



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
UNIDADE ACADÊMICA DE EDUCAÇÃO A DISTÂNCIA E TECNOLOGIA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Sistema IoT de Monitoramento do Nível da Água de Rios para Cidades Inteligentes

Por

João Luiz da Silva Ramos

Recife,
Fevereiro/2021



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
UNIDADE ACADÊMICA DE EDUCAÇÃO A DISTÂNCIA E TECNOLOGIA
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JOÃO LUIZ DA SILVA RAMOS

Sistema IoT de Monitoramento do Nível da Água de Rios para Cidades Inteligentes

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação da Unidade Acadêmica de Educação a Distância e Tecnologia da Universidade Federal Rural de Pernambuco como requisito parcial à obtenção do grau de Bacharel.

Orientador: Prof. Dr. Obionor de Oliveira Nóbrega

Recife,
Fevereiro/2021

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

- R175s Ramos, João Luiz da Silva
Sistema IoT de Monitoramento do Nível da Água de Rios para Cidades Inteligentes / João Luiz da Silva Ramos. -
2021.
42 f. : il.
- Orientador: Obionor de Oliveira Nobrega.
Inclui referências, apêndice(s) e anexo(s).
- Trabalho de Conclusão de Curso (Graduação) - Universidade Federal Rural de Pernambuco, Bacharelado em
Sistemas da Informação, Recife, 2021.
1. Cidades Inteligentes. 2. Internet das Coisas. 3. Nível da água. 4. Arduino. I. Nobrega, Obionor de Oliveira,
orient. II. Título

CDD 004

UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
UNIDADE ACADÊMICA DE EDUCAÇÃO A DISTÂNCIA E TECNOLOGIA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

JOÃO LUIZ DA SILVA RAMOS

**Sistema IoT de Monitoramento do Nível da Água de Rios para Cidades
Inteligentes**

Trabalho de Conclusão de Curso julgado adequado para obtenção do título de Bacharel em Sistemas de Informação, defendido e aprovado por unanimidade em 23/02/2021 pela banca examinadora.

Banca Examinadora:

Prof. Dr. Obionor de Oliveira Nóbrega
Orientador
Universidade Federal Rural de Pernambuco

Prof. Ma. Bianca Carneiro Ribeiro
Universidade Federal Rural de Pernambuco

Prof. Dra. Juliana Regueira Basto Diniz
Universidade Federal Rural de Pernambuco

Dedico esta pesquisa aos meus pais,
meus maiores e melhores orientadores na
vida.

AGRADECIMENTOS

Quero agradecer primeiramente a Deus, pois sem Ele nada disso seria possível.

Todo esse trabalho dedico aos meus pais Luciana e Luiz, minhas irmãs Luana e Vitória, minha avó M^a. Luiza e a todos os meus amigos, em especial Carlos Júnior, Eveline Amélia Ribas, Luís Carlos e Reidylla Lima. Vocês me motivam todos os dias. Muito obrigado.

Agradeço ao IFPE – Campus Palmares, a instituição que gerou em mim o interesse em seguir carreira na área de tecnologia. Também sou extremamente grato ao Polo UAB – Palmares, instituição pela qual frequentei presencialmente as aulas do curso de Sistemas de Informação, ofertado pela UFRPE. Macir Reinaldo, Geymeesson Silva e a tantos outros membros do polo. Vocês foram fundamentais para a conclusão deste trabalho.

Meus sinceros agradecimentos a Universidade Federal Rural de Pernambuco e ao corpo docente da mesma, em especial ao Prof^o. Dr^o. Obionor de Oliveira Nóbrega, por ter me orientado durante toda a elaboração deste trabalho.

“Eu sou; eu fui. Eu quero ser.”

(Joe Hill)

RESUMO

Atualmente, as fortes chuvas são as principais causas de desastres naturais no Brasil, ocasionando o transbordamento dos rios e transtornos para a comunidade atingida por este problema. Diante disso, desenvolveu-se o protótipo de um dispositivo de baixo custo, cujo principal objetivo é monitorar e transmitir os dados obtidos (volume do nível da água dos rios). Considerando a transmissão e análise dos dados, o dispositivo está habilitado no conceito de Cidades Inteligentes e Internet das Coisas, pois utiliza-se da tecnologia da informação para coletar, analisar e transmitir dados. Para a construção do protótipo, utilizou-se uma placa microcontroladora Arduino e alguns outros componentes eletrônicos. O código-fonte do dispositivo foi desenvolvido através do Arduino IDE, um compilador de códigos, baseado na linguagem de programação C/C++, onde os dados recebidos pelo hardware serão interpretados e analisados. Um teste de acurácia foi aplicado para identificar e relacionar os valores obtidos pelo dispositivo, onde os resultados foram satisfatórios, considerando que os dados de distância capturados pelo dispositivo foram precisos, comparados com a grandeza física real. Foi realizada uma simulação em um recipiente cilíndrico (onde tal cenário pode ser feita uma co-relação com um rio), onde os resultados atingiram as metas esperadas: através do código desenvolvido, calcular a distância entre o nível da água e o dispositivo, interpretar o valor calculado e acender o LED relacionado à situação do nível da água (LED vermelho para o nível alto, LED amarelo para o nível médio ou LED verde para o nível baixo). Diante desse resultado, é possível validar a funcionalidade do código desenvolvido e do *hardware* construído.

Palavras-chave: Cidades Inteligentes, Internet das Coisas, Nível da Água, Arduino.

ABSTRACT

Currently, heavy rains are the main causes of natural disasters in Brazil, causing rivers to overflow and disturbances for the community affected by this problem. Therefore, the prototype of a low-cost device was developed, whose main objective is to monitor and transmit the data obtained (volume of the water level of the rivers). Considering the transmission and analysis of data, the device is enabled in the concept of Smart Cities and Internet of Things, as it uses information technology to collect, analyze and transmit data. For the construction of the prototype, an Arduino microcontroller board and some other electronic components were used. The source code of the device was developed through Arduino IDE, a code compiler, based on the C / C ++ programming language, where the data received by the hardware will be interpreted and analyzed. An accuracy test was applied to identify and relate the values obtained by the device, where the results were satisfactory, considering that the distance data captured by the device were accurate, compared to the actual physical quantity. A simulation was carried out in a cylindrical container (where such a scenario can be made a relationship with a river), where the results reached the expected goals: using the developed code, calculate the distance between the water level and the device, interpret the calculated value and turn on the LED related to the water level situation (red LED for high level, yellow LED for medium level or green LED for low level). Given this result, it is possible to validate the functionality of the developed code and the built *hardware*.

Keywords: Internet Of Things, Smart Cities, Water Level, Arduino.

