



CENTRO UNIVERSITÁRIO DE BRASÍLIA- UniCEUB
FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS –
FATECS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

GABRIEL DE OLIVEIRA MARQUES

SISTEMA DE CAPTAÇÃO DE ÁGUAS PLUVIAIS PARA UMA RESIDÊNCIA PARTICULAR

BRASÍLIA – DF

1º SEMESTRE DE 2014

GABRIEL DE OLIVEIRA MARQUES

SISTEMA DE CAPTAÇÃO DE ÁGUAS PLUVIAIS PARA UMA RESIDÊNCIA PARTICULAR

Trabalho apresentado ao Centro
Universitário de Brasília (UniCEUB) como
pré-requisito para a obtenção de Certificado
de Conclusão de Curso de Engenharia de
Computação.

Orientador: Prof.^a. MSc. Luciano Henrique
Duque

Brasília

Dezembro, 2014

GABRIEL DE OLIVEIRA MARQUES

SISTEMA DE CAPTAÇÃO DE ÁGUAS PLUVIAIS PARA UMA RESIDÊNCIA PARTICULAR

Trabalho apresentado ao Centro
Universitário de Brasília (UniCEUB) como
pré-requisito para a obtenção de Certificado
de Conclusão de Curso de Engenharia de
Computação.

Orientador: Prof.^a. MSc. Luciano Henrique
Duque

Brasília, 09 de novembro de 2014.

Banca Examinadora

Prof. Miguel Mosquera

Prof. Irene Joffily

Dedico este trabalho ao meu Deus, meus pais e meu irmão,
Gabriel de Oliveira Marques.

“Quando vocês estiverem unidos a mim e eu a vocês, num relacionamento íntimo e orgânico, não imaginam que colheita terão. Separados, vocês não podem produzir. Qualquer um que se separa de mim é um ramo morto, que é apanhado e jogado na fogueira” João 15.5-7.

Gabriel de Oliveira Marques.

Sumário

Capítulo 1 - Introdução.....	14
1.1.Motivação.....	15
1.2.Objetivo.....	16
1.2.1.Objetivo Geral.....	16
1.2.2.Objetivos Específicos.....	16
1.3.Metodologia.....	16
1.4.Resultados esperados.....	18
1.5.Organização.....	18
Capítulo 2 - Referencial teórico	20
2.1.A água doce no mundo.....	20
2.2.Os riscos das águas pluviais.....	20
2.3.Norma de Instalações prediais de águas pluviais (NBR 10844).....	21
2.3.1.Definições.....	21
2.3.1.1.Área de contribuição.....	21
2.3.1.2.Vazão de projeto.....	22
2.3.1.3.Tempo de concentração.....	23
2.3.2.Fatores Meteorológicos.....	23
2.4.O sistema de captação de águas pluviais.....	23
2.4.1.Cisternas.....	24
2.4.2.Calhas.....	25
2.4.3.Condutores.....	28
2.4.3.1.Condutores verticais.....	28
2.4.3.2.Condutores horizontais.....	28
2.5.Tratamento da água de chuva.....	29
2.6.Arduino.....	30
2.6.1.Arduino Mega.....	31

2.6.2.Shield Ethernet.....	32
2.6.3.IDE Arduino.....	33
2.7.Módulo Relé.....	35
2.8.Linguagem HTML.....	36
2.9.Sistemas de Controle.....	37
2.9.1.Controle em malha aberta.....	37
2.9.2.Controle em malha fechada.....	38
2.10.Sensores.....	39
2.10.1.Sensores analógicos.....	39
2.10.2.Sensores digitais.....	40
2.11.Engenharia verde.....	40
Capítulo 3 – Desenvolvimento do Projeto.....	42
3.1.Apresentação geral do projeto.....	42
3.2.Programação na IDE.....	44
3.3.Placa de Circuito.....	56
3.3.1.Sensor de nível de água.....	57
3.3.2.Sensor de chuva.....	60
3.3.3.Acionamento da bomba d’água.....	61
3.3.4.Módulo de alerta.....	64
3.3.5.Display LCD.....	66
3.3.6.Conexão da rede local.....	67
3.4.Protótipo.....	68
Capítulo 4 – Testes realizados.....	71
4.1.Cenário 1.....	71
4.2.Cenário 2.....	73
Capítulo 5 – Conclusão	74
5.1.Conclusões do Projeto.....	74

5.2.Aplicações do sistema.....	76
5.3.Sugestões de melhoria.....	76
Referências.....	77
Anexo – Código Fonte do Sistema	81

Lista de Tabelas

Tabela 2.1 – Capacidade dos reservatórios cilíndricos (Fonte:São José do Cerrito, 2003).....	24
Tabela 2.2 – Intensidade Pluviométrica (Fonte : Crepani, 2001).....	26
Tabela 2.3– Declividade das calhas (Fonte: Crepani, 2001).....	27
Tabela 2.4 – Coeficientes multiplicativos da vazao de projeto (Fonte:arq.ufsc.br).....	27
Tabela 2.5 – Capacidade de condutores horizontais (Fonte:NBR 10844, dez 1989).....	29
Tabela 4.1 – Cenário de testes (Fonte:Autor).....	71

Lista de Figuras

Figura 2.1 – Superfície horizontal (Adaptado de:NBR 10844, dez 1989).....	22
Figura 2.2 – Superfície vertical (Adaptado de:NBR 10844, dez 1989).....	22
Figura 2.3 – Influência do vento na inclinação da chuva (Adaptado de:NBR 10844, dez 1989).....	23
Figura 2.4 – Esquema de cisterna (Fonte:Autor).....	25
Figura 2.5 – Tipos de calhas (Fonte:Autor).....	27
Figura 2.6 – Separador de primeiras águas de chuva (Fonte:sempresustentavel.com.br).....	30
Figura 2.7 – Arduino Mega (Fonte: store.arduino.cc/product/A000067).....	31
Figura 2.8 – Shield Ethernet (Fonte: oarduino.com).....	32
Figura 2.9 – Serial Monitor do Arduino (Fonte: orionmultitouch.blogspot.com).....	34
Figura 2.10 - IDE Arduino (Fonte: roboticaifal.webnode.com.br).....	34
Figura 2.11 – Esquema do módulo relé (Fonte:Autor).....	36
Figura 2.12 – Controle em malha aberta (Adaptado de:WWW.ece.ufrgs.br).....	38
Figura 2.13 – Controle em malha fechada (Adaptado de:pt.wikipedia.org).....	38
Figura 2.14 – Variação da temperatura (Fonte:Autor).....	40
Figura 3.1 – Modelo do projeto (Fonte:Autor).....	42
Figura 3.2 – Diagrama de blocos (Fonte:Autor).....	43
Figura 3.3 – Esquema de desenvolvimento do projeto (Fonte:Autor).....	44
Figura 3.4 – Interface WEB (Fonte:Autor).....	56
Figura 3.5 – Esquema físico de desenvolvimento do projeto (Fonte:Autor).....	57
Figura 3.6 – Placa de circuito (Fonte:Autor).....	57
Figura 3.7 – Circuito do sensor de nível de água (chave aberta) (Fonte:Autor).....	58
Figura 3.8 – Circuito do sensor de nível de água (chave fechada) (Fonte:Autor).....	58
Figura 3.9 – Tensão dos sensores de níveis de água (Fonte:Autor).....	59
Figura 3.10 – Sensores de níveis de água (Fonte:Autor).....	59
Figura 3.11 – Resistência da placa do sensor de chuva (Fonte:Autor).....	60
Figura 3.12 – Ligação do sensor de chuva no Arduino Mega (Fonte:Autor).....	61
Figura 3.13 – Modo de instalação do sensor de chuva no telhado (Fonte:Autor).....	61
Figura 3.14 – Contatos NA e NF do relé (Fonte:Autor).....	62

Figura 3.15 – Ligação do módulo relé no Arduino e na bomba d'água (Fonte:Autor).....	62
Figura 3.16 – Acionamento/desativação da bomba d'água (Fonte:Autor).....	63
Figura 3.17 – Desativação da bomba d'água durante o abastecimento (Fonte:Autor).....	64
Figura 3.18 – Componentes do LED (Fonte:Autor).....	65
Figura 3.19 – Componentes do Buzzer (Fonte:Autor).....	65
Figura 3.20 – Esquema de ligação do módulo de alerta (Fonte:Autor).....	66
Figura 3.21 – Camadas do display LCD (Fonte:Autor).....	66
Figura 3.22 – Configuração dos pinos digitais do LCD (Fonte:Autor).....	67
Figura 3.23 – Ligação do display LCD (Fonte:Autor).....	67
Figura 3.24 – Esquema de conexão da rede (Fonte:Autor).....	68
Figura 3.25 – Protótipo do projeto (Fonte:Autor).....	69
Figura 3.26 – Medidas da área de contribuição do protótipo (Fonte:Autor).....	69
Figura 3.27 – Instalação da calha e do condutor no protótipo (Fonte:Autor).....	70
Figura 4.1 – Sinal do sensor de chuva no osciloscópio (Fonte:Autor).....	72
Figura 4.2 – Queda e aumento do sinal do sensor de chuva no osciloscópio (Fonte:Autor).....	72
Figura 4.3 – Gráficos do sinal digital dos sensores de níveis e do módulo relé (Fonte:Autor).....	73
Figura 5.1 – Gráfico de economia no uso da água potável 1 (Fonte:Autor).....	74
Figura 5.2 – Gráfico de economia no uso da água potável 2 (Fonte:Autor).....	75

Lista de siglas e abreviaturas

LCD	Liquid Crystal Display – Display de Cristal Líquido.
IDE	Integrate Development Environment – Ambiente de Desenvolvimento de Integração.
LED	Light Emitting Diode – Diodo de Emissão de Luz.
HTML	Hyper Text Markup Language – Linguagem de Marcação de Hiper Texto.
HTTP	Hyper Text Transfer Protocol – Protocolo de Transferência de Hiper Texto.
GND	GROUND – Terra .
VCC	Continued Current Voltage – Voltagem de Corrente Contínua.
RISC	Reduced Instruction Set Computer – Computador com um Conjunto Reduzido de Instruções.
GPS	Global Positioning System – Sistema de Posicionamento Global.
RAM	Random Access Memory – Memória de Acesso Aleatório.
ROM	Read Only Memory – Memória Apenas de Leitura.
MHz	Mega Hertz - um milhão de hertz por segundo.
IP	Internet Protocol – Protocolo de Internet.
TCP	Transmission Control Protocol – Protocolo de Controle de Transmissão.
UDP	User Datagram Protocol - Protocolo de Datagramas de usuários.

