



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO CEARÁ
IFCE *CAMPUS* TAUÁ
CURSO DE TECNOLOGIA EM TELEMÁTICA

BRUNO MONTES DA SILVA RIBEIRO

**AUTOMAÇÃO DE PROCESSO RESIDENCIAL: EXPERIMENTO DE USO
OTIMIZADO E AUTOMÁTICO DE TORNEIRA TRADICIONAL BASEADO EM
ARDUINO.**

TAUÁ-CE
2020

BRUNO MONTES DA SILVA RIBEIRO

AUTOMAÇÃO DE PROCESSO RESIDENCIAL: EXPERIMENTO DE USO
OTIMIZADO E AUTOMÁTICO DE TORNEIRA TRADICIONAL BASEADO EM
ARDUINO.

Trabalho de Conclusão de Curso (TCC)
apresentado ao curso de Tecnologia em
Telemática do Instituto Federal de
Educação, Ciência e Tecnologia do Ceará
(IFCE) – Campus Tauá, como requisito
parcial para obtenção do Título de
Tecnólogo em Telemática. Área de
concentração: Sistemas Embarcados.

Orientador: Prof. Me. Adonias Caetano de
Oliveira.

Coorientador: Prof. Francisco Luciano
Castro Martins Júnior

TAUÁ-CE

2020

Dados Internacionais de Catalogação na Publicação
Instituto Federal do Ceará - IFCE
Sistema de Bibliotecas - SIBI
Ficha catalográfica elaborada pelo SIBI/IFCE, com os dados fornecidos pelo(a) autor(a)

R484a Ribeiro, Bruno Montes da Silva.

Automação de processo residencial: experimento de uso otimizado e automático de torneira tradicional baseado em arduino./ Bruno Montes da Silva Ribeiro. - 2020.
76 f. : il. color.

Trabalho de Conclusão de Curso (Graduação) - Instituto Federal do Ceará,
Tecnologia em Telemática, Campus Tauá, 2020.
Orientação: Prof. Me. Adonias Caetano de Oliveira..
Coorientador Prof. Francisco Luciano Castro Martins Júnior

1. Água . 2. Torneira automatizada. 3. Sistemas embarcados com arduino. I. Título.

CDD 621.382

BRUNO MONTES DA SILVA RIBEIRO

AUTOMAÇÃO DE PROCESSO RESIDENCIAL: EXPERIMENTO DE USO
OTIMIZADO E AUTOMÁTICO DE TORNEIRA TRADICIONAL BASEADO EM
ARDUINO.

Monografia apresentada ao curso de Tecnologia em Telemática do Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) - *Campus* Tauá, como requisito parcial para obtenção do Título de Tecnólogo em Telemática. Área de concentração: Sistemas Embarcados.

Aprovado (a) em: ____ / ____ / ____.

BANCA EXAMINADORA

Prof. Me. Adonias Caetano de Oliveira (Orientador)

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) – *Campus* Tianguá

Prof. Esp. Francisco Luciano Castro Martins Júnior (Coorientador)

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) - *Campus* Tauá

Prof. Me. Eduardo de Olivindo Cavalcante

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) – *Campus* Tauá

Prof. Esp. Lucas Ferreira Mendes

Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) – *Campus* Tauá

DEDICATÓRIA

À Deus por me proporcionar esse momento.

À minha família, por todo apoio e compreensão.

AGRADECIMENTOS

A Deus, por me abençoar com o dom da vida, me proporcionar motivação e força para buscar meus objetivos, me permitir chegar até este momento e me agraciar com sabedoria para enfrentar as dificuldades encontradas no percurso.

A minha família, por me apoiar e demonstrar que são orgulhosos das minhas conquistas, isso me deu e me dá muita força sempre.

Em especial meus avós, Maria Ferreira e Francisco Ferreira, por me ensinar a ser a exata pessoa que sou hoje. Minha mãe, Francislene Ferreira Montes, que sempre foi forte e batalhou para minha criação, por todo o amor e compreensão, por me ensinar princípios como dignidade, empatia, simplicidade e felicidade, me proporcionando tudo o que podia e estava ao seu alcance, apesar das dificuldades. Meu pai, Francisco da Silva Ribeiro, que mesmo distante, sempre apoia minhas escolhas. Meu irmão, Breno Montes que me ajudou sempre que precisei. Meu sobrinho Arthur, minhas irmãs Nadja, Nicole e Nazaré.

De forma muito especial, agradeço a minha companheira de vida, que veio para somar minha trajetória, me fortalecer cada vez mais e me fazer sonhar sempre mais alto, minha noiva Ana Luiza Ferreira dos Santos.

Agradeço também aos professores Lucas e Eduardo por fazer parte da banca avaliadora, participando e contribuindo com esse momento importante, e ao professor Rôney Reis por propor a ideia inicial do desenvolvimento.

Agradeço de forma honrosa aos meus mestres, meu orientador prof. Adonias Caetano e coorientador prof. Luciano Martins, por compartilhar seus conhecimentos, pela disponibilidade, orientação, compreensão e forma de tratamento, sou extremamente grato.

E por fim, agradeço aos demais familiares e amigos.

“Sonhar grande e pequeno dá o mesmo
trabalho”

(Jorge Paulo Lemann)

RESUMO

A água é um recurso essencial para a vida e por ser limitada é uma causa direta para combater o desperdício hídrico. Um meio bastante comum para utilização desse recurso em residências é através de torneiras, que quando mal utilizada causa gasto desnecessário desse recurso natural. A fim de reduzir o desperdício de água pode-se utilizar métodos que interfiram no comportamento do usuário alterando diretamente sua interação com dispositivos como é o caso da torneira. Por meio de tecnologias de sistemas embarcados e plataformas de prototipagem, é possível o desenvolvimento de uma torneira automatizada capaz de atuar diretamente na maneira em que o usuário utiliza o objeto. Para isso foi desenvolvido um protótipo baseado na plataforma Arduino com a finalidade de automatizar uma torneira de acionamento comum. Por meio de um conjunto de testes de validação da torneira automatizada em comparação com a torneira convencional permitiu a obtenção de dados de consumo da água. Essas informações de consumo foram dispostas em uma tabela, para melhor visualização, que continha valores individuais de uso e valor total de teste em litros, juntamente com a atividade exercida. Uma análise comparativa dos dados possibilitou mensurar quanto o usuário desperdiçou simplesmente pela má utilização de uma torneira por acionamento manual, ou seja, mostrou a diferença do consumo de uso efetivo da água e consumo com desperdício mostrando que o protótipo proposto pode ser um aliado no combate ao desperdício de água.

Palavras-chave: Água. Torneira automatizada. Sistemas embarcados com arduino.

ABSTRACT

Water is an essential resource for life and because it is limited it is a direct cause to fight water waste. A very common way to use this resource in homes is through taps, which when misused causes unnecessary expenditure of this natural resource. In order to reduce water waste, methods can be used that interfere with the user's behavior by directly changing their interaction with devices such as the tap. Through technologies of embedded systems and prototyping platforms, it is possible to develop an automated tap capable of acting directly in the way in which the user uses the object. For this, a prototype was developed based on the Arduino platform in order to automate a common tap. Through a set of validation tests of the automated tap compared to the conventional tap, it was possible to obtain water consumption data. This consumption information was displayed in a table, for better visualization, which contained individual usage values and total test value in liters, along with the activity performed. A comparative analysis of the data made it possible to measure how much the user wasted simply due to the misuse of a faucet by manual activation, that is, it showed the difference in the consumption of effective water use and consumption with waste showing that the proposed prototype can be an ally in combat to the waste of water.

Keywords: Water. Automated faucet. Embedded systems with arduino

LISTA DE FIGURAS

Figura 1 - Microcontrolador ATmega328P	22
Figura 2 - Sensor de obstáculos infravermelho E18-D80NK	25
Figura 3 - Válvula Solenoide de 12 volts com adaptador $\frac{3}{4}$ " para $\frac{3}{4}$ "	25
Figura 4 - Módulo Relé Simples	26
Figura 5 - Arduino UNO Rev 3	28
Figura 6 - Arduino MEGA 2560	28
Figura 7 - Diagrama de pinagem do ATmega328	29
Figura 8 - Diagrama de pinagem do ATmega2560	29
Figura 9 - Diagrama de processamento da placa Arduino	32
Figura 10 - Imagens superiores e inferiores da placa de controle de relês	33
Figura 11 - Solenoide conectada a mangueira de água.	33
Figura 12 - Esquema do projeto	37
Figura 13 - Diagrama de caso de uso do protótipo	51
Figura 14 - Diagrama de funcionamento da automação da torneira.	52
Figura 15 - Arquitetura Física do protótipo	55
Figura 16 - Data Logger	56
Figura 17 - Bloco da torneira automatizada	57
Figura 18 - Conexão do sistema de passagem de água.	60
Figura 19 - Conexão do sistema de data logger.	62
Figura 20 – Esquema de LEDs e resistores.	63
Figura 21 - Acoplamento para relé e conjunto de LEDs	64
Figura 22 - Copo medidor	65
Figura 23 - Conexão do sensor de fluxo com o Arduino Uno	65
Figura 24 - Torneira com sensor reposicionado	66
Figura 25 - Protótipo pronto para uso	67
Figura 26 – Mão suja / Mão limpa.	68
Figura 27 - Display ao finalizar um teste do protótipo	68

LISTA DE TABELAS

Tabela 1 - Calendário das Chuvas do Estado do Ceará: 2011 a 2019	14
Tabela 2 - Consumo da torneira com acionamento manual	70
Tabela 3 - Consumo da torneira automatizada.....	70

LISTA DE QUADROS

Quadro 1 - Horários abastecimento bloqueado.....	38
Quadro 2 - Dispositivos essenciais do protótipo.....	48
Quadro 3 - Estágios de funcionamento primário do protótipo	53
Quadro 4 - Bibliotecas utilizadas	58

LISTA DE ABREVIATURAS E SIGLAS

COGERH	Companhia de Gestão dos Recursos Hídricos
COGERH	Companhia de Gestão dos Recursos Hídricos
CS	Seleção de Chip
DEMAE	Departamento Municipal de Água e Esgoto de Caldas Novas
DEMAE	Departamento Municipal de Água e Esgoto de Caldas Novas
FEEC	Faculdade de Engenharia Elétrica e Computação
FUNCEME	Fundação Cearense de Meteorologia e Recursos Hídricos
I2C	<i>Inter-Integrated Circuit</i>
IDE	<i>Integrated Development Environment</i>
MISO	<i>Master IN Slave OUT</i>
MOSI	<i>Master OUT Slave IN</i>
ONU	Organização das Nações Unidas
RAM	<i>Random Access Memory</i>
ROM	<i>Read Only Memory</i>
RTC	<i>Real Time Clock</i>
SCK	<i>Serial Clock</i>
SCL	<i>Serial Clock Line</i>
SDA	<i>Serial Data Line</i>
SNIS	Sistema Nacional de Informações Sobre Saneamento
SPI	<i>Serial Peripheral Interface</i>
SS	<i>Slave Select</i>
UML	<i>Unified Modeling Language</i>
UNICAMP	Universidade Estadual de Campinas
USB	<i>Universal Serial Bus</i>

SUMÁRIO

1	INTRODUÇÃO.....	14
1.1	Justificativa.....	16
1.2	Objetivos.....	17
1.2.1	<i>Objetivo geral.....</i>	18
1.2.1	<i>Objetivos específicos.....</i>	18
1.3	Estrutura do trabalho.....	18
2	FUNDAMENTAÇÃO TEÓRICA.....	20
2.1	História e conceito de automação.....	20
2.2	Sistemas embarcados e microcontrolador.....	21
2.2.1	<i>Componentes.....</i>	23
2.2.1.1	<i>Sensores.....</i>	24
2.2.1.2	<i>Atuadores.....</i>	25
2.2.1.3	<i>Fonte de energia.....</i>	26
2.3	Arduino.....	27
3	TRABALHOS RELACIONADOS.....	31
3.1	Procedimentos de busca e seleção dos trabalhos.....	31
3.2	A Utilização da Placa Arduino na Criação de Um Sistema de Controle de Desperdício de Água e Acessibilidade nas Escolas.....	31
3.3	Consumo de Água das Torneiras dos Banheiros da FEEC.....	34
3.4	Controlando o consumo de água através da internet utilizando Arduino.....	36
4	DESENVOLVIMENTO DO PROTÓTIPO.....	39
4.1	Revisão de Literatura.....	39
4.2	Requisitos do protótipo.....	40
4.2.1	<i>Levantamento de requisitos.....</i>	40
4.2.2	<i>Especificação de requisitos.....</i>	41

4.2.2.1	<i>Requisitos funcionais</i>	41
4.2.2.2	<i>Requisitos não funcionais</i>	45
4.2.2.3	<i>Requisitos operacionais</i>	50
4.2.3	<i>Diagrama de Casos de Uso</i>	50
4.2.4	<i>Diagramas de Estados</i>	52
4.3	Projeto e Implementação do protótipo	53
4.3.1	<i>Arquitetura Lógica</i>	53
4.3.2	<i>Arquitetura Física</i>	55
4.3.3	<i>Ambiente de Programação</i>	57
4.3.4	<i>Implementação</i>	58
4.3.4.1	<i>Conexão dos componentes e funcionamento</i>	59
4.3.5	<i>Verificação de Componentes</i>	62
4.3.6	<i>Validação do protótipo</i>	66
4.3.6.1	<i>Metodologia de teste</i>	67
5	RESULTADOS E DISCUSSÕES	69
5.1	Tratamento de dados	69
5.2	Resultados obtidos e análise	69
6	CONCLUSÃO	72
	REFERÊNCIAS	73
	APÊNDICE – CÓDIGO FONTE DO SISTEMA DE TORNEIRA AUTOMATIZADA	76

1 INTRODUÇÃO

De acordo com o *General Comment No. 15* (2003) da Organização das Nações Unidas – ONU, a água potável é um dos requisitos básicos para garantir uma vida digna, pois seu uso adequado proporciona uma melhor qualidade de vida, visto que é essencial. No entanto, este recurso natural é limitado, o que torna as abordagens sobre o tema de desperdício cada vez mais comuns.

No caso de algumas regiões, garantir a utilização da água é uma situação constantemente desafiadora por conta da captação desse recurso, que nem sempre é satisfatória em volume devido a poucas chuvas.

De acordo com a Tabela 1 no ano de 2012, o Calendário das Chuvas do Estado do Ceará apresentou um desvio de -51,4%, com um valor observado de 388,8 mm, menos da metade da normal esperada de 800,6 mm. Já o ano de 2013 apresentou um desvio final de -31,2%, uma melhora em relação ao ano anterior, porém ainda bem abaixo do esperado.

Em 2019 a situação foi melhor, pois houve um desvio de 5,7%, ou seja, pouco maior que o esperado, atingindo assim um dos melhores resultados dos últimos 9 anos, ficando atrás apenas do ano de 2011 com um desvio de 29,2%, conforme os dados obtidos pela Fundação Cearense de Meteorologia e Recursos Hídricos – FUNCEME e mostrados na Tabela 1.

Tabela 1 - Calendário das Chuvas do Estado do Ceará: 2011 a 2019

Ano	Normal (mm)	Observado (mm)	Desvio (%)
2011	800,6	996,6	24,5
2012	800,6	362,1	-54,8
2013	800,6	546,9	-31,7
2014	800,6	546,1	-31,8
2015	800,6	523,1	-34,7
2016	800,6	554,6	-30,7
2017	800,6	698,2	-12,8
2018	800,6	802,2	0,2
2019	800,6	845,9	5,7

Fonte: FUNCEME (2019).

Apesar da melhora apresentada no ano de 2019, ainda é preciso estar alerta, pois o estado já vem sofrendo constantemente com a seca desde 2012, ocasionando a baixa captação de recurso hídrico.

A situação das reservas de água no estado do Ceará tem sido um ponto crítico e para constatar esse fato, a Companhia de Gestão dos Recursos Hídricos – COGERH, analisou os 153 açudes monitorados pela Companhia até o ano de 2017 e concluiu que de 18,64 bilhões de metros cúbicos de capacidade total, foram apresentados apenas 6,3% desse volume. Mesmo com uma melhora significativa nos índices de chuva em 2018, de 155 açudes monitorados em 2019, 75 apresentam volume inferior a 30% de suas capacidades (COGERH, 2019).

Dentro dessas dificuldades vários municípios promovem campanhas contra o desperdício hídrico e adotam estratégias para tentar combater a crise já sofrida durante anos, como na zona urbana do município de Tauá (CE), localizado na região dos Inhamuns, onde a distribuição de água é feita por setor. Então cada setor deve se preocupar com o consumo e estoque de água até que ela possa ser distribuída novamente naquele setor. Tem-se então a necessidade de práticas constantes de redução do consumo hídrico.

Problemas como o consumo excessivo de água, em grande parte são ocasionados por usuários e com os avanços do meio tecnológico podem ser amenizados. Um exemplo é o uso de sistemas embarcados, que são sistemas com finalidade específica como controlar o uso de determinado dispositivo, e que devido ao uso de componentes eletrônicos podem executar uma tarefa mais rápido que um ser humano, mesmo precisando do ser humano para funcionar como ativador do sistema.