LISTA DE FIGURAS

Figura 1 - Cenário - Simulação	20
Figura 2 - Visão frontal Arduino Uno	21
Figura 3 - Visão frontal Sensor Ultrassônico HC-SR04	22
Figura 4 - Visão frontal <i>Protoboard</i>	23
Figura 5 - Visão frontal <i>Jumpers</i>	23
Figura 6 - Fluxograma do projeto	25
Figura 7 - Modelagem do protótipo	26
Figura 8 - Simulação do projeto	28
Figura 9 - Variações dos resultados obtidos durante o teste	28

LISTA DE QUADROS

Quadro 1 - Componentes utilizados no projeto	19
Quadro 2 - Representação dos LEDs	22
Quadro 3 - Representação da modelagem do protótipo	24
Quadro 4 - Funções desenvolvidas	27

LISTA DE ABREVIATURAS E SIGLAS

C/C++	Linguagem de programação.
GND	Filtro de densidade neutra (<i>Graduated Neutral Density filter</i>)
HC-SR04	Modelo – Sensor Ultrassônico
IDE	Interface de desenvolvimento.
IOT	Internet das Coisas (<i>Internet Of Things</i>)
LED	Diodo Emissor de Luiz (<i>Light Emitting Diode</i>)
TIC	Tecnologia da Informação e Comunicação
VCC	Tensão em Corrente Contínua (<i>Voltage Common Collector</i>)

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Contexto e Motivação	13
1.2	Objetivos	14
1.2.1	Objetivo Geral	14
1.2.2	Objetivos Específicos	14
1.3	Estrutura do trabalho	15
2	REFERENCIAL TEÓRICO	16
2.1	Cidades Inteligentes	16
2.2	Arduino	17
2.3	Trabalhos relacionados	17
3	METODOLOGIA	20
3.1	Componentes utilizados	21
3.1.1	Arduino UNO R3	21
3.1.2	Sensor Ultrassônico HC-SR04	22
3.1.3	Protoboard	22
3.1.4	Jumpers	23
3.1.5	LEDs	24
4	PROTÓTIPO	25
4.1	Modelagem	25
4.2	Desenvolvimento do código-fonte	26
4.3	Teste de acurácia	28
5	RESULTADOS E DISCUSSÕES	30
6	CONCLUSÃO	32
	REFERÊNCIAS BIBLIOGRÁFICAS	33
	APÊNDICE A – CÓDIGO DO PROJETO	35
	APÊNDICE B – CAPTURA DE DISTÂNCIA	38
	APÊNDICE C – PROTÓTIPO CONSTRUÍDO	39
	APÊNDICE D – RESULTADOS DA SIMULAÇÃO	40

1 Introdução

1.1 Contexto e Motivação

Eventualmente ao redor do mundo ocorrem tragédias ocasionadas por desastres naturais. Tais desastres são decorrentes das ações do homem para com a natureza, potencializando o aquecimento global e o efeito estufa e, conseqüentemente aumentando a quantidade de eventos trágicos causados pela própria natureza (TODA MATÉRIA, 2020). Esses eventos trazem vários danos e prejuízos para a sociedade no mundo. Os desastres naturais são divididos em classificações, tais como: terremotos, furacões, erupções vulcânicas, seca e tempestades.

No Brasil, a principal causa de desastres naturais que acomete em quase todos os anos da história recente são as fortes chuvas que ocasionam transbordamento dos rios, e, conseqüentemente grande parte dos incidentes ocasionados pelas mesmas. Tais desastres ocorrem com bastante frequência no país, causando transtornos nos grandes centros urbanos e eventualmente ocasionando em tragédias fatais em certas ocasiões. No norte e nordeste do país ocorreram dois eventos de transbordamentos de rios com forte impacto repercussão. O primeiro ocorreu em abril de 2009, onde 8 estados (Amazonas, Bahia, Ceará, Pará, Pernambuco, Piauí, Maranhão e Rio Grande do Norte) foram atingidos pelas fortes chuvas, contabilizando 19 mortes e 186 mil desabrigados (BBC, 2009). A segunda ocasião ocorreu em maio de 2017 no estado de Pernambuco, onde as regiões mais atingidas foram a Zona da Mata Sul e o Agreste. Nesse evento foram contabilizados 2656 desabrigados e 42.145 desalojados, na época, obrigando alguns municípios atingidos decretarem estado de calamidade pública (AGÊNCIA BRASIL, 2017).

Grande parte desses municípios não possuem um acompanhamento e monitoramento dos rios locais. O monitoramento do nível da água dos rios seria de extrema importância para a diminuição dos impactos ocasionados pelas fortes chuvas durante os períodos de tempestades nas regiões citadas.

Considerando o paradigma da *Internet Of Things* (IoT), juntamente com o *Smart Cities*, foi idealizado o desenvolvimento de uma solução que realize o monitoramento do nível da água dos rios, com a finalidade de ajudar a defesa civil na tomada de decisão e gestão de incidentes, assim como, a população no acompanhamento em tempo real do nível de rios para mitigar os transtornos consequentes do transbordamento dos rios, causados pelas fortes chuvas.

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo deste trabalho é desenvolver uma solução inteligente que monitorea em tempo real o nível da água de um rio utilizando Arduino, de modo a conseguir calcular a distância do nível da água.

1.2.2 Objetivos Específicos

- Realizar a modelagem lógica do projeto;
- Analisar os componentes utilizados;
- Estudar a biblioteca utilizada para desenvolvimento do código;
- Desenvolver código-fonte do projeto;
- Realizar teste de acurácia;
- Simular situações de uso do projeto;
- Discutir os resultados obtidos.

1.3 Estrutura do trabalho

O trabalho está dividido em seis capítulos. O primeiro refere-se a introdução, que contém o contexto e motivação, os objetivos e a estrutura do trabalho. O segundo capítulo aborda o referencial teórico, onde são levantados alguns pontos sobre Cidades Inteligentes, Arduino e trabalhos relacionados. O terceiro capítulo é sobre a metodologia, discutindo sobre os componentes utilizados, a prototipagem do projeto e os resultados obtidos durante os testes. No quarto capítulo é apresentado o desenvolvimento do protótipo. O quinto capítulo aborda os resultados obtidos e as discussões acerca destes resultados. Encerra-se o trabalho no sexto capítulo, onde o mesmo refere-se ao capítulo de conclusão.

2 Referencial Teórico

2.1 Cidades Inteligentes

O conceito abrangente de *Smart Cities* (Cidades Inteligentes) consiste em fazer “um esforço consciente para usar a tecnologia da informação para transformar a vida e o trabalho dentro de seu território de forma significativa e fundamental” (BARBOSA et al, 2013, p. 26). Segundo AGUIRRE (2017), “são cidades que, por meio de coleta e análise de dados, conseguem proporcionar serviços públicos com a qualidade desejada pelos cidadãos, consumindo o mínimo de recursos financeiros e econômicos.”