RESUMO

Esse trabalho propõe o desenvolvimento de um sistema capaz de controlar e monitorar o armazenamento de águas pluviais. O sistema coleta a água da chuva, e direciona para um reservatório. Esse reservatório possui sensores de níveis de água que informa ao usuário se ele está vazio ou não. A água é coletada por calhas nos telhados, armazenada em uma cisterna e direcionada para uma caixa d'água de uma residência através de uma bomba d'água, que alimenta uma rede de encanamentos que abastecem locais que não necessitem de água potável como vasos sanitários, torneiras de quintal e jardim. Essa caixa d'água também possui sensores de níveis que informa se ela precisa ser abastecida ou se está completa. O sistema informa o usuário todos os dados em tempo real sobre o sistema em um display LCD ou por uma página Web disponibilizada em uma rede Wi-Fi local. Essa rede é criada através de um Shield Ethernet, um roteador e o Arduino. O Arduino é uma plataforma de hardware livre, de fácil implementação e baixo custo, podendo ser integrado em diversos tipos de projeto. Nos telhados da residência é instalado um sensor de chuva que detecta a presença de água e armazena o tempo de duração da chuva para o cálculo da intensidade pluviométrica e da vazão do projeto. Dependendo da região em que o usuário mora isso irá contribuir com problemas relacionados á escassez de recursos hídricos e enchentes nos grandes centros urbanos além de gerar uma economia no uso da água potável.

Palavras chaves: Arduino, Ethernet Shield, LCD.

ABSTRACT

This paper proposes the development of a system capable of controlling and monitoring the storage of rainwater. The system collects rainwater and directs to a reservoir. This reservoir has water level sensors and informs the user if it is empty or not. Water is collected by gutters on the roofs, stored in a cistern and directed to a water tank of a house through a water pump that feeds a network of pipelines that supply sites that do not require potable water as toilets, yard taps and garden. This water tank also has water level sensors that tells if it's complete or not. The system informs the user all the data in a real-time system on a LCD display or through a Web page available in a local Wi-Fi network. This network is created via an Ethernet Shield, a router and the Arduino. The Arduino is an open hardware platform, easy to implement and low cost, and can be integrated into different types of projects. On the roofs of Residence a rain sensor that detects the presence of water is installed and stores the duration of the rain for calculation of rainfall intensity and output. Depending on the region where user lives, it will contribute to problems related to water scarcity, floods in urban major centers and will generate savings in the use of potable water.

Keywords: Arduino, Ethernet Shield, LCD.

Capítulo 1 – Introdução

O crescimento populacional, o processo de industrialização e o conseqüente aumento da demanda por água nos grandes centros urbanos, têm causado a insuficiência e degradação de água. Além do crescimento da população e da industrialização, a urbanização também amplia significativamente a demanda de água. Cerca de 40% da população mundial convive com a escassez de água por falta de um manejo dos recursos hídricos(Segala,2014).

Relatórios da Organização das Nações Unidas(ONU) repetem o diagnóstico cada vez mais alarmante: mais de 1 bilhão de pessoas, o equivalente a 18% da população mundial não têm acesso a uma quantidade mínima aceitável de água potável, ou seja, água segura para uso humano. Se nada mudar no padrão de consumo, dois terços da população do planeta em 2025, o equivalente a 5,5 bilhões de pessoas poderão não ter acesso á água limpa. Em 2050, apenas um quarto da humanidade vai dispor de água para satisfazer suas necessidades básicas. Com isso há a necessidade de se utilizar novas técnicas para o melhor aproveitamento dessa água. Por essa razão, não devemos descartar nenhuma fonte alternativa de água. A água de chuva permite que seja utilizada e as experiências nos conduzem á captação e ao manejo da água da chuva para fins humanos e agrícolas também. A água disponível na superfície da terra após se evaporar abandona os sólidos e sobe pura para a atmosfera na forma de vapor. Essa água sendo bem captada, tratada e armazenada pode ficar disponível para utilização para os mais diversos fins não potáveis(Revista Atualidades Vestibular,2008).

As instalações de sistemas de captação de águas pluviais têm sido cada vez mais constantes nas residências não só do Brasil, mais em vários outros países. O preço de um sistema de aproveitamento de água de chuva varia de acordo com o projeto, uma vez que cada área de telhado, que através das calhas instaladas em seus extremos, conduz a água para o reservatório, demanda configurações de componentes diversificadas. Esses sistemas têm a solução para residências em que o consumo de água é alto, gerando economia e retornando o valor de investimento com o decorrer do tempo (Lorenzete, 2011).

Esse sistema de captação requer uma série de equipamentos a começar pelas calhas que serão instaladas no telhado da residência. Elas servem para coletar a água da chuva que cai no telhado. Conectado a essas calhas haverá uma tubulação que conduz essa água para um reservatório e antes de chegar lá, a água passa por um separador de primeiras águas que funciona como um filtro. Esse filtro elimina as impurezas da água como fezes de animais e folhas. Essa água que é eliminada é conhecida como as primeiras águas de chuva(Manual de construções e instalações Versão 1.1, jan 2012) . A cisterna que armazena a água é normalmente instalada abaixo do solo, apenas com a tampa de inspeção para fora para evitar que entre bichos ou que escorra água contaminada para dentro da cisterna(São José do Cerrito, 2003)..

Para esse sistema desenvolvido, é utilizada a tecnologia Arduino que é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR (Michael McRoberts, 2013). O Arduino controla os sensores de níveis, o sensor de chuva, a bomba de água, uma tela LCD e uma Shield Ethernet que transmite informações do sistema para uma interface WEB. Os sensores de níveis são instalados na cisterna e na caixa d'água da residência e controlam o nível dessa água. A bomba é ligada ou desligada pelo módulo rele no instante em que um dos sensores for ultrapassado pelo nível de água. O módulo rele é um interruptor eletromecânico. Sua movimentação física ocorre quando a corrente elétrica percorre as espiras de sua bobina, criando um campo magnético. Esse dispositivo transforma a tensão da bomba em uma tensão que o Arduino possa suportar. O sensor de chuva é instalado no telhado da residência e possui uma resistência sensível a fluídos. Caso ele detecte a presença de água o sistema calcula o tempo dessa chuva e usa esse valor como variável para o cálculo da vazão do projeto.

1.1 . Motivação

Os sistemas de captação de água de chuva estão voltados mais para as soluções dos problemas que o meio ambiente enfrenta, não se voltando muito para o benefício individual, mais sim coletivo. Embora esses sistemas tenham sido desenvolvidos principalmente para que a água doce seja preservada e a água de chuva não seja desperdiçada, esses sistemas conseqüentemente nos trazem um retorno a longo prazo do valor do investimento, gerando uma economia na conta de água. A motivação deste projeto está relacionada com o resultados levados ao meio ambiente, que se resume em drenar as águas da chuva para que diminua o escoamento evitando enchentes e ao fator econômico preservando a água doce utilizando a água da chuva para fins não potáveis.

1.2.Objetivos

1.2.1.Objetivo Geral

Desenvolver um sistema de captação de água da chuva para utilizá-la em fins não potáveis gerando economia no uso da água e drenando a água da chuva evitando enchentes nos grandes centros urbanos.

1.2.2.Objetivos Específicos

Para atingir o objetivo geral, os objetivos específicos pautam-se em:

- Alocar os sensores de níveis no reservatório de águas pluviais e na caixa d'água da residência.
- Instalar um sensor de chuva nos telhados da residência.
- Montar um display LCD para visualizar as informações.
- Montar um módulo de componentes para alertar o usuário sobre os dados disponibilizados no sistema em um dado momento.
- Montar um módulo relé para o acionamento da bomba elétrica.

- Desenvolver um código em linguagem C para controlar o circuito da placa.
- Desenvolver uma interface WEB em linguagem HTML para o monitoramento do sistema.
- Realizar testes para que a integração do software e do hardware atendam as especificações do sistema.
- Desenvolver um protótipo que simule o sistema em funcionamento.

1.3. Metodologia

Para facilitar o desenvolvimento do sistema e atingir aos objetivos específicos, o sistema foi dividido nas seguintes etapas:

Etapa 1: Desenvolvimento do hardware pautando-se nas seguintes atividades:

- Instalar os sensores de níveis de água nos reservatórios para retornarem 0 ou 1 caso esteja com a bóia fora ou dentro d'água. Esses sensores possuem duas saídas onde uma é ligada no 5v do Arduino e outra em uma de suas portas digitais passando por um resistor conectado a porta GND para criar uma resistência a passagem da corrente.
- Um sensor de chuva é instalado no telhado para detectar a presença de fluído em sua resistência. Caso a resistência esteja em nível baixo, o programa realiza uma contagem do tempo de duração dessa chuva e armazena em uma variável para o cálculo da intensidade pluviométrica e da vazão.
- Para a montagem do display LCD de 16x2, temos 16 saídas que são distribuídas nas portas digitais, na porta 5v e na porta Ground (GND) ou terra do Arduino
- O esquema de alerta é composto por um conjunto de LEDs e um BUZZER. Eles serão conectados as portas digitais e na porta GND do Arduino e passarão pelos resistores que criam uma resistência na passagem de corrente, se opondo a uma corrente infinita.
- O modulo rele serve para acionar uma bomba d'água de 220 v apenas usando o 5v do Arduino, pois ele isola essa tensão. Essa bomba é acionada automaticamente caso um dos sensores de níveis estiver retornando valor 0.
- As portas da Shield Ethernet são conectadas as portas do Arduino e através de um cabo de rede é conectada a um roteador. Assim, uma rede local é criada na residência.

Etapa 2: O desenvolvimento do software pautando-se nas seguintes atividades:

- Desenvolver em linguagem HTML uma interface WEB. Essa interface irá receber os dados do Arduino por meio da Shield Ethernet.
- Desenvolver um código em linguagem C na IDE do Arduino para controlar o circuito do projeto.

Etapa 3: os teste operacionais de software e hardware são realizados em conjunto com a implementação pois se deixarmos para o final, temos o risco de acumular muitos erros. Os teste são realizados através da compilação de cada parte nova do código e de cada nova integração do hardware. A medida em que o projeto estiver sendo implementado e estiver evoluindo, os testes serão freqüentes.

Etapa 4: o desenvolvimento do protótipo gera um modelo em miniatura que permite a visualização de um sistema de captação de água de chuva e facilita o entendimento de cada atividade que compõe o funcionamento do sistema.

1.4.Resultados esperados

Com o desenvolvimento deste projeto espera-se um sistema capaz de transmitir a água coletada da chuva para uma caixa d'água residencial que abastece uma rede de encanamentos para fins não potáveis. Isso se fará sem o uso manual do usuário, será tudo automatizado podendo dar informações sobre o processo desse sistema ao cliente via uma interface WEB. Para tanto será utilizada algumas tecnologias como Arduino e Shield Ethernet.

