o Python geralmente já faz parte do sistema e é bem mais fácil e prático de baixar os pacotes necessários para o desenvolvimento de sistemas e de realizar a programação em si.

2.5.7. Bomba SP-500

Para realizar o processo de irrigação das plantas, foi utilizada a bomba de aquário SP-500 da JAD Aquarium Co., Ltd, <http://www.jadaquarium.com/>. A bomba tem o circuito aberto pelo relé que, após receber o sinal do Arduino, fecha o circuito ativando a bomba e realizando a irrigação. A bomba utilizada promove a circulação da água em aquários. Além disso, tem o seu funcionamento de forma submersa e a água sai pelo tubo superior, como observa-se na figura 13. Por isso, funcionará para impulsionar a água de acordo com o desejado e esperado.

Segundo o fabricante do produto, a bomba utilizada funciona em uma tensão de 220V, uma vazão de 150L/h, potência de 6W e frequência entre 50 – 60Hz. Abaixo, na figura 14, mostra-se a bomba de aquário utilizada no projeto proposto.

Figura 13: Representação do funcionamento da bomba de aquário



Fonte: Elaborado pelos autores deste trabalho (2017)

Figura 14: Bomba de aquário



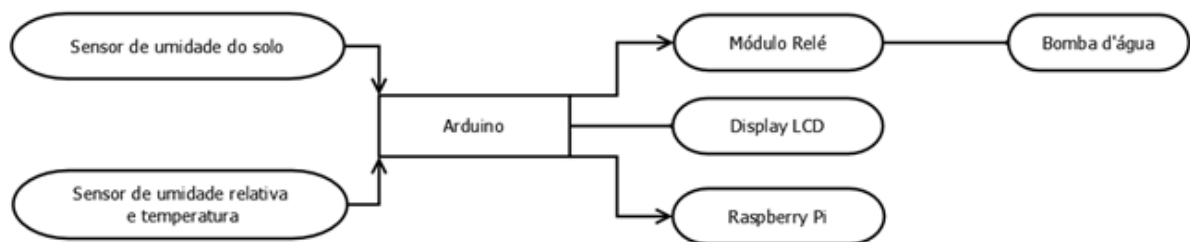
Fonte: Elaborado pelos autores deste trabalho (2016)

3. MATERIAIS E MÉTODOS

O projeto será construído de forma modular, ou seja, será utilizada “uma estratégia para construir processos ou produtos complexos a partir de pequenos subsistemas que podem ser desenvolvidos individualmente, mas que funcionam como um conjunto integrado” (TAHARA, 2009). Assim facilitando e simplificando a elaboração da proposta como um todo. Em suma, o projeto constrói-se por várias etapas que ao final serão integradas.

O hardware é formado por um Arduino UNO genérico (BlackBoard), um módulo relé, sensores de umidade do solo, do ar e temperatura, um display LCD e o minicomputador Raspberry Pi. Além desses, o projeto conta também com uma bomba d’água. A figura 15 apresenta as relações entre os componentes.

Figura 15: Integração entre os componentes utilizados



Fonte: Elaborado pelos autores deste trabalho (2017)

A proposta é utilizar a plataforma Arduino para a elaboração do protótipo. No projeto proposto, a plataforma ativará a bomba d’água quando os sensores transmitirem os dados para essa ativação, como informações que indicam que o solo está seco. Para isso o microcontrolador da placa precisa estar programado.

Após o Arduino estar programado, o sensor de umidade do solo envia dados analógicos para o Arduino. Na placa é definido se o solo está seco ou úmido de acordo com algum critério escolhido pelo usuário. Com o solo seco, um sinal é enviado ao módulo relé que fecha o circuito da bomba d’água e a ativa, permitindo a irrigação. Pois, a bomba está conectada ao NA (normal aberto) do relé.

Com base nas tabelas 2 e 3 do capítulo anterior, observou-se o intervalo de tensão entre uma temperatura/umidade e outra. Assim, fez-se uma média aritmética com os valores para

encontrar um intervalo que funcionasse como padrão entre as tensões. Depois que as informações forem tratadas, elas serão exibidas no display LCD, a fim de fornecer ao usuário informações acerca do ambiente em que as plantas se encontram.

O módulo relé serve para controlar o circuito da bomba, que funciona com 220V. O relé mantém o circuito aberto até receber o sinal digital do Arduino quando o sensor identificar que o solo está seco. Por isso, uma parte deve ser conectada ao C e outra deve ser conectada ao NA (normal aberto). A vantagem no uso de relés é que podem ser usados circuitos de baixas correntes (do Arduino) para controlar circuitos de correntes mais altas (da bomba d'água).

Em seguida, tem-se o display LCD que informa a umidade do solo, a temperatura, em graus celsius e a umidade relativa do ar. Para o controle do display é utilizada a biblioteca LiquidCrystal (no código `#include <LiquidCrystal.h>`). Em seguida define-se os pinos que serão utilizados para o funcionamento do display, seguindo o formato: `LiquidCrystal lcd (<pino RS>, <pino enable>, <pino D4>, <pino D5>, <pino D6>, <pino D7>)`. Para habilitar a escrita no display conectou-se o pino 5 (pino R/W) ao terra e para habilitar a recepção de dados conectou-se o pino 4 (pino RS) ao pino 12 do Arduino. Já o pino 6 (pino enable) habilita ou não o processamento da informação, esse pino está conectado ao pino 11 do Arduino.

Para o uso do display utiliza-se um potenciômetro para o controle do brilho do componente conectado ao pino 3. No setup do código inicializa-se o display com o comando `lcd.begin(16,2)`, os números informados são, respectivamente, a quantidade de colunas e quantidade de linhas. Com o uso da biblioteca o código torna-se mais simples.

Enfim, o Raspberry Pi recolhe as informações vindas dos sensores por uma comunicação serial com o Arduino. A partir de um código Python o minicomputador armazena os dados dos sensores em um arquivo de texto simultaneamente à hora atual, que o computador identifica pela conexão com a internet. O sistema operacional utilizado no minicomputador é um sistema Linux, que normalmente já tem a linguagem Python como parte do sistema. Além disso, o Raspberry Pi nesse projeto também tem a função de fornecer a alimentação para o Arduino.

3.1. DESENVOLVIMENTO DO PROTÓTIPO

3.1.1. Apresentação do sistema de irrigação proposto

O projeto propõe a irrigação automática, a fim de garantir o melhor aproveitamento da água se adequando as necessidades de hortaliças. Com isso, analisa-se a umidade do solo para identificar se o solo se encontra seco ou úmido garantindo uma irrigação melhor e inteligente. Além disso, a umidade do ar e temperatura também serão coletados para fins de consulta e análise de dados, tanto do usuário quanto do sistema. Nas entradas analógicas do projeto têm-se: o sensor de umidade do solo e do ar e sensor de temperatura. Esses componentes coletam os dados que, ao serem utilizados no sistema, funcionam atendendo as necessidades das plantas.

Enquanto que o módulo relé, o display LCD e o Raspberry Pi receberão as informações de saída. O módulo relé ativará ou não a bomba para a irrigação. O display LCD fornece ao usuário informações acerca do ambiente em que as plantas se encontram, apresentando a umidade do solo, do ar e a temperatura. Já o computador Raspberry Pi recebe as informações dos sensores e salva elas em um arquivo de texto juntamente com a hora em que as informações foram coletadas.

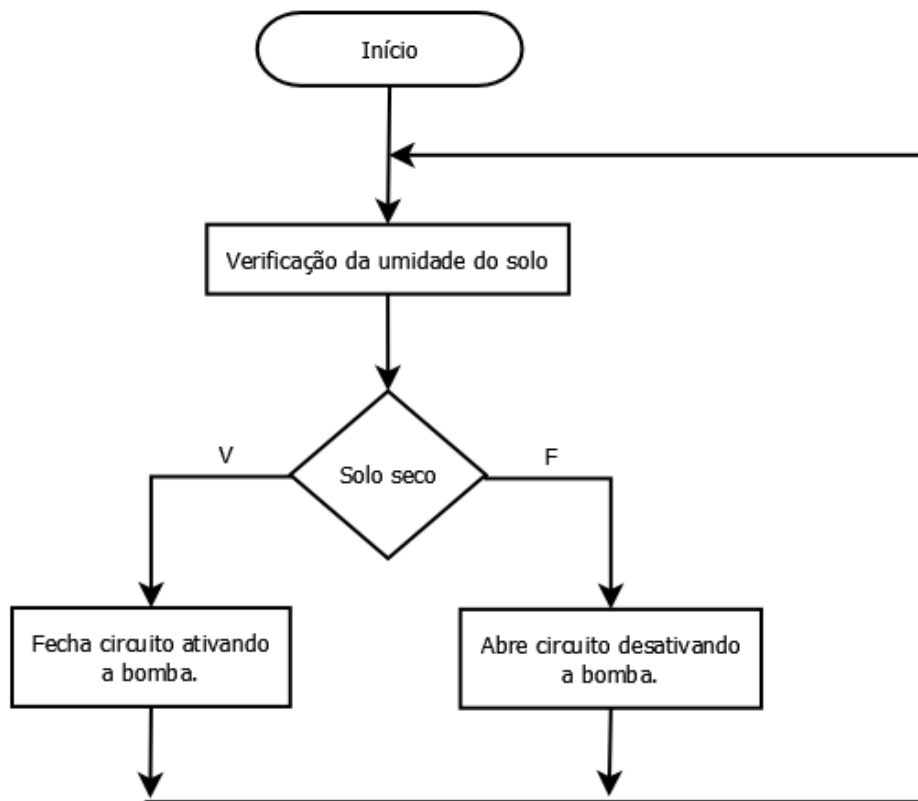
Logo, o sistema proposto consiste em duas partes. A primeira tratando da coleta de dados buscando controle da umidade do solo, possibilitando a ativação ou não da bomba para irrigação. Já a segunda parte, trata-se do uso de todos os dados coletados, estes serão exibidos no display LCD para o usuário e serão gravados em um arquivo no Raspberry Pi.

3.1.2. Sistema de irrigação

O sistema proposto tem como responsabilidade a verificação da umidade do solo a partir do sensor. Quando o sensor identificar que o solo se encontra seco o software envia o sinal para o relé que ativa a bomba d'água. Quando o solo estiver devidamente irrigado o relé desativa a bomba. Com isso, pretende-se manter níveis adequados de irrigação, suprimindo a necessidade das plantas.