O monitoramento de rios tem como objetivo evitar possíveis tragédias e transtornos que possam ocorrer em consequência dos transbordamentos dos rios, considerando que em várias cidades do país não fazem uso de ferramentas tecnológicas para realizar o acompanhamento do nível da água dos rios para detectar a situação dos mesmos.

No contexto atual, utiliza-se de dispositivos inteligentes para inúmeras funções para melhorar a qualidade de vida de uma determinada localização, fazendo da Tecnologia da Informação e Comunicação (TIC) e da IoT as principais ferramentas para realizar as funções aplicadas, dependendo do contexto. Para PRADO e SANTOS (2014):

“Para alcançar esse conceito é preciso, primeiramente, entender o contexto na qual foi concebida esta nova forma de se relacionar com a cidade. Para tal contextualização, utilizar-se-ão as tendências observadas nos últimos anos ou décadas, não só relacionadas a demografia e aspectos sociais, mas principalmente ao avanço tecnológico no qual se fundamentam as Smart Cities. ”

Para C40 (2011):

“As cidades são consideradas inteligentes quando são identificadas contendo investimentos inteligentes ao longo dos eixos: economia, mobilidade, meio ambiente, recursos humanos e estilos de vida inteligentes. Os significativos avanços tecnológicos e das tecnologias da informação e comunicação (TIC) agora fazem das plataformas tecnológicas embarcadas um instrumento potencialmente significativo para sensorizar e monitorar a funcionalidade e o desempenho das cidades, permitindo ampliar sobremaneira suas capacidades de gerenciar recursos com mais

eficiência e prover conectividade e informações de forma transparente aos seus cidadãos e visitantes”

Considerando que o projeto do protótipo desenvolvido neste trabalho seja implementado futuramente utilizando um servidor *web*, informando os dados coletados através de protocolos de rede, o mesmo poderá ser enquadrado como um dispositivo aplicável em *Smart Cities*.

2.2 Arduino

O Arduino UNO R3 é um dispositivo eletrônico de código aberto que permite que sejam configurados dispositivos de entrada e saída, podendo-os conectá-los entre si (através do microcontrolador Arduino) e podendo desenvolver projetos, de acordo com a idealização do usuário (ARDUINO, 2021). Para KIANE (2019):

“O Arduino é uma plataforma eletrônica open-source composta por hardware e software. É uma ferramenta fácil para prototipagem rápida, destinada a estudantes sem formação em eletrônica e programação. O hardware consiste em uma placa com um microcontrolador (microcomputador de um único circuito, onde é processado o código desenvolvido).”

Considerando que os dados reconhecidos pelo Arduino são apenas sinais elétricos (MARGOLIS, 2011), utiliza-se o Arduino IDE (Interface de Desenvolvimento) para criar o código fonte do projeto desenvolvido, onde a linguagem de programação, que é baseada em C/C++ será transferida para o Arduino realizar suas operações, podendo ser desde acender uma LED até disparar um sinal ultrassônico para medir distâncias.

2.3 Trabalhos relacionados

A temática IoT, Arduino e monitoramento do nível da água são recorrentes no meio acadêmico que cresce gradativamente ao decorrer dos anos. Esta seção tem como objetivo apresentar, trazer alguns pontos positivos e negativos e analisar alguns trabalhos relacionados de acordo com o tema em questão.

Gonçalves (2019) apresenta em seu trabalho o desenvolvimento de um sistema de monitoramento local de nível de óleo de tanques industriais, utilizando sensor ultrassônico. O enfoque da pesquisa é pautada na automação industrial, cujo objetivos são centrados em melhorar a eficiência, as condições de trabalho e aumentar a qualidade de produção e serviço. As fábricas inteligentes já são uma realidade no mundo atual, utilizando a aplicação em conjunto da indústria 4.0, A principal ferramenta usada nesse contexto é a CLP (Controlador Lógico Programável), um computador capaz de executar funções específicas para cada tarefa. Para a obtenção de resultados, Gonçalves utiliza como cenário três tanques, um conjunto de sensores de níveis (para uma maior precisão no resultado dos valores) e uma CLP didática. Por tratar-se de um experimento em um ambiente que possui uma maior variação de resultados, houve um alto custo para o aquisição dos materiais desenvolvidos para o protótipo.

A proposta de projeto sugerida por Loffi et al. (2019) visa como o principal objetivo monitorar o nível da água para a prevenção de enchentes utilizando um protótipo e o mesmo realiza o monitoramento e tratamento de dados obtidos, onde há o estudo de padrões de informações. Entretanto, o custo financeiro do projeto é elevado para a proposta sugerida. Por ser um trabalho desenvolvido por 5 pessoas, a obtenção de recursos torna-se um pouco menos dificultoso. O contexto de estudo de caso foi realizado no estado de Santa Catarina, mais precisamente no Rio Itajaí, rio este que já trouxe inúmeros prejuízos para a população ribeirinha devido a enchentes causados pelas fortes chuvas.

Os testes do protótipo do projeto foi realizado localmente, onde a transferência dos dados obtidos pelo dispositivo utilizado realizou-se através de um cabo USB e um computador. O protótipo detecta o nível da água do rio e de acordo com a próxima detecção, o monitoramento realiza o cálculo, em mm, da precipitação da chuva. Há o intervalo de 15 minutos para a análise dos dados e os resultados foram satisfatórios, comprovando a eficácia do protótipo desenvolvido.

Antoniolli (2019) desenvolve um sistema de monitoramento utilizando automação em IoT, com a finalidade de manter o controle de qualidade da água. Seu protótipo faz uso de uma placa micro controladora Arduino e o dispositivo de obtenção de dados é um sensor de boia, que é acoplado na superfície de um sistema aquapônico, para estabilizar a economia da água e controlar agentes químicos na irrigação dos cultivos biológicos. O projeto em questão é voltado para o setor agroindustrial, com a

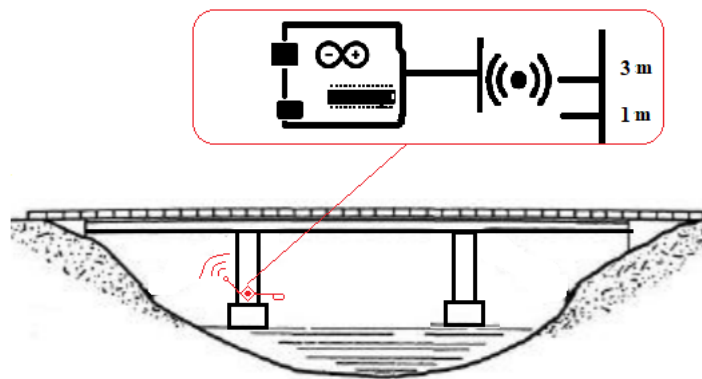
proposta de realizar um bom funcionamento na irrigação das plantas e economia da água a ser utilizada, sendo realizada as medições de acordo com as automações programadas.