O sistema terá que contar com um reservatório que armazene a água da chuva e calhas nos telhados das residências para coletar essas águas e direcioná-las por meio de condutores verticais e horizontais até a cisterna abaixo do solo. Essa água será filtrada por meio de um separador de primeiras águas de chuva dispensando as impurezas. A água captada é transmitida para a caixa d'água residencial por meio de uma bomba d'água que é acionada pelo modulo relé conectado ao Arduino.

O controle de abastecimento e de desativação da bomba de água será controlado por sensores de níveis de água tanto no reservatório de águas pluviais como na caixa d'água residencial. O tempo de duração da chuva é obtido através de um sensor de chuva. A cada período de chuva, o sistema realiza o cálculo da vazão que ocorreu no projeto. Essa vazão é calculada a partir do período chuvoso detectado pelo sensor de chuva, da pluviosidade média da região, da intensidade pluviométrica e da área de contribuição do telhado.

Com esse sistema, espera-se um retorno de economia em relação aos gastos de água em uma residência, pois ao invés de se usar água potável para descargas, lavagens de quintais, jardins e carros, será usada a água coletada da chuva. Utilizando esse sistema, o usuário não estará contribuindo só com si mesmo, mais com toda a população, pois ele diminui o escoamento de águas pluviais nos centros urbanos. Se a maioria das residências, mais não apenas uma, adotasse esse sistema, o risco de enchentes seria muito menor, pois minimizaria o escoamento de água nos pavimentos.

1.5.Organização

O trabalho está organizado conforme os capítulos a seguir:

1. Capítulo 2: Apresenta o referencial teórico sobre o sistema de captação de água de chuva, a descrição do problema a ser resolvido e a descrição das tecnologias usadas nesse sistema

2. Capítulo 3: Apresenta o desenvolvimento do projeto e as suas funcionalidades.

3. Capítulo 4: Apresenta os testes realizados durante e ao final do desenvolvimento do projeto.

4. Capítulo 5: Apresenta as considerações finais, as conclusões e melhorias que podem ser feitas para que esse projeto atinja proporções maiores e aumente sua aplicabilidade.

Capítulo 2 – Referencial teórico

Os sistemas de captação de águas pluviais vem sendo aprimorado ao longo dos anos pela necessidade de ter uma solução para problemas com excesso de água das chuvas, causando vários tipos de desastres e também com o elevado consumo e falta de gerenciamento de água potável do planeta.

2.1.A água doce no mundo

A quantidade existente de água no planeta é equivalente a 1.385.984.610 km³. Essa quantidade se encontra na forma de lagos, rios, oceanos e geleiras e sem contar com os 23.400.000 km³ de água abaixo do solo e 12.900 km³ em vapor. Pesquisas revelam que 97,5% desta água se concentra nos oceanos e os 2,5% restantes são de água doce que equivalem a 24.486.310 km³. Desse valor, 68,70% se dividem em geleiras e neves, 30% são de águas subterrâneas e 0,26% estão em lagos e rios.(Sperancetta *et al*, 2004).

O aumento da escassez de recursos hídricos é consequência do mau gerenciamento do consumo de água que se dá pelo crescimento da população, da indústria e da agricultura. O programa de Avaliação Mundial informa que a escassez de água está em 40% das pessoas em todo o mundo. Alexander Costa, chefe do departamento de geografia física da Universidade Estadual do Rio de Janeiro diz que a água doce não é finita pelo fato de se renovar a partir da evaporação dos mares. Contudo a velocidade com que a água se degrada é superior á velocidade de renovação. “*A velocidade de renovação é extremamente lenta, podendo durar décadas ou séculos. O processo humano de degradação das águas é contínuo e cada vez mais intenso*”(Moss, 2013).

O desperdício é outro fator que contribui para a escassez de água no planeta. O consumo de água está totalmente ligado a agropecuária e a indústria, cerca de 70% e 20 % respectivamente. Apenas 10% se destina ao consumo doméstico de acordo com o Programa de Avaliação Mundial da Água. Com relação a esses dados, Costa cita que “*os consumidores de água no país são a agricultura e a indústria. O consumo doméstico é muito menor*” (Moss, 2013).

2.2.Os riscos das águas pluviais

As águas provenientes das chuvas, chamadas águas pluviais podem trazer agravantes principalmente para grandes centros urbanos, pois há grande adensamento populacional. Durante o tempo de chuva forte e contínua essas águas podem transbordar e causar enchentes e alagamentos de rios e lagos, trazendo várias doenças para a população. Os danos podem ser grandes, de acordo com o volume de águas que saem do leito normal e de acordo com a densidade populacional. Tudo isso é ocasionado pelo excesso de escoamento que vai para a cidade, pois essa água não é coletada e/ou drenada(Pena, 2014).

Os problemas causados pelas enchentes podem ocorrer por causas naturais, através de rios intermitentes, que nunca secam durante o ano. Eles possuem dois tipos de leitos: um menor e principal, por onde a água corre durante a maior parte do tempo, e um maior e complementar, que é inundado apenas em períodos de cheias. Por outro lado, o problema das enchentes pode e quase sempre está relacionado a interferência humana, que são as causas antrópicas das enchentes. Essas causas estão relacionadas a poluição por parte da população mais esse não é o único caso. A maioria dos casos está relacionada ao mau uso do espaço urbano pela população, por falta de um sistema de drenagem, pois a água que deveria ser infiltrada no solo, escorre pelos pavimentos e asfaltos das cidades. Caso esse escoamento fosse diminuído através de uma drenagem ou coleta dessa água por parte da maioria das residências de uma cidade, essas enchentes diminuiriam bastante seu impacto ou dependendo da gravidade, evitaria ela(Pena, 2014).

2.3.Norma de Instalações prediais de águas pluviais (NBR 10844)

Essa norma cita várias definições e critérios para dimensionar calhas e condutores em sistemas de águas pluviais. Ela fixa exigências e critérios necessários aos projetos das instalações desses sistemas, visando garantir níveis aceitáveis de funcionalidades, segurança, higiene, conforto, durabilidade e economia. Se aplica a drenagem de águas pluviais em coberturas, terraços e quintais.(NBR 10844, 89).

2.3.1.Definições

Para os efeitos desta norma, são adotadas algumas definições dentre outras:

2.3.1.1.Área de contribuição

Soma das áreas das superfícies que, interceptando a chuva, conduzem as águas para determinado ponto da instalação (NBR 10844, 89).

No cálculo da área de contribuição, devem-se considerar os incrementos devidos á inclinação da cobertura e ás paredes que interceptam água de chuva que também deve ser drenada pela cobertura. A norma contribui com dois tipos de superfícies mostradas nas figuras 2.1 e 2.2 (NBR 10844, 89).

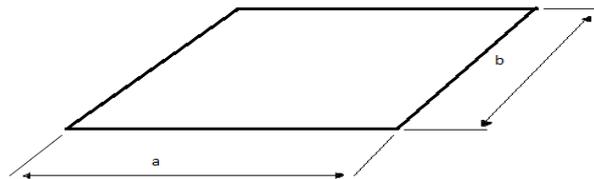


Figura 2.1 – Superfície horizontal

(Adaptado de:NBR 10844, dez 1989)

Podemos chegar ao valor da área de contribuição através da eq(1):

$$A = a \cdot b \quad eq(1).$$

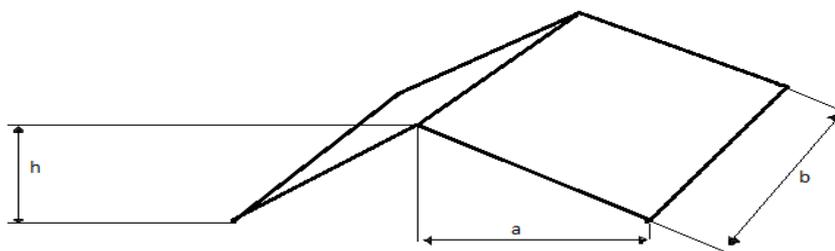


Figura 2.2 – Superfície vertical

(Adaptado de:NBR 10844, dez 1989)

A área de contribuição da superfície vertical se dá pela eq(2):

$$A = \left(a + \frac{h}{2} \right) \cdot b \quad eq(2).$$

2.3.1.2. Vazão de projeto

A vazão do sistema de captação de águas pluviais é calculada pela equação eq(3) logo abaixo(NBR 10844, 89):

$$Q = I \cdot A_c \quad eq(3).$$

- Q = vazão de escoamento em (m^3/min) ou (L/min).
- A_c : área de contribuição(m^2).
- I = intensidade pluviométrica(m/min).

2.3.1.3.Tempo de concentração

É o intervalo de tempo que decorre do início da chuva e o momento em que toda a área de contribuição passa a contribuir para determinada seção transversal de um condutor ou calha(NBR 10844,89).

2.3.2.Fatores Meteorológicos

Para o cálculo da quantidade de chuva que será interceptada por superfícies inclinadas ou verticais, é necessário levar em conta a ação dos ventos e adotar um ângulo de inclinação da chuva em relação á horizontal. Esse ângulo deve ser igual a arc tg^2x como mostra a equação 4 abaixo. O vento deve ser considerado na direção que ocasionar maior quantidade de chuva interceptada pelas superfícies(NBR 10844,89).

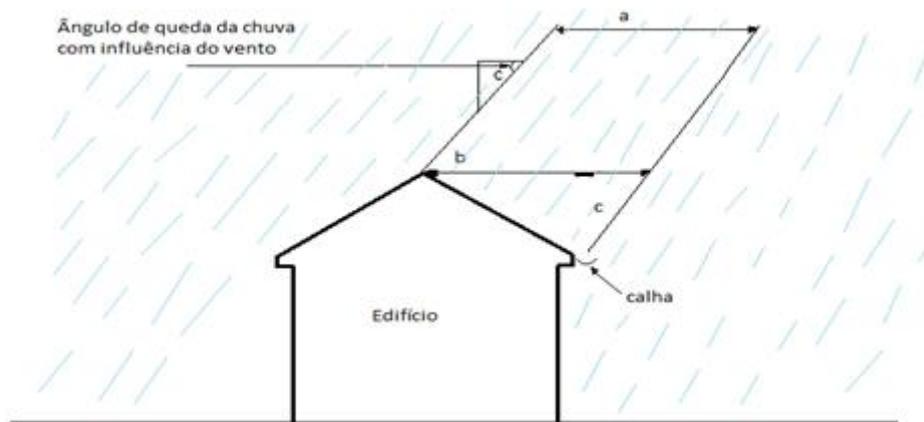


Figura 2.3 – Influência do vento na inclinação da chuva.