A figura 16 abaixo ilustra o fluxograma da principal funcionalidade do projeto.

Figura 16: Fluxograma do sistema de irrigação



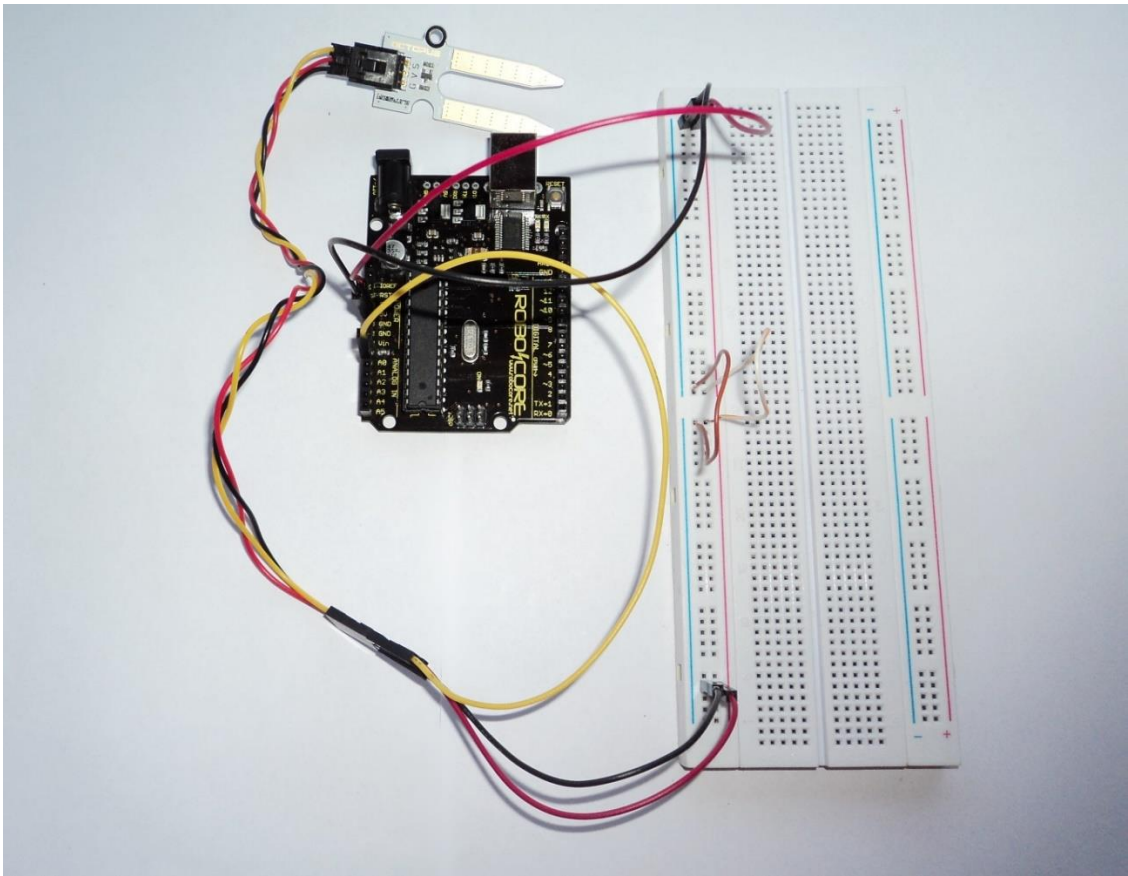
Fonte: Elaborado pelos autores deste trabalho (2016)

3.1.2.1. Montagem do sistema de irrigação

O projeto é constituído por um Arduino UNO genérico (BlackBoard), um módulo relé, sensores de umidade do solo e do ar e temperatura, um display LCD e o computador Raspberry Pi.

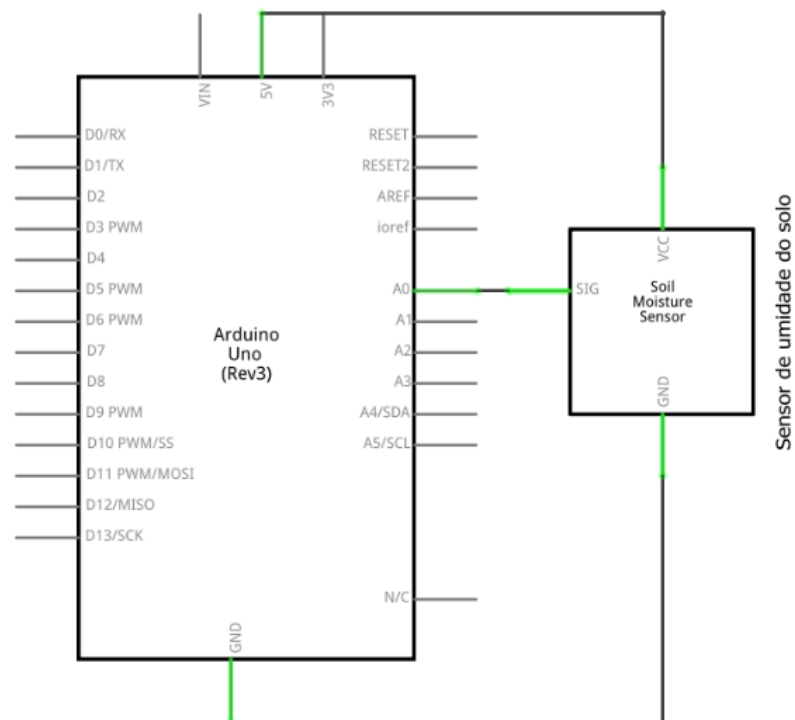
Primeiramente, é inserido no sistema o sensor de umidade do solo, pois os dados desse sensor são os principais meios de ativação da bomba para a irrigação, ao verificar as condições de umidade do solo. Apresenta três pinos: um GND, um VCC e uma saída analógica que está conectada ao pino analógico A0 do Arduino. A figura 17 e 18 ilustram essa conexão.

Figura 17: Conexão do sensor de umidade do solo com o Arduino



Fonte: Elaborado pelos autores deste trabalho (2016)

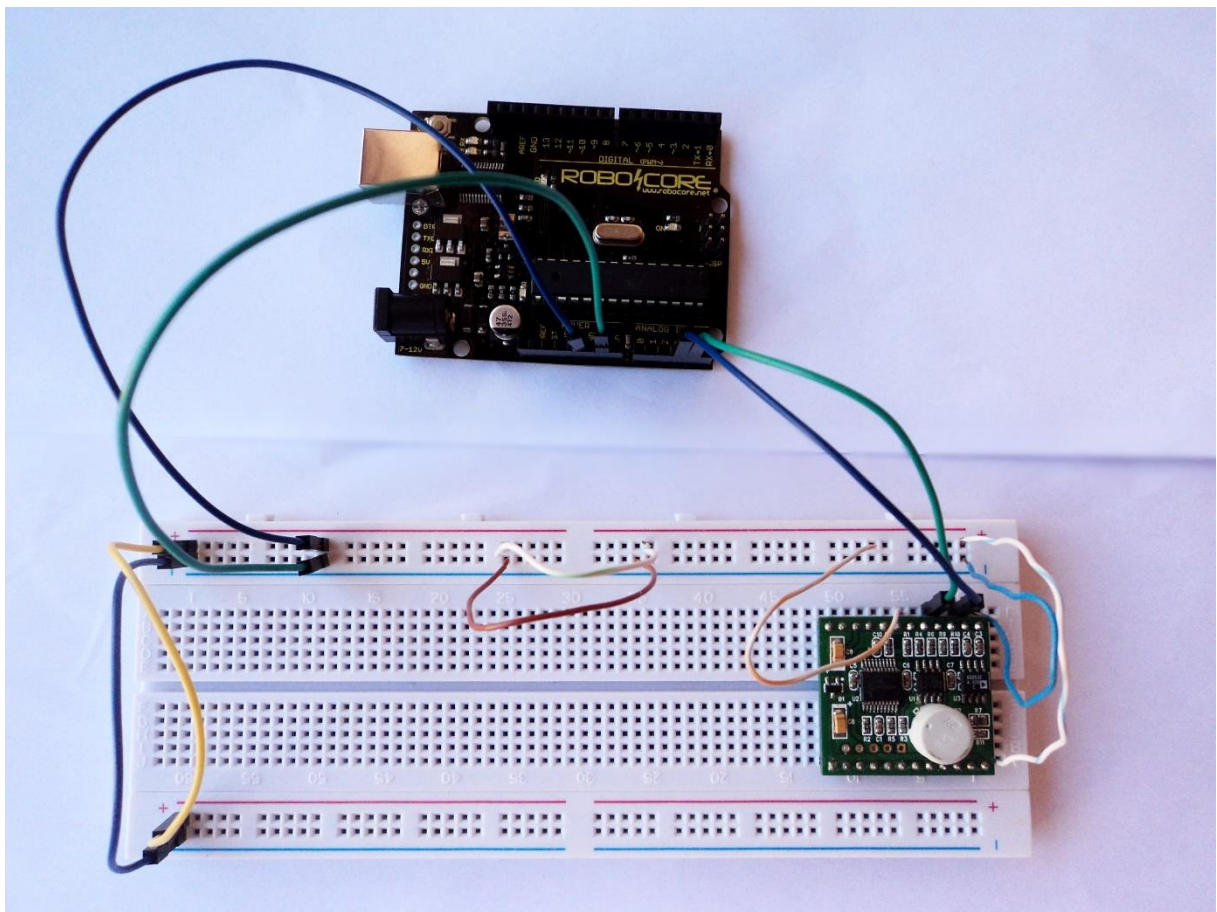
Figura 18: Exemplificação das conexões entre Arduino e sensor de umidade do solo



Fonte: Elaborado pelos autores deste trabalho (2017)