3 Metodologia

A pesquisa desenvolvida foi realizada utilizando a técnica exploratória, onde foi detectado um problema em comum (desastres naturais – enchentes ocasionadas pelas fortes chuvas) e realizando testes de detecção de nível da água.

Utilizou-se uma placa microcontroladora Arduino UNO R3, em comunicação com alguns componentes eletrônicos e um código desenvolvido em C que irá realizar as funções do dispositivo. Foi realizada uma simulação (visualização do cenário na Figura 1), com a finalidade de obter resultados para confirmar e validar o funcionamento do dispositivo desenvolvido.

Figura 1 - Cenário - Simulação



Fonte: (Próprio Autor, 2021)

3.1 Componentes utilizados

No Quadro 1, os componentes escolhidos para o desenvolvimento do projeto.

Quadro 1 - Componentes utilizados no projeto

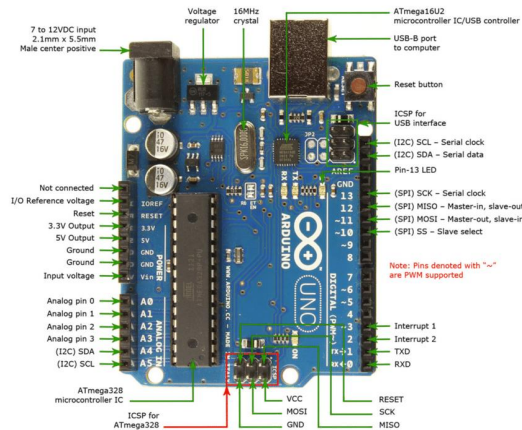
Componente	Quantidade
Microcontrolador Arduino UNO R3	1
Sensor Ultrassônico HC-SR04	1
Protoboard	1
Jumpers	9
LEDs	3

Fonte: (Próprio Autor, 2021)

Tais componentes são essenciais para o funcionamento do protótipo, pois cada um estabelece uma função específica para realizar o objetivo do desenvolvimento do protótipo.

3.1.1 Arduino UNO R3

Figura 2 - Visão frontal Arduino Uno



Fonte: (RESEARCHGATE, 2020).

O dispositivo principal para desenvolvimento do protótipo é o Arduino UNO R3 (Figura 2). Seu principal objetivo é ler as instruções (código-fonte na linguagem de

programação C) que serão transmitidas para ele e realizar a comunicação entre os outros componentes que a ele estarão conectados.

3.1.2 Sensor Ultrassônico HC-SR04

O sensor ultrassônico HC-SR04 (Figura 3) é o dispositivo que recebe as informações acerca do nível da água, onde o mesmo dispara uma onda sonora – emitida pelo pino *Trig* – e ao ricochetear em um objeto, o mesmo retorna mais uma onda sonora – recebida pelo pino *Echo*. No entanto, seu alcance é limitado a 4 metros de distância. Ultrapassando desse limite, o dispositivo não é capaz de receber o *Echo* propagado pelo *Trig*. Outro ponto que merece ser destacado é que a partir de uma determinada distância, o seu valor pode variar em milímetros (mm), não havendo uma precisão sobre a distância de um determinado objeto – no caso do propósito a ser obtido, o nível da água.

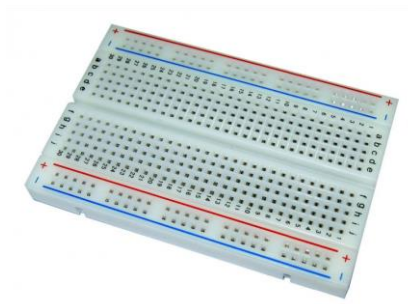
Figura 3 - Visão frontal Sensor Ultrassônico HC-SR04



Fonte: (ARDUINO, 2020)

3.1.3 Protoboard

A função da *protoboard* (Figura 4) é facilitar a prototipagem de projetos, facilitando a inserção e manipulação de vários componentes, ou seja, é uma espécie de ponte entre o microcontrolador e os componentes. No protótipo, possui a finalidade de acoplar o sensor ultrassônico e os LEDs.

Figura 4 - Visão frontal *Protoboard*

Fonte: (ROBOCORE, 2020)

3.1.4 Jumpers

Os *jumpers* (Figura 5) utilizados foram macho-macho, ou seja, eles irão acoplar tanto o microcontrolador quanto a *protoboard*. Também são muito utilizados em prototipagens, evitando o contato direto entre o componente e o Arduino. No protótipo, os *jumpers* são utilizados para conectar o sensor e os LEDs ao microcontrolador (ambos conectados na *protoboard*).

Figura 5 - Visão frontal *Jumpers*

Fonte: (VIDA DE SILÍCIO, 2020)

3.1.5 LEDs

O *Light Emitting Diode* (LED) é um componente eletrônico composto por materiais semicondutores capaz de emitir luz visível. No protótipo, sua função é indicar o estado de alerta em relação ao nível da água. No Quadro 2 é possível compreender o significado da ativação de cada cor de LED durante o funcionamento do protótipo.

Quadro 2 - Representação dos LEDs

Cor	Estado
Verde	Baixo nível de água
Amarelo	Médio nível de água
Vermelho	Alto nível de água

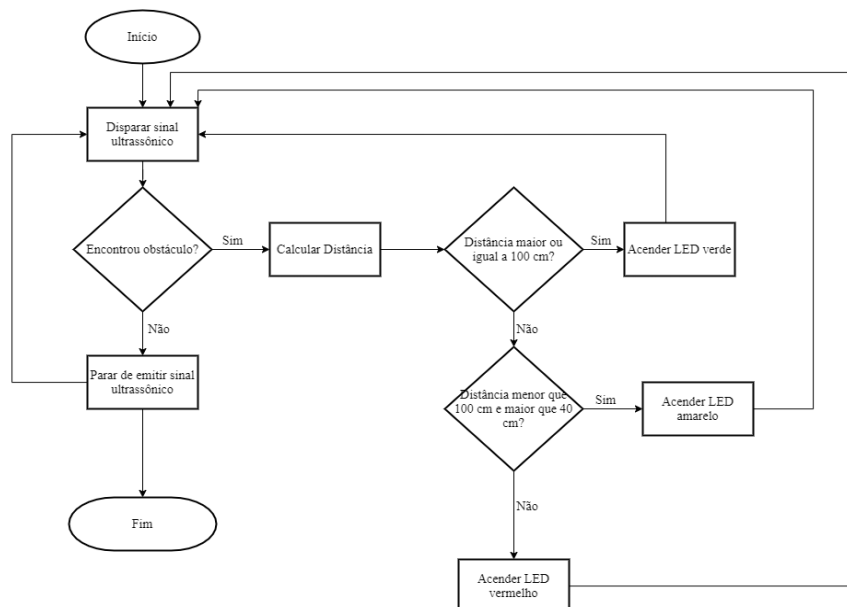
Fonte: (Próprio Autor, 2021)