(Adaptado de:NBR 10844, dez 1989)

$$c = \text{arcTg}^2(x) \quad \text{eq(4)}.$$

2.4.O sistema de captação de águas pluviais

A partir do momento em que uma sociedade decide captar águas da chuva para utilizá-las, não estará somente tendo um retorno de economia, mais também estará combatendo grandes ciclos de escassez de água e de enchentes nas grandes cidades. O resultado de um sistema como esse gera menos impacto de enchentes, pois minimiza grande parte do escoamento de água para os encanamentos pluviais. Essa água é

coletada para depois ser utilizada. Esse sistema funciona também como uma drenagem urbana, pois essa água é escoada por meio de tubos e armazenada em cisternas, diminuindo o escoamento nos pavimentos e asfaltos. O desperdício é menor, pois se utiliza a água da chuva em vez da água doce para consumo de fins não potáveis. Assim, essa solução contribui com a diminuição de escassez de água, preservando a água doce e desagravando e /ou evitando os problemas de enchentes em grandes cidades(Prefeitura de São Paulo, 2012).

Esses tipos de instalações pluviais são compostas por calhas instaladas em telhados, condutores que dirigem as águas a um reservatório ou cisterna, normalmente acoplados abaixo do nível do solo, apenas com sua tampa de inspeção para fora. Antes de ser armazenada na cisterna, a água passa por um separador de primeiras águas de chuva, que funciona como um filtro, eliminando as impurezas da água como fezes de animais, folhas e resíduos. As tubulações envolvidas nessas instalações requerem cuidados, como o fato de que as tubulações das águas pluviais terem de ser utilizadas separadamente das tubulações das águas do abastecimento público, evitando levar poluição ou diminuição da qualidade da água para a cidade. FENDRICH (2002) estabeleceu uma medida preventiva com relação a esse problema, que seria pintar a tubulação das águas pluviais de verde escuro para sua fácil identificação, pois se trata de uma solução ambiental(Sperancetta *et al*, 2004).

São necessárias algumas atividades para operar um sistema como esse:

- Coletar águas pluviais através de calhas em telhados, coberturas.
- Armazenar essa água em reservatórios, cisternas.
- Abastecer o local onde será utilizada as águas pluviais.
- Eliminar as primeiras águas de chuva, preservando a qualidade dessa água.

2.4.1.Cisternas

A cisterna deve ser dimensionada com cuidado pois é o principal custo do sistema. O cliente deve está ciente da capacidade de armazenamento e o local em que será instalada. Para o seu dimensionamento deve levar em consideração os seguintes critérios: custos totais de implantação, demanda de água e a área do telhado da residência. A distribuição temporal anual das chuvas é uma importante variável a ser considerada no dimensionamento do reservatório. É importante a cisterna estar instalada acima do lençol freático. A distância mínima do lençol para sua base de assentamento é de 1,00m. A profundidade da escavação é determinada pela altura dos tanques. No processo de escavação, é importante a análise do tipo de solo ou terreno onde o tanque será instalado. A qualidade do solo e suas características de expansibilidade determinam o aterramento(SNatural, 1989-2011).

Geralmente a cisterna possui um formato cilíndrico e é feita de ferrocimento. Isso garante a manutenção da qualidade da água por longos períodos de tempo. A tabela abaixo facilita o dimensionamento da cisterna(São José do Cerrito , 2003).

Tabela 2.1 – Capacidade dos reservatórios cilíndricos.

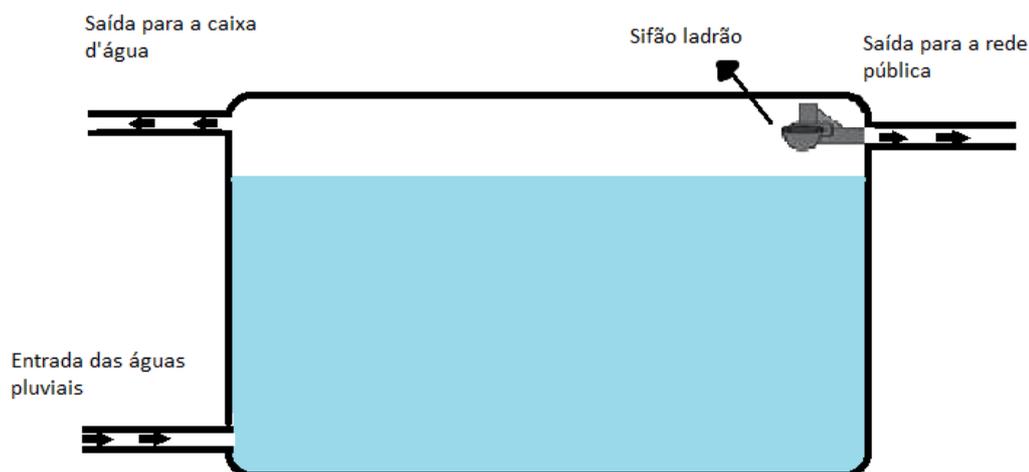
Altura da peça	0,50m	1,00m	1,50m	2,10m
Diâmetro da base				
1,00m	400 litros	800 litros	1.200 litros	1.600 litros
1,50m	900 litros	1.800 litros	2.600 litros	5.700 litros
2,00m	1.600 litros	3.100 litros	4.700 litros	6.600 litros
2,50m	2.500 litros	4.900 litros	7.400 litros	10.300 litros
3,00m	3.500 litros	7.000 litros	10.600 litros	14.800 litros
3,50m	4.800 litros	9.600 litros	14.400 litros	20.200 litros
4,00m	6.300 litros	12.600 litros	18.800 litros	26.400 litros
4,50m	8.000 litros	15.900 litros	23.800 litros	33.400 litros

(Fonte: São José do Cerrito, maio de 2003)

O reservatório possui 3 saídas de água:

- A primeira direciona a água coletada para a caixa d'água residencial através de uma bomba d'água que fica no fundo do reservatório.
- A segunda serve para que a água coletada pelas calhas nos telhados possa entrar no reservatório.
- A terceira serve para evitar o excesso de água no reservatório. A água será escoada para a rede pública por meio de um sifão ladrão.

O sifão ladrão impede a entrada de insetos e roedores no reservatório. É instalado dentro da cisterna e faz o controle do nível para que não haja transbordamento enviando o excesso de água para a rede pública. A figura 2.4 mostra um esquema de cisterna com três saídas.

**Figura 2.4-Esquema de cisterna**

(Fonte: Autor)

2.4.2. Calhas

A idéia de usar calhas é a de captar a água da chuva antes que cheguem no solo, onde normalmente se contamina e fica imprópria para uso. As calhas têm por objetivo coletar as águas de chuva que caem sobre os telhados e conduzi-las aos condutores verticais e horizontais. As seções das calhas possuem varias formas, a depender das condições imposta pelo projeto e dos materiais empregados em sua confecção. Muitos dos danos estruturais de casas e prédios são resultados direto da ação da água e da ausência de calhas(Lorenzete, 2011).

Conforme a norma de instalações prediais de águas pluviais NBR 10844, as calhas devem ser feitas de chapas de aço galvanizado, (NBR 7005, NBR 6663), folhas-de-flandres (NBR 6647), chapas de cobre (NBR 6184), aço inoxidável, alumínio, fibrocimento, PVC rígido, fibra de vidro , concreto ou alvenaria (NBR 10844, 89).

As calhas não devem ser acanhadas ou ter pouca queda, elas são interligadas a um cano condutor que leva a água até o nível do solo chegando a cisterna. A capacidade de vazão das calhas e dos bocais são dados utilizados para o dimensionamento do sistema de coleta de chuvas de telhados e coberturas. Nos cálculos de dimensionamento de sistemas de coleta de águas pluviais, é preciso calcular a vazão máxima do projeto, que vai depender do regime de chuvas da região onde está a construção e da área de contribuição do trelhado (Enciclopédia Tigre, 2010).

Para calcular a vazão do projeto, utiliza-se a equação eq(3) da seção 2.3.1.2:

$$Q = I. Ac.$$

Para determinar o valor da intensidade pluviométrica máxima, a tabela 2.2 apresenta os dados de pluviosidade média anual e de duração média do período chuvoso para as regiões do país(INPE, 2004).

Tabela 2.2 – Intensidade Pluviométrica

Região	Pluviosidade Média Anual(mm)	Duração do Período Chuvoso(meses)	Intensidade Pluviométrica Máxima(mm/mês)
Norte	3400	7	485,7
Nordeste	2000	1	2000,0
Sudeste	3600	6	600,0
Sul	2000	9	222,2
Centro-Oeste	3000	7	428,5

(Fonte : Crepani, 2001)

A intensidade pluviométrica é dada pela equação eq(5):

$$I = \frac{P}{D} \quad eq(5),$$

- I = Intensidade Pluviométrica(mm/mês). Para obter a vazão do projeto em L/min, é necessário converter a unidade da intensidade pluviométrica em m/min. Para isso converte-se o período D em minutos e a pluviosidade P em m.
- D=Duração do Período Chuvoso(meses).
- P=Pluviosidade Média Anual(mm).

A pluviosidade média é a medida da precipitação de quantidade de chuva em um determinado local(INPE, 2004).

Destacam-se dois tipos principais de calhas de acordo com a figura 2.5.

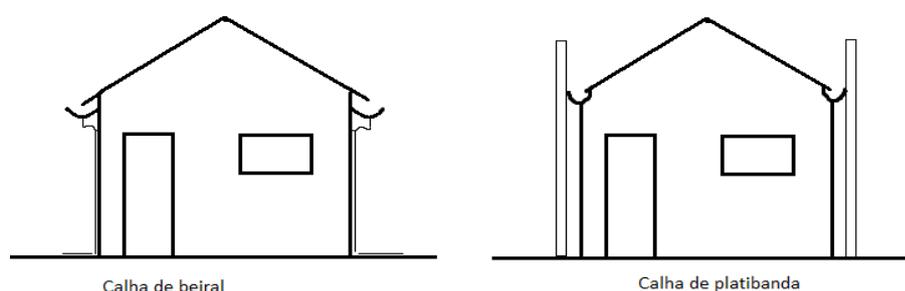


Figura 2.5 – Tipos de calhas.

(Fonte:Autor)

Sempre que possível as calhas de platibanda e beiral devem ser fixadas centralmente sob uma extremidade de uma cobertura. Suas inclinações devem ser uniformes, tendo como valor mínimo 0,5% que equivale a 5mm de inclinação por metro(NBR 10844,89).

A forma da calha influencia na sua capacidade de escoamento. Normalmente ela é retangular ou semicircular. Em função da forma da calha a norma NBR 10844 fornece sua capacidade hidráulica. Note que a declividade deve ser de no mínimo 0,5% para que a água escoe com maior rapidez para o ponto de coleta (NBR 10844,89).