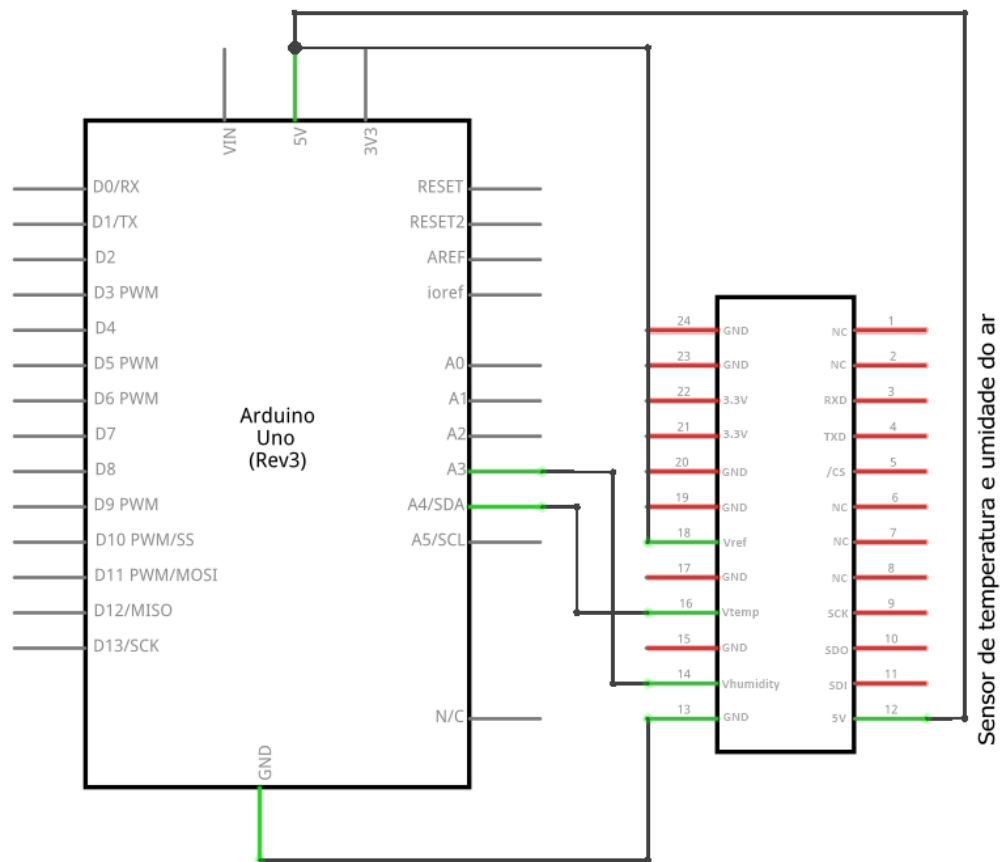
Em seguida, o sensor de temperatura e umidade do ar é incluído ao projeto, para verificar a umidade do ar e a temperatura do ambiente em que as plantas se encontram. Dos pinos existentes nesse componente, utilizou-se 5 pinos: 5V, pino 12; o GND, pino 13; a saída analógica de umidade relativa, pino 14, ligado ao pino analógico A3 do Arduino; a saída analógica de temperatura, pino 16, conectado ao pino analógico A4 do Arduino; e, por fim, o pino que servirá de referência de voltagem para os dados analógicos (Vref), pino 18, conectado ao VCC com 5V do Arduino. A figura 19 e 20 ilustram as conexões com o Arduino.

Figura 19: Conexão do sensor de umidade relativa e temperatura com o Arduino



Fonte: Elaborado pelos autores deste trabalho (2016)

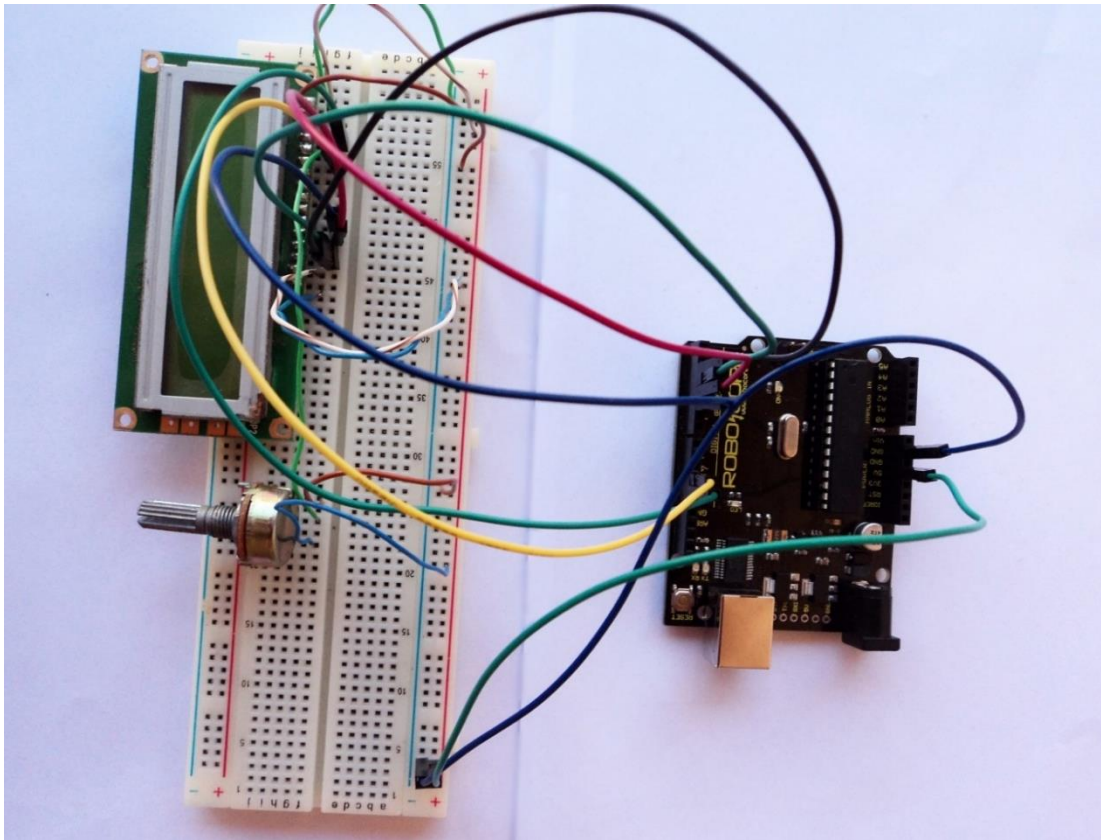
Figura 20: Exemplificação das conexões entre Arduino e sensor de umidade relativa e temperatura



Fonte: Elaborado pelos autores deste trabalho (2017)

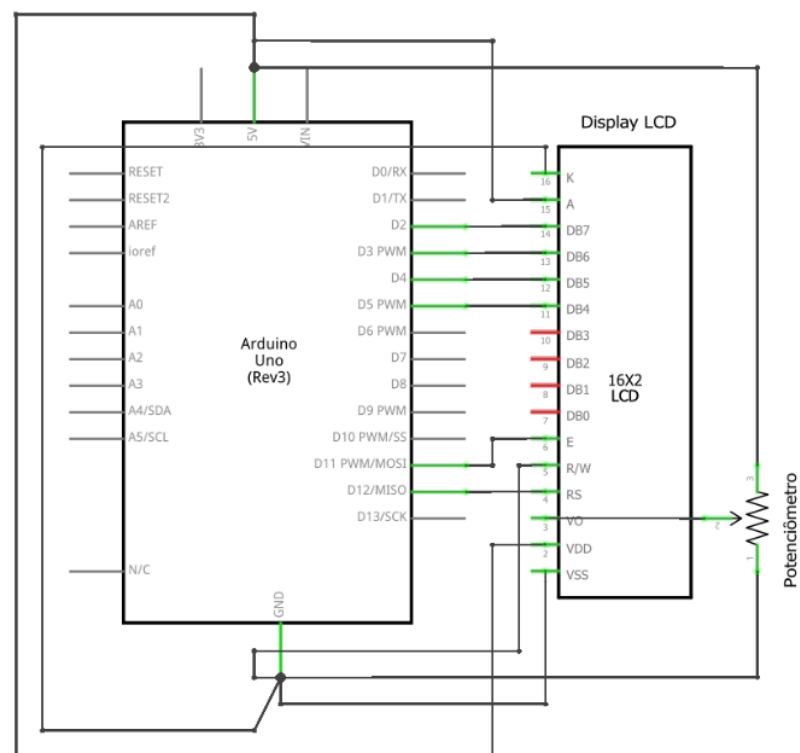
Logo após, tem-se o display LCD que informa para o usuário se o solo está seco ou não, e também a temperatura, em graus celsius, e a umidade relativa do ar. Para o uso do display utiliza-se um potenciômetro de 10K Ω ligado à 5V e ao GND. O potenciômetro tem como função o controle do brilho do display, que possui letras pretas e fundo verde. O pino central do potenciômetro conecta-se ao pino 3 do display. Para a ligação entre display e Arduino, utilizou-se seis pinos digitais do Arduino (12, 11, 5, 4, 3 e 2), VCC e GND. A figura 21 e 22 apresentam as conexões do display LCD com o Arduino.