4 Protótipo

4.1 Modelagem

A modelagem lógica do circuito durante a prototipagem é essencial antes de realizar a conexão física entre os componentes e o microcontrolador Arduino. Durante esse processo, foi realizado inicialmente testes entre os LEDs para verificar se o modelo lógico estava de acordo com as conexões realizadas na conexão física. O fluxograma do projeto encontra-se na Figura 6.

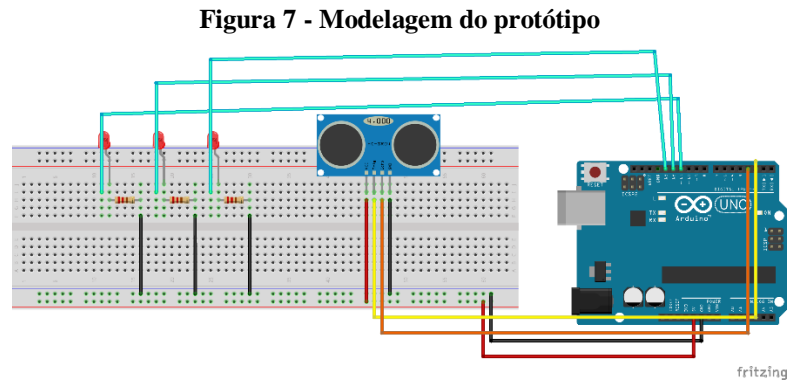
Figura 6 - Fluxograma do projeto



Fonte: (Próprio Autor, 2021)

O software utilizado para realizar a modelagem do projeto foi o *Fritzing* (FRITZING, 2020), uma iniciativa para hardwares de código-aberto que possibilita o usuário realizar testes de hardware utilizando componentes virtuais do Arduino. É possível criar layouts, circuitos eletrônicos e realizar testes de códigos que podem ser transmitidos para o protótipo virtual que o usuário desenvolveu no próprio *Fritzing*. Entretanto, não é possível

simular o circuito desenvolvido pelo software. O protótipo foi modelado utilizando todos os componentes presentes no projeto. Na Figura 7 e no Quadro 3 é possível visualizar a modelagem do protótipo e a representação dos elementos inseridos graficamente.



Fonte: (Próprio Autor, 2021)

Quadro 3 - Representação da modelagem do protótipo

Cor do fio	Representação
Ciano	Pinos dos LEDs (11, 12 e 13).
Preto	Pinos GND (energia neutra).
Vermelho	Pino VCC (5V).
Amarelo	Pino <i>Trigg.</i>
Laranja	Pino <i>Echo</i>

Fonte: (Próprio Autor, 2021)

4.2 Desenvolvimento do código-fonte

Em seguida, foi desenvolvido o código-fonte que irá ler os dados obtidos pelo sensor ultrassônico e apresentar no Monitor Serial do Arduino IDE, interpretador utilizado para construir a estrutura do código repassado para o microcontrolador Arduino.

O Arduino possui duas funções principais: a *setup* e *loop*, onde na primeira são feitas todas as atribuições iniciais, como determinar constantes que representam o pino de um determinado componente. No protótipo em questão, por exemplo, o pino que está configurado para emitir uma onda sonora (*Trig*) é o pino de número 2, e o pino que receberá o retorno da mesma onda (*Echo*) é o pino de número 3. Da mesma forma com os LEDs: os pinos

referentes às luzes são os de número 11, 12 e 13. A segunda função é responsável por todas as atividades que o Arduino faz repetidamente e infinitamente (enquanto ele estiver sendo alimentado eletricamente), ou seja, onde toda a atividade do projeto é desenvolvida. Nesta função são criadas várias outras em sua estrutura para que haja o funcionamento correto do projeto desenvolvido. No Quadro 4 são descritas as funções criadas na estrutura do *loop*.

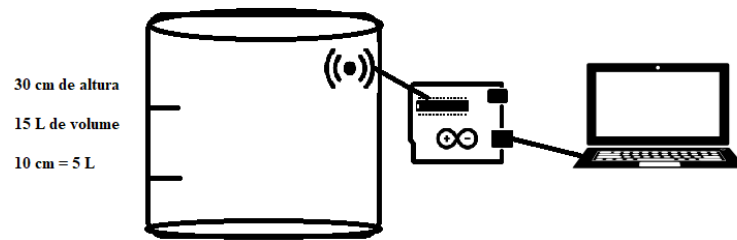
Quadro 4 - Funções desenvolvidas

Funções criadas no projeto	
AcenderLEDs	Tem como objetivo indicar qual o nível da água no momento, onde a cor verde representa nível baixo, amarelo representa nível médio e vermelho representa nível alto.
EnviarOndaSonora	Função designada para que o sensor HC-SR04 dispare um sinal ultrassônico a cada 0,5 segundos.
Distancia	Responsável por realizar o cálculo de distância e exibirá os valores no monitor serial do Arduino IDE a cada 0,5 segundos. Esta função possui algumas condicionais para determinar qual LED será aceso de acordo com a distância encontrada pelo sensor.

Fonte: (Próprio Autor, 2021)

Para a construção do código foi necessário utilizar a biblioteca *Ultrasonic.h* (SIMÕES, 2020), que tem como constantes os valores para calcular a distância em centímetros e em milímetros. Entretanto, no projeto as medidas utilizadas para monitoramento foram em centímetros e metros, para manter uma melhor precisão nos dados obtidos pelo sensor HC-SR04.