Tabela 2.3 – declividade das calhas

Diâmetro interno D(mm)	Declividades		
	0,5%	1%	2%
100	130	183	256
125	236	339	466
150	384	541	757
200	829	1.167	

(Fonte:arq.ufsc.br)

Curvas na calha também influenciam na sua capacidade de escoamento. As calhas curvas terão um fator de decréscimo de sua capacidade se compararmos com as

calhas retas. Para calcular o dimensionamento da calha curva a norma NBR 10844 optou por multiplicar a vazão do projeto por um fator maior que 1, conforme a tabela abaixo(NBR 10844,89).

Tabela 2.4 – coeficientes multiplicativos da vazão de projeto

Tipo de curva	Curva a menos de 2m da saída da calha	Curva entre 2 e 4m da saída da calha
Canto reto	1,20	1,10
Canto arredondado	1,10	1,05

(Fonte:arq.ufsc.br)

2.4.3. Condutores

Condutores conduzem as águas pluviais dos telhados, terraços e áreas abertas por meio de tubos. Essa água é conduzida ao local de lançamento por coletores. O local de lançamento pode ser um coletor público, uma galeria de águas pluviais, uma caixa de ralo na via pública, um canal ou rio(Macintyre, 2010).

Existem 2 tipos de condutores: verticais e horizontais.

2.4.3.1. Condutores verticais

Nos condutores verticais, devem ser empregados tubos e conexões de ferro fundido (NBR 8161), fibrocimento, PVC rígido (NBR 10843, NBR 5680), aço galvanizado (NBR 6663, NBR 7005), folhas-de-flandres(NBR 6647), chapas de cobre(NBR 6184), aço inoxidável, alumínio ou fibra de vidro.

Os condutores podem ser instalados externos ou internos ao edifício. Isso depende de considerações de projeto, do uso e da ocupação do edifício e do material dos condutores. O diâmetro interno mínimo dos condutores verticais de seção circular é 70mm(NBR 10844,89).

O condutor vertical pode ser ligado, na sua extremidade superior, diretamente a uma calha, ou receber um ralo quando se trata de terraços ou calhas largas, onde se receia a obstrução do condutor por folhas, papéis, e detritos diversos (Macintyre, 2010).

O dimensionamento dos condutores verticais deve ser feito a partir dos seguintes dados:

Q = Vazão de projeto, em L/min.

H = Altura da lâmina de água na calha, em mm.

L = Comprimento do condutor vertical, em m(NBR 10844, 89).

2.4.3.2. Condutores horizontais

Os condutores horizontais devem ser projetados com declividade uniforme, com valor mínimo de 0,5%. Neles, devem ser empregados tubos e conexões de ferro fundido (NBR 8161), fibrocimento(NBR 8056), PVC rígido(NBR 10843, NBR 5680), aço galvanizado(NBR 5580, NBR 5885), cerâmica vidrada(NBR 5645), concreto(NBR 9794), cobre, canais de concreto ou alvenaria.

Outro fator importante nos condutores horizontais de seção circular é o seu dimensionamento. A NBR já fornece a capacidade em função da declividade, diâmetro e rugosidade (n) que é uma medida das variações do relevo de uma superfície(NBR 10844,89).

Tabela 2.5-capacidade de condutores horizontais

Diâmetro interno D(mm)	n=0,011				n=0,012				n=0,013			
	0,5%	1%	2%	4%	0,5%	1%	2%	4%	0,5%	1%	2%	4%
1	2	3	4	5	6	7	8	9	10	11	12	13
50	32	45	64	90	29	41	59	83	27	38	54	76
75	95	133	188	267	87	122	172	245	80	113	159	226
100	204	287	405	575	187	264	372	527	173	243	343	486
125	370	521	735	1.040	339	478	674	956	313	441	622	882
150	602	847	1.190	1.690	552	777	1.100	1.550	509	717	1.010	1.430
200	1.300	1.820	2.570	3.650	1.190	1.670	2.360	3.350	1.100	1.540	2.180	3.040
250	2.350	3.310	4.660	6.620	2.150	3.030	4.280	6.070	1.990	2.800	3.950	5.600
300	3.820	5.380	7.590	10.800	3.500	4.930	6.960	9.870	3.230	4.550	6.420	9.110

(Fonte:NBR 10844, dez 1989)

A rugosidade é definida como erros microgeométricos, é um conjunto de irregularidades, pequenas saliências e reentrâncias que caracterizam uma superfície. A rugosidade influencia bastante na qualidade de deslizamento, resistência ao desgaste entre outros fatores (Fabiani, 2014).

2.5.Tratamento da água de chuva

O tratamento da água da chuva armazenada é realizado se for feito corretamente. Se a utilização pretendida para a água da chuva é interior, para usos domésticos potáveis, devem ser aplicadas práticas apropriadas de filtração e desinfecção. Se a água da chuva é utilizada no exterior, para irrigação do jardim ou lavagens de quintais e carros, ou até mesmo para uso interior não potável como descargas de vasos sanitários, as exigências de tratamento são menos restritas(Weierbacher, 2008).

Para esse projeto a água da chuva é utilizada para fins não potáveis. Nesse caso usa-se o separador de primeiras águas de chuva que separa e descarta as primeiras águas de chuvas fortes que fazem a lavagem da atmosfera, do telhado, calhas e tubulações. Após esse descarte, as próximas águas vão estar mais limpas e então serão direcionadas para a cisterna. Esse componente é instalado no condutor de descida de água da calha do telhado e é dividido em três partes:

- Reservatório temporário para descarte da primeira água de chuva forte.
- Saída para descarte das águas do reservatório temporário.
- Desvio da água de chuva forte para a cisterna (Manual de construções e instalações Versão 1.1, 2012).

Separador de primeiras águas de chuva.



Figura 2.6 – Separador de primeiras águas de chuva

(Fonte:sempresustentavel.com.br)

2.6.Arduino

O projeto Arduino se iniciou na Itália em 2005. No começo, o Arduino tinha sido criado para fins acadêmicos, com o intuito de interagir em projetos escolares de forma a ter um orçamento menos que outros sistemas de prototipagem disponíveis naquela época. De 2005 até 2008 teve uma marca de 50.000 placas vendidas e em 2006 recebeu uma menção honrosa na categoria Comunidades Digitais pela Prix Ars estabelecendo então o começo de seu sucesso(Monk, 2013).

Arduino é um projeto simples, muito popular e acessível. Possui eletrônica e programação embarcada de alto nível. Ele se divide no componente de placa de hardware e o software IDE que é um programa de computador que se usa para desenvolver o código e fazer o upload para a placa Arduino. É um computador programável para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele. O sistema do Arduino interage com um ambiente por meio de hardware e software. Uma de suas utilidades é desenvolver objetos interativos que são independentes. Podem também enviar um conjunto de dados recebidos de sensores para um site(Banzi 2010,McRoberts 2011).

Esse pequeno computador é de fonte aberta tanto para hardware como para software. Isso significa que o código, os esquemas e o projeto podem ser utilizados livremente por qualquer pessoa e com qualquer propósito. Ele é composto por um Atmel AVR que é um microcontrolador RISC de chip único que tem programas e dados armazenados em sistemas de memória física separados, mas possui a habilidade de ler

os itens da memória do programa. Esse microcontrolador funciona como um computador completo, com memória RAM, memória de programa(ROM), uma unidade de processamento de aritmética e os dispositivos de entrada e saída. A maioria das placas inclui um regulador linear de 5V e um oscilador de cristal de 16 MHz. O regulador linear é um regulador de tensão que funciona como um resistor variável que ajusta continuamente uma rede divisora de tensão para manter uma voltagem de saída constante. O oscilador de cristal é um componente eletrônico e ele utiliza a ressonância de um cristal quando está em vibração para criar um sinal elétrico com frequência precisa. O oscilador funciona como um relógio e usa essa frequência para medir precisamente o tempo, assim permitindo sua operação na velocidade correta. As placas Arduino possuem saídas USBs que conectam a um computador para fazer upload ou recuperação dos dados do sistema e podem ser conectados a diversos objetos como LEDs, displays, botões, interruptores, motores, receptores GPS, módulos Ethernet e vários tipos de sensores como os de temperatura, de pressão e de distância(McRoberts, 2011).

Esse incrível microcontrolador também pode ter suas funcionalidades estendidas apenas utilizando as Shields. São placas de circuitos que contém outros dispositivos como receptores GPS, displays de LCD e módulos de Ethernet. Essas placas de circuitos se conectam ao Arduino para obter outras funcionalidades(McRoberts, 2011).

2.6.1.Arduino Mega

O Arduino Mega é uma placa de microcontrolador que possui 54 pinos de entradas e saídas digitais, 16 entradas analógicas, 4 UARTs(portas seriais de hardware), um oscilador de cristal de 16 MHz, uma entrada de alimentação, uma conexão USB e uma conexão ICSP. (Arduino, 2014).

Tantos os pinos digitais como os analógicos possuem funções variadas. Podem ser de entrada ou de saída. Alguns podem servir para leituras analógicas e também como entradas digitais. Todas essas funções são escolhidas pelo programador(McRoberts, 2011).



Figura 2.7-Arduino Mega.

(Fonte: <http://store.arduino.cc/product/A000067>)

Os pinos digitais e inclusive os analógicos podem ser programados para serem entradas digitais. Os pinos são programados para servirem de entradas digitais através do programa escrito pelo seu desenvolvedor e nele é inserido um comando que ao ser

executado efetua leitura de tensões aplicadas ao pino que está sendo lido. Após essa leitura, é possível saber se o pino encontra-se em um estado alto ou baixo. As entradas digitais apenas informam se existe ou não uma tensão aplicada em seu pino, elas não conseguem determinar o quanto de tensão está sendo aplicada no pino, só entregam 0 ou 1, com ou sem tensão. Na prática o programa pode saber se um pino está alimentado com 0 ou 5 Volts. Além das entradas digitais, temos as saídas digitais. Elas fazem com que um pino libere 0 ou 5 Volts. Através de um pino programado como saída podemos acender vários dispositivos como LEDs, relés e motores (JACEE, 2012).

As entradas analógicas são capazes de medir a tensão que foi aplicada. Através dela conseguimos utilizar sensores que convertem várias grandezas físicas ou químicas em um valor de tensão que depois é lido pela entrada analógica(JACEE, 2012).

2.6.2.Shield Ethernet

O Arduino pode controlar sensores ou enviar informações remotamente. A Shield Ethernet possibilita o acesso a essas informações em uma rede local ou até mesmo na internet e permite o seu monitoramento em qualquer lugar do mundo e é compatível com o Arduino Uno e Mega. Essa placa é acoplada a placa de hardware Arduino. Para estabelecer uma conexão é necessário utilizar um cabo de rede conectado a Shield e configurar um IP válido da rede. Esse cabo de rede é um RJ45, o mesmo usado em modems de acesso a conexão de banda larga(Arduino, 2014).