Figura 21: Conexão do display LCD com o Arduino



Fonte: Elaborado pelos autores deste trabalho (2016)

Figura 22: Exemplificação das conexões entre Arduino e Display LCD

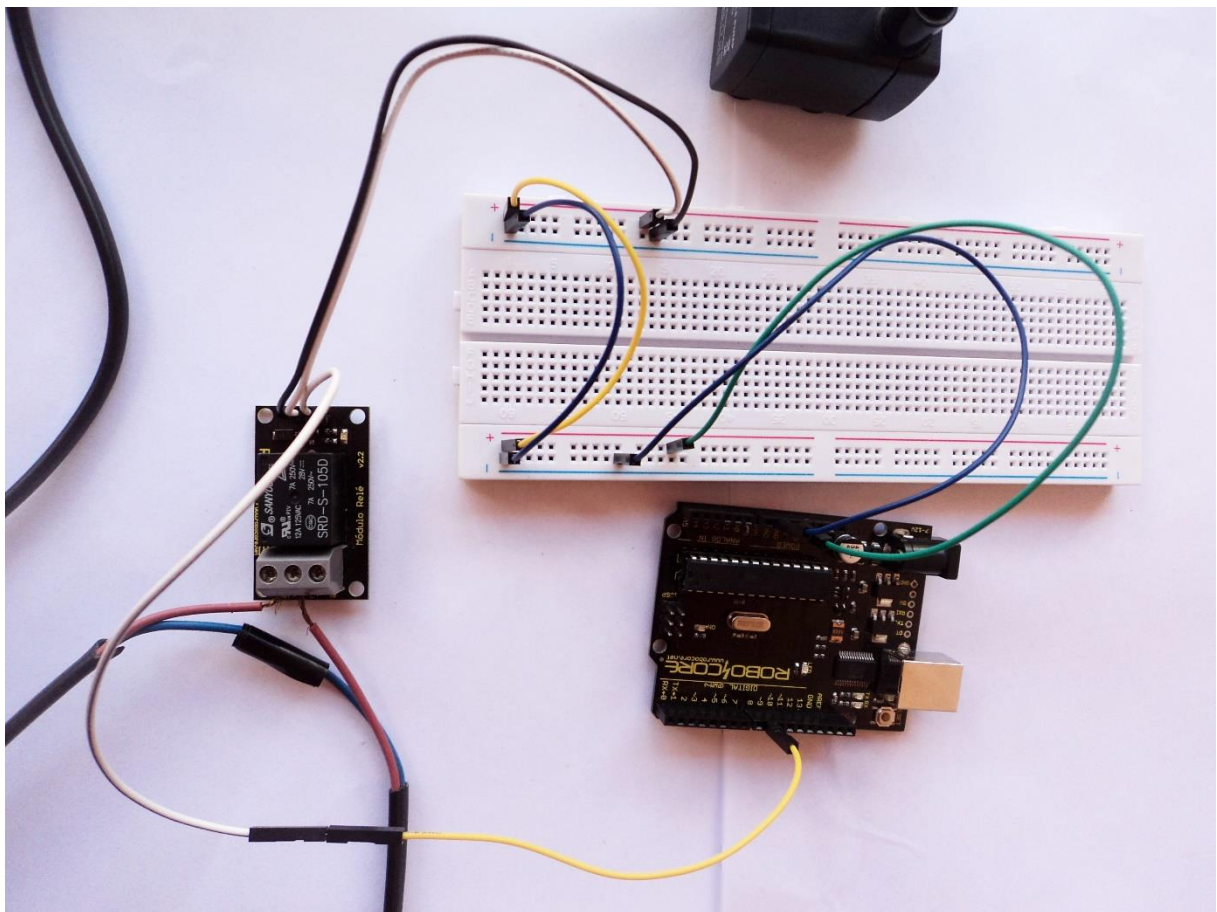


Fonte: Elaborado pelos autores deste trabalho (2017)

O módulo relé tem a função de realizar o acionamento ou desligamento da bomba d'água com base no sensor de umidade do solo e no software desenvolvido.

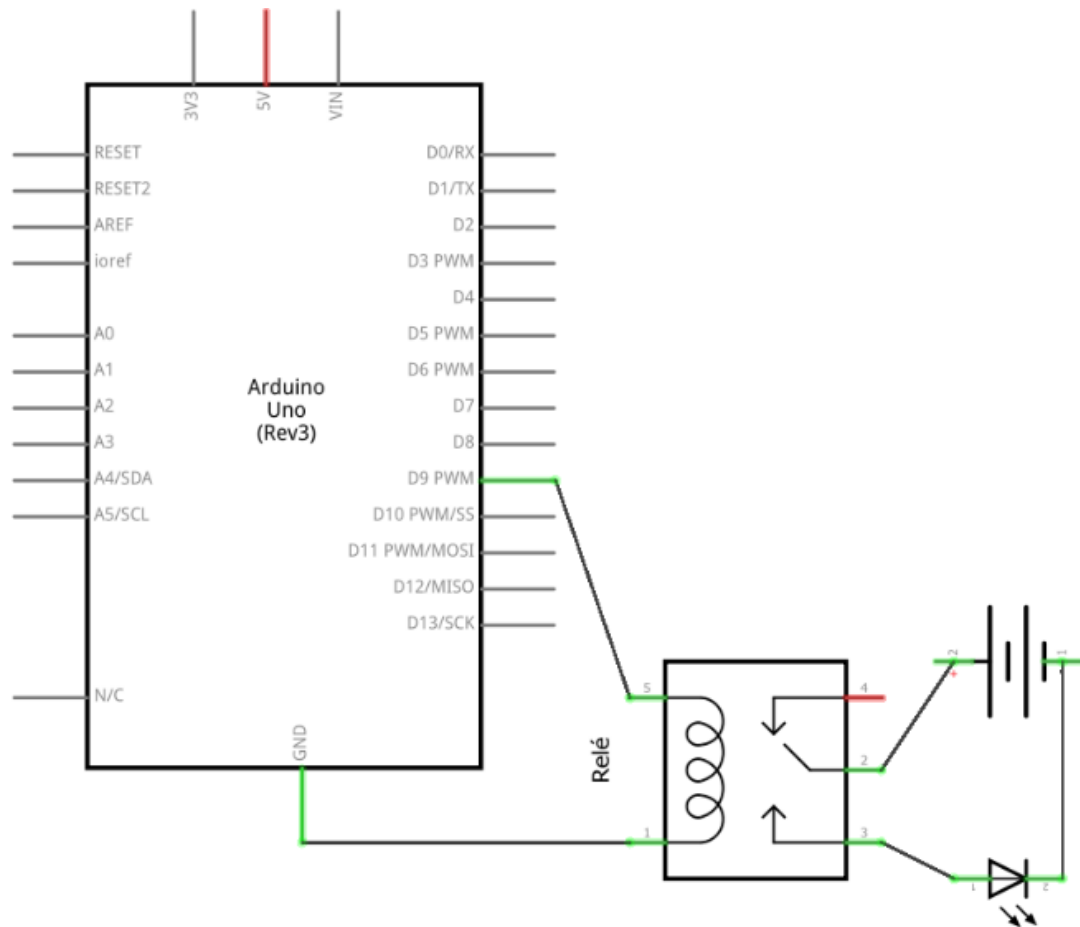
O módulo relé tem três pinos: o GND, o de alimentação VCC e a entrada que obtém os comandos do Arduino, que se encontra conectada ao pino digital 9 do Arduino. E, para abrir e fechar o circuito da bomba d'água, o fio fase da bomba está dividido com parte conectada ao C e outra parte ao NA (normal aberto). Para, então, o circuito fechar ao receber sinal do Arduino. A figura 23 e 24 mostram as conexões do módulo relé.

Figura 23: Conexão do módulo relé com o Arduino



Fonte: Elaborado pelos autores deste trabalho (2016)

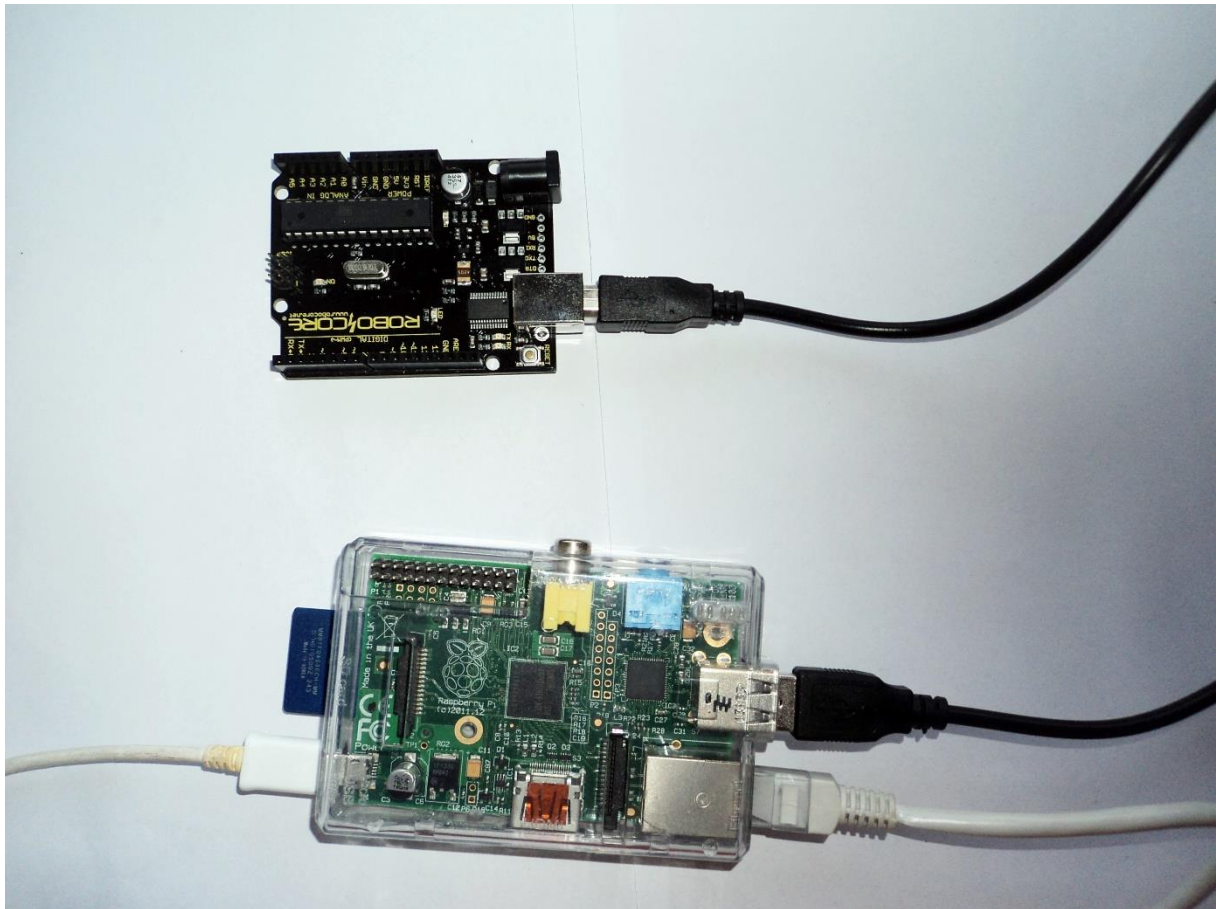
Figura 24: Exemplificação das conexões entre Arduino e Display LCD



Fonte: Elaborado pelos autores deste trabalho (2017)

Ao final, conecta-se o Raspberry Pi ao Arduino. As informações coletadas pelos sensores utilizados são enviadas para esse minicomputador e ele grava essas informações em um arquivo de texto. Assim, possibilitando o armazenamento de dados para uma possível consulta. Ele se conecta ao Arduino por uma comunicação serial pela conexão USB e, assim, ele também alimenta o Arduino. A figura 25 apresenta o Raspberry Pi e o Arduino.