Figura 8 - Simulação do projeto



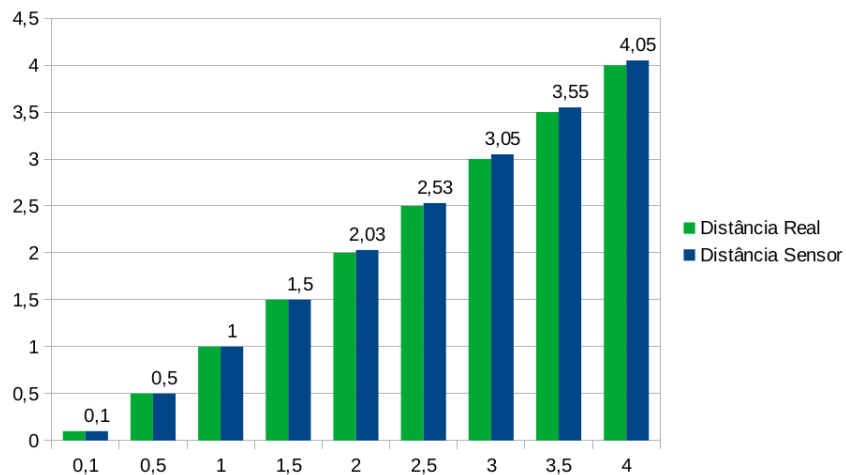
Fonte: (Próprio Autor, 2021)

4.3 Teste de acurácia

Durante alguns testes de aproximações de objetos ao sensor, detectou-se que não havia uma precisão correta do valor da distância dos objetos em relação a medidas superiores a 2 metros. Como é demonstrado no gráfico da Figura 9, há uma variação de 3 a 5 milímetros entre os valores de 2 a 4 metros.

A solução encontrada foi adicionar uma estrutura condicional no código, onde se a distância do sensor for maior que 2 metros e menor que 3 (três) metros, será acrescido 3 centímetros (que foi a variação para menos do valor) no resultado do cálculo da distância.

Figura 9 - Variações dos resultados obtidos durante o teste



Fonte: (Próprio Autor, 2021)

Utilizou-se uma placa micro controladora Arduino Uno R3, em comunicação com alguns componentes eletrônicos e um código desenvolvido em C/C++ que irá realizar as funções do dispositivo. Foi realizada uma simulação (visualização do cenário na Figura 1), com a finalidade de obter resultados para confirmar e validar o funcionamento do dispositivo desenvolvido.

5 Resultados e discussões

Realizou-se testes de medição de nível da água utilizando um recipiente cilíndrico (balde) com capacidade de 15 litros de água, possuindo 32 cm de altura e 10 cm de diâmetro. Durante o experimento foram realizadas marcações no balde, referente às medidas de distância, onde colocou-se 3 níveis de medição. 10 cm do balde é relacionado a 5L de água, restando 2 cm do balde para inserir o sensor sobre o mesmo.

Tabela 1 - Relação entre o nível de medição e a distância

Nível de medição (classificação)	Distância
Baixo nível	30 – 20 cm
Médio nível	19,9 – 10 cm
Alto nível	9,9 – 0 cm

Fonte: (Próprio Autor, 2021)

Na Tabela 1, é possível visualizar os dados obtidos de acordo com a simulação realizada. O sensor ultrassônico foi acoplado na superfície do balde, com a finalidade de realizar a medição da distância do nível da água, de acordo com a quantidade de água que será adicionada ao balde. A relação água – distância é inversamente proporcional, considerando que à medida que a água for adicionada no recipiente cilíndrico, a distância entre a mesma e o sensor será diminuída. Inicialmente houve uma pequena variação do resultado. Tal variação ocorreu pela movimentação da água no balde. Entretanto, após a água permanecer em um estado de não-movimento, o valor detectado foi preciso.

Os LEDs acenderam de acordo com as condicionais configuradas correlacionadas aos níveis pré-estabelecidos para este experimento, onde o LED verde permaneceu aceso com o balde vazio até o momento em que o sensor HC-SR04 detectou que havia mais de 5L de água no recipiente, ou seja, a água foi elevada até os 10 cm do balde, previamente marcado e medido. No monitor serial do Arduino IDE foi detectado que a água estava a 20 cm de distância do sensor ultrassônico, com mínimas variações enquanto a água estava em movimento no balde, sendo assim, detectando baixo nível na água.

Ao inserir mais água, o LED verde apagou-se e foi aceso o LED amarelo, onde representa a detecção de um médio nível de água no recipiente, ou seja, adicionando água até os 20 cm presente na marcação, o LED permaneceu aceso e no monitor serial do Arduino IDE também foi verificado a precisão da medição desse segundo nível, exibindo que a água estava a 10 cm de distância do sensor ultrassônico.

Ao ultrapassar esse nível, o LED amarelo foi desligado e foi acionado o LED vermelho, detectando um alto nível da água, em que houve uma aproximação menor que 10 cm e ultrapassando os 10L de água, que para o experimento tal medida é o limite para detectar um suposto perigo de desastre.

6 Conclusão

O transtorno decorrente das enchentes causadas pelas fortes chuvas ainda assola boa parte da sociedade brasileira. Considerando este problema, idealizou-se o desenvolvimento do protótipo de um dispositivo que é capaz de medir a distância do nível da água, utilizando uma placa micro controladora de baixo custo, alguns componentes eletrônicos e a criação de um código para a funcionalidade deste dispositivo.

O arquitetura do dispositivo foi segmentado em *hardware* e *software*, onde na primeira realizou-se a montagem e infraestrutura do protótipo; e a segunda desenvolveu-se o código-fonte do protótipo.

Um teste de acurácia foi realizado para identificar e constatar a eficácia da medição do sensor ultrassônico, componente este que emite ondas para realizar os cálculos de distância.

As simulações de validação do protótipo foram realizadas em um recipiente cilíndrico, com a finalidade de realizar a medição do nível da água de acordo com as medidas previamente estabelecidas para os resultados de baixo, médio e alto nível. Os resultados das simulações foram satisfatórios, considerando que as medidas foram calculadas corretamente e o dispositivo emitiu saídas visuais (utilizando LEDs) de acordo com o valor da distância do nível da água obtido.

Para trabalhos futuros, deseja-se realizar melhorias no dispositivo, como alimentação elétrica utilizando energia solar, disponibilidade dos dados monitorados através de um *web server*, estudo dos dados coletados, realizando a análise de padrões obtidos e a implementação do dispositivo em um cenário real.

REFERÊNCIAS BIBLIOGRÁFICAS

AGÊNCIA BRASIL. **Com 44 mil pessoas afetadas, municípios de PE contabilizam prejuízos após chuvas.** 2017. Disponível em: <<https://agenciabrasil.ebc.com.br/geral/noticia/2017-05/com-44-mil-pessoas-afetadas-municipios-de-pe-contabilizam-prejuizos-apos>>. Acesso em: 5 de fev. de 2021.

AGUIRRE, Tiago Ferraz de. **Gerência de Risco de Enchentes em Cidades Inteligentes.** 86 f. Trabalho de Conclusão de Curso (Pós-Graduação). Pontifícia Universidade Católica de Campinas. Campinas, 2017.