Isso é possível devido ao avanço tecnológico que proporcionou o uso de controladores cada vez mais rápidos, barateamento de componentes e a facilidade de adquirir e programar dispositivos. Desenvolver um sistema embarcado tornou-se uma tarefa mais simplificada por conta de plataformas de prototipagem como o Arduino, proporcionando aos desenvolvedores a possibilidade de criar sistemas como automatização de lâmpadas, acesso utilizando biometria, entre outros, de maneira barata.

De acordo com Gonçalves (2006, p. 292), “dadas as poucas possibilidades de redução do consumo no campo da tecnologia, restaram as alternativas de cunho comportamental”, ou seja, o uso de tecnologias como os sistemas embarcados ocasionará maior efeito se afetar a maneira no qual o usuário está de fato utilizando a água.

Um objeto simples e utilizado na maioria das residências de médio e pequeno

porte do município de Tauá, como uma torneira de pia, não permite muitas alterações em seu projeto para reduzir o consumo de água permitindo uma utilização satisfatória, porém há possibilidade de explorar o lado comportamental dos usuários para atingir esse objetivo, pois grande parte do desperdício ocorre por esse meio, com ações como deixar a torneira acionada enquanto não é utilizada (GONÇALVES 2006).

Esperar a água do chuveiro elétrico esquentar é uma prática que gera perdas, que é caracterizada por Gonçalves (2009, p. 42) como: “[...] a quantidade de água prevista para a realização de um ou mais usos, mas que não é utilizada devido a deficiências técnicas, operacionais, econômicas ou de outro tipo. ”

Segundo o Sistema Nacional de Informações Sobre Saneamento – SNIS, mediante práticas desse modelo, é sabido que há tentativas de conscientização dos usuários, no entanto é notado que o consumo de água por habitante não teve grandes alterações (SNIS, 2016).

Visto isso, torna-se pertinente a instalação de um protótipo de torneira automática, que incentive o usuário a ativá-la apenas em sua real utilização, e posteriormente seja comparada com o uso da mesma torneira em sua forma original. Os resultados dessa comparação irão definir se a torneira automatizada pode possibilitar o baixo consumo hídrico em relação a torneira de acionamento tradicional.

1.1 Justificativa

Uma publicação feita no site oficial do Departamento Municipal de Água e Esgoto de Caldas Novas – DEMA E, mostra que durante uma escovação são gastos 12 litros de água em uma torneira que permanece aberta pelo período de 5 minutos. Sabe-se então que durante o processo de escovação a torneira que permanece aberta por 5 minutos estará em grande parte desse tempo desperdiçando água, pois este processo poderia ser realizado com aproximadamente 0,5 litros se ela fosse fechada quando não utilizada, o que representa uma redução de aproximadamente 95,83% no consumo de água, ou então simplesmente usar um copo com água (DEMAE, 2019).

Diante de todos os problemas encontrados sobre a captação e distribuição de água no estado, o Sistema Nacional de Informações sobre Saneamento – SNIS

(2016) mostra que no Ceará, o valor médio de consumo de água é de 125 litros por habitante ao dia, números um pouco acima do valor de 110 litros determinado pela Organização das Nações Unidas – ONU como suficiente para atender as necessidades básicas de uma pessoa.

Como citado anteriormente, para tentar resolver esse problema devem ser utilizados métodos que interfiram no comportamento do usuário alterando diretamente sua interação com dispositivos como uma torneira. Este trabalho propõe então a pesquisa e desenvolvimento de um sistema que possa gerenciar a abertura e fechamento de uma torneira a fim de garantir seu acionamento apenas quando for de fato demandada sua utilização e assim seu consumo seja reduzido em práticas cotidianas de um usuário comum.

Esta pesquisa utilizará os conhecimentos de automação, eletrônica e microcontroladores, e a partir do laboratório de eletrônica do IFCE campus Tauá, desenvolver um experimento que torne possível automatizar a abertura e fechamento de uma torneira tradicional.

Os conceitos de automação são importantes no desenvolvimento do projeto, pois os procedimentos de abrir e fechar a torneira passarão a ser executados por uma válvula solenoide de 12 volts denominada de atuador do sistema, que segundo Roggia e Cardozo (2016, p. 21) é um “elemento que atua para alterar fisicamente uma variável manipulada”. Com isso o processo citado não será feito pelo ser humano, caracterizando o sistema como automatizado. Entretanto, ainda há a necessidade de que um usuário forneça dados de entrada para a unidade controladora do sistema e assim dar início ao seu funcionamento (GROOVER, 2011).

Para isso, será necessário utilizar uma placa Arduino Uno Rev 3, que de acordo com Mota (2017), funciona de maneira semelhante a um mini computador, podendo programar suas portas de entrada e saída para comunicação com os componentes externos, enviando ou recebendo dados de acordo com a aplicação e componentes conectados ao sistema. Posteriormente será conectado a placa, um sensor de obstáculos infravermelho E18-D80NK, que servirá para detectar o usuário, gerando dados de entrada no sistema para que o sistema microcontrolado gere o sinal necessário para acionamento ou desligamento da válvula solenoide.

O projeto será mais detalhado no capítulo de desenvolvimento, bem como suas funcionalidades desenvolvidas para melhor interação com o usuário.

1.2 Objetivos

Essa seção apresenta os objetivos a que o protótipo é desenvolvido, e servirá como um guia para seu desenvolvimento.

1.2.1 Objetivo geral

O principal objetivo deste trabalho é desenvolver um protótipo de um sistema embarcado baseado em Arduino para automatização e controle do uso de uma torneira, comparando seu desempenho às torneiras tradicionais.

1.2.2 Objetivos específicos

- a) Idealizar o protótipo proposto para a problemática sugerida, a partir de conceitos adquiridos em bibliografias das áreas de automação industrial, microcontroladores, eletrônica e lógica de programação;
- b) Planejar a proposta de automação de uma torneira com base nas informações adquiridas no item acima, vivenciando o problema e objeto a ser automatizado, e com base em dados que confirmem o desperdício hídrico no contexto residencial por utilização de uma torneira;
- c) Testar o protótipo (torneira automatizada) em situações cotidianas, com a finalidade de gerar resultados para que assim possam ser comparados com os dados do projeto em sua forma original (torneira comum);
- d) Analisar os dados obtidos da torneira automatizada e apontar possíveis pontos no qual serão identificados os motivos do desgaste hídrico na utilização da torneira comum, e após comparar os sistemas, dizer se o protótipo tem potencial para reduzir o consumo hídrico.

1.3 Estrutura do trabalho

Este trabalho da área de sistemas embarcados é formado por seis capítulos. Neste primeiro capítulo foi feito uma contextualização do tema tratado, a justificativa do trabalho e a descrição dos objetivos geral e específicos.

O segundo capítulo é o referencial teórico utilizado como base para o desenvolvimento da proposta, abordando assuntos como sistemas embarcados, microcontroladores, sensores, atuadores e fontes de energia.

Os trabalhos relacionados sobre projeto ou desenvolvimento de sistemas de controle de desperdício de água são apresentados no terceiro capítulo.

No quarto capítulo, é apresentado o projeto do sistema, descrevendo suas etapas, materiais e recursos utilizados e a metodologia de testes adotada.

O quinto capítulo descreve a realização dos testes, os resultados coletados e a respectiva análise destes.

Finalmente, o sexto capítulo faz as considerações finais sobre a proposta desta monografia, ressaltando possíveis desdobramentos dos trabalhos.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os conceitos de automação, sistemas embarcados, microcontroladores e Arduino. Esses conceitos servem como base para o desenvolvimento e compreensão do projeto. Ainda aqui, são apresentados exemplos de dispositivos como sensores, atuadores e placas de prototipagem.

Este capítulo será dividido em três seções, sendo elas a 2.1 responsável por descrever o conceito e a história de automação, como e por quem esse termo foi denominado, e ainda exemplos de sistemas automatizados existentes antes de sua definição. Os conceitos de sistemas embarcados e microcontroladores serão descritos na seção 2.2 juntamente com os conceitos de atuadores, sensores e fontes de energia. Finalizando o capítulo, na seção 2.3 será abordado o conceito de Arduino, placas de prototipagem da plataforma e suas características.

2.1 História e conceito de automação

O conceito de automação surgiu no ano de 1945 e foi criado por Del Harder, um engenheiro da *Ford Motor Company*, com a necessidade da descrição dos diversos dispositivos automáticos que foram naquela época implementados na empresa (GROOVER, 2011). Percebe-se então o motivo da utilização do termo **automação industrial**.

Conforme Roggia e Cardozo (2016, p. 16), “[...] a automação industrial começou a conquistar destaque na sociedade no século XVIII, com o início da Revolução Industrial, originada na Inglaterra”. Junto a esse fator, os avanços tecnológicos das máquinas industriais permitiram o aumento produtivo das mercadorias em uma escala bem maior.

O processo de automação desenvolveu-se ao longo do tempo bem antes de seu embasamento, pois com o desenvolvimento de dispositivos mecânicos, surgiram os chamados elementos básicos da automação: *a energia*, *o sistema de controle* e *o programa de instruções*. O terceiro elemento pode ser confirmado pelo uso do tear de Jacquard, uma das primeiras máquinas programáveis, criadas na década de 1800, em que o padrão de tecelagem era definido por um tipo programa de instruções (GROOVER, 2011).

Groover (2011, p. 56) diz que: “A automação pode ser definida como a

tecnologia por meio da qual um processo ou procedimento é alcançado sem a assistência humana”. Porém, muitos sistemas automatizados necessitam da entrada de dados para que seu funcionamento seja iniciado e neste caso os dados podem ser fornecidos por um humano.

Uma das maneiras do sistema adquirir os dados de entrada é com a utilização de sensores, que de acordo com Roggia e Cardozo (2016), são dispositivos conectados a uma variável controlada, que alteram suas propriedades na medida em que ocorrem mudanças no processo.

Roggia e Cardozo (2016, p. 20) define processo como “conjunto de atividades ou passos que objetivam atingir uma meta. Utilizado para criar, inventar, projetar, transformar, produzir, controlar, manter e usar produtos ou sistemas”.

Tendo em vista essa definição, nota-se que os objetivos deste trabalho envolvem conceitos de automação aplicados em um processo, que no caso é a abertura e fechamento de uma torneira. Portanto o termo automação de processo substituirá automação industrial, pois apesar de ser originado na indústria, a automação pode ser implementada também a um processo não industrial.

Basta interconectar componentes para obter como principal função a realização de um ou mais processos e tem-se então um sistema de automação, que por sua vez é alterado de acordo com resposta ao estado imediato e as variações dos eventos causadas nos equipamentos (DE NEGRI, 2004).

2.2 Sistemas embarcados e microcontrolador

A ampla variedade de aplicação de sistemas automatizados, só se tornou possível devido a introdução dos computadores, que passaram a armazenar e fazer a leitura de programas de instruções, tornando-se assim o controlador dos processos, o que possibilitou também o uso de atualizações para otimizar e corrigir programas de controle (GROOVER, 2011).

Por conta da queda de custo dos microcontroladores após a década de 80, os sistemas embarcados têm sido cada vez mais utilizados, como em aparelhos de TV, DVD e relógios (OLIVEIRA, 2006 apud VARGAS, 2007).

Vargas (2007, p. 11) diz “Sistemas embarcados ou embutidos são aqueles que manipulam dados dentro de sistemas ou produtos maiores, não possuem interface com o usuário e executam uma função específica.”

Para Santos (2006), uma das principais características dos sistemas embarcados, é a interação com o ambiente, no qual são utilizados sensores para coleta e tratamento de dados, para que posteriormente possam executar sua função. Outra característica importante é que muitos sistemas são capazes de operar com valores digitais e analógicos devido a conectividade de componentes que utilizam valores contínuos e discretos, isso aumenta a capacidade de aplicação por conta da compatibilidade de sensores e atuadores que podem ser usados de acordo com a demanda do projeto.

Microcontrolador é um dispositivo que possui todos os componentes básicos de um computador, como microprocessador, memórias voláteis¹ e não voláteis², portas de entrada e saída, tudo integrado em um único *chip* (PENIDO; TRINDADE, 2013).

Normalmente possuem aplicações bem definidas por conta da limitação dos seus recursos computacionais, como menor capacidade de memória, o que permite que seu projeto de construção seja de baixo custo comparado a um computador *desktop*, por exemplo (GIMENEZ, 2002).

Um microcontrolador utilizado para sistemas embarcados é o ATmega328P por conta de sua plataforma de prototipagem. A Figura 1 exibe esse microcontrolador.

Figura 1 - Microcontrolador ATmega328P



Fonte: (MICROCHIP, 2019)

De acordo com Penido e Trindade (2013) a Intel foi a primeira empresa a lançar um microcontrolador e isso ocorreu no ano de 1977. Ela é também

¹ Tipo de memória que perde seu conteúdo ao ser desligada da energia (GIMENEZ, 2002, p. 9).

² Tipo de memória que não perde seu conteúdo ao ser desligada da energia (GIMENEZ, 2002, p. 9).

responsável por criar a família 8051 utilizada em muitos sistemas de automação.

Garcia (2016) mostra que a família 8051 possui uma CPU de 8 bits, 128 bytes de memória *Random Access Memory* – RAM para armazenamento de dados, 4 KB de memória *Read Only Memory* – ROM. Sua arquitetura era baseada em Harvard, logo instruções e dados não compartilham o mesmo espaço quando armazenadas, diferente da arquitetura de Von Neuman em que o sistema possui uma única área de memória que armazena dados e programas.

A utilização de um microcontrolador como sistema de controle na automação é possível, pois esse tipo de projeto é utilizado quando um sistema automatizado se torna capaz de realizar ações na medida em que os sinais de entrada são modificados (DE NEGRI, 2004), além deste elemento ter a capacidade de armazenar e ler o programa de instruções, o que o caracteriza como controlador de processos.

Esses programas são de toda responsabilidade do desenvolvedor do sistema, visto que é preciso conhecer a linguagem utilizada e de acordo com a necessidade, escrever o algoritmo do programa, que só pode ser implementado graças a lógica de programação utilizada nos computadores.

De acordo com Albani (2010, p. 16), “Lógica de programação pode ser definida como um conjunto de técnicas para encadear pensamentos a fim de atingir determinado objetivo”. Além de ter de conhecer a linguagem compatível com o sistema, o programador terá que conhecer seu *hardware*, para que o código do programa seja feito de maneira lógica, pensando no passo a passo, até a finalização do processo, o que permitirá escrever o programa com total convicção de seu funcionamento, ou então o processo poderá ocorrer de forma não planejada.

Um sistema embarcado utilizado diariamente por diversos usuários é o controle remoto de TV. Esse sistema consiste em um dispositivo alimentado por pilhas, que possui codificações armazenadas, e dependendo do botão que o usuário acionar, a unidade controladora envia uma codificação para a TV por meio do sensor infravermelho. Pode-se então por meio desse exemplo, reconhecer todos os componentes básicos definidos para sistemas embarcados, como fonte energética, sensor, atuador e microcontrolador.

2.2.1 Componentes

O microcontrolador é o componente mais importante de um sistema embarcado. Como foi visto, ele possui integrado a unidade central de processamento, memória e portas para conexão com outros componentes externos à unidade microcontroladora. Esses componentes são controlados pelo microcontrolador e possuem papel fundamental dentro de uma aplicação.

Visto isso, se faz necessário a abordagem de cada um dos componentes para desenvolver muitos tipos de sistemas microcontrolados, assim se torna fácil o aperfeiçoamento ou troca de algum, visando o melhor desempenho.

2.2.1.1 Sensores

De acordo com Groover (2011), um sensor é um dispositivo usado para converter uma ação física em um sinal a ser utilizado no sistema, que em sua maioria é o sinal elétrico. Essa conversão ocorre com a transformação do sinal contínuo, detectado pelo sensor, em um sinal discreto processado pelos computadores. Seu uso é imprescindível, pois no sistema proposto há a necessidade da entrada de dados para seu funcionamento, o que será feito através de um sensor de obstáculos infravermelho E18-D80NK semelhante à Figura 2, sensor capaz de detectar objetos através do envio e captação de luz infravermelha.

Informações disponíveis no *datasheet*³ do E18-D80NK, mostram que são necessários 5 volts para seu funcionamento e uma corrente de 100mA que pode ser fornecido por uma placa Arduino por exemplo, e seu campo de atuação possui limites ajustáveis de 6 a 80 cm.

³ https://cdn.awsli.com.br/945/945993/arquivos/Manual_IR-Sensor%20Switch%20E18.pdf

Figura 2 - Sensor de obstáculos infravermelho E18-D80NK



Fonte: (Elaborada pelo autor, 2020).

2.2.1.2 Atuadores

Um atuador é um dispositivo capaz de transformar um sinal enviado por um controlador do sistema em uma ação física, fazendo com que o objetivo final de uma automação seja alcançado (GROOVER, 2011). Nesse caso, será utilizada uma válvula solenoide de duas vias, com a bitola de $\frac{3}{4}$ " x $\frac{3}{4}$ " como o da Figura 3. Esse dispositivo necessita de uma alimentação externa de 12 volts e será usado entre a encanação e a torneira. Assim, independente do modelo da torneira o sistema funcionará, bastando apenas que utilize um adaptador de acordo com o diâmetro da rosca da torneira, e que siga o requisito de manter o registro da torneira sempre aberto.