Figura 2.8-Shield Ethernet

(Fonte:arduino.com)

A Shield Ethernet possui um componente chamado Wiznet W5100 e através dele o usuário tem acesso á rede IP nos protocolos TCP ou UDP e seu uso pode ser facilitado pelo uso de bibliotecas Ethernet como a Library e SD Library. Esse componente hospeda interfaces BUS que são infinitamente variáveis podendo ser simples ou complexas e possui memória de 16Kbytes interno para processar pacotes de rede TCP/IP. O Wiznet é capaz de dar suporte ao Auto-MDIX que permite com que o equipamento descubra se o tipo de cabo conectado é cross ou reto e depois disso configura automaticamente a interface para aceitar esse tipo de cabo. Além desse componente, a Shield Ethernet possui um slot para cartão micro-SD que pode ser usado para armazenar arquivos que vão servir na rede. O cartão micro-SD possui 16 Gb de capacidade e armazenamento de alta qualidade. Esse cartão gerencia recursos de um

modo bastante avançada e através disso ele possibilita ao usuário portabilidade e maior desempenho, além de uma velocidade de leitura e gravação de até 30 MegaBytes por segundo(Silva 2013).

As Shields atuais possuem o módulo Power over Ethernet. Essa tecnologia é descrita pelo padrão IEEE 802.3 como uma tecnologia que transmite energia elétrica junto com os dados para qualquer dispositivo remoto, através de cabamentos em uma rede Ethernet. A Ethernet Shield é de fácil aplicação e compatível com o Arduino, com tensão de operação de 5V e velocidade de conexão de 10/100Mb. Ela estabelece conexão com o Arduino por meio da porta SPI(Serial Peripheral Interface) que é um protocolo que permite a comunicação do microcontrolador com diversos outros componentes, formando uma rede. A Ethernet contém uma série de LEDs informativos de acordo com Silva (2013):

- PWR: indica que a placa e o shield estão sendo alimentados.
- LINK: indica a presença de um link de rede e pisca quando o shield transmite ou recebe dados.
- FULLD: indica que a conexão de rede é full duplex, onde o modo de operação se resume em transmitir e receber dados simultaneamente.
- 100M: indica a presença de uma ligação de rede 100Mb/S.
- RX: pisca quando o shield envia dados.
- COLL: pisca quando são detectados colisões de rede.

2.6.3.IDE Arduino

A IDE do Arduino é um software que utiliza uma interface gráfica desenvolvida em Java e fornece uma capacidade de estender várias bibliotecas para o melhor desenvolvimento do código-fonte. O programa desenvolvido na IDE é diretamente enviado para a placa Arduino. A IDE compilar códigos em linguagens C e C++ através de um compilador GCC que na verdade é um conjunto de compiladores de linguagens de programação produzido pelo projeto chamado GNU que tem como objetivo criar sistemas operacionais baseados em software livre. O GCC permite compilar o código-fonte em binários executáveis para as várias plataformas de informática mais comuns sendo então bastante flexíveis (JACEE, 2012).

A IDE do Arduino possui uma ferramenta muito útil chamada Serial Monitor (Figura 2.9). O Serial faz a depuração do código e através do seu monitor exhibe os dados enviados de seu Arduino. É possível também pelo monitor enviar dados de volta para o Arduino. O Serial exhibe uma taxa de transmissão que é a taxa por segundo em que alterações de estado ou bits são enviados para a placa e possui uma configuração padrão de 9.600 bauds(McRoberts, 2013).

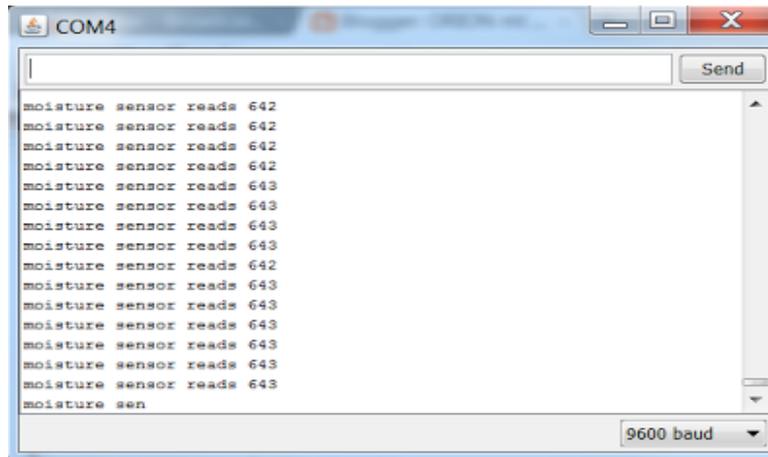


Figura 2.9-Serial Monitor do Arduino

(Fonte: orionmultitouch.blogspot.com)

Para desenvolver um software na IDE, o programa normalmente é dividido em duas partes principais que são as funções Setup e Loop. Na função Setup, a placa e o programa são inicializados e executados uma vez sempre que a placa é ligada ou resetada. Nessa função se define as configurações do hardware por exemplo: o pino 10 da placa Arduino será um pino de saída digital . Esse pino pode está conectado a algum dispositivo como um LED ou um relé. A função Loop executa o programa indefinidamente. Ao terminar a execução do último comando dessa função o programa inicia novamente a partir do primeiro comando dela e essa função permanece em execução até que a placa seja desligada ou resetada (JACEE, 2012).

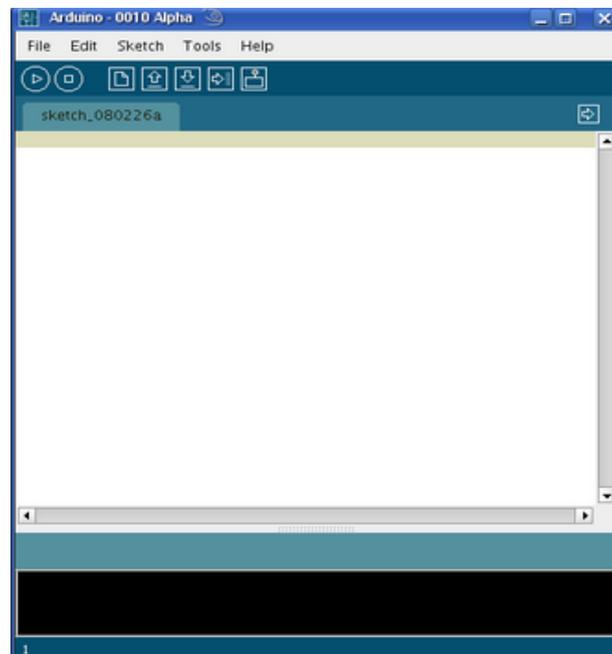


Figura 2.10-IDE Arduino

(Fonte: roboticaifal.webnode.com.br)

2.7.Módulo Relé

O estudo do relé iniciou-se por Joseph Henry, considerado o criador do relé, quando estudava fenômenos eletromagnéticos, indução eletromagnética, auto indutância e eletroímãs. Depois da morte de Joseph, por volta de 50 anos depois o relé começou a ser utilizado em uma escala ampla, para diversas aplicações. Os relés são considerados uma espécie de antepassados dos transistores, pois esses tinham suas aplicações muito limitadas, caras e bastante lentas. Esses dispositivos estão isolados totalmente dos seus sistemas de controle permitindo com que eles trabalhem com diversos valores de tensão (Cunha 2009, Santos 2014).

O relé é usado em diversas aplicações, pois representa uma quantidade ampla de equipamentos e dispositivos, com diferentes formas de construção e operação. Essa gama de dispositivos e operações são utilizadas para diversas aplicações a depender da importância, do porte e da segurança da instalação que for considerada. O relé carrega várias formas construtivas, cada uma utilizada com fundamentos básicos. Algumas dessas formas, de acordo com Filho (2005) são:

- Relés de indução.
- Relés digitais.
- Relés térmicos.
- Relés fluidodinâmicos.
- Relés eletromagnéticos.

O relé eletromagnético é utilizado em uma variedade de aplicações, incluindo alarmes e sensores e inclusive é usado nesse projeto de captação de águas pluviais. Esse relé isola sinais e sua movimentação física depende da corrente elétrica percorrida na espira da bobina interna que ao passar por ela cria um campo magnético. O campo magnético gerado atrai a alavanca responsável pela mudança de estado dos contatos. O relé eletromagnético possui três contatos: comum, normalmente fechado(NF) e normalmente aberto(NA). Caso a bobina esteja sem energia, o contato comum estará conectado no normalmente fechado. Quando essa bobina recebe uma corrente, o contato comum se desloca do normalmente fechado passando a se conectar no normalmente aberto. A bobina não tem nenhuma ligação elétrica entre os contatos, ou seja, ela é totalmente isolada. A figura 2.11 ilustra o esquema do relé(Branco, 2011).

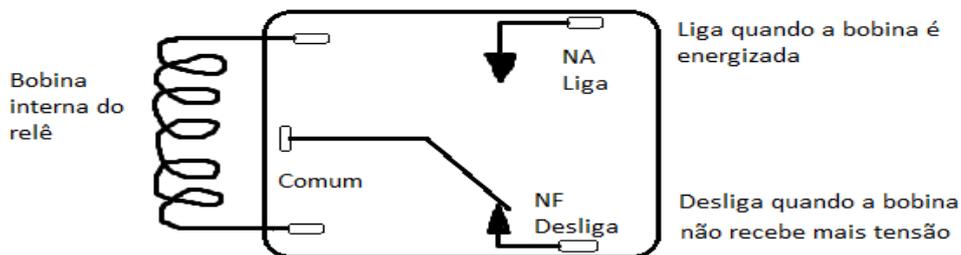


Figura 2.11-Esquema do módulo relé

(Fonte:Autor)

2.8.Linguagem HTML.

O HTML foi criado em 1991 no CERN, localizado na Suíça, por Tim Berners-Lee. Ela surgiu junto com o protocolo HTTP que ambos possibilitaram a internet ser popularizada. No começo essa linguagem foi desenvolvida para interligar instituições de pesquisas próximas e compartilhar documentos com mera facilidade. Um ano depois a biblioteca de desenvolvimento WWW(World Wide Web) que é uma rede de alcance mundial foi liberada. Assim, junta com o HTML ela proporcionou o uso em escala mundial da WEB(Pacievitch, 2014).