Figura 25: Conexão do Raspberry Pi com o Arduino



Fonte: Elaborado pelos autores deste trabalho (2016)

3.1.2.2. Desenvolvimento do software

Em todo o projeto utilizou-se a plataforma Arduino, menos na parte do Raspberry Pi. O Raspberry Pi utiliza o sistema operacional Linux e a linguagem de programação usada para essa parte do projeto é Python. Já na plataforma Arduino o software é construído com linguagem de programação C/C++. Nessa seção serão apresentadas partes específicas importantes para o projeto, tanto do sketch quanto do código Python.

3.1.2.2.1. Código do Arduino

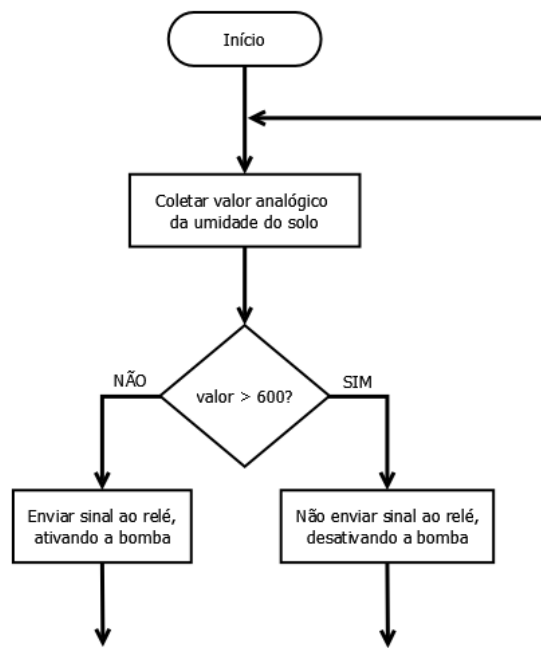
De início são incluídas as bibliotecas necessárias e são definidas algumas variáveis utilizadas. No projeto tem-se a diretiva `#include <LiquidCrystal.h>`, que é a biblioteca que auxilia na manipulação do display LCD. Ademais, os pinos utilizados pelo LCD também são definidos anteriormente à estrutura `setup()`.

Na estrutura void setup () do sketch é definido o pino de saída que enviará sinal de ativação para o módulo relé e o display lcd é inicializado. Inicializa-se também a comunicação serial, com taxa transmissão 9600bps.

Os valores de umidade do solo, do ar e a temperatura serão definidos no void loop (), a estrutura que executará o código repetitivamente enquanto o Arduino estiver ativo. Desse modo, esses valores serão identificados a todo momento.

O valor do sensor da umidade do solo já é utilizado assim que identificado. Pois, com a faixa 0 – 1023 do analógico já é possível identificar se o solo está seco ou úmido. Então, com simples blocos condicionais pode-se definir se a bomba d'água será ativada ou não. Em que quanto mais perto do 0 mais seco e quanto mais perto do 1023 mais úmido. No projeto definiu-se que quando menor e igual que 600 o solo encontra-se seco e quando maior que 600 o solo está úmido. A figura 26 e 27 explanam esse processo.

Figura 26: Fluxograma exemplificando parte da utilização dos dados do sensor de umidade do solo



Fonte: Elaborado pelos autores deste trabalho (2017)

Figura 27: Pseudocódigo exemplificando parte da utilização dos dados do sensor de umidade do solo

```

algoritmo "sensor umidade do solo"
// Função :
// Autor :
// Data : 28/01/2017
// Seção de Declarações
var
umidSolValue : real
sensorUmidSol : real
inicio
// Seção de Comandos
ESCREVA ("Informe o valor do umidade do solo:")
LEIA (sensorUmidSol)
umidSolValue := sensorUmidSol
SE umidSolValue < 600 ENTÃO
    ESCRIVA ("Ativando bomba")
SENÃO
    ESCRIVA("Desativando bomba")
FIMSE

finalgoritmo

```

Fonte: Elaborado pelos autores deste trabalho (2017)

Entretanto, para coletar a temperatura e a umidade relativa, é necessário um processo mais elaborado. Primeiramente, ao receber um valor analógico entre 0 – 1023, converte-se o valor identificado para um valor equivalente a uma tensão de 0 – 4.95V. No código utilizou-se algo semelhante a uma regra de três ($\text{float voltUmidAr} = \text{umidArValue} * (4.95 / 1023.0)$;). Existe um exemplo na IDE chamado ReadAnalogVoltage que segue a mesma lógica. Além dessa forma, a função `map()`, do Arduino, poderia ter sido utilizada.

Logo após a obtenção da tensão referente à temperatura e umidade relativa, utilizou-se a função `map()` para definir a conversão do valor de tensão para o número esperado. Para a conversão, tem-se como base a tabela apresenta no capítulo 3. Por ela, é possível conhecer qual temperatura ou umidade é referente a uma dada tensão.

A função `map()` é no formato `map(value, fromLow, fromHigh, toLow, toHigh)`. Escolheu-se valores de referência com base na análise das tabelas 3 e 4 do capítulo 3 e elaborou-se as funções para temperatura ($\text{float temp} = \text{map}(\text{voltTemp}, 1.52, 3.62, 0, 50)$;) e umidade ($\text{float umidAr} = \text{map}(\text{voltUmidAr}, 0.52, 3.88, 10, 80)$;).

Em seguida, os valores devem ser enviados ao Raspberry Pi. Inicialmente, um bloco condicional verifica se a comunicação serial está ativa. Então, lê-se a mensagem recebida pela

comunicação serial com `Serial.read()`. Se a mensagem recebida for a esperada enviada pelo Raspberry Pi, então o Arduino transmite os valores dos sensores para o minicomputador, com `Serial.print()`.

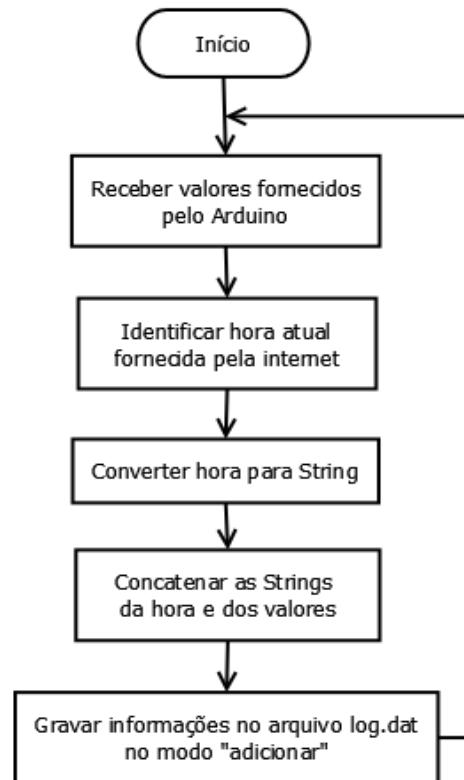
Por fim, a umidade do solo e do ar e a temperatura devem ser exibidas pelo display LCD. Com a biblioteca a programação fica bastante simples. Primeiro, para a mensagem aparecer no display é necessário posicionar a coluna e a linha da mensagem informada com o comando `lcd.setCursor(coluna, linha)`. Em seguida, imprime a String na tela com `lcd.print("")`. Logo após, é utilizado o comando `lcd.scrollDisplayLeft()`, para mover os caracteres para a esquerda. Com ele é utilizado um laço que vai de 0 a 40, que é a quantidade caracteres que vão ser movidas. Ao final, o comando `lcd.clear()` limpa o display para uma nova mensagem.

O código completo encontra-se no apêndice A.

3.1.2.2.2. Código Python

Inicialmente inclui-se as bibliotecas que serão importantes. Logo em seguida, o scheduler, biblioteca que permite que uma tarefa seja executada periodicamente, `BlockingScheduler()` é passado para uma variável. Ele indica que o scheduler é a única coisa que irá rodar no programa. Então, com a variável, é utilizado o método `add_job()` para definir a tarefa que será executada, se é periódica e o tempo do intervalo. Posteriormente, o scheduler é iniciado por `start()`. Todo o processo é ilustrado nas figuras 28 e 29.

Figura 28: Fluxograma exemplificando os processos principais do código Python



Fonte: Elaborado pelos autores deste trabalho (2017)

Figura 29: Pseudocódigo exemplificando os processos principais do código Python

```

algoritmo "Escrever no arquivo"
// Função :
// Autor :
// Data : 29/01/2017
// Seção de Declarações
var
valoresArduin : caractere
strgHora : caractere
strgArq : caractere
inicio
// Seção de Comandos
ESCREVA ("Forneça os valores dos sensores:")
LEIA (valoresArduin)
ESCREVA ("Forneça a hora atual:")
LEIA (strgHora)
strgArq := strgHora + "      " + valoresArduin
ESCREVAL ("Adicionando ao arquivo:", strgArq)
ESCREVA ("Repitir operação a cada 30 minutos")
finalgoritmo
  
```

Fonte: Elaborado pelos autores deste trabalho (2017)

A tarefa que o scheduler deve realizar é a execução da função `some_job()`. Nessa função, informações dos sensores vindas de outra função (`quisicao()`) são recebidas e armazenadas em uma variável. Sendo assim, a hora atual é identificada e, após o tipo ser passado para String, os valores dos sensores e o horário atual são concatenados e transferidos para outra função.

A função que recebe a String chama-se `escrever_arquivo(BUFFER)`. Nela o arquivo que armazenará as informações é aberto no modo “adicionar”, que não sobrescreve a informação anterior. Então, a informação é escrita e depois fecha-se o arquivo.