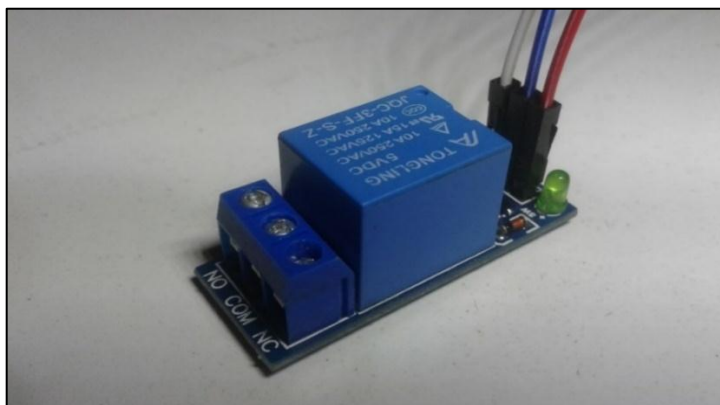
Figura 3 - Válvula Solenoide de 12 volts com adaptador $\frac{3}{4}$ " para $\frac{3}{4}$ "



Fonte: (Elaborada pelo autor, 2020).

Por necessitar de uma fonte externa, será necessário usar um módulo relé simples similar à Figura 4, que atuará entre o microcontrolador e a válvula.

Figura 4 - Módulo Relé Simples



Fonte: (Elaborada pelo autor, 2020).

Cunha (2009, p. 55) diz que: “O relé é um dispositivo elétrico destinado a produzir modificações súbitas e predeterminadas em um ou mais circuitos elétricos de saída, quando alcançadas determinadas condições no circuito de entrada, que controla o dispositivo”. Sabendo disso, no circuito microcontrolado proposto, o relé irá permitir ou não que a energia da fonte externa (12 volts) chegue até a válvula solenoide.

2.2.1.3 Fonte de energia

Por fatores como disponibilidade, capacidade de conversão em outros tipos de energia e a possibilidade de ser utilizada para transmissão de sinais, a eletricidade é estabelecida como principal fonte de energia para sistemas automatizados. Além das possibilidades já citada, pode também ser armazenada e dependendo do dispositivo de armazenamento, utilizada por um longo tempo, principalmente em projetos que demandam uma menor quantidade de energia (GROOVER, 2011).

Apesar da possibilidade de ser utilizada outras fontes de energia, (GROOVER, 2011, p. 59) afirma que: “Em muitos casos, quando fontes alternativas de energia são utilizadas na condução do processo em si, a energia elétrica é empregada nos controles que automatizam a operação”. Fazendo assim, com que ela continue sendo a fonte primária de energia para funcionamento do sistema.

A aplicação de todos esses conceitos, darão embasamento para desenvolver o projeto junto ao problema de desperdício hídrico causado pela utilização padrão de

uma torneira de abertura e fechamento manual, no qual será utilizado conceitos de automação de processo para modificar seu modo de acionamento, assim necessitando de um ser humano apenas para fornecer dados de entrada para dar início ao seu funcionamento.

Esse modo de acionamento passará a ser feito por um atuador, nesse caso uma válvula solenoide que utilizará 12 volts de tensão. O Arduino por sua vez, fornece no máximo 5 volts na maioria de suas portas digitais e analógicas, tornando-se assim necessária a utilização de uma fonte externa para a válvula. Já no caso da captura de dados de entrada, 5 volts são suficientes para o sensor de obstáculos infravermelho E18-d80NK funcionar de forma adequada. Além disso, há também a alimentação do microcontrolador, que no modelo Arduino Uno Rev 3 basta utilizar 5 volts através de um cabo *Universal Serial Bus* – USB ou de 7 a 12 volts através do conector de energia.

2.3 Arduino

Um Arduino é um dispositivo programável, no qual conectado a componentes externos é capaz de processar dados de entrada e saída dentre essa interconexão, podendo assim enviar sinais de controle e receber dados através de sensores que interagem com seu ambiente (MCROBERTS, 2011).

Esse fator também é comprovado por Souza *et al.* (2011), no qual mostra que essa plataforma foi desenvolvida para interagir com o ambiente, sendo utilizada por meio de programação em conjunto com dispositivos eletrônicos que podem ser conectados às suas portas de entrada e saída.

É importante ressaltar que de acordo com seu site⁴ oficial, o Arduino é uma plataforma eletrônica com *software* e *hardware* fáceis de serem utilizados, voltado para usuários que buscam construir projetos interativos.

Há vários modelos de placas Arduino disponíveis, como o Arduino Leonardo, Arduino Nano, Arduino Micro, sendo a Arduino UNO Rev 3 a mais famosa e documentada placa de prototipagem dos modelos, segundo o site⁵ oficial da plataforma, ou seja, existem várias versões derivada do modelo como o da

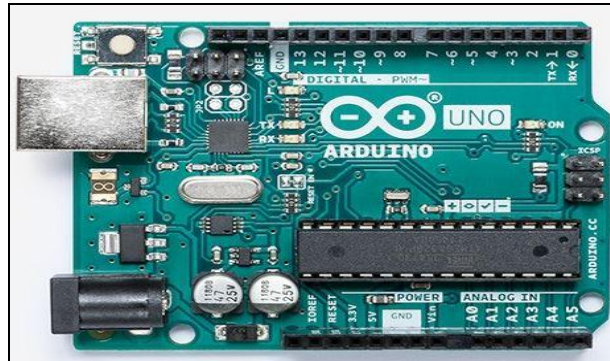
Figura 5.

⁴ <https://www.arduino.cc/en/Guide/Introduction#>

⁵ <https://store.arduino.cc/usa/arduino-uno-rev3>

No mesmo site podem ser encontrados vários modelos de placas, bem como a documentação de cada uma.

Figura 5 - Arduino UNO Rev 3

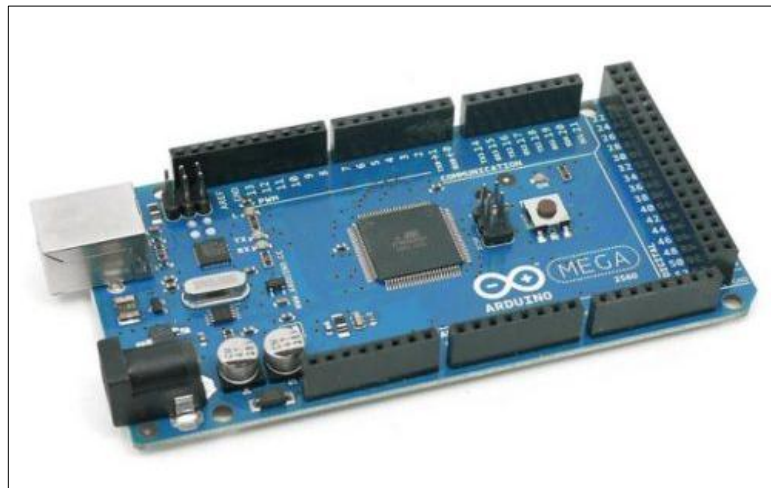


Fonte: (ARDUINO.CC, 2020).

Para projetos mais robustos a placa mais utilizada da plataforma é a Arduino Mega 2560, em uma rápida comparação da

Figura 5 com a Figura 6 pode-se notar que a placa Arduino MEGA 2560 se trata de um hardware com maior número de portas, ideal quando o número de portas da Arduino Uno, por exemplo, não for o suficiente.

Figura 6 - Arduino MEGA 2560



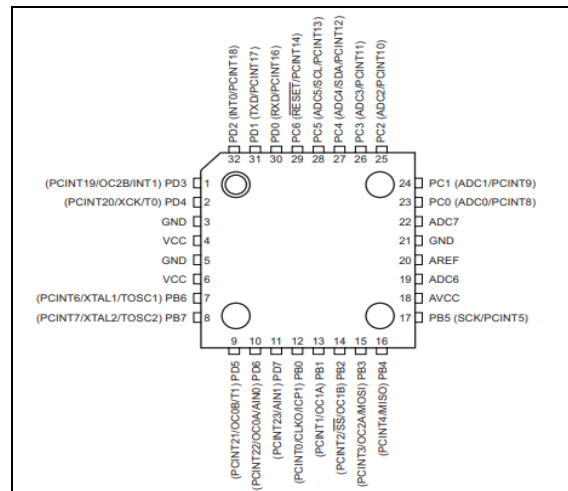
Fonte: (SOUZA, 2014).

Segundo Souza (2014), o Arduino Mega 2560 utiliza o microcontrolador ATmega2560 de 8 bits que trabalha em uma frequência de 16 MHz e possui 256 KB de Flash, 8 KB de RAM e uma EEPROM com capacidade de 4 KB. Sua placa dispõe de 54 pinos digitais no qual 15 possibilitam o uso de PWM. Ademais, possuem 16

pinos analógicos, 4 portas para comunicação serial e é uma placa utilizada normalmente para projetos mais elaborados.

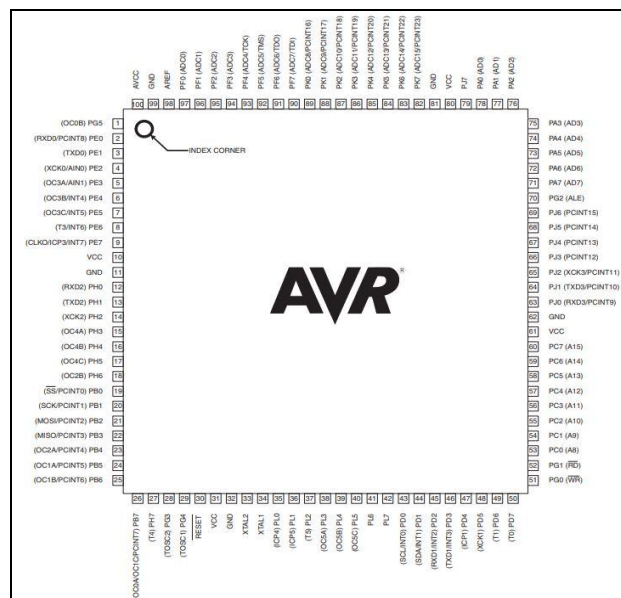
Ao comparar as pinagens das Figura 7 e Figura 8, é nítido a quantidade de recursos superior no microcontrolador ATmega2560 em relação ao mais famoso da plataforma, o ATmega328.

Figura 7 - Diagrama de pinagem do ATmega328



Fonte: (Atmel, 2020).

Figura 8 - Diagrama de pinagem do ATmega2560



Fonte: (Atmel, 2020).

A programação de um sistema baseado em Arduino é feita em uma linguagem semelhante a C/C++ por meio da Ambiente de Desenvolvimento Integrado (*Integrated Development Environment* – IDE) do Arduino, e ela é uma plataforma

open source, o que permite uma maior gama de compartilhamento de projetos entre usuários e construção de dispositivos externos compatíveis (SOUZA *et al.* 2011).

Mediante a essas afirmações, o Arduino se enquadrou nos requisitos para ser o dispositivo utilizado neste trabalho de automação de processo como controlador de processos, trabalhando em conjunto com um dispositivo de entrada de dados para acionamento do sistema e com um dispositivo de saída que acionará ou não o sistema de acordo com a programação utilizada.

3 TRABALHOS RELACIONADOS

Devido à importância da água para humanidade e as questões relacionadas ao seu desperdício, alguns trabalhos foram desenvolvidos com o intuito de economizar e controlar o seu uso. Projetos utilizando o auxílio tecnológico para redução e controle do uso da água tornam-se cada vez mais comuns, devido a fácil acessibilidade de dispositivos microcontrolados e microcontroladores, como uma placa Arduino, que além de ter um custo baixo, é de código aberto. Além disso, estudos relatando a preocupação sobre o desperdício e maneiras de amenizar esse problema, bem como alguns projetos neste âmbito, são abordados neste capítulo.

3.1 Procedimentos de busca e seleção dos trabalhos

Para localizar trabalhos com o mesmo propósito de automação e economia de água, foi utilizado o buscador Google e inserido primeiramente a *string* "consumo de água em torneiras" seguido das *strings* "arduino" e "artigo" o que possibilitou o acesso aos artigos descritos na seção 3.2, este selecionado pela ênfase em desperdício de água em escolas e automação de um sistema para esse fim e 3.3, selecionado por abordar o uso de torneiras de acionamento por mola em uma faculdade e ainda possibilitar a viabilidade de substituição delas.

Em seguida, a busca foi por algum trabalho que possibilitasse o controle de recursos hídricos de forma remota, para isso o mesmo buscador foi utilizado e com a *string* "controlando o consumo de água remotamente.pdf", foi localizado o trabalho descrito na seção 3.4, que se trata do uso da tecnologia de internet das coisas para controle e monitoramento remoto de um sistema de liberação e bloqueio residencial de água.

A busca e leitura dos trabalhos selecionados possibilitou uma abordagem sobre eles que será descrita nas próximas seções.

3.2 A Utilização da Placa Arduino na Criação de Um Sistema de Controle de Desperdício de Água e Acessibilidade nas Escolas

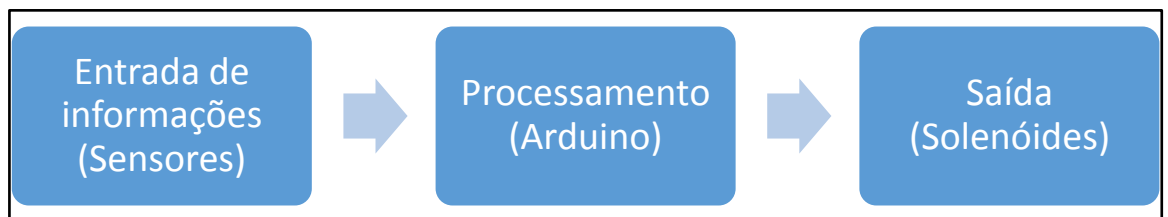
Esse projeto titulado "A Utilização da Placa Arduino na Criação de Um

Sistema de Controle de Desperdício de Água e Acessibilidade nas Escolas” dos autores (FELIX; VILELA; SILVA, 2012) consiste em criar um sistema para controlar o uso de água nas escolas, bem como tornar acessível a sua utilização para discentes com deficiências físicas, utilizando uma placa Arduino para gerenciar a demanda hídrica em banheiros e bebedouros.

Segundo o autor, o sistema proposto facilita o uso de torneiras, que por estarem automatizadas poderão ser utilizadas por usuários com deficiência motora, tornando-se também um sistema democrático no quesito usabilidade.

A plataforma selecionada para ser o centro operacional do sistema foi o Arduino, devido a sua vasta gama de aplicabilidade, quantidade de pinos, *hardware* e *software* livre. Para que a placa Arduino siga seu fluxo de funcionamento como mostra a Figura 9, ela necessita de um código que pode ser escrito e editado em diferentes tipos de sistemas operacionais, no qual é relatado por Fonseca (2010, p.3, apud FELIX; VILELA; SILVA, 2012, p.28).

Figura 9 - Diagrama de processamento da placa Arduino

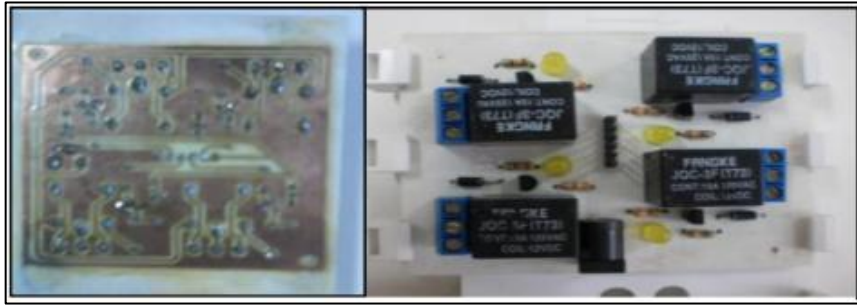


Fonte: Adaptado de Felix, Vilela e Silva (FELIX; VILELA; SILVA, 2012)

É relatado no artigo, que caso necessário, poderá ser utilizado outros tipos de sensores, como o de calor, porém foi previsto o sensor de movimento para fazer intermédio entre o usuário e a placa microcontroladora.

Foi confeccionado uma placa de controle de relês mostrada na Figura 10, que basicamente funciona como acionador ou interruptor da válvula solenoide mostrada na Figura 11, que por sua vez será o registro de passagem de água, permitindo ou não que a água passe para as torneiras, por exemplo.

Figura 10 - Imagens superiores e inferiores da placa de controle de relês



Fonte: (FELIX; VILELA; SILVA, 2012)

Figura 11 - Solenoide conectada a mangueira de água.



Fonte: (FELIX; VILELA; SILVA, 2012)

É dito por Felix, Viela e Silva (2012, p. 28) que em sistemas de automação as válvulas solenoides recebem cargas de energia emitidas pela placa Arduino. No entanto, o protótipo proposto alimenta a válvula por meio da rede elétrica de 110 volts e não utilizando a placa Arduino diretamente, já que seu acionamento é feito por relês, os quais são controlados pela placa central.

O protótipo libera a passagem de água somente quando as válvulas solenoides forem acionadas pelos relês, que servirão como interruptores para elas. Estes estão ligados diretamente ao Arduino, que só enviará o sinal de acionamento, quando receber um sinal de movimento enviado para a placa através dos sensores instalados próximo às torneiras. Da mesma maneira, quando a placa não receber informação de movimento enviado pelo sensor, os relês não fecharão o circuito da válvula com a rede elétrica, e quando não é alimentada a válvula retornará a sua posição inicial, fechando a passagem e impedindo o desperdício de água.