ANTONIOLLI, Alessandro. **Sistema de monitoramento automatizado para controle de qualidade de água em sistema aquapônico.** 95 f. Trabalho de Conclusão de Curso (Graduação). Universidade do Vale do Taquari. Lajeado, 2019.

ARDUINO. **Arduino**, 2020. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 11 de nov. de 2020.

ARDUINO. **Arduino**, 2021. Disponível em: <<https://www.arduino.cc/en/guide/introduction>>. Acesso em: 6 de fev. de 2021.

BARBOSA, Gabriel Bernal; URQUIZA, Murilo Ferreira; CÂNDIDO, Matheus Bolela B.; PUGLIESI, Jaqueline Brigladori. **Tecnologia Integrada às Áreas Para o Desenvolvimento de Cidades Inteligentes.** Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica, v. 3, n. 1, 2013.

BBC. **BBC News**, 2009. Disponível em: <<http://news.bbc.co.uk/2/hi/americas/8033351.stm>>. Acesso em: 15 de jan. de 2021.

C40 SÃO PAULO CLIMATE SUMMIT. **Síntese do C40 São Paulo Climate Summit 2011.** São Paulo. Prefeitura de São Paulo. 2011.

FRITZING. **Fritzing**, 2020. Disponível em: <<https://fritzing.org/faq/>>. Acesso em: 10 de nov. de 2020.

GOLÇALVES, Matheus Rocha. **Controle de nível de planta didática usando controlador lógico programável.** 40 f. Trabalho de Conclusão de Curso (Graduação). Universidade Federal de Ouro Preto. Ouro Preto, 2019.

KIANE, Rayse. VIA - Estação Conhecimento. **Arduino: uma plataforma de prototipagem eletrônica.** 2019. Disponível em: <<https://via.ufsc.br/arduino-e-o-seu-papel-no-makerspace/>>. Acesso em: 6 de fev. de 2021.

LOFFI, Leandro *et al.* **Monit-RIO - Tecnologia da informação de comunicação para monitoramento de rios em casos de cheias.** 1. ed. Resende. AEDB, 2016. Disponível em <<https://www.aedb.br/seget/arquivos/artigos16/13424296.pdf>>. Acesso em: 6 de fev. de 2021.

MARGOLIS, Michael. *Arduino Cookbook - Recipes to Begin, Expand and Enhance Your Projects*. O'Reilly. Sebastopol, 2011.

PRADO, K. C. D.; SANTOS, P. E. **Smart Cities: Conceito, Iniciativas e o Cenário Carioca**. 133 f. Trabalho de Conclusão de Curso (Graduação). Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2014.

RESEARCHGATE. **Arduino UNO Board in Reality**. 2018. Disponível em: <https://www.researchgate.net/figure/Arduino-UNO-Board-in-Reality_fig1_323119599>. Acesso em: 7 de fev. de 2021.

ROBOCORE. **Robocore**, 2020. Disponível em: <<https://www.robocore.net>>. Acesso em: 4 de dez. de 2020.

SIMÕES, Erick. **Arduino Libraries**. Disponível em: <<https://www.arduinolibraries.info/libraries/ultrasonic>>. Acesso em: 4 de dez. de 2020.

TODA MATÉRIA. **Desastres Naturais**. 2020. Disponível em: <<https://www.todamateria.com.br/desastres-naturais/>>. Acesso em: 5 de fev. de 2021.

VIDA DE SILÍCIO. **Vida de Silício**, 2020. Disponível em: <<https://www.vidadesilicio.com.br>>. Acesso em: 15 de dez. de 2020.

APÊNDICE A – Código do projeto

```
#include <Ultrasonic.h>

#define pino_emi 2

#define pino_rec 3

Ultrasonic Sensor (pino_emi, pino_rec);

int led_vermelho = 11;

int led_amarelo = 12;

int led_verde = 13;

void setup() {

  pinMode (led_vermelho, OUTPUT);

  pinMode (led_amarelo, OUTPUT);

  pinMode (led_verde, OUTPUT);

  Serial.begin (9600);

  delay (500);

  Serial.println ("Iniciando medições...");

}

void AcendeLEDVerde(){

  digitalWrite (led_verde, HIGH);

  digitalWrite (led_vermelho, LOW);

  digitalWrite (led_amarelo, LOW);

}
```

```
void AcendeLEDAmarelo(){  
digitalWrite (led_amarelo, HIGH);  
digitalWrite (led_verde, LOW);  
digitalWrite (led_vermelho, LOW);  
}
```

```
void AcendeLEDVermelho(){  
digitalWrite (led_vermelho, HIGH);  
digitalWrite (led_verde, LOW);  
digitalWrite (led_amarelo, LOW);  
}
```

```
void Distancia() {  
float dist, dist_m;  
long ms = Sensor.timing();  
dist = Sensor.convert(ms, Ultrasonic::CM);  
dist_m = dist / 100;
```

```
if (dist >= 2000 && dist <= 3000){  
dist = dist - 0.03;  
dist_m = dist / 100;  
}
```

```
if (dist > 3000 && dist <= 4000){  
dist = dist - 0.05;  
dist_m = dist / 100;
```

```
}
```

```
if (dist > 0 && dist <= 10) {
```

```
  AcendeLEDVermelho();
```

```
}
```

```
if (dist > 10 && dist <= 20) {
```

```
  AcendeLEDAmarelo();
```

```
}
```

```
if (dist > 20 && dist <= 30) {
```

```
  AcendeLEDVerde();
```

```
}
```

```
Serial.println("-----");
```

```
Serial.print("Distância em cm: ");
```

```
Serial.println(dist);
```

```
Serial.print("Distância em m: ");
```

```
Serial.println(dist_m);
```

```
Serial.println("-----");
```

```
delay(500);
```


```
}
```

```
void loop() {
```

```
  Distancia();
```

```
}
```