É uma linguagem de marcação de hipertexto e seus arquivos são interpretados pelos browsers. Um arquivo HTML possui a extensão .html, porém elas podem ser geradas dinamicamente com algumas linguagens como o PHP,ASP e PYTHON. Alguns casos essa extensão pode ser alterada para a extensão da linguagem utilizada para gerar a página HTML Essa linguagem trabalha com o sistema de tags que são estruturas de linguagens de marcação que consistem em breves instruções tendo uma marca de início e outra de fim (Oriente, 2014).

Existe vários tipos de Tags:

- <html> - documento.
- <head> - cabeçalho.
- <title> - título da página.
- <body> - corpo do documento.
- <p> - novo parágrafo.

O texto HTML é armazenado em um arquivo localizado em um lugar onde um servidor web possa lê-lo. Um navegador recupera o conteúdo desse arquivo a partir de um servidor web. Depois disso, o navegador lê o conteúdo retornado pelo servidor e o

apresenta como um texto formatado. Apenas o navegador e não o servidor interpreta o texto em HTML. O servidor apenas informa ao navegador sobre o tipo de conteúdo que está retornando, para distingui-lo por exemplo, de um documento em PostScript(Coulouris, 2007).

2.9.Sistemas de Controle

Os sistemas de controles começaram a ser desenvolvidos no século XVIII, quando James Watt iniciou o desenvolvimento de um controlador centrífugo para controle de velocidade de uma máquina a vapor. Passado um tempo outros trabalhos foram desenvolvidos por Minorsky, Hazen e Nyquist que colaboraram com a evolução desses sistemas. Com o desenvolvimento contínuo dos sistemas de controle, métodos de resposta de frequência tornaram possível aos engenheiros projetar sistemas a malha fechada que satisfazia os requisitos de desempenho e esses métodos proporcionaram sistemas mais estáveis, que satisfaziam vários requisitos arbitrários. Hoje os sistemas de controle são muito importantes nos processos de indústria por conta da disponibilidade de computadores que começou a surgir na década de 60. Além de integrarem os processos de indústria, são importantes nas operações de controle de pressão, temperatura, umidade, viscosidade e vazão (Ogata, 2000).

Sistemas de controle contém subsistemas e outros processos desenvolvidos com um certo objetivo de se obter uma saída que foi estimada com um desempenho desejado a partir de uma entrada específica. São na verdade um conjunto de dispositivos que estão relacionados entre si e que gerenciam o comportamento de outros dispositivos. Os processos dos sistemas de controle possuem sinais aplicados de controle e sinais de saída dos processos. Os sinais são aplicados na entrada de controle para que os sinais de saída atinjam um valor pré-determinado. Esses processos estão sujeitos a perturbações também conhecidas como ruídos que são influências externas ao sistema(C.Dorf, 2013).

2.9.1.Controle em malha aberta

Nesse sistema de controle, um sinal pré-determinado(X) ou sinal de controle é aplicado na entrada do sistema. A partir da aplicação desse sinal, espera-se atingir um determinado comportamento ou um valor específico no sinal de saída(Y). Em um sistema de malha aberta não é observada a evolução do processo para a determinação ou correção de um sinal de controle aplicado ao sistema. Em outras palavras, o sinal de controle não é calculado por uma medição de valor do sinal de saída. Esse tipo de controle é sensível a perturbações externas e internas, não possuem habilidades para corrigir seus efeitos e a variável controlada pode modificar o seu valor em função da entrada de perturbações. O controlador em malha aberta atua como se não houvesse ocorrido qualquer perturbação, por isso a saída desejada irá ocorrer somente se não houver nenhuma perturbação externa. Os sistemas em malha aberta são imprecisos, não se adaptam a variações externas, não possuem realimentação, são fáceis de implementar e possuem baixo custo pelo fato de não envolverem equipamentos sofisticados para medição ou determinação de sinais de controle(Gomes 2000).

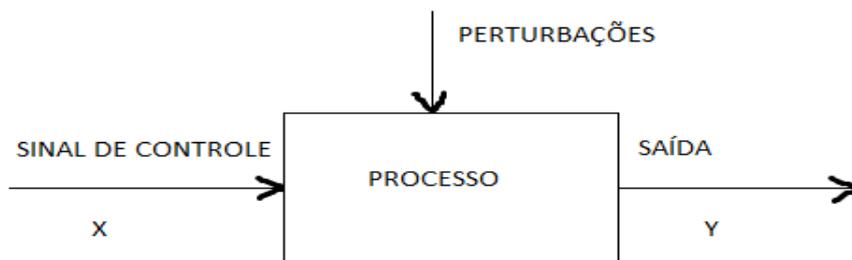


Figura 2.12-Controle em malha aberta

(Adaptado de:WWW.ece.ufrgs.br)

No sistema de malha aberta não é permitido realizar compensações para qualquer perturbação. Um exemplo disso é se um amplificador eletrônico qualquer for atingido por um ruído(perturbação), então o efeito disso irá corromper o sinal de saída(S. Nise, 2012).

2.9.2.Controle em malha fechada

O sistema em malha fechada é também chamado de retroativo pois é necessário fazer uma realimentação do sistema. A partir dessa realimentação, que é feita da saída para a entrada, o sistema coleta informações sobre como a saída de controle está evoluindo e essas informações são utilizadas para determinar o sinal de controle que deve ser aplicado a um processo em um certo instante. O sinal de controle(X) que deve ser aplicado no processo em execução é determinado pelo desvio chamado de erro(E), que é gerado pela comparação do sinal de saída(Y) com um sinal de referência(Z), que também é chamado de set-point. Logo o sinal de controle é determinado para poder corrigir esse desvio entre o sinal de saída e o sinal de referência. As informações da saída do controlador são coletas através de elementos sensores ou dos transdutores. A comparação dos sinais de saída com os de referência torna o sistema mais preciso, fazendo com que ele reaja melhor as perturbações externas. A realimentação do controle de malha fechada resulta no aumento da precisão do sistema, melhora da dinâmica do funcionamento, melhor rejeição de perturbações, fácil controle e ajuste da resposta do sistema e melhor redução do erro em regime permanente do sistema. O processo de malha fechada é ilustrado pela figura 2.13 (Gomes, 2000).

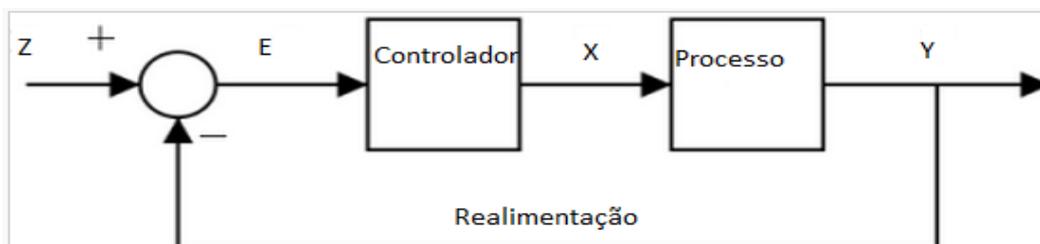


Figura 2.13 – Controle em malha fechada

(Adaptado de:pt.wikipedia.org)

O sinal que é gerado através da proporção do erro entre a resposta desejada que é o sinal de entrada(Z) e a resposta real que é o sinal de saída(Y) resulta em uma sequência de operações em malha fechada que é chamada de realimentação. Esse sinal gerado(E), é definido pela equação 6 de acordo com (C. Dorf, 2013) :

$$E = Z - Y \quad eq(6),$$

Nos sistemas de malha fechada é feita a compensação do efeito das perturbações, medindo a resposta de saída, realimentando essa medida através da malha de realimentação e comparando essa resposta com a entrada. Existindo alguma diferença entre elas, o sistema irá acionar o processo para fazer a correção. Caso não haja nenhuma diferença o processo não é acionado, pois a resposta dele já é a desejada. Considerando essa característica, a malha fechada possui vantagens de apresentar uma exatidão maior que os sistemas em malha aberta. São menos sensíveis a ruídos, perturbações e alterações do ambiente (S. Nise, 2012).

2.10.Sensores

Com o avanço da tecnologia, pôde ser possível dar maior versatilidade e durabilidade as aplicações modernas através dos sensores que auxiliam na automação de máquinas, equipamentos e processos. Sensores são comumente chamados de transdutores que alteram a sua característica física interna devido a fenômenos físicos externos. Eles vieram para substituir as antigas chaves de acionamento mecânico e estão aptos a mudar seu comportamento sob a ação de uma grandeza física. Os sensores fornecem um sinal que indica essas grandezas físicas e convertem elas em um sinal elétrico, ou seja, esses sinais físicos recebidos são convertidos em sinais elétricos pelos sensores. Os sinais elétricos que são gerados pelos sensores podem ser sinais digitais ou sinais analógicos (Argolo *et al*, 210).

2.10.1.Sensores analógicos

Esses tipos de sensores geram um conjunto de dados no seu sinal de saída ao longo do tempo, que variam dentro de uma certa faixa de atuação determinada. Essa variedade de dados são medidas e calculadas por elementos sensíveis com circuitos eletrônicos. Algumas dessas grandezas físicas assumem diversos valores ao longo do tempo: pressão, temperatura, umidade, vazão, força, dentre outros(Wendling 2010, Argolo *et al*, 2010).

Uma grandeza física pode sofrer várias alterações durante uma faixa de operação em sensores analógicos como mostra a figura 2.14 que representa uma variação de temperatura.

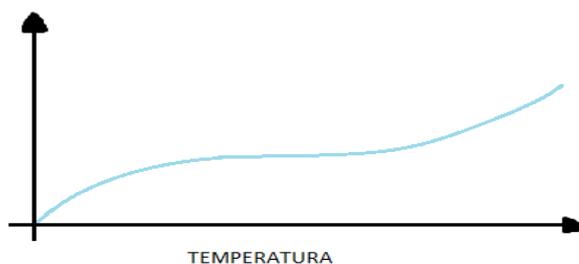


Figura 2.14 – Variação da temperatura

(Fonte:Autor)

2.10.2.Sensores digitais

Os sensores digitais apenas assumem valores 0s ou 1s em suas saídas digitais dentro de um intervalo de tempo. Grandezas físicas não assumem esses valores binários e por causa disso são convertidos por circuitos eletrônicos para serem assim mostrados aos sistemas de controle. Esse tipo de sensor geralmente é utilizado em ocasiões como a necessidade de detectar uma passagem de objetos, ou no caso dos sensores utilizados nos sistemas de captação de águas pluviais que informam se a quantidade de água ultrapassou o nível almejado (Wendling, 2010).