De fato, automatizar uma torneira pode proporcionar acessibilidade e combater o desperdício hídrico gerado em torneiras e banheiros, porém falta neste artigo, algum estudo ou teste que comprove essa eficiência. Outro fator, é que não

há como mensurar o consumo individual de cada torneira ou setor em que seja instalado o sistema. Ademais, não é especificado o tempo de uso das torneiras por exemplo, se existe um tempo variável com limite máximo por uso ou um tempo fixo, ponto relevante para definir o consumo hídrico final, pois caso seja com um tempo fixo, ainda poderá gerar desperdício se o usuário precisar fazer uso do sistema durante 3 segundos e o tempo ativo do sistema seja de 7 segundos, por exemplo.

3.3 Consumo de Água das Torneiras dos Banheiros da FEEC

Este artigo dos autores Persona, Mandelli e Inagaki, (2012) trata-se de um estudo sobre as torneiras de acionamento por mola utilizada nos banheiros da Faculdade de Engenharia Elétrica e Computação (FEEC) e o desperdício de água causado pelo seu mal funcionamento.

O estudo tem início abordando questões relacionada a água, como em FAO (2008, apud PERSONA; MANDELLI; INAGAKI, 2012, p. 1) no qual diz que: “[...] estimam que até 2025 um terço da população mundial experimentará efeitos extremos de sua escassez.”

Um fato importante apresentado é a mudança de abordagem nas estratégias de conscientização a partir da década de 1990, que passou a ter mais foco no uso de equipamentos que consomem menor volume de água, porém a má regulagem de alguns desses aparelhos pode trazer um efeito contrário que aumentará o desperdício em relação aos sistemas comuns.

De acordo com Persona, Mandelli e Inagaki (2012), é comum as torneiras da instituição estudada estarem mal reguladas, como o jato forte que chega a jorrar água em seus usuários ou até mesmo no chão, além do tempo que permanece acionada sem necessidade. A má regulagem aliada ao desperdício de água no momento de sua utilização, incentiva estudos de alternativas para que seja reduzido o consumo das torneiras em uma determinada instituição.

Para gerar resultados, a metodologia foi dividida em partes, dando início na utilização de um cronômetro e béqueres⁶ para descobrir o tempo de funcionamento de cada uma das 42 torneiras *Docol Pressmatic Alfa* depois de ativa, bem como o volume de água gasto cada vez que ela é acionada. Esse modelo de torneira possui

⁶ Recipiente simples utilizado em laboratório e geralmente de formato cilíndrico com fundo chato e um bico em sua parte superior. Eles são graduados e oferecem medidas pouco precisas

fechamento por mola.

Em seguida, cinco torneiras foram utilizadas por 25 pessoas para que fosse calculado o tempo médio no qual uma pessoa utiliza para lavar as mãos nos banheiros do local estudado. Nessa etapa foram cronometrados os tempos efetivos e de desperdício de uso das torneiras. Também foi registrado o tempo total associado à força de uso do acionador, devido a variação no tempo que a torneira permanece aberta.

Os dados da etapa 1 foram usados para calcular as médias da vazão, volume de água gasto por ativação da torneira, e tempo em que ela permanece ativa. Estes ao serem analisados, observou-se que os valores médios encontrados são: 1,07L de água utilizada a cada acionamento, 7,71s de funcionamento e 8,27L/min de vazão.

Posteriormente, o tempo de uso efetivo de 25 pessoas foi calculado, junto com o tempo no qual a torneira permaneceu ativa.

Depois da análise dos dados anteriores, foi inserido no estudo, uma torneira de fechamento eletrônico, denominada de Torneira para Lavatório de Mesa *Formatta – DocolEletric*, no qual foi calculado o tempo médio de fechamento depois que o usuário retirava as mãos do sensor.

Persona, Mandelli e Inagaki (2012), concluem que a média de tempo de uso efetivo de água é de 8,2s, no qual são gastos 1,13L e a torneira permanece aberta em média durante 14,14s, consumindo 1,95L, desperdiçando assim 0,82L, que equivale a 42% da água que passa pela torneira, cada vez que um usuário lava as mãos nesse sistema.

A torneira eletrônica proposta por Persona, Mandelli e Inagaki (2012) teve seu fechamento regulado para 0,4s após retirada das mãos e sua vazão regulada para 8,3L/min, para comparar com a vazão da torneira de fechamento por mola (8,27L/min).

Os autores afirmam ainda que com isso, o tempo médio de lavagem de mãos seria de 0,8s a mais que o efetivo, e assim a torneira gastaria 1,33L por um tempo de 9 segundos ativa, que geraria 0,2L de desperdício nesse ato, ou seja, 18% da água utilizada. O que torna viável a troca das torneiras de fechamento por mola pelas torneiras de fechamento eletrônico para reduzir o desperdício hídrico nas torneiras da Faculdade de Engenharia Elétrica e Computação da Universidade Estadual de Campinas – UNICAMP.

O uso de torneiras eletrônicas tem vantagem no quesito combate ao

desperdício hídrico quando comparado com torneiras de fechamento por mola. Ademais, ela funciona de maneira mais rápida por possuir regulagens devido a parte eletrônica envolvida e seu sistema de fechamento. Mesmo com seu custo mais elevado, essa versão eletrônica apresenta maior eficácia do que modelos tradicionais.

Apesar de possuir modelos no mercado disponível para a compra, o sistema pode ser feito e personalizado de acordo com a situação e modo de uso por meio de microcontroladores e sensores disponíveis.

3.4 Controlando o consumo de água através da internet utilizando Arduino

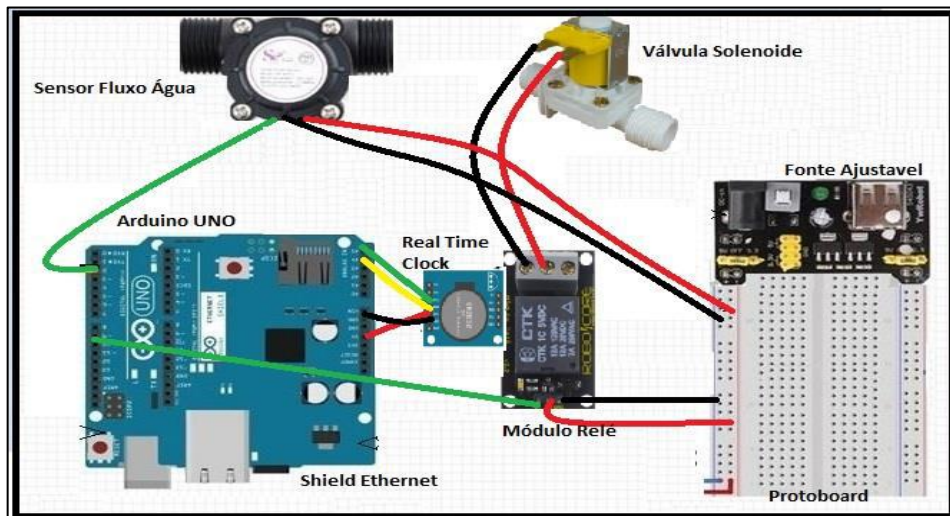
Este artigo dos autores (GROSSKOPF; PYKOSZ, 2017) relata o desenvolvimento de um protótipo que visa o monitoramento e controle do consumo de água em tempo real nas residências e indústrias. O seu gerenciamento pode ser feito à distância por meio da internet devido ao uso do Arduino, sensores, módulos e da tecnologia *Internet of Things* (IOT).

IOT torna possível a interação entre usuários e aparelhos eletrônicos através de envio ou recebimento de comandos, conforme explicam Grosskopf e Pykosz (2017). Também é possível até transformar-se em aparelhos inteligentes capazes de coletar e processar as informações onde estão conectados (OLIVEIRA, 2017 apud GROSSKOPF; PYKOSZ, 2017).

Este artigo teve como objetivo desenvolver um sistema a partir do artigo titulado de Monitoramento do consumo de água utilizando ferramenta *open source* (GROSSKOPF, 2016), ampliar funcionalidades, como um módulo *Real Time Clock* – RTC para obter maior precisão na gravação de hora e data, visando a otimização sobre o monitoramento do uso de água e dessa forma incentivar o seu uso consciente.

Foram usados como componentes para desenvolvimento do sistema: um Arduino UNO R3, uma *protoboard* de 400 pontos, um sensor de fluxo de água de ½, uma placa *shield ethernet* W5100, uma válvula solenoide, fonte para *protoboard*, módulo RTC, modelo DS1307 e um módulo relé, montados conforme a Figura 12.

Figura 12 - Esquema do projeto



Fonte: (GROSSKOPF; PYKOSZ, 2017)

O uso do *shield ethernet* é essencial para que a configuração do sistema possibilite que seu acesso possa ser feito através de um *browser* por onde será realizado o acompanhamento do consumo de água em tempo real.

Uma das funcionalidades do sistema é a configuração de horários e dias para liberação ou bloqueio do consumo de água, de acordo com os horários de maior uso. O bloqueio também pode ocorrer por decisão da análise de dados do consumo, que por meio de parâmetros define se o consumo está fora do normal, impedindo assim a passagem de água pela válvula.

O Quadro 1 foi pré-definido com horários de bloqueio e liberação de água com base em horários de uso. Uma observação importante a ser feita é que o sistema foi instalado junto ao hidrante de uma residência, o que possibilita um sistema mais eficiente no caso do bloqueio, que apesar de bloqueado, a residência possui um caixa d'água para eventuais utilizações em horários de bloqueio.

Quadro 1 - Horários abastecimento bloqueado

Horário	00:01 - 06:00	06:01 - 07:00	07:01 - 12:00	12:01 - 13:00	13:01 - 17:00	17:01 - 00:00
Dia semana						
Domingo	Bloqueado	Bloqueado	Liberado	Liberado	Liberado	Liberado
Segunda-Feira	Bloqueado	Liberado	Bloqueado	Liberado	Bloqueado	Liberado
Terça-Feira	Bloqueado	Liberado	Bloqueado	Liberado	Bloqueado	Liberado
Quarta-Feira	Bloqueado	Liberado	Bloqueado	Liberado	Bloqueado	Liberado
Quinta-Feira	Bloqueado	Liberado	Bloqueado	Liberado	Bloqueado	Liberado
Sexta-Feira	Bloqueado	Liberado	Bloqueado	Liberado	Bloqueado	Liberado
Sábado	Bloqueado	Bloqueado	Liberado	Liberado	Bloqueado	Liberado

Fonte: (GROSSKOPF; PYKOSZ, 2017)

Grosskopf e Pykosz (2017) conclui que os objetivos principais foram alcançados com a implementação e uso do sistema. E ainda mostra que projetos utilizando Arduino não são mais tão complexos há alguns anos, devido a inúmeras bibliotecas compartilhadas entre usuários da ferramenta, tutoriais, demonstrações, que tornou fácil o campo da criação de projetos para interessados nesse universo.

Esse projeto apresenta eficiência em sua proposta, devido a inserção de IOT que possibilita gerenciar o sistema à distância. O modelo de bloqueio e liberação cumpre seu papel de acordo com o algoritmo programado, porém não se torna viável caso seja aplicado em lugares onde a circulação de água é limitada, como em regiões do Nordeste. Nesses locais, as residências tendem a possuir um sistema para armazenar a maior quantidade de água possível.

Em locais como o município de Tauá no estado do Ceará, é possível através de uma tabela disponibilizada pela companhia de abastecimento, prever o dia em que será liberado a água para um determinado setor, porém não é possível prever o horário, o que dificulta ainda mais a implementação desse tipo de sistema.

4 DESENVOLVIMENTO DO PROTÓTIPO

Os procedimentos metodológicos adotados neste trabalho estão relacionados com a aplicação de metodologias de engenharia de software – ES, uma subárea da ciência da computação que estuda todos os aspectos da produção de software, desde os estágios iniciais de especificação até sua manutenção, quando este entrar em operação (SOMMERVILLE, 2011).

Neste trabalho, o desenvolvimento do projeto proposto foi organizado em quatro etapas, sendo elas:

- a) Revisão de Literatura para dar base teórica por meios de conceitos e trabalhos relacionados (ver segundo e terceiro capítulo) para a proposta desta monografia. A seção 3.1 do terceiro capítulo explica como foi realizada a busca dos trabalhos relacionados;
- b) Requisitos do protótipo (seção 4.2 Requisitos do protótipo), no qual foram obtidos os requisitos funcionais e não funcionais, bem como, os diagramas comportamentais;
- c) Projeto e Implementação do protótipo (seção 4.3), etapa em que foi construída a proposta com auxílio de diagrama de arquitetura, componentes eletrônicos e tecnologias de desenvolvimento do ambiente Arduino. A seção 4.3 também descreve e justifica as escolhas realizadas durante a implementação do projeto;
- d) Validação do protótipo (seção 4.3.6), fase em que foram adotados métodos para validação do protótipo e obtenção dos resultados relatados no quinto capítulo.

4.1 Revisão de Literatura

Este trabalho teve início com uma revisão de literatura de autores, obras e contribuições primeiramente na área de automação industrial para fundamentar o desenvolvimento da pesquisa.

Em seguida outras áreas foram exploradas para adquirir informações necessárias a fim de automatizar uma torneira de uso comum mediante a

problemática. Para isso utilizou-se das áreas de sistemas embarcados, microcontroladores e eletrônica.

Após a revisão bibliográfica, uma aplicação prática foi desenvolvida com base em obras revisadas e nas informações coletadas. Essa prática partiu da área de automação para selecionar cada parte do protótipo que iniciou com a escolha de uma torneira com acionamento/fechamento manual.

Depois, foram utilizados os conceitos de microcontroladores e sistemas embarcados para a escolha da plataforma, o que determinou também o tipo de linguagem de programação, e os sensores e componentes necessários.

4.2 Requisitos do protótipo

Nesta seção serão apresentados os requisitos levantados e os diagramas produzidos para melhor compreensão do funcionamento do protótipo.

4.2.1 Levantamento de requisitos

Os requisitos do sistema foram fundamentados juntamente com a idealização da proposta e basearam-se inicialmente no uso de uma torneira comum para tarefas casuais como lavar as mãos, lavar o rosto, escovar os dentes, entre outras.

Após levantar os requisitos de uso cotidiano de uma torneira comum, foram adicionados os requisitos que compõe a parte de automação e gerência, para que o sistema funcione de modo minimamente satisfatório.

A escolha dos requisitos ocorreu em sua grande parte de forma observacional, de modo a possibilitar que o usuário utilizasse a torneira automatizada da forma mais semelhante possível do uso de uma torneira convencional, porém focando no uso controlado da água.

Depois da definição dos requisitos da torneira automatizada, foram levantados os requisitos para um sistema de *Data logger*, que será acoplado na torneira antes e depois de automatizada a fim de gerar dados para comparação entre ambas. Esse sistema armazenará dados sobre o gasto de água em um micro SD, possibilitando a criação de tabelas dispostas na seção resultados deste trabalho.

4.2.2 Especificação de requisitos

Os requisitos funcionais e não funcionais levantados na fase anterior são dispostos nesta subseção.

Foi escolhido utilizar o nível de descrição classificado como requisitos de usuário, isto é, “declarações, em uma linguagem natural com diagramas, de quais serviços o sistema deverá fornecer a seus usuários e as restrições com as quais este deve operar” (SOMMERVILLE, 2011). Para estabelecer a prioridade dos requisitos foram adotadas as denominações “essencial”, “importante” e “desejável” (SILVA, 2016).

Essencial é o requisito sem o qual o sistema não entra em funcionamento. Requisitos essenciais são requisitos imprescindíveis, que têm que ser implementados impreterivelmente.

Importante é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implementados, mas, se não forem, o sistema poderá ser implantado e usado mesmo assim.

Desejável é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser deixados para versões posteriores do sistema, caso não haja tempo hábil para implementá-los na versão que está sendo especificada.

4.2.2.1 Requisitos funcionais

[RF001] Iniciar o protótipo

Após ser ligado numa fonte de alimentação energética adequada, o protótipo verifica se tem o cartão de memória inserido e inicia o módulo RTC, o sensor de fluxo de água e configura os pinos do sensor de distância e LEDs.

Entradas e pré-condições: O protótipo com uma fonte de alimentação adequada.

Saídas e pós-condições: O *display* LCD 16x2 deverá exibir “erro de SD Card” se o cartão não for detectado ou “Sistema Ok” caso contrário, durante 2s, e depois exibe

a hora atual no formato “H:M:S off”, na linha 00 onde $H \in \{00, 01, 02, \dots, 23\}$ e $M, S \in \{00, 01, 02, \dots, 59\}$ e “Gasto: 0.00L/min” na linha 01, e inicia a detecção de usuários.

Fluxos secundários (alternativos e de exceção): Nenhum.

Prioridade: Essencial.

[RF002] Detectar usuário

Deve ser detectado a presença de usuários da torneira dentro da zona de leitura de distância mínima de 6 cm e máxima de 30 cm.

Entradas e pré-condições: O protótipo com uma fonte de alimentação adequada e acoplado corretamente na torneira.

Saídas e pós-condições: Abre a torneira.

Fluxos secundários (alternativos e de exceção): Nenhum.