APÊNDICE B – Captura de distância

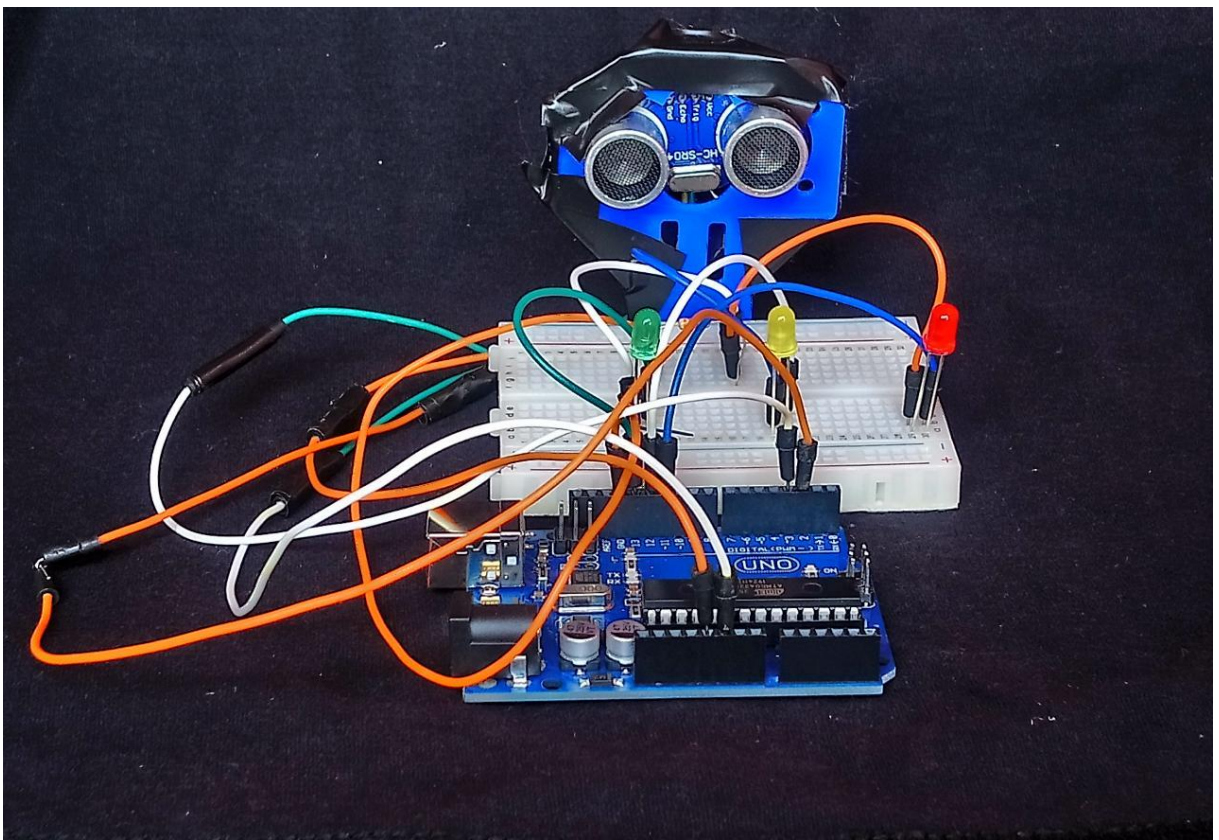


The screenshot shows a terminal window titled 'COM3' with a search bar and an 'Enviar' button. The terminal output displays the following text:

```
Iniciando medições...
-----
Distância em cm: 12.49
Distância em m: 0.12
-----
Distância em cm: 14.23
Distância em m: 0.14
-----
Distância em cm: 13.41
Distância em m: 0.13
-----
Distância em cm: 13.01
Distância em m: 0.13
-----
Distância em cm: 12.40
Distância em m: 0.12
-----
Distância em cm: 11.37
```

At the bottom of the window, there are several controls: a checked checkbox for 'Auto-rolagem', an unchecked checkbox for 'Show timestamp', a dropdown menu for 'Nova-linha' (set to 'Nova-linha'), a dropdown menu for '9600 velocidade' (set to '9600 velocidade'), and a 'Deleta a saída' button.

APÊNDICE C – Protótipo construído



APÊNDICE D – Resultados da simulação



COM3

Distância em cm: 25.43
Distância em m: 0.25

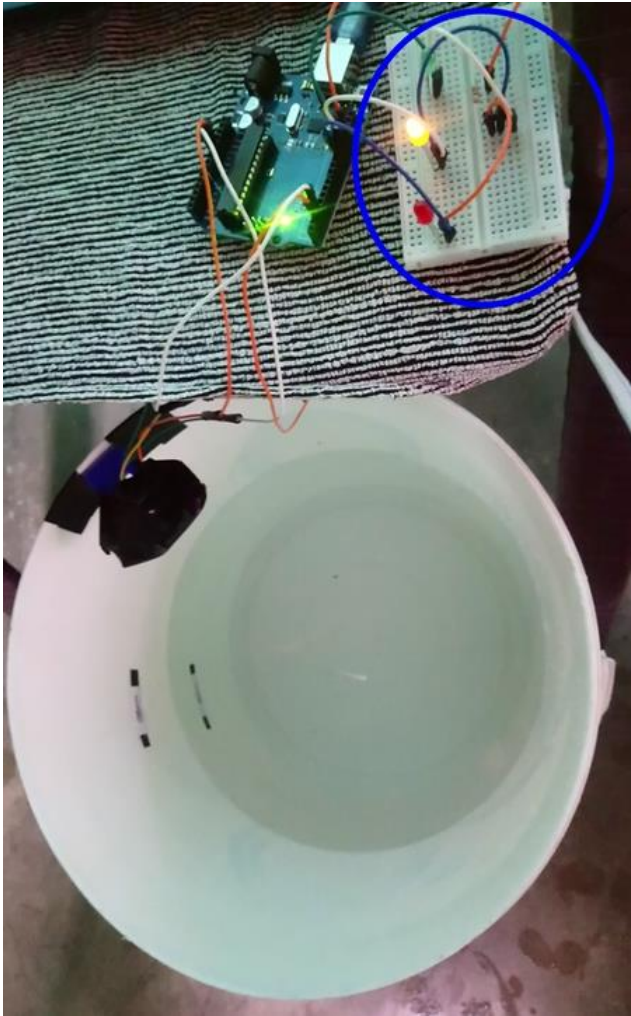
Distância em cm: 25.49
Distância em m: 0.25

Distância em cm: 25.54
Distância em m: 0.26

Distância em cm: 25.49
Distância em m: 0.25

Distância em cm: 25.47
Distância em m: 0.25

Auto-rolagem Show timestamp



```
COM3  
|  
-----  
Distância em cm: 16.02  
Distância em m: 0.16  
-----  
Distância em cm: 15.95  
Distância em m: 0.16  
-----  
Distância em cm: 15.95  
Distância em m: 0.16  
-----  
Distância em cm: 15.96  
Distância em m: 0.16  
-----  
Distância em cm: 15.96  
Distância em m: 0.16  
-----  
 Auto-rolagem  Show timestamp
```



```
COM3
|
-----
Distância em cm: 5.36
Distância em m: 0.05
-----
Distância em cm: 5.36
Distância em m: 0.05
-----
Distância em cm: 5.36
Distância em m: 0.05
-----
Distância em cm: 5.36
Distância em m: 0.05
-----
Distância em cm: 5.47
Distância em m: 0.05
-----
 Auto-rolagem  Show timestamp
```