Entretanto, antes do arquivo ser escrito foi preciso realizar a comunicação serial com o Arduino. A função `quisicao()`, mencionada anteriormente, realiza a comunicação e recebe os dados necessários. Primeiro, iniciou-se a conexão serial com o método `Serial(porta serial, taxa de transmissão)`. Logo então, o caracter ‘l’ é enviado e em um bloco condicional, no Arduino, os valores esperados são transmitidos para o Raspberry Pi. Este, por sua vez, lê a informação, retorna ela e fecha a comunicação serial. O retorno é utilizado na função `some_job()`.

O código encontra-se no apêndice B.

4. RESULTADOS OBTIDOS

Nesta seção apresenta-se os resultados encontrados na implementação do sistema de irrigação sugerido em ambiente similar aos ambientes propostos. Com isso, pretende-se avaliar o funcionamento do projeto e realizar análises acerca dos resultados obtidos. Para, enfim, garantir que a proposta funciona como esperado.

O projeto tem a função de realizar a irrigação automática dependendo do nível de umidade do solo, que é identificado pelo sensor. Mas além disso, ele também realiza a coleta de dados da umidade do solo, do ar e da temperatura. Esses dados são coletados e armazenados em um arquivo pelo Raspberry Pi a partir de um código na linguagem Python. A tabela abaixo apresenta os materiais utilizados na construção do protótipo para implementação e as funções que cada componente deve exercer:

Tabela 5: Funções dos elementos do projeto

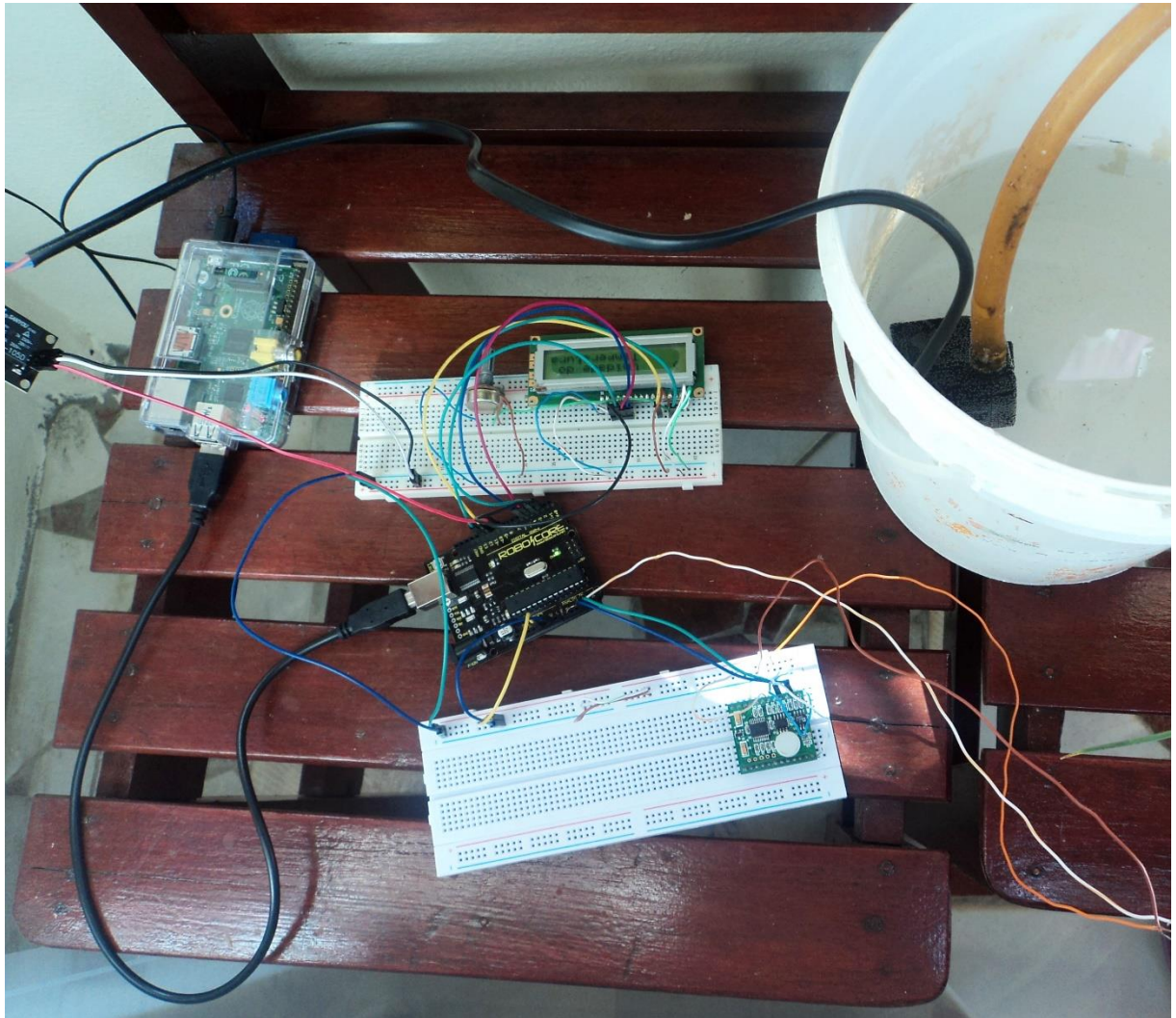
COMPONENTES	FUNÇÃO
Arduino BlackBoard	Controle de todo o projeto
Sensor de umidade do solo	Medir a umidade presente no solo das plantas
Sensor de umidade relativa e temperatura	Medir a umidade relativa e temperatura do ambiente das plantas
Display LCD	Exibir as informações relevantes ao usuário acerca do cultivo
Módulo Relé	Fechar ou abrir o circuito de ativação da bomba de aquário
Raspberry Pi	Realizar a comunicação serial com o Arduino para receber informações
Bomba de aquário	Realizar a irrigação quando o solo estiver seco

Fonte: Elaborado pelo autor deste trabalho (2016)

Para a realização da implementação, ligou-se o Raspberry Pi e a bomba de aquário. Entretanto, a bomba só deve ligar de fato, ao fechar o circuito, quando o relé recebe o sinal do

Arduino. Então, após a aquisição de uma pequena planta, verificou-se a operação do projeto elaborado. A figura 30 juntamente com a figura 31 apresentam o protótipo montado e em funcionamento.

Figura 30: Protótipo montado em ambiente de teste



Fonte: Elaborado pelo autor deste trabalho (2016)

Figura 31: Sensor de umidade do solo em teste

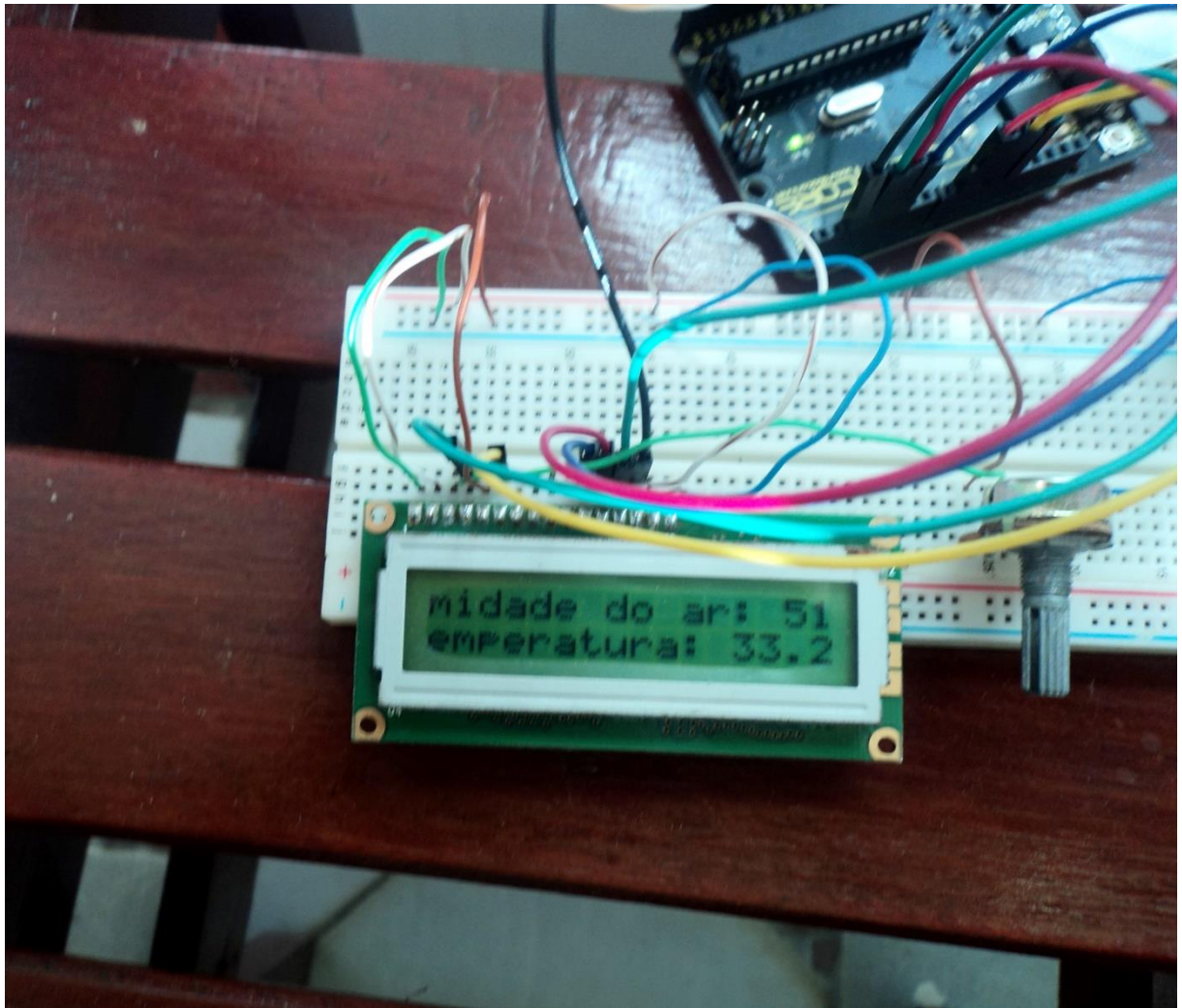


Fonte: Elaborado pelo autor deste trabalho (2016)

Após a experimentação, observa-se que o sistema de irrigação tem grande potencial na economia de água. Pois, quando o solo está seco, ele fica rapidamente úmido pela velocidade que a água é impulsionada pela bomba utilizada. Já que esta, por sua vez, apresenta vazão de 150L/h. E o solo, dependendo da extensão do solo irrigado, demora para chegar ao seco. Ademais, analisou-se que o valor anteriormente escolhido para solo seco (menor que 300) não seria viável. Porque, quando o sensor identificasse um valor abaixo de 300, o solo já se encontraria muito seco. Logo, isso poderia ser prejudicial às plantas. Portanto, com base nas observações da experimentação, assumiu-se um novo valor para solo seco: menor que 600. Mas, é importante lembrar que a quantidade de umidade ideal será diferente para cada espécie de planta utilizada. Por isso, cada usuário pode alterar os valores dependendo das necessidades encontradas.