Prioridade: Essencial.

[RF003] Ativar automaticamente a torneira.

A torneira deverá ser ativada automaticamente pelo protótipo a fim de liberar o fluxo da água.

Entradas e pré-condições: A manopla da torneira estar aberta e ter detectado a presença de usuário.

Saídas e pós-condições: Inicia a leitura de dados sobre o consumo de água.

Fluxos secundários (alternativos e de exceção): O *display* LCD 16x2 deverá exibir a mensagem “Erro de gravação” se houver erro ao criar o arquivo e a gravação de dados não é efetuada.

Prioridade: Essencial.

[RF004] Contar tempo de consumo

O temporizador deve iniciar a contagem de tempo quando a torneira é acionada e o

display LCD 16x2 altera o status “off” para “on” no formato “H:M:S on”. Também deve sinalizar os estágios por meio de LEDs. Sendo t o tempo em segundos contado pelo temporizador, os quatros períodos são descritos pelos seguintes sub requisitos:

[RF00401]: Com $0s > t \leq 3s$ é ligado o LED verde;

[RF00402]: Com $3s < t \leq 6s$ são ligados os LEDs verde e amarelo;

[RF00403]: Com $6s < t \leq 9s$ são ligados os LEDs verde, amarelo e vermelho;

[RF00404]: Com $t > 9s$ é ligado apenas o LED vermelho desligando os LEDs verde e amarelo vermelho e bloqueando o protótipo.

Entradas e pré-condições: Torneira aberta.

Saídas e pós-condições: Somente o LED vermelho ligado e iniciar o fechamento da torneira.

Fluxos secundários (alternativos e de exceção): a gravação de dados é interrompida junto ao RF00404. Necessário não detectar usuário para zerar a contagem.

Prioridade: Essencial.

[RF005] Bloquear automaticamente a torneira

Após 9 segundos, a torneira deverá ser bloqueada automaticamente pelo protótipo a fim de bloquear o fluxo da água e o *display* LCD 16x2 altera o status “on” para “off” no formato “H:M:S off”.

Entradas e pré-condições: Torneira ativada após 9s.

Saídas e pós-condições: Torneira desativada, salvar dados será interrompido.

Fluxos secundários (alternativos e de exceção): Detectar usuário. Se o usuário for detectado o modo bloqueado permanecerá.

Prioridade: Essencial.

[RF006] Desativar a torneira

Ao iniciar o protótipo, seu estado natural é desativado e permanecerá assim

enquanto não identificar usuário.

Entradas e pré-condições: Torneira não sendo utilizada.

Saídas e pós-condições: RF003 em execução.

Fluxos secundários (alternativos e de exceção): Hora marcada no display e todos os *LEDs* apagados.

[RF007] Salvar dados de consumo de água

Dados sobre o consumo da água serão salvos em um arquivo de extensão “.txt” no *SD Card* enquanto o protótipo estiver ligado. Os dados a serem salvos são: data, hora, vazão em tempo real e consumo acumulado.

Entradas e pré-condições: Torneira ativada.

Saídas e pós-condições: Arquivo “.txt” com os dados salvos.

Fluxos secundários (alternativos e de exceção): O *display* LCD 16x2 deverá exibir a mensagem descrita no RF001 caso não haja o *SD Card*.

Prioridade: Importante.

[RF008] Exibir mensagens

Após conectado a torneira, o protótipo deve exibir algumas mensagens de aviso para o usuário por meio de um *display* 16x2. Essas mensagens são:

- a) “Sistema ok”, ao iniciar e não constatar erro;
- b) “Erro de *SD Card*”, ao ligar e identificar erro ou ausência do cartão de memória;
- c) “Erro de gravação”, quando houver erro ao criar um arquivo no cartão de memória para inserir dados;
- d) “00:00:00 off”, exibida na linha 0 após a iniciação do protótipo;
- e) “00:00:00 on”, exibida na linha 00 após o RF002;
- f) “Gasto: X.XXL/min”, exibido na linha 01 após o protótipo iniciar a leitura. O X representa dígitos de 0 a 9.

Prioridade: Desejável.

4.2.2.2 Requisitos não funcionais

Esta seção descreve os requisitos não funcionais do protótipo. Para uma melhor organização desta seção, os requisitos foram agrupados pelas categorias:

- a) **Usabilidade:** descreve os requisitos não funcionais associados à facilidade de uso do protótipo.
- b) **Desempenho:** contém os requisitos que descrevem os requisitos não funcionais associados à eficiência, uso de recursos e tempo de resposta do sistema.
- c) **Confiabilidade:** descreve os requisitos não funcionais associados à frequência de falha, e a robustez do sistema na recuperação destas falhas.
- d) **Padrões:** descreve os requisitos não funcionais associados a padrões ou normas que devem ser seguidos pelo sistema ou pelo seu processo de desenvolvimento.
- e) **Hardware e Software:** descreve os requisitos não funcionais associados ao hardware e software usados para desenvolver ou para executar o sistema.

Devido à pouca interatividade do protótipo com o usuário de fato, foi definido apenas um requisito não funcional para as categorias usabilidade e desempenho.

O RNF001 sendo implementado é preferível que ocorra uma única vez, tornando este um requisito de usabilidade única. O RNF002 possui um baixo grau de usabilidade, pois se tratando de um tempo baixo não irá interferir na experiência de uso.

[RNF001] Acoplamento rosqueável

Deve ser possível o acoplamento rosqueável para torneiras cromadas ou de plástico com acionamento manual por alavanca ou manopla.

Categoria: Usabilidade

Entradas e pré-condições: Torneira ser manual.

Saídas e pós-condições: Torneira pronta para uso.

Prioridade: Essencial.

[RNF002] Tempo de resposta

Ao detectar presença/ausência do usuário o sistema deve ser acionado/bloqueado em um tempo médio de 2 segundos.

Categoria: Desempenho.

Entradas e pré-condições: Acoplamento correto na torneira aberta manualmente e com fonte de alimentação adequada.

Saídas e pós-condições: Funcionamento adequado.

Prioridade: Importante.

[RNF003] Disponibilidade do protótipo

Deverá estar disponível 24h por dia durante os sete dias da semana para detectar o acionamento automaticamente, realizar contagens de tempo, fechar torneira e salvar dados de consumo se estiver o SD Card inserido.

Ocorrerá o desligamento se houver queda de energia.

Após alimentado novamente o reinício leva 6 segundos se um cartão de memória estiver inserido e cerca de 12 segundos caso esteja sem cartão de memória.

Categoria: Confiabilidade.

Entradas e pré-condições: Acoplamento correto na torneira aberta manualmente e com fonte de alimentação adequada.

Saídas e pós-condições: Funcionamento adequado.

Prioridade: Essencial.

[RNF004] Linguagem de programação

Todo o firmware será escrito através da linguagem de programação C++.

Categoria: Padrões.

Entradas e pré-condições: Nenhuma

Saídas e pós-condições: Firmware escrito no padrão da plataforma Arduino.

Prioridade: Essencial

[RNF005] Paradigma de programação

Para uma melhor capacidade de reuso de códigos deve se adotar um paradigma de programação estruturada ou orientada a objetos na escrita do firmware.

Categoria: Padrões.

Entradas e pré-condições: Nenhuma

Saídas e pós-condições: Firmware escrito de acordo com o paradigma adequado.

Prioridade: Desejável

[RNF006] Acurácia

O valor de capacidade consumida exibido no *display* deve ser em litros com precisão de duas casas decimais.

Categoria: Padrões.

Entradas e pré-condições: Acoplamento correto na torneira aberta manualmente e com fonte de alimentação adequada.

Saídas e pós-condições: Dados salvos e exibidos com precisão.

Prioridade: Importante.

[RNF007] Formato de armazenamento de dados

Os dados lidos durante o uso da torneira são armazenados no formato: “data, hora, total: 0,00 litros, Vazão: 0,00 L/min”.

Categoria: Padrões.

Entradas e pré-condições: Torneira acionada.

Saídas e pós-condições: Arquivo .txt capaz de ser aberto por softwares de planilha, onde cada “,” será a separação de uma célula.

Prioridade: Importante.

[RNF008] Dispositivos

A proposta deste trabalho é um sistema embarcado baseado na plataforma Arduino Uno Rev3 aplicado na automatização e controle do uso de uma torneira, o protótipo deve ser composto por dispositivos descritos pelo

Quadro 2. O microcontrolador ATmega328 será responsável pela interação entre os componentes do protótipo, e o único dado de usuário que será processado pelo microcontrolador é a detecção de algo físico captado pelo sensor de obstáculos infravermelho E18-D80NK quando for detectado dentro da zona de ativação da torneira.

Categoria: Hardware e Software

Entradas e pré-condições: Nenhuma

Saídas e pós-condições: Funcionamento satisfatório.

Prioridade: Essencial

Quadro 2 – Dispositivos essenciais do protótipo

Dispositivo	Tipo	Modo de ativação	Finalidade
Sensor de Obstáculos Infravermelho E18-D80NK	Sensor	Acionado ao ligar o protótipo	Faz a leitura contínua para identificar se há algo dentro da zona de leitura.
Módulo Relé simples	Atuador	Acionado de acordo com o sensor E18-D80NK	A partir da leitura do sensor de obstáculos, o relé é ativado/desativado chaveando energia para passagem ou bloqueio de água.
Válvula solenoide de 2 vias	Atuador	Acionado por meio do relé	Caso o relé seja acionado, ele fechará o contato da fonte 12 V com a válvula e permitirá a

			passagem da água.
<i>LEDs</i>	Atuador	Acionado quando o protótipo estiver em uso	<p>Ao entrar na zona de leitura, dentro da distância de 50 cm, o <i>LED</i> verde acenderá;</p> <p>Permanecendo por um período de 3 segundo a 6 segundos, o <i>LED</i> amarelo acenderá;</p> <p>Permanecendo por um período de 6 segundo a 9 segundos, o <i>LED</i> vermelho acenderá;</p> <p>Permanecendo por um período maior que 9 segundos, somente o <i>LED</i> vermelho acenderá.</p>
<i>Display 16x2</i>	Atuador	Acionado ao ligar o protótipo	Exibe informações para o usuário do protótipo, definidas pelo RF006.
Módulo <i>SD Card</i>	Atuador	Acionado após o protótipo iniciar	Armazena dados sobre o uso da torneira.
Módulo RTC DS3231	Sensor	Acionado ao ligar o protótipo	Funciona como relógio do sistema para auxiliar na gerência dos dados.
Torneira de acionamento comum	Atuador	Acionada junto com a válvula solenoide	<p>Responsável por ajustar a vazão do sistema.</p> <p>Obs: deve ficar sempre aberta para a passagem de água.</p>
Sensor de fluxo YF-S201	Sensor	Acionado ao ligar o protótipo	Responsável por captar o fluxo e quantidade gasta de água.

Fonte: (Elaborado pelo autor, 2020).

[RNF09] Tipo de *SD Card* compatível

Poderá ser utilizado cartões de memória dos tipos *SD card* ou micro SD com o

mínimo recomendado de 1 GB de armazenamento.

Categoria: Hardware e Software

Entradas e pré-condições: Nenhuma

Saídas e pós-condições: Espaço para armazenar dados sobre o consumo.

Prioridade: Desejável.

4.2.2.3 Requisitos operacionais

A fonte de energia elétrica escolhida deve garantir as seguintes condições:

- a) A placa controladora deve utilizar de 7 a 12V caso seja alimentada pelo conector Jack ou 5V se alimentada por meio de um cabo USB.
- b) A Alimentação pode ocorrer através de uma rede elétrica por meio de fontes que disponibilizam apenas a tensão determinada ou baterias.
- c) Tensão abaixo de 7V pelo conector Jack pode deixar o uso da placa Arduino UNO Rev3 instável. Caso a tensão seja acima de 12V pode ocorrer sobreaquecimento do regulador de tensão.
- d) A válvula solenoide deve ser utilizada com uma tensão de 12V.

Prioridade: Essencial.

4.2.3 Diagrama de Casos de Uso

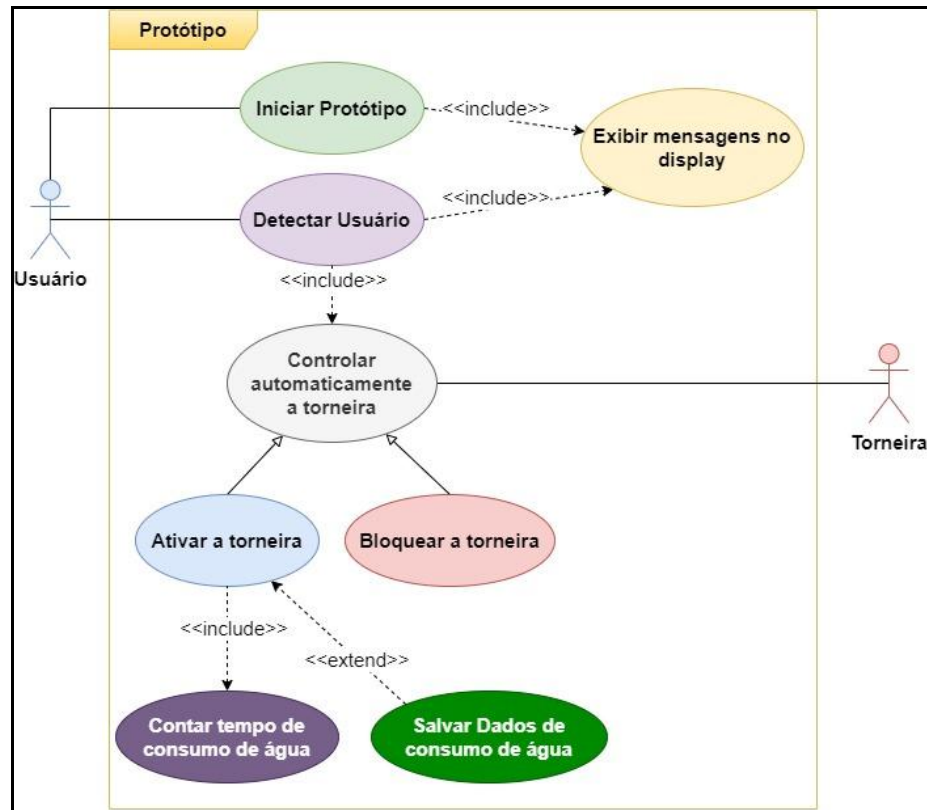
A fim de compreender melhor o comportamento do protótipo, foram desenvolvidos os diagramas de casos de uso (ver Figura 13) e estados (ver Figuras 14) da Linguagem de Modelagem Unificada (*Unified Modeling Language*, UML). Para isto foi utilizado o software online denominado de Draw.io⁷, que permite criar diversos diagramas da UML e de outras linguagens.

O diagrama mostrado na Figura 13 foi definido para exemplificar de forma mais clara, o funcionamento do protótipo. Os atores de um diagrama de casos de uso representam os papéis desempenhados por usuários ou qualquer outro componente externo ao protótipo (por exemplo, hardware e software) e que de

⁷<https://www.draw.io/>

alguma maneira utilizam os seus serviços e funções.

Figura 13 - Diagrama de caso de uso do protótipo



Fonte: (Elaborada pelo autor, 2020).

Neste caso, o ator Usuário trata-se de pessoas que irão interagir com as funções de iniciar o protótipo e a mais recorrente, detectar usuário. Ao iniciar o protótipo, obrigatoriamente será exibido mensagens no *display*, como já definido nas saídas e pós condições do RF001.

A função de detectar usuário possui como funções obrigatórias, exibir mensagens no *display* e controlar automaticamente a torneira (ator Torneira), sendo a segunda a funcionalidade que caracteriza a automação.

Controlar automaticamente a torneira possui dois modos de atuação, caso o usuário seja detectado, a funcionalidade ativada será “Ativar a torneira”, se essa condição não for satisfeita, “Bloquear a torneira” entrará em vigor.

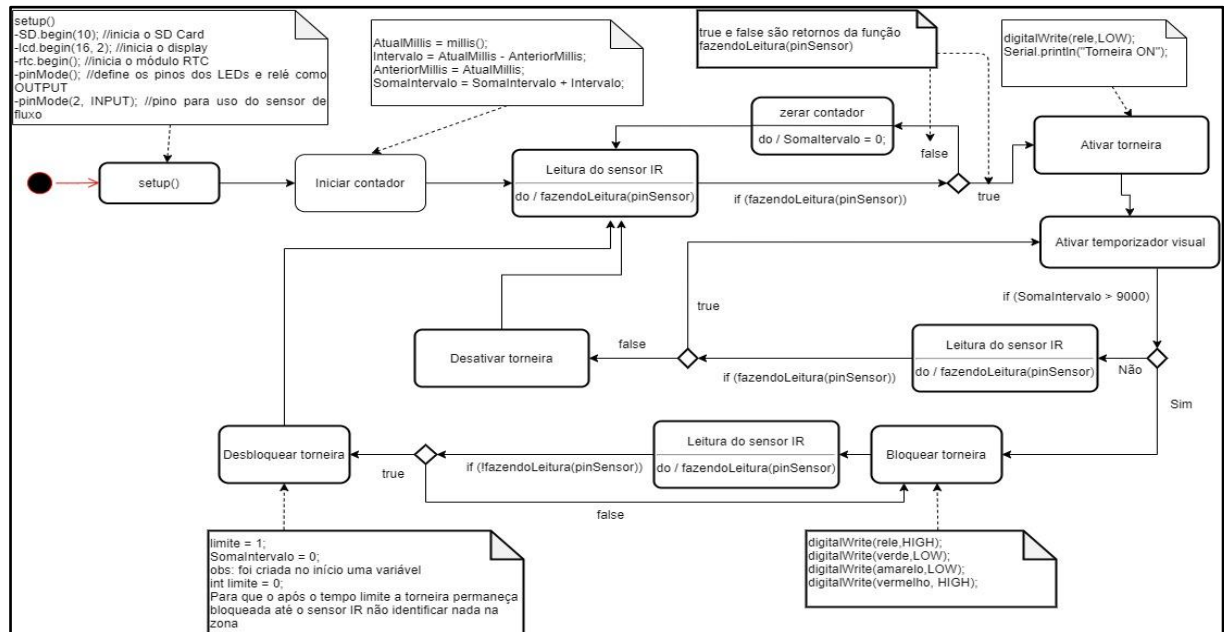
Ao ativar a torneira, pode-se ver no diagrama que obrigatoriamente a funcionalidade “Contar tempo de consumo de água” entra em funcionamento. Esta função serve para garantir o RF004, mantendo um controle de limite de uso constante do protótipo.