2.11.Engenharia Verde

A engenharia verde está relacionada a questões globais como água potável, sustentabilidade, reduções de emissões, minimização do uso de energia e outros fatores que contribuem para preservação do meio ambiente. Seu objetivo é projetar diversas tecnologias que cooperem com o meio ambiente reduzindo os impactos negativos, minimizem a poluição, o risco da saúde humana e conservem os recursos naturais. A função dos sistemas de controle dentro dessa engenharia continuará a se expandir à medida em que as questões globais enfrentadas requererem níveis maiores de automação e precisão. De acordo com C.Dorf (2013), um dos princípios da engenharia verde e que se aplica ao sistema de captação de águas pluviais são:

- Conservar e melhorar os ecossistemas naturais enquanto se protege a saúde humana e o bem-estar.
- Utilizar o conceito de ciclo de vida em todas as atividades de engenharia.
- Procurar prevenir arduamente o desperdício.

A engenharia verde possui aplicações divididas em cinco áreas principais que atendem os seus princípios (C. Dorf, 2013)

- Monitoramento Ambiental.
- Sistemas de Armazenamento de Energia.
- Monitoramento da Qualidade da Energia.

- Energia Solar.
- Energia Eólica.

Os sistemas de captação de águas pluviais se encaixam em aplicações de monitoramento ambiental pois suas atividades estão totalmente voltadas as boas práticas ambientais de preservação de recursos ambientais como a água de chuva que é usada para fins não potáveis e de minimização de enchentes nas vias pavimentadas dos grandes centros urbanas. Tudo isso está relacionado ao objetivo citado acima de conservar e melhorar os ecossistemas naturais enquanto se protege a saúde humana e o bem-estar. O monitoramento ambiental está voltado para as atividades tanto de controles como de avaliação e serve para conhecer o estado e as tendências qualitativas e quantitativas dos vários recursos naturais envolvidos e as influências que são exercidas pelas atividades humanas e por fatores naturais sobre o meio ambiente. Assim, os seus resultados poderão compor medidas de planejamento, controle, recuperação, preservação e conservação do ambiente do projeto(Ramos, 2014).

Capítulo 3 – Desenvolvimento do Projeto

3.1. Apresentação geral do projeto

O sistema de captação de águas pluviais tem a função de coletar a água da chuva através de calhas nos telhados e direcioná-las por meio de condutores verticais e horizontais para uma cisterna abaixo do solo. A água é bombeada por uma bomba d'água da cisterna para um reservatório em uma residência particular. Esse bombeamento é acionado a partir do nível de água que os sensores detectarem e será controlado por uma placa de circuito.

Nos telhados da residência há um sensor de chuva que possui a simples função de informar se esta chovendo ou não e assim realizar a contagem do tempo dessa chuva. Esse dado é utilizado junto com a pluviosidade média para o cálculo da intensidade pluviométrica. Com a área do telhado da residência e o valor dessa intensidade, é possível então o sistema obter o cálculo da vazão do projeto. Os dados do sistema são visualizados em um display LCD e em uma página WEB que é acessada por um IP dentro de uma rede local. A figura 3.1 esclarece o projeto.

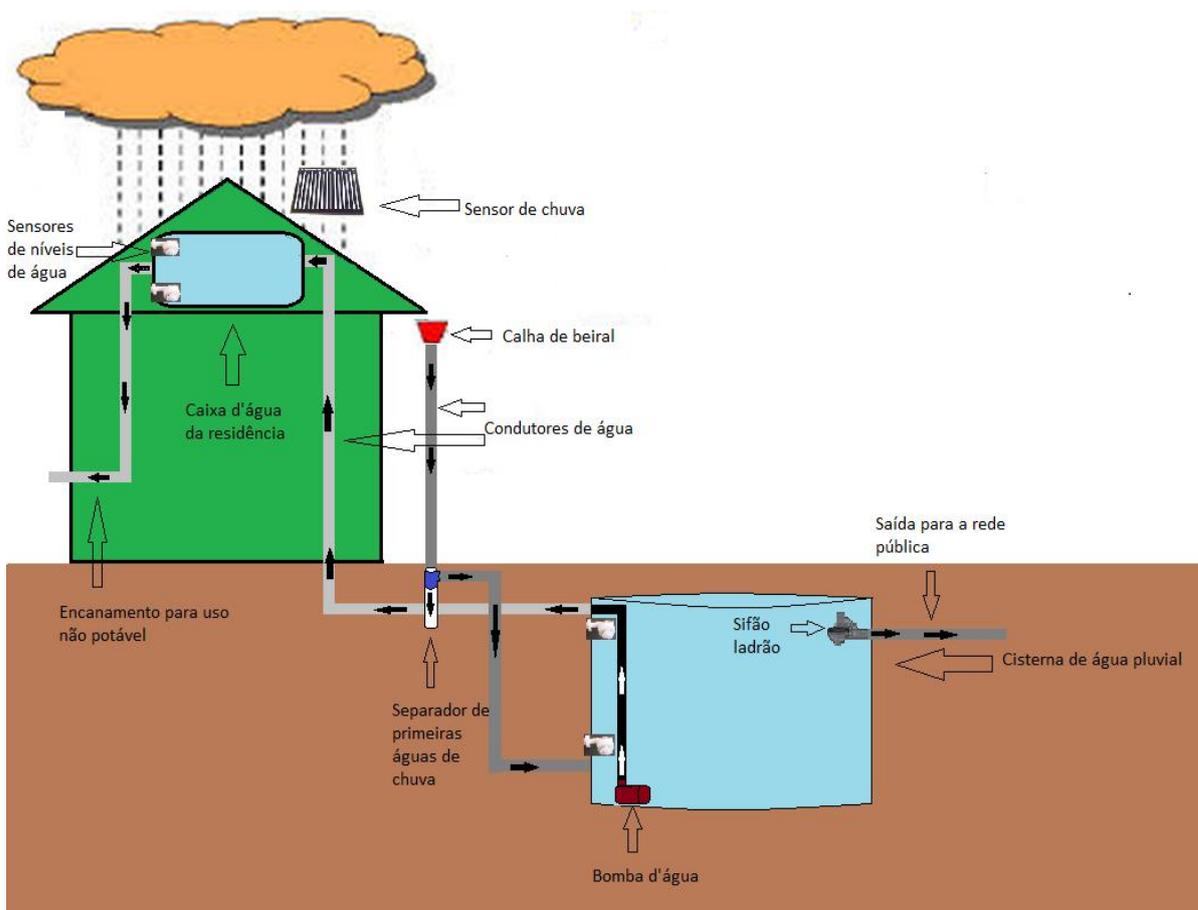


Figura 3.1-Modelo do projeto

(Fonte:Autor)

O desenvolvimento do projeto está organizado em 3 partes sendo que a segunda parte se divide em várias etapas. O diagrama de blocos (Figura 3.2) demonstra a lógica de funcionamento de cada parte do sistema.

1. Programação: a programação do projeto compõe o desenvolvimento do código para gerar a página WEB e o código para controlar e monitorar o hardware do sistema. Essa programação se resume em uma única classe programada em linguagem de programação C e em linguagem HTML na IDE do Arduino.
2. Placa de circuito: essa placa(hardware do sistema) integra o microcontrolador Arduino, a Shield Ethernet, o módulo relé, o módulo de alerta(LEDs e buzzer), os sensores de níveis e de chuva e um display LCD. O Arduino simplesmente têm a função de controlar todos esses dispositivos por meio de suas portas digitais pois para o desenvolvimento desse projeto não foi necessário usar nenhuma das portas analógicas.
3. Protótipo: No desenvolvimento do protótipo foi necessário construir uma maquete de madeira simulando o funcionamento do sistema dentro de uma casa. O circuito desenvolvido no Protoboard foi transportado para uma placa de fenolite.

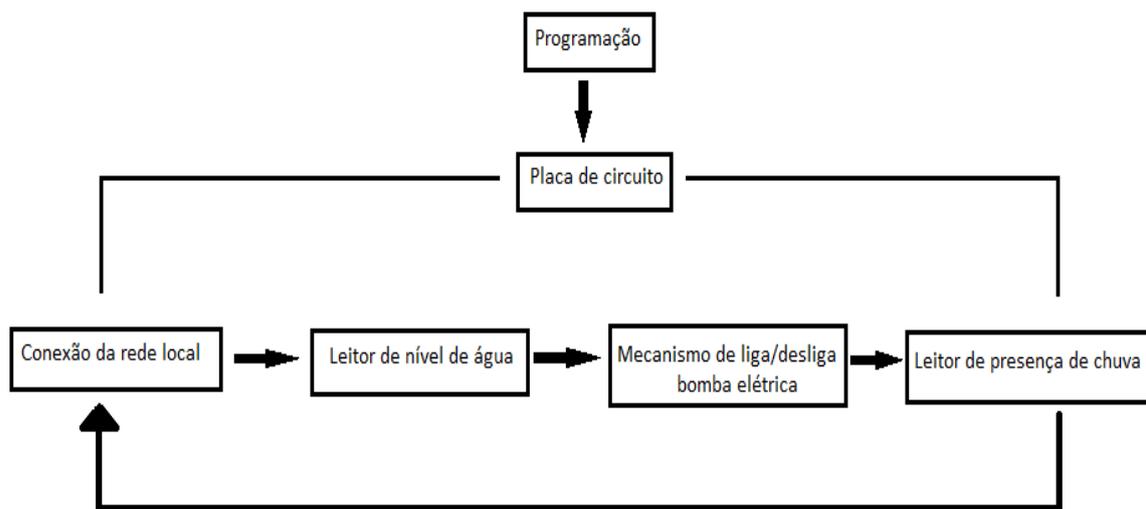


Figura 3.2-Diagrama de blocos

(Fonte:Autor)

A automação do projeto é basicamente responsável pelo Arduino que controla todos os sensores(de chuva e de nível de água) e converte suas grandezas físicas em sinais digitais e assim opera com eles para acionamento de outros dispositivos como o relé, que no caso é energizado pelo Arduino ao receber um sinal do sensor e assim passa a ligar a bomba de água. Através dos sinais dos sensores o Arduino também dá o comando de realizar operações de cálculos matemáticos a exemplo o da vazão do projeto. O display LCD é conectado ao Arduino para mostrar todos os dados e

informações do sistema como o cálculo de vazão e mensagens informando se o sistema está estável, se está reabastecendo, se está chovendo ou se o reservatório de chuva está vazio. Um modelo de alerta é instalado na placa e controlado pelo Arduino. Esse modelo é composto por LEDs e um buzzer que dependendo do valor do sinal de saída dos sensores, são acionados para alertar o usuário sobre o estado do sistema.

Por último a Shield Ethernet é responsável por estabelecer uma conexão do Arduino com um roteador de internet. O roteador se encarrega de manter uma rede local por onde será acessada a página WEB. Essa página é desenvolvida em código HTML através da IDE do Arduino e por ela é possível o usuário monitorar o sistema de captação. Um esquema do hardware do projeto é apresentado abaixo para facilitar o entendimento do funcionamento do sistema.