O display LCD também tem o papel essencial de expor as informações do ambiente ao usuário. Este consegue concluir seu papel, como ilustra a figura 32 abaixo.

Figura 32: Display LCD rolando caracteres informativos



Fonte: Elaborado pelo autor deste trabalho (2016)

Por fim, avaliou-se a coleta de dados pela comunicação serial com o Raspberry Pi. No teste realizado, a função `add_job()`, do código Python, executa o método `some_job()` a cada 4 segundos. Além disso, apenas o sensor de umidade de solo foi escolhido para a verificação do funcionamento da coleta e armazenamento de dados, pois o teste realizou-se separadamente aos testes expostos anteriormente. Porém, é evidente que com os outros dados também funciona. Abaixo a figura 33 ilustra o resultado encontrado:

Figura 33: Arquivo log.dat com as informações coletadas

GNU nano 2.2.6		File: log.dat
17:48:46	291	
17:48:50	291	

Fonte: Elaborado pelo autor deste trabalho (2016)

Com isso, a efetuação dos testes conclui o objetivo de validar o projeto nos quesitos de sistema de irrigação automática e na coleta e armazenamento de dados.

A tabela 6 apresenta os preços dos componentes utilizados

Tabela 6: Orçamento do projeto

COMPONENTES	PREÇO
Arduino BlackBoard	R\$ 85,00
Sensor de umidade do solo	R\$ 7,90 (com base no preço em dólar)
Sensor de umidade relativa e temperatura	R\$ 53,73 (com base no preço em dólar)
Módulo relé	R\$ 12,90
Módulo display LCD	R\$ 18,90
Raspberry Pi	R\$ 299,90 (por ser importado)
Bomba de aquário SP-500	R\$ 35,00
Potenciômetro	R\$ 2,40
TOTAL	R\$ 515,73

Fonte: Elaborado pelos autores deste trabalho

5. CONSIDERAÇÕES FINAIS

Com a agricultura sendo uma das principais fontes de renda e sustento das pessoas que a praticam, o sistema propõe um auxílio aos agricultores familiares na irrigação de plantas de pequeno porte. Assim, o projeto pretende a elaboração de um sistema de irrigação simples e de fácil manuseio. Além disso, o sistema ainda apresenta a função de coleta e armazenamento de dados para futuras análises e acompanhamentos do ambiente das plantas.

Para o controle das tarefas do Arduino, desenvolveu-se um software em C/C++. O circuito do projeto possui sensores que verificam a umidade do solo, umidade relativa e temperatura. Com isso, pretende-se analisar como o ambiente das plantas se encontram. Além disso, essas informações serão transmitidas por meio de comunicação serial para o minicomputador Raspberry Pi para, então, serem armazenadas em um arquivo. Ademais, os dados dos sensores devem ser exibidos pelo display LCD. Pois, é muito importante que os agricultores conheçam a situação do local de seu cultivo.

Então, o sistema passou por situações de teste para verificação do funcionamento dos sensores, display LCD e Raspberry Pi. A partir disso, montou-se o protótipo para realização dos testes em ambiente de simulação de uma plantação. Então, tornou-se possível verificar o controle do Arduino sobre os sensores e sobre o módulo relé e, consequentemente, o ativamento/desligamento da bomba de aquário. Simultaneamente, observou-se a comunicação serial entre Arduino e Raspberry Pi, além de o minicomputador também possuir a função de fonte de alimentação.

Nos testes notou-se a eficiência do sistema na economia da água utilizada. Pois, com a velocidade da água, o solo fica rapidamente úmido e demora para ficar seco (dependendo da extensão irrigada). Os sensores também funcionam corretamente, o que permite o funcionamento do sistema e a observação do lugar da plantação. Logo, a observação dos dados torna-se possível pela exibição de caracteres do módulo display LCD e pela leitura do arquivo na memória secundária do minicomputador. O único problema apresentado foi que o Raspberry Pi precisa de conexão com a internet para conhecer o horário atual e registrar juntamente aos dados dos sensores.

Além disso, os gastos com o projeto são altos, já que alguns componentes são importados. Sendo assim, não é um projeto de baixo custo e de fácil acessibilidade. Então, a proposta pode acabar sendo inviável dependendo das necessidades do usuário.

Com isso, concluiu-se o objetivo de realizar a automatização do processo de irrigação das plantas com a plataforma Arduino de forma simples e prática. Além disso, a função de

monitoramento e registro das características do ambiente para fim de estudos e conhecimento realizou-se com sucesso. O projeto elaborado consegue realizar a irrigação no tempo ideal, sem desperdício de água e realiza as funções essenciais sem causar danos às plantas. Ademais, as funcionalidades propostas ocorreram como planejado, possibilitando a irrigação mais prática, eficaz e despreocupada. Garantindo, assim, um benefício aos agricultores familiares e suas culturas.

5.1. TRABALHOS FUTUROS

Apesar do sistema elaborado corresponder com as expectativas e realizar a função da irrigação automática e monitoramento de dados, ainda há o que melhorar. Segue abaixo algumas aplicações que planejava-se implementar e outras que são opções para o aprimoramento do projeto:

- Realizar testes monitorados e intensos com plantas, a fim de determinar e estudar os parâmetros para o melhor desenvolvimento das plantas.
- Empregar painel solar para fornecimento de energia ao projeto, utilizando assim energia sustentável e renovável.
- Utilizar um módulo RTC com o Raspberry Pi para garantir que o minicomputador atualize as informações de data e hora sem depender de rede ou Internet.
- Aplicar o sistema com maiores quantidades de sensores, a fim de analisar o funcionamento do projeto em grande escala ao atender as necessidades de muitas plantas.
- Desenvolver um aplicativo móvel ou para desktop para maior controle e monitoramento do usuário e possibilitar a ativação da irrigação pelo próprio agricultor pelo seu aparelho.
- Utilizar um módulo cartão micro SD com um módulo RTC como substituição à função do Raspberry Pi, pois, como o minicomputador é importado, o uso dos módulos torna o custo com o projeto mais acessível.
- Receber informações do sensor de umidade do ar e temperatura por dados digitais, para maior precisão da informação.

REFERÊNCIAS

AFE – APRENDA FÁCIL EDITORA. **Umidade relativa do ar interfere na escolha de áreas para plantio.** Disponível em: <<http://www.afe.com.br/noticia/1123/umidade-relativa-do-ar-interfere-na-escolha-de-areas-para-plantio>>. Acesso em 20 de novembro de 2016.

ARDUINO.CC. **Arduino/Genuino UNO.** Disponível em: <<https://www.arduino.cc/en/Main/ArduinoBoardUno>>. Acesso em 20 de novembro de 2016.

ARDUINO.CC. **Arduino - Introduction.** Disponível em: <<https://www.arduino.cc/en/Guide/Introduction>>. Acesso em 29 de junho de 2016.

BARBACENA, I. L., FLEURY, C. A. **Display LCD.** Disponível em: <<ftp://ftp.dca.fee.unicamp.br/pub/docs/ea079/complementos/Lcd.pdf>>. Acesso em 23 de novembro de 2016

BORGES, L. E. **Python para Desenvolvedores.** 2ª ed. Rio de Janeiro, RJ: Edição do Autor, 2010.

BRAGA, N. C. **Medidor de umidade.** Disponível em: <<http://www.newtoncbraga.com.br/index.php/mini-projetos/167-bancada-e-instrumentacao/4404-min053>>. Acesso em 10 de novembro de 2016.

BRAGA, N. C. **Tudo Sobre Relés (livro completo).** Disponível em: <<http://www.newtoncbraga.com.br/index.php/como-funciona/597-como-funcionam-os-reles?showall=&limitstart=>>>. Acesso em 29 de junho de 2016.

BURKE, A. **Como uma bomba de aquário funciona?** Disponível em: <http://www.ehow.com.br/bomba-aquario-funciona-como_5822/>. Acesso em 20 de novembro de 2016.

COLUSSI, J. **Produtos sem agrotóxicos, alimentos orgânicos buscam mais espaço na mesa do consumidor.** Disponível em: <<http://zh.clicrbs.com.br/rs/noticias/campo-e>>

lavoura/noticia/2015/06/produzidos-sem-agrotoxicos-alimentos-organicos-buscam-mais-espaco-na-mesa-do-consumidor-4772643.html>. Acesso em 01 de julho de 2016.

COSTA, A. R. **As relações hídricas das plantas vasculares**. Disponível em: <<http://www.angelfire.com/ar3/alexcosta0/RelHid/Rhw7.htm>>. Acesso em 28 de janeiro de 2017.

CULTURA MIX. **Como Fazer Irrigação Por Gotejamento**. Disponível em: <<http://flores.culturamix.com/dicas/como-fazer-irrigacao-por-gotejamento>>. Acesso em 10 de novembro de 2016.

DEBIAN. **O que é GNU/Linux?** Disponível em: <https://www.debian.org/releases/etch/arm/ch01s02.html.pt_BR>. Acesso em 23 de novembro de 2016.

JAD AQUARIUM CO., LTD. **Instruction Book of Submersible Pump for Aquarium**. Raoping Guangdong, 2015. 6p.

LIMA, C. B. de. **Os poderosos μ controladores AVR**. Florianópolis, SC, 2009. 143p.

LIMA, F. **Água na dose certa**. Disponível em: <<http://revistagloborural.globo.com/Revista/Common/0,,EMI332287-18283,00-AGUA+NA+DOSE+CERTA.html>>. Acesso em 10 de novembro de 2016.