Há também outra função atrelada ao “Ativar torneira” denominada de “Salvar

dados de consumo de água”, porém esta necessita de memória de acordo com o RNF009 que deve ser obedecido para entrar em funcionamento.

4.2.4 Diagramas de Estados

Figura 14 - Diagrama de funcionamento da automação da torneira.



Fonte: (Elaborada pelo autor, 2020).

A Figura 14 apresenta o diagrama de funcionamento da automação da torneira, ou seja, os estados possíveis do protótipo. O estado “Leitura do sensor IR” depende do sensor de obstáculos infravermelho E18-D80NK, responsável por limitar a zona de leitura e identificar se há ou não usuário durante todo o período em que o protótipo estiver ligado. Determinado pelo [RF001], a distância máxima de 50 cm é definida com auxílio de uma régua, pois o sensor utiliza um ajuste físico na parte de trás por meio de uma chave de fenda e o processo de limitação de distância não necessita ser feito via software.

O módulo relé inicialmente mantém o circuito entre fonte externa e solenoide aberto, com o comando *HIGH* enviado pela placa Arduino. Assim que o usuário entrar na área de ativação, o microcontrolador processará essa informação transmitida pelo E18-D80NK e enviará o comando *LOW* para a porta conectada ao relé, fazendo com que o ele feche o circuito entre fonte externa e solenoide.

Quando o módulo relé receber o comando *LOW*, outra estrutura condicional será ativada. Essa estrutura funciona como um contador e foi desenvolvida

utilizando a função *millis()* que retorna um valor em milissegundos no qual o programa atual está em execução. Esse contador foi dividido em quatro estágios explicados no Quadro 3.

Quadro 3 - Estágios de funcionamento primário do protótipo

Estágio	Período	Descrição
1	Maior que 0 ms	O controlador enviará um sinal de LOW para o pino relé e ativará o LED verde, indicando que a torneira está em funcionamento.
2	Maior que 3000 ms	Ainda com o LED verde acionado, será ativado o LED amarelo, que indicará tempo de uso baixo.
3	Maior que 6000 ms	Todos os LEDs se mantêm acesos, indicando para o usuário que o tempo de uso da torneira está médio e o sistema poderá travar caso continue utilizando-a.
4	Maior que 9000 ms	No estágio 4 apenas o LED vermelho permanecerá aceso, indicando sistema bloqueado. Junto a isso será enviado o comando HIGH para o relé desativar a válvula e enquanto a função de uso do E18-D80NK retornar <i>true</i> , o sistema permanecerá bloqueado.

Fonte: (Elaborado pelo autor, 2020).

Quando estiver no último estágio e não identificar algo na zona de ativação, o protótipo irá reiniciar o contador e sairá do bloqueio voltando à posição inicial, com todos os LED apagados.

4.3 Projeto e Implementação do protótipo

Esta seção descreve o desenvolvimento do protótipo, ferramentas utilizadas, tecnologias envolvidas, arquiteturas, bibliotecas e algumas funções.

4.3.1 Arquitetura Lógica

Há duas funções padrões na IDE Arduino denominadas “*void setup(){}*” e “*void loop(){}*” no qual a estrutura lógica é implementada sobre elas.

No escopo da função “*void setup(){}*” ocorre a inicialização do módulo RTC, do *display*, a configuração do sensor de fluxo, a checagem da inserção de um *SD card* e a declaração dos pinos de LEDs e relé como saída, pois eles somente receberão sinal da placa controladora.

No escopo da função “*void loop(){}*” é localizado a maior parte da arquitetura lógica do protótipo, pois é nessa estrutura que o software permanece executando para que o protótipo continue em funcionamento. Nela ocorre chamadas de funções como a denominada “*ativaSensor*”, sua sintaxe é “*bool ativaSensor(int porta);*”. Ela é uma função booleana, no qual seu retorno pode ser “*true*” ou “*false*” e foi criada para a utilização do sensor de obstáculos E18-D80NK. O retorno “*true*” ocorre quando o sensor detecta algo através do seu feixe infravermelho e “*false*” caso nada seja detectado.

A função “*ativaSensor*” é chamada através de uma estrutura condicional “*if (ativaSensor(pinSensor))*”, e no caso do retorno satisfazer a condição é criado um arquivo nomeado de “*DATA.txt*” que dentro da estrutura irá escrever uma *String* com o padrão definido pelo RNF008 caso nenhum erro ocorra na criação. Caso apresente falha ela será reportada no *display*. Se a condição do *if* não for satisfeita a estrutura indicará apenas mensagens no *display* sendo uma delas “*off*”, como dito no RF005.

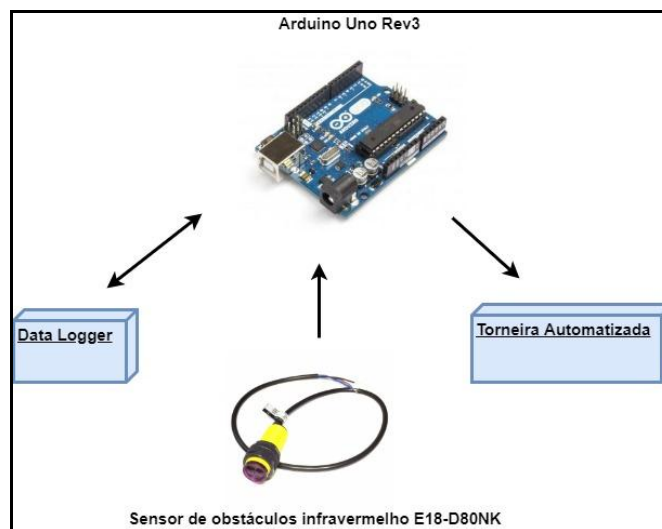
Após tentar salvar dados de uso do protótipo, é disparado um contador utilizando uma estrutura condicional para limitar o tempo de uso contínuo. Quando o contador atingir o valor máximo, o programa entra em uma estrutura definida por “*while (limite == 0)*”, “*limite*” é uma variável de controle definida com o valor “0” na parte inicial do programa e servirá para desativar a torneira nessa fase, e acender apenas o LED vermelho através das funções “*digitalWrite(rele,HIGH);*” e “*digitalWrite(vermelho,HIGH);*” respectivamente, para indicar uso máximo por vez alcançado. Para desbloquear, a função “*ativaSensor*” é chamada novamente e caso não identifique usuário, a variável de controle “*limite*” tem seu valor alterado para “1”, saindo da condicional “*while*”.

Ainda na função “*void loop(){}*” ocorrem os cálculos necessários para

contabilizar a quantidade de água de acordo com o sensor de fluxo, onde seu funcionamento ocorre por meio hélices que giram ao passar algo no sentido indicado, e por meio de um sensor de imã é possível saber quantos giros ocorreram. Para isso é necessário uma função chamada de “incpulso()” (um incrementador importante para calcular o número de giros) juntamente com a constante “5.5” disponível no *datasheet* do sensor, possibilitando saber o valor em litros/min ao multiplica-los. Ainda no “*loop()*{}”, também ocorrem os cálculos para gerar o contador que utiliza a função “*millis()*,” como ponto de partida (função no qual retorna um valor em milissegundos a partir do instante em que o programa atual está em execução).

4.3.2 Arquitetura Física

Figura 15 - Arquitetura Física do protótipo

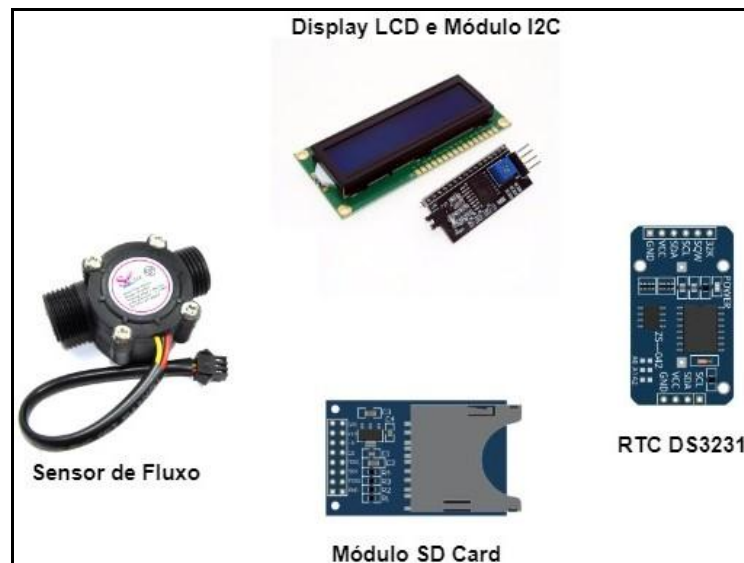


Fonte: (Elaborada pelo autor, 2020).

A Figura 15 retrata a composição do protótipo, que é formado por quatro partes principais, no qual a Placa Arduino Uno Rev3 é responsável por controlar o fluxo de instruções, envio de comandos, e interligação entre o funcionamento de cada bloco e componentes contido neles.

O Sensor de obstáculos infravermelho E18-D80NK é essencial por se tratar do componente que servirá de gatilho para funcionamento da maior parte do protótipo, detectando usuário. Sempre enviará algum sinal para a placa Arduino Uno, seja de detecção de usuário ou não.

Figura 16 - *Data Logger*

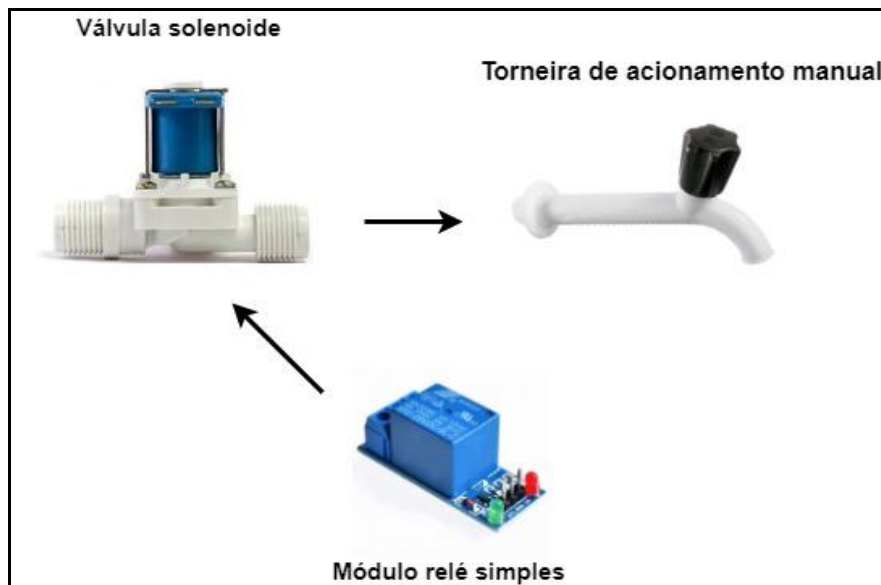


Fonte: (Elaborada pelo autor, 2020).

O bloco denominado *data logger* foi feito para gravar e mostrar dados sobre o funcionamento do protótipo. Mostrado na Figura 16, ele é composto por cinco componentes:

- a) **Sensor de fluxo:** utilizado para medir o fluxo de água, e com o auxílio de cálculos determinar a quantidade de água gasta;
- b) **RTC DS3231:** responsável por ser o relógio e calendário, possui importância relevante para o acompanhamento e análise dos dados de uso, já que devido ao seu funcionamento é possível distinguir os dados gerados;
- c) **Módulo SD Card:** acoplado um SD card é possível utilizá-lo como armazenamento de dados, gravando nele dados do sensor de fluxo e RTC;
- d) **Display LCD 16x2:** designado para mostrar dados sobre o funcionamento do protótipo, como hora (gerada pelo RTC) e quantidade de água gasta (gerada com o auxílio do sensor de fluxo);
- e) **I2C:** facilita a conexão do *display* LCD 16x2 com o a placa Arduino.

Figura 17 - Bloco da torneira automatizada



Fonte: (Elaborada pelo autor, 2020).

O bloco da torneira automatizada mostrado na Figura 17, é formado pelo uso de três componentes:

- a) Módulo relé: recebe um sinal da placa Arduino e com isso fecha ou abre o circuito do solenoide com a fonte de alimentação;
- b) Válvula solenoide: conectado na encanação é utilizado para barrar ou abrir a passagem de água para a torneira de acordo com o módulo relé;
- c) Torneira de acionamento manual: direciona a água para o usuário e possibilita regular o fluxo através da chave de registro.

4.3.3 Ambiente de Programação

O *firmware* do protótipo foi desenvolvido sob a IDE Arduino 1.8.12⁸ e posteriormente instalado no microcontrolador por meio de um cabo USB. A IDE contém muitos exemplos e bibliotecas para uso de componentes, e isso propicia maior agilidade para desenvolvimento do sistema de modo geral, teste de componentes e alterações de modo funcionamento.

Da IDE foram utilizados os recursos de *debug* para identificar erro de linguagem no código fonte, compilação, importação de bibliotecas, e o monitor serial

⁸ Disponível para download em <https://www.arduino.cc/en/main/software>.

que auxiliou para visualizar erros estruturais e lógicos durante o desenvolvimento. Ela também foi utilizada para exportar o firmware para a placa controladora.

Para que fosse possível utilizar alguns dispositivos descritos pelo

Quadro 2 foi preciso incluir no código determinadas bibliotecas mostradas pelo Quadro 4.

Quadro 4 - Bibliotecas utilizadas

Biblioteca	Finalidade
DS3231.h	Disponibiliza funções em C para utilizar o <i>Real Time Clock</i> (RTC) DS 3231, isto é, um relógio de tempo real de alta precisão e baixo consumo de energia. Essa biblioteca usa somente formato de 24 horas e que os alarmes não são implementados.
Wire.h	Essa biblioteca permite o acesso a dispositivos de protocolo I2C (<i>Inter-Integrated Circuit</i>) com endereçamento de 7 bits.
LiquidCrystal_I2C.h	Permite utilizar o módulo I2C com um <i>display</i> LCD 16x2.
SPI.h	Disponibiliza funções relacionadas com a comunicação serial de protocolo <i>Serial Peripheral Interface</i> (SPI).
SD.h	Contém funções de leitura e escrita de dados em cartão SD.

Fonte: (Elaborada pelo autor, 2020).

4.3.4 Implementação

O protótipo foi implementado com base em duas partes principais, sendo elas a automação de uma torneira e um sistema de *data logger* para gerenciamento.

A primeira parte do protótipo foi desenvolver a automação do uso de uma torneira convencional utilizando a plataforma Arduino para uso de maneira satisfatória por parte do usuário.

Inicialmente sua implementação ocorreu com o uso de um sensor ultrassônico HC-SR04 para detectar o usuário, mas devido a inconstância do sensor e sua fácil exposição a água, foi substituído por um sensor que se adequasse melhor ao propósito e então foi selecionado o E18-D80NK, no qual foi utilizado na versão final

do protótipo.

Posteriormente foi implementado um sistema de *data logger* que passou por algumas alterações em sua lógica de funcionamento.

A alteração mais significativa foi que inicialmente após ser verificado no sistema o *SD Card* permanecia ativo e gravando dados até o sistema ser desligado da energia ou que ocorresse algum erro na gravação de dados. Todavia, isso gera um mal aproveitamento do sistema de *data logger*, pois essa gravação de dados ocorria a cada 1 segundo, mesmo se o sistema não estivesse sendo utilizado.

Devido a esse problema, a lógica de funcionamento foi alterada para que os dados do funcionamento da torneira só fossem gravados no *data logger* a partir do momento em que a torneira fosse ativada. Essa modificação ocasionou um melhor aproveitamento de memória do *SD Card* e permitiu uma melhor visualização na parte de gerenciamento para saber o momento exato no qual o sistema foi ativado e desligado. O tempo de gravação de 1 segundo permaneceu na versão final do sistema, no entanto a alteração citada anteriormente foi adotada.

4.3.4.1 Conexão dos componentes e funcionamento.

Para desenvolvimento do protótipo foi utilizado uma *protoboard*, pois alguns componentes compartilham o uso dos pinos GND (terra) e 5 volts.

O componente responsável por captar dados externos e enviá-los para o microcontrolador é o sensor de obstáculos infravermelho E18-D80NK, pois estará constantemente fazendo leitura da zona determinada pelo RF001 e será primordial para o funcionamento geral da torneira, pois funcionará como dado base para gatilho do sistema. Ele foi conectado à placa Arduino utilizando o pino 7 para dados, assim como no teste do sensor, e os pinos VCC e GND foram ligados na *protoboard* que por sua vez estava ligada à placa e permitindo a conexão de qualquer dispositivo que viesse a utilizar os pinos 5 volts e GND.

Os LEDs verde, amarelo e vermelho foram ligados respectivamente nos pinos 9, 10 e 11 com um resistor de 330 Ω cada e foram utilizados para fazer um temporizador visual para o usuário, seguindo os estágios de funcionamento do Quadro 3 disponível na subseção 4.2.4.

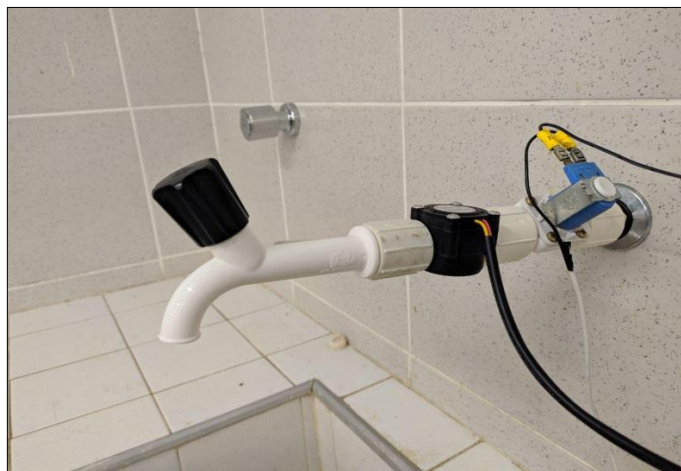
O módulo relé foi inserido entre a placa e a válvula solenoide, funcionando como chave assim como é ilustrado na Figura 17. Seus pinos VCC e GND foram

ligados na *protoboard* e o pino de sinal foi conectado na porta 6 da placa. Um dos fios da fonte de 12 volts é conectado na porta COM do módulo relé e o outro fio é conectado a válvula solenoide. Na porta NO (*Normal Open*) é conectado um fio diretamente no outro conector da válvula, permitindo que, ao ser acionado, o relé feche um circuito entre fonte e solenoide.

O sensor de fluxo foi instalado junto ao sistema de passagem de água, especificamente antes da torneira, e para isso ele foi rosqueado em um adaptador de $\frac{1}{2}$ " para $\frac{3}{4}$ " possibilitando sua conexão com a válvula solenoide na parte em que a água entra no sensor. Sua conexão com a placa controladora ocorreu utilizando o pino de sinal na porta 2, e o fios GND e 5 volts na *protoboard*.

Na saída de água da encanação foi inserido a válvula solenoide acompanhado do sensor de fluxo e em seguida de uma torneira para tanque e pia de acionamento e fechamento manual, com bitola de $\frac{1}{2}$ ". Todas essas conexões ocorreram por meio de adaptadores rosqueáveis. A composição da parte do sistema responsável pela passagem de água pode ser vista montada na Figura 18

Figura 18 - Conexão do sistema de passagem de água.



Fonte: (Elaborada pelo autor, 2020).

As informações geradas sobre a quantidade de água consumida e a data e hora da utilização da torneira, foram armazenadas em um cartão micro SD com 4 GB de armazenamento.

Isso foi possível graças ao MH-SD Card Module, que por meio de uma biblioteca já inclusa na IDE Arduino possibilita a criação de arquivos dentro de um cartão micro SD e o armazenamento de dados dentro desse arquivo.

Para gerar dados foram armazenados uma *String* composta por data e hora

geradas pelo RTC, quantidade em litros passadas pelo sensor e vazão em tempo real geradas pelo sensor de fluxo.

Esses dados serão salvos em ciclos de 1 segundo, com isso, a cada 1 segundo é gravada uma nova informação com as mesmas características, permitindo a consulta futura da informação em qualquer segundo, dado que as informações não são sobrescritas.

Esse módulo utiliza as portas de comunicação serial SPI para interação com as placas Arduino, sendo elas os pinos CS.

- a) MISO (*Master IN Slave OUT*) - Dados do *Slave* para *Master*;
- b) MOSI (*Master OUT Slave IN*) - Dados do *Master* para *Slave*;
- c) SCK (*Serial Clock*) - *Clock* de sincronização para transmissão de dados entre o *Master* e *Slave*;
- d) SS (*Slave Select*) - Seleciona qual *Slave* receberá os dados;
- e) CS - Seleção de *Chip*.

Devido a isso deve-se obter maior atenção para a conexão desse componente, pois as portas SPI das placas Arduino variam conforme o modelo utilizado.

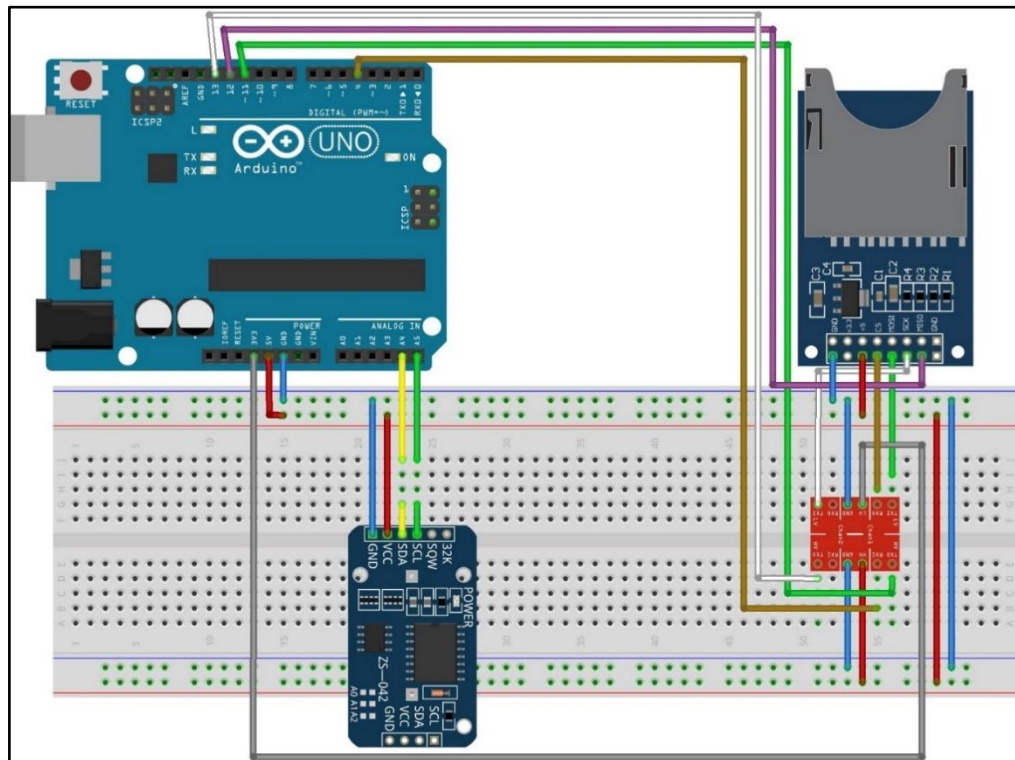
Como no protótipo a placa utilizada foi a Arduino Uno Rev3, o módulo *SD Card* teve seus pinos MOSI, MISO, SCK e SC conectados respectivamente nas portas 11, 12, 13 e 4 como mostra a Figura 19.

Para o módulo *SD Card* foi necessário a utilização de um conversor de nível lógico, pois os pinos de sinal trabalham com uma tensão de 3,3 volts.

O protocolo I2C (ou Inter-IC) é um barramento bidirecional simples que utiliza apenas duas linhas de comunicação para controle entre os CIs: a linha serial de dados (*Serial Data Line*, ou SDA) e a linha serial do clock (*Serial Clock Line*, ou SCL) (MCROBERTS, 2011). O módulo RTC utiliza comunicação I2C e para isso se comunica com a placa através desses pinos.

Na placa Arduino Uno Rev3 esses pinos estão definidos nas portas A4 e A5 respectivamente.

Figura 19 - Conexão do sistema de *data logger*.



Fonte: (Elaborada pelo autor, 2020).

4.3.5 Verificação de Componentes

Os componentes do sistema dispostos no Quadro 2, passaram pelos testes de verificação descritos nesta seção.

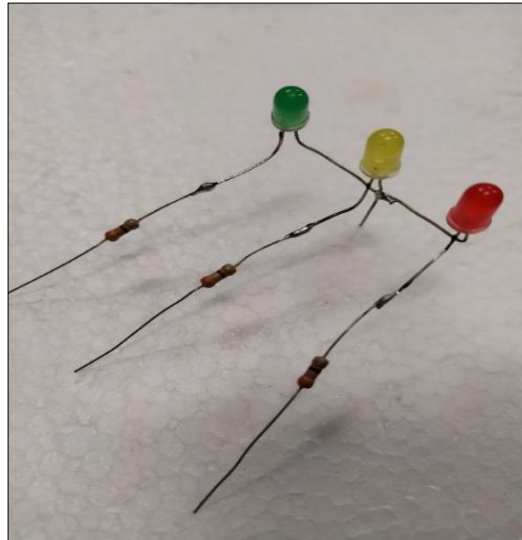
Dentre os modelos de Arduino, o utilizado foi o Arduino Uno Rev3, por questão de usabilidade, pois é extremamente aplicável para projetos em fase de desenvolvimento por possuir vários pinos de entrada e saída, bem como portas de alimentação onde uma delas é usada para conexão com o computador que será usado para programar o código e carregá-lo na placa.

Uma ligação entre o sensor de obstáculos infravermelho E18-D80NK e placa Arduino Uno Rev3 foi feita para teste de precisão do sensor utilizando uma régua para conferir o resultado e validá-lo. Para ajustar a distância do E18-D80NK basta utilizar uma régua para conferir a distância e com o uso de uma chave de fenda ir girando um pequeno parafuso que se encontra atrás do sensor, enquanto houver algo sendo detectado um *LED* vermelho posicionado no sensor acenderá, caso nada seja detectado, ele apagará.

A conexão entre a placa e sensor ocorreu utilizando a porta 8 do Arduino Uno

para controle de detecção e as portas GND e 5V para a alimentação do sensor. Para maior interatividade com o usuário, foram introduzidos três *LEDs* seguindo o esquema de cores de um semáforo igualmente a Figura 20.

Figura 20 – Esquema de LEDs e resistores.



Fonte: (Elaborada pelo autor, 2020).

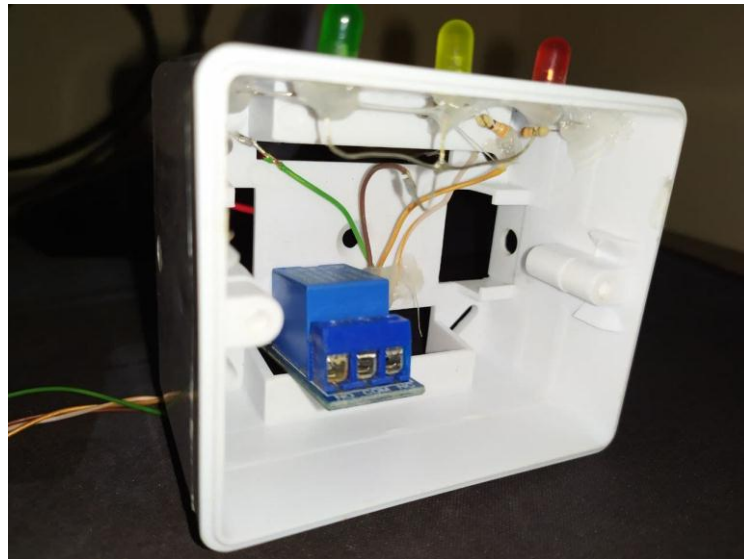
Os LEDs foram usados em conjunto com a função *delay(unsigned long ms)* a fim de pausar a execução do sistema durante um tempo em milissegundos.

Para validar o conjunto de LEDs foi desenvolvido um esquema de LED sequencial de tal forma que cada LED ficava aceso durante 1s na ordem verde, amarelo, vermelho e depois reiniciava a sequência.

Os próximos a serem testados foram o módulo relé e a válvula solenoide, bastando apenas declarar o pino de conexão como saída dentro do escopo da função *void setup()* para o relé e no escopo da função *void loop()*, foi desenvolvido um código para teste de tal maneira que o relé fechasse o circuito entre a válvula e a fonte durante 1s e deixasse aberto durante 5s utilizando a função *delay()*.

Após a validação dos LEDs e relé, eles foram acomodados dentro de uma caixa de tomada de 6,3 cm x 7,5 cm x 4,7 cm, como mostrado na Figura 21.

Figura 21 - Acoplamento para relé e conjunto de *LEDs*



Fonte: (Elaborada pelo autor, 2020).

Para o módulo RTC DS3231 a verificação ocorreu de tempos em tempos, no qual teve que ser gravado inicialmente no módulo a data e hora atual e em momentos aleatórios ocorria uma comparação com o relógio do computador utilizado para testes.

As informações geradas sobre a quantidade de água consumida, bem como a data e hora da utilização da torneira, foram armazenadas em um cartão micro SD com 4 GB de armazenamento.

Isso foi possível graças ao *MH-SD Card Module*, que por meio de uma biblioteca já inclusa na IDE Arduino possibilita a criação de arquivos dentro de um cartão micro SD e o armazenamento de dados dentro desse arquivo.

Para gerar dados foram armazenados uma *String* composta por data, hora, quantidade de água em litros passadas pelo sensor e vazão em tempo real.

Esses dados são salvos em ciclos de 1 segundo, com isso, a cada 1 segundo é gravada uma nova informação com as mesmas características, permitindo a consulta futura da informação em qualquer segundo, dado que as informações não são sobrescritas.

Figura 22 - Copo medidor

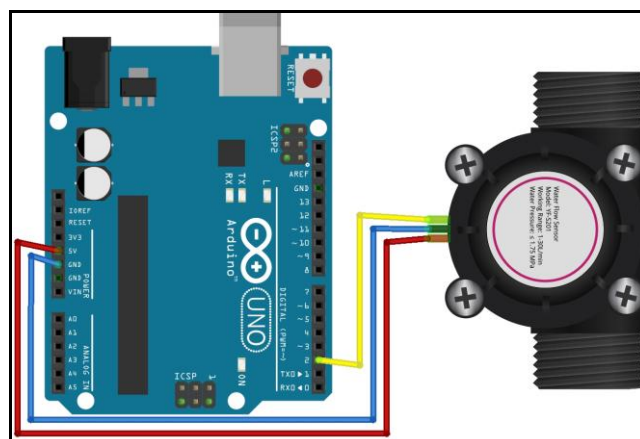


Fonte: (Elaborada pelo autor, 2020).

Capaz de medir um fluxo de até 30L/min, o sensor de fluxo YF-S201 ficou responsável por medir a quantidade de água consumida e o fluxo em tempo real, visando essa função utilizou-se para sua verificação um copo medidor igualmente o da Figura 22 com marcas que medem mililitros, para que fossem comparados com a quantidade mostrada de acordo com o YF-S201.

A ligação do sensor de fluxo com a placa Arduino ocorreu pela porta 2, e por meio desta é possível habilitar a interrupção para que o sensor funcione de forma adequada. Para entender melhor a conexão desse dispositivo a conexão citada pode ser vista na Figura 23.

Figura 23 - Conexão do sensor de fluxo com o Arduino Uno



Fonte: (Elaborada pelo autor, 2020).

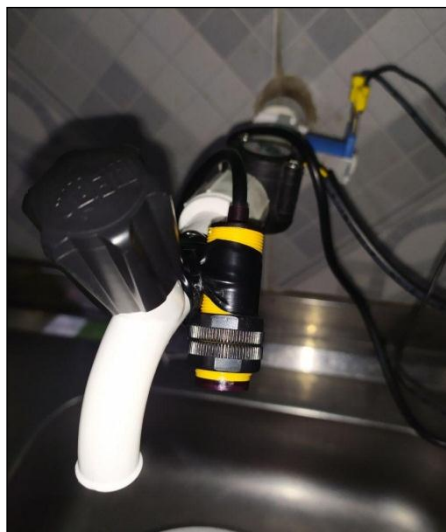
Para finalizar a verificação dos componentes, o último testado foi o *display* 16x2, nele foi inserido a frase “Torneira automatizada”, que permanecia sendo

mostrada durante 2 segundos e desaparecia durante 1 segundo, repetindo esse ciclo enquanto o sistema estivesse ligado.

Para a conexão do *display* foi utilizado um módulo I2C, com isso o número de fios conectado a placa controladora foi bastante reduzido, facilitando a conexão e possibilitando que o arduino Uno Rev 3 seja utilizado na versão final do protótipo.

4.3.6 Validação do protótipo

Figura 24 - Torneira com sensor reposicionado



Fonte: (Elaborada pelo autor, 2020).