MOLLO, L. **Efeito da temperatura no crescimento, no conteúdo e na composição de carboidratos não-estruturais de plantas de *Alcantarea imperialis* (Carrière) Harms (Bromeliaceae) cultivadas in vitro**. 2009. 90 f. Dissertação (Mestrado em Biodiversidade Vegetal e Meio Ambiente) - Instituto de Botânica da Secretaria de Estado do Meio Ambiente, São Paulo. 2009.

O TEMPO. **Hortas em casa são tendência e podem refletir uma nova maneira de pensar a alimentação**. Disponível em: <<http://www.otempo.com.br/cidades/hortas-em-casa-s%C3%A3o-tend%C3%Aancia-e-podem-refletir-uma-nova-maneira-de-pensar-a-alimenta%C3%A7%C3%A3o-1.934464>>. Acesso em 24 de junho de 2016.

PIMENTEL, C. **A relação da planta com a água**. Seropédica, RJ: Edur, 2004. 191 p.

PIRES, R. C. et al. Agricultura Irrigada. **Revista Tecnologia & Inovação Agropecuária**, São Paulo, v. 1, n. , p. 98-111, jun. 2008.

RASPBERRY PI BRA. **Conhecendo o Raspberry Pi Modelo B – Especificações**. Disponível em: <<http://raspberrypibra.com/conhecendo-o-raspberry-pi-modelo-b-especificacoes-336.html>>. Acesso em 23 de novembro de 2016.

RASPBERRY PI BRA. **O que é Raspberry Pi?**. Disponível em: <<http://raspberrypibra.com/o-que-e-raspberry-pi-4.html>>. Acesso em 23 de novembro de 2016.

RASPBERRY PI BRA. **Todos os modelos de Raspberry Pi**. Disponível em: <<http://raspberrypibra.com/todos-os-modelos-de-raspberry-pi-921.html>>. Acesso em 23 de novembro de 2016.

RIBEIRO et al. **Estresse por altas temperaturas em trigo**: impacto no desenvolvimento e mecanismos de tolerância. R. Bras. Agrociência, Pelotas, v.18 n. 2-4, p.133-142, abr-jun, 2012.

ROSSI, M. **O “alarmante” uso de agrotóxicos no Brasil atinge 70% dos alimentos**. Disponível em: <http://brasil.elpais.com/brasil/2015/04/29/politica/1430321822_851653.html>. Acesso em 01 de julho de 2016.

SANTOS, D. **Vantagens e desvantagens dos principais tipos de irrigação**. Disponível em: <<https://www.agrosmart.com.br/blog/vantagens-tipos-de-irrigacao/>>. Acesso em 10 de novembro de 2016.

SANYOU RELAY. **Miniature power relay/ SRD-series**. Richmond, B.C, 2011. 3p.

SOUZA, F. **Arduino UNO**. Disponível em: <<https://www.embarcados.com.br/arduino-uno/>>. Acesso em 20 de novembro de 2016.

SOUZA, F. **Introdução ao Arduino.** Disponível em: <http://www.embarcados.com.br/arduino-primeiros-passos/>. Acesso em 29 de junho de 2016.

SUHOGUSOFF, V. G. et al. Influência da temperatura e da umidade relativa do ar e determinação do estado trófico em *Tillandsia stricta* Lindl, (Bromeliaceae), ocorrente no Parque Estadual da Ilha Anchieta, Ubatuba-SP. **O Mundo da Saúde São Paulo**, São Paulo, v. 32, n. 3, p. 339-346, jul/set. 2008.

SURE ELECTRONICS. **Temperature and Relative Humidity Sensor Module User's Guide.** Nanjing, 2009. 14p.

TAHARA, S. **Produto Modular.** Disponível em: <http://www.portaldeconhecimentos.org.br/index.php/por/content/view/full/9676>. Acesso em 29 de junho de 2016.

THE COUNCIL FOR AGRICULTURAL SCIENCE AND TECHNOLOGY. Irrigation Systems for Every Applications. Tradução de Fernando Braz Tangerino Hernandez (1996). **Agribusiness Worldwide**, Iowa, v. 11, n. 6, p. 20-30, 1989.

THOMSEN, A. **Controlando um LCD 16x2 com Arduino.** Disponível em: <http://blog.filipeflop.com/display/controlando-um-lcd-16x2-com-arduino.html>. Acesso em 9 de setembro de 2016.

THOMSEN, A. **Primeiros passos com o Raspberry Pi.** Disponível em: <http://blog.filipeflop.com/embarcados/tutorial-raspberry-pi-linux.html>. Acesso em 23 de novembro de 2016.

UPTON, E.; HALFACREE, G. **Raspberry Pi® User Guide.** Chichester, 2012. 262p.

VIVA O LINUX. **O que é GNU/Linux.** Disponível em: <https://www.vivaolinux.com.br/linux/>. Acesso em 23 de novembro de 2016

ZUCARELI, V. et al. Fotoperíodo, temperatura e reguladores vegetais na germinação de sementes de *Passiflora cincinnata* Mast. **Revista Brasileira de Sementes**, Londrina-PR, v. 31, n. 3, p. 106-114, abr. 2009.

APÊNDICE A – Código do Arduino

```
//inicializando biblioteca
#include <LiquidCrystal.h>

//identificação de pinos
#define sensorUmidSol A0
#define sensorUmidAr A3
#define sensorTemp A4

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

//definindo e inicializando variáveis
float umidSolValue = 0;
int umidArValue = 0;
int tempValue = 0;
int mensagem = 0;
char charBuf [6];

void setup() {
  pinMode(9, OUTPUT);
  lcd.begin(16, 2);
  Serial.begin(9600);

  Serial.println("Setup finalizado, aguardando comando.");
}

void loop() {

  //leitura de sensores
  umidSolValue = analogRead(sensorUmidSol);
  umidArValue = analogRead(sensorUmidAr);
  tempValue = analogRead(sensorTemp);
```

```

//mudança de valor analógico para numeração convencional
float voltUmidAr = umidArValue * (4.95 / 1023.0);
float umidAr = map(voltUmidAr, 0.52, 3.88, 10, 80);

float voltTemp = tempValue * (4.95 / 1023.0);
float temp = map(voltTemp, 1.52, 3.62, 0, 50);

//converter float para String
String umid_sol_strg = dtostrf(umidSolValue, 6, 2, charBuf);
String umid_ar_strg = dtostrf(umidAr, 6, 2, charBuf);
String temp_strg = dtostrf(temp, 6, 2, charBuf);

if (Serial.available() > 0) {
    mensagem = Serial.read();

    if ( mensagem == 'I' ) {
        Serial.println(umid_sol_strg + "\t" + umid_ar_strg + "\t" + temp_strg);
    } //if para enviar uma mensagem ao receber 'I'

} //if para verificar se a comunicação serial está disponível

if (umidSolValue <= 600) {
    lcd.setCursor(0, 0);
    lcd.print("Solo seco :(");
    lcd.setCursor(2, 1);
    lcd.print(" (");
    lcd.print(umidSolValue);
    lcd.print(")");

    lcd.setCursor(15, 0);
    lcd.print("Umididade do ar: ");
    lcd.print(umidAr);
    lcd.print("%");
}

```

```

lcd.setCursor(15, 1);
lcd.print("Temperatura: ");
lcd.print(temp);
lcd.print("C");

```

```

for (int posicao = 0; posicao < 40; posicao++)
{
    lcd.scrollDisplayLeft();
    delay(300);
} //laço que permite que os caracteres "deslizem" pelo display LCD

```

```

} else if (umidSolValue > 600) {
    lcd.setCursor(0, 0);
    lcd.print("Solo úmido :)");
    lcd.setCursor(3, 1);
    lcd.print(" (");
    lcd.print(umidSolValue);
    lcd.print(")");
}

```

```

lcd.setCursor(18, 0);
lcd.print("Umidade do ar: ");
lcd.print(umidAr);
lcd.print("%");

```

```

lcd.setCursor(18, 1);
lcd.print("Temperatura: ");
lcd.print(temp);
lcd.print("C");

```

```

for (int posicao = 0; posicao < 40; posicao++)
{
    lcd.scrollDisplayLeft();
    delay(300);
} //laço que permite que os caracteres "deslizem" pelo display LCD

```

```
//if para definição de solo seco ou úmido, para ser exibido no display LCD

if (umidSolValue <= 600) {
    digitalWrite(9, HIGH);
} else {
    digitalWrite(9, LOW);
} //if para ativar ou não a bomba

delay(500);

// "limpa" a tela do display
lcd.clear();

}
```

APÊNDICE B – Código Python

```
#!/usr/bin/python
# -*- coding: iso-8859-1 -*-
#Bibliotecas
import time
import serial
import datetime
from apscheduler.schedulers.blocking import BlockingScheduler

def aquisicao():
    # Iniciando conexao serial
    comport = serial.Serial('/dev/ttyUSB0', 9600)

    VALUE_SERIAL=comport.readline()

    print '\nRetorno da serial: %s' % (VALUE_SERIAL)

    print 'Enviando comando...'

    PARAM_CHARACTER='I'

    # Enviando 'I'
    comport.write(PARAM_CHARACTER)

    print 'Comando enviado!'

    # Lendo mensagem recebida após envio de 'I'
    VALUE_SERIAL=comport.readline()
```

```

print '\nValores: %s' % (VALUE_SERIAL)

# Fechando conexao serial
comport.close()

# Retornando mensagem recebida
return VALUE_SERIAL

def escrever_arquivo(BUFFER):
    # Abre arquivo no modo "adicionar"
    f = open('log.dat','a')
    f.write(BUFFER)
    f.write('\n')
    f.close()
    return

def some_job():
    valor = aquisicao()
    hora = datetime.datetime.now().time()
    print hora
    string_hora = str(hora)
    string_hora = string_hora[0:8]
    aux = string_hora + '\t' + str(valor)
    escrever_arquivo(aux)
    return

# Começando o main aqui
f = open('log.dat','w')
f.close()

scheduler = BlockingScheduler()
#scheduler.add_job(some_job, 'interval', hours=1)

```



```
#Ejecutando a cada 30 minutos  
scheduler.add_job(some_job, 'interval', minutes=30)  
#scheduler.add_job(some_job, 'interval', seconds=5)  
scheduler.start()
```