Antes da validação, houve um reposicionamento do sensor E18-D80NK. Sua antiga localização era embaixo do sistema de passagem de água apontado para o usuário. Na sua versão final como mostra a Figura 24 o sensor foi posicionado na torneira por meio de fita isolante para prendê-lo, apontado para baixo.

O sistema pode ser visto de maneira um pouco mais ampla na Figura 25, que mostra o sistema conectado e pronto para uso.

Figura 25 - Protótipo pronto para uso



Fonte: (Elaborada pelo autor, 2020).

4.3.6.1 Metodologia de teste.

Para validar o protótipo, foi desenvolvida uma metodologia de testes, visando o uso de uma torneira comum para atividades do dia a dia.

Foram selecionados três usuários de maneira aleatória para uma série de testes dividida em três atividades, sendo elas:

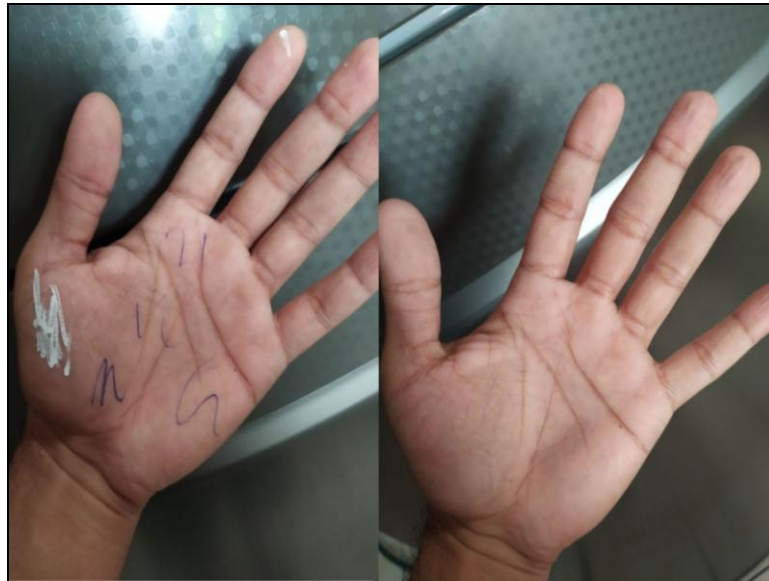
- a) Lavar as mãos com detergente neutro;
- b) Lavar o rosto com sabonete líquido;
- c) Sujar as mãos e lavar em seguida.

Os testes descritos foram feitos usando a torneira convencional e a mesma torneira, porém automatizada, ambas com o *data logger* inserido. Com isso torna-se possível a análise e comparação dos dados de ambas as torneiras. Ficou a critério do usuário ajustar a vazão da água por meio chave de registro.

Na atividade denominada “sujar as mãos e lavar em seguida”, foi dado ao usuário uma caneta e um corretivo branco para que ele riscasse e sujasse uma de suas mãos e então com o auxílio da torneira e detergente neutro, pudesse ser limpa como mostra a

Figura 26.

Figura 26 – Mão suja / Mão limpa.



Fonte: (Elaborada pelo autor, 2020).

Após coletar o arquivo de dados disponíveis no *SD card*, foi utilizado o Microsoft Excel para abri-lo. Com o software citado, é possível delimitar as colunas por meio de vírgulas e espaços, o que possibilita maior velocidade para tratar os dados. Outro fator importante, é a ferramenta de buscas da plataforma, com ela é possível selecionar uma data ou hora específica e encontrar os dados relacionados.

Para auxiliar nos testes, outro artifício utilizado foi a fotografia do *display* como na Figura 27, antes de iniciar um teste e ao finalizar, assim no momento de tratamento dos dados bastou selecionar os dados do intervalo fotografado.

Figura 27 - *Display* ao finalizar um teste do protótipo



Fonte: (AUTOR, 2020).

5 RESULTADOS E DISCUSSÕES

Este capítulo apresenta os resultados obtidos por meio do *data logger* da torneira de acionamento manual e da torneira automatizada, e por fim será feita uma análise comparativa dos dados.

Devido a pandemia de coronavírus enfrentada atualmente, os testes foram feitos com um número extremamente reduzido de usuários, no qual foi relatado para eles como o sistema funcionava, mas não era sabido o real propósito, no caso a economia de água.

5.1 Tratamento de dados

Aplicando a metodologia de testes descrita na subseção 4.3.6.1 e usando o *data logger* do sistema, foram obtidos os dados sobre o consumo da torneira em ambos os estados, automatizada e por acionamento manual. Os dados foram abertos utilizando o *software* Excel, no qual pode-se dividir as células por espaço, gerando uma tabela de dados e facilitando o tratamento deles.

Todas as linhas no qual a vazão estava em “0.00 L/min” foram excluídas, pois isso significa que a torneira não estava em uso.

Os dados foram separados entre uso da torneira comum e uso da torneira automatizada e dentro de cada categoria foram separados por etapa de testes. Posteriormente foi calculado a diferença do total gasto entre o início e o fim de cada etapa, obtendo o total gasto individualmente por etapa.

5.2 Resultados obtidos e análise

As Tabelas, Tabela 2 e

Tabela 3 contêm os dados já tratados sobre o consumo da torneira. Em ambas é possível visualizar as quantidades consumidas por usuário em cada teste e o total por teste.

Tabela 2 - Consumo da torneira com acionamento manual

	Teste a (L)	Teste b (L)	Teste c (L)
Usuário 1	0,62	2,74	1,71
Usuário 2	0,97	2,56	5,81
Usuário 3	2,27	3,48	9,09
Total (L)	3,86	8,78	18,38

Fonte: (Elaborada pelo autor, 2020).

Tabela 3 - Consumo da torneira automatizada

	Teste a (L)	Teste b (L)	Teste c (L)
Usuário 1	0,49	0,56	0,79
Usuário 2	0,73	0,56	1,52
Usuário 3	1,04	0,68	0,94
Total (L)	2,26	1,8	3,25

Fonte: (Elaborada pelo autor, 2020).

Analisando as tabelas, é possível visualizar resultados próximos e resultados com valores distantes um do outro, comparando os valores obtidos no mesmo modelo de teste e pelo mesmo usuário, porém com sistemas diferentes.

No “teste a” todos os usuários aplicaram pouco produto e como suas mãos não estavam sujas, foi possível retirar rapidamente a espuma causada pelo produto, por isso os resultados foram próximos.

No “teste b” houve uma economia hídrica geral de aproximadamente 79%. Isso ocorreu devido ao processo de fechar os olhos ao passar sabonete líquido no rosto, pois os olhos fechados causam dificuldade para fechar o registro da torneira. Essa dificuldade foi sanada pelo sistema automatizado, pois o usuário não precisou identificar a chave de registro para fechar a torneira e sempre ao colocar as mãos no rosto para o processo de lavagem, a torneira desligava automaticamente.

No “teste c” os usuários 2 e 3, permaneceram com a torneira ligada por mais tempo como podemos ver na Tabela 2, porém ao utilizar a torneira automatizada, o consumo caiu consideravelmente. Um dos motivos para isso, ocorreu pelo fato de os usuários esfregarem bastante as mãos fora da zona de leitura da torneira automatizada e utilizá-la apenas para retirar o detergente junto com a sujeira das mãos. O mesmo ocorreu com a torneira comum, porém como suas mãos estavam

sujas e coberta de detergente, foi preferível não tocar na chave de registro.

Coletando depoimentos dos usuários sobre a experiência de uso, foi relatado pelos três usuários que o uso foi satisfatório. O “usuário 1” relatou que a água da torneira estava saindo com muita pressão e isso fez com que a água espalhasse ao bater na sua mão e o molhando. Esse problema foi resolvido por ele ao ajustar a vazão da torneira. O relato foi constatado apenas pelo “usuário 1”, pois o ajuste da vazão foi satisfatório para os outros usuários.

Outro relato foi o fato de a torneira estar posicionada distante da parede fazendo com que a água pudesse respingar no chão durante a utilização da torneira automatizada. Isso pode ser resolvido dependendo do ambiente instalado e um exemplo de fácil utilização é em torneiras de bancadas, no qual a válvula solenoide e o sensor de fluxo pode ser instalado embaixo da pia, fazendo com o que não haja alteração no posicionamento da torneira.

Ainda sobre os relatos coletados, foi sugerido que o *display* ficasse mais próximo do usuário para uma melhor visualização dos dados. Para isso deve ser desenvolvido uma case contra a proteção de água para o sistema, pois ele só havia sido posicionado a uma certa distância devido a exposição dos componentes eletrônicos como o *display* LCD que podia queimar caso entrassem em contato com a água.

Sobre a sensação do consumo, todos não perceberam o efeito causado pelo uso da torneira automatizada, ou seja, o baixo consumo hídrico causado pelo uso da torneira não afetou a tarefa a ser realizada se comparado com a torneira convencional.

6 CONCLUSÃO

Baseado nos resultados apresentados no capítulo anterior e coleta dos depoimentos dos usuários, a torneira automatizada mostrou potencial para o uso de uma menor quantidade de água e seus resultados apontam para um sistema capaz de auxiliar no combate contra o consumo inadequado de água em residências, pois alterando diretamente a maneira comportamental com que o usuário interage com a torneira foi capaz de reduzir o uso desnecessário de água.

O sistema de *data logger*, é uma parte importante do projeto, pois além de captar os dados do consumo, mostram eles para os usuários, com uma possibilidade a mais de impactar o usuário e gerenciar sua utilização.

O temporizador utilizado como uma maneira visual simplificada também atua de forma eficaz, pois as visualizações de cores acesas são vistas com maior facilidade do que os dados apresentados no *display*, além de servir como sistema de segurança no caso algo permaneça na frente do sensor de obstáculos por muito tempo.

O protótipo está passível de alterações que poderão ser futuramente implementadas, como a compactação do sistema, no qual tornará mais fácil o manuseio e deixará os itens sensíveis ao contato com a água, protegido.

Outra implementação que poderá ser feita, é a conexão da placa Arduino com a *internet*, assim poderá ser desenvolvido aplicações que armazenam os dados em nuvem, e isso tornará poderá elevar a gama de tratamento e visualização dos dados, não necessitará de *display* para visualização e nem cartão de memória. Essa conexão poderá ser feita por meio de um módulo *wifi* para Arduino, ou uma placa similar como a ESP 32 que possui *bluetooth* e *wifi* integrado.

REFERÊNCIAS

ALBANI, V. D. C. **Lógica de programação**: Curso Técnico em Informática. Colatina: CEAD / IFES, 2010. 104 p. Disponível em: http://proedu.rnp.br/bitstream/handle/123456789/704/Logica_Programacao_web.pdf?sequence=3&isAllowed=y. Acesso em: 24 set. 2020.

ARDUINO IDE. Versão 1.8.9. 2019. Disponível em: <https://www.arduino.cc/en/main/OldSoftwareReleases>. Acesso em: 24 set. 2020

ARDUINO.CC. **ARDUINO UNO REV3**. Disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3>. Acesso em: 24 set. 2020.

ATMEL. **ATmega328P**. Disponível em: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf. Acesso em: 24 set. 2020.

ATMEL. **ATmel ATmega640/V-1280/V-1281/V-2560/V-2561/V**. Disponível em: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf. Acesso em: 24 set. 2020.

COGERH. **Situação Hídrica Semanal**. 2017. Disponível em: <http://www.hidro.ce.gov.br/nt/noticia/show/4849>. Acesso em: 20 maio 2019.

CUNHA, L. **Relés e contadores**. O setor elétrico, outubro 2009. Disponível em: http://www.instalacoeseltricas.com/download/Radiografia_reles_contatores_out09.pdf. Acessado em: 25 jun. 2018.

DE NEGRI, V. J. **Introdução aos sistemas para automação e controle industrial**. Florianópolis: LASHIP/EMC/UFSC, 2004.

DEMAE. **Consumo de água**. Disponível em: <https://www.demae.go.gov.br/projetos/consumo-de-agua/>. Acesso em: 24 set. 2020.

FELIX, B.; VILELA, T. Z.; SILVA, E. M. DA. **A utilização da placa arduino na criação de um sistema de controle de desperdício de água e acessibilidade nas escolas**. Anais da II Mostra Nacional de Robótica (MNR) . 2012. Disponível em: <http://sistemaolimpico.org/midias/uploads/1dec6034810e559f5d5d269dacfe3587.pdf>. Acesso em: 24 set. 2020.

FUNCEME. **Calendário das Chuvas no Estado do Ceará**. Disponível em: <http://www5.funceme.br/app/calendario/produto/ceara/media/anual>. Acesso em: 01 set. 2020.

GARCIA, F. D. **Intel 8051**. Disponível em: <https://www.embarcados.com.br/intel-8051/>. Acesso em: 5 dez. 2019.

General Comment No. 15: **The Right to Water** (Arts. 11 and 12 of the Covenant), 2003. Disponível em: <https://www.refworld.org/pdfid/4538838d11.pdf>. Acesso em: 01 set. 2020.

GIMENEZ, S. P. **Microcontroladores 8051**. São Paulo: Pearson Education do Brasil Ltda., 2002.

GONÇALVES, R. F. **Uso Racional da Água em Edificações**. Vitória: ABES, 2006.

GONÇALVES, R. F. **Conservação de água e energia em sistemas prediais e públicos de abastecimento de água**. Rio de Janeiro: ABES, 2009. 352p.

GROOVER, M. **Automação industrial e sistemas de manufatura**. 3. ed. São Paulo: Pearson Prentice Hall, 2011.

GROSSKOPF, P.; PYKOSZ, L. C. Controlando o consumo de água através da Internet utilizando Arduino. In: Congresso Nacional de Inovação e Tecnologia. 2. **Anais** [...]. São Bento do Sul: SCINOVA, 2017. Disponível em: <https://docplayer.com.br/59485179-Controlando-o-consumo-de-agua-atraves-da-internet-utilizando-arduino.html>. Acesso em: 24 set. 2020

MCROBERTS, Michael. **Arduino básico**. São Paulo: Novatec, 2011.

MICROCHIP. **Atmega328p**. Disponível em: <https://www.microchip.com/wwwproducts/en/atmega328p>. Acesso em: 24 set. 2020.

MOTA, A. **O que é arduino e como funciona?**. 14 de maio de 2017. Disponível em: <https://portal.vidadesilicio.com.br/o-que-e-arduino-e-como-funciona/> Acesso em: 24 set. 2020.

PENIDO, É. DE C. C.; TRINDADE, R. S. **Microcontroladores**. Ouro Preto - MG: Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, 2013. Disponível em: http://estudio01.proj.ufsm.br/cadernos/ifmg/tecnico_automacao_industrial/microcontroladores.pdf. Acesso em: 24 set. 2020.

PERSONA, G.; INAGAKI, G. Y. M. **Consumo de água nas torneiras dos banheiros da feec**. Ciências do ambiente. UNICAMP. 2012.

ROGGIA, L.; CARDOZO, R. F. **Automação industrial**. Santa Maria: Universidade Federal de Santa Maria, 2016.

SANTOS, D. M. **Projeto de sistemas embarcados**: Um estudo de caso baseado em microcontrolador e seguindo AOSD. 2006. 54 f. Trabalho de Conclusão de Curso (Graduação Ciências da Computação) – Universidade Federal de Santa Catarina, Centro Tecnológico - Florianópolis SC, 2006. https://repositorio.ufsc.br/bitstream/handle/123456789/183926/danillo-moura_monografia.pdf?sequence=-1&isAllowed=y. Acesso em: 24 set. 2020.

SILVA, Felipe R. **Documento de Requisitos do Sistema**: Módulo de Avaliação

Acadêmica do Siga Ensino (Versão 0.1). Universidade Federal Juiz de Fora - Campus Governador Valadares, 2016. Disponível em: <https://www2.ufjf.br/diavi/files/2016/07/DocumentosdeRequisitosdoSistema.pdf>. Acesso em: 27 set. 2020.

SNIS - Sistema Nacional de Informação sobre Saneamento. **Diagnóstico dos Serviços de Água e Esgotos**. 2016. Disponível em: <http://www.snis.gov.br/diagnostico-agua-e-esgotos/diagnostico-ae-2016>. Acesso em: 23 jun. 2018.

SOMMERVILLE, Ian. **Engenharia de Software**. 9.ed. São Paulo, SP: Pearson Prentice Hall, 2011.

SOUZA, Anderson R. et al. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC. **Revista Brasileira de Ensino de Física**, [s.l.], v. 33, n. 1, p.01-05, mar. 2011. FapUNIFESP (SciELO). Disponível em: <http://dx.doi.org/10.1590/s1806-11172011000100026>. Acesso em: 24 set.2020

SOUZA, F. **Arduino MEGA 2560**: Embarcados. 04 de março de 2020. Disponível em: <https://www.embarcados.com.br/arduino-mega-2560/>. Acesso em: 24 set. 2020.

VARGAS, R. S. **Sistemas Embarcados**: Acoplamento do Soft-Core Plasma ao barramento OPB de um PowerPC 405. Florianópolis-SCUniversidade Federal de Santa Catarina, 2007.

APÊNDICE – CÓDIGO FONTE DO SISTEMA DE TORNEIRA AUTOMATIZADA

Link do github para acesso ao código fonte do sistema desenvolvido nesta monografia: <https://github.com/brunomontes/TorneiraAutomatizadaFinalVer.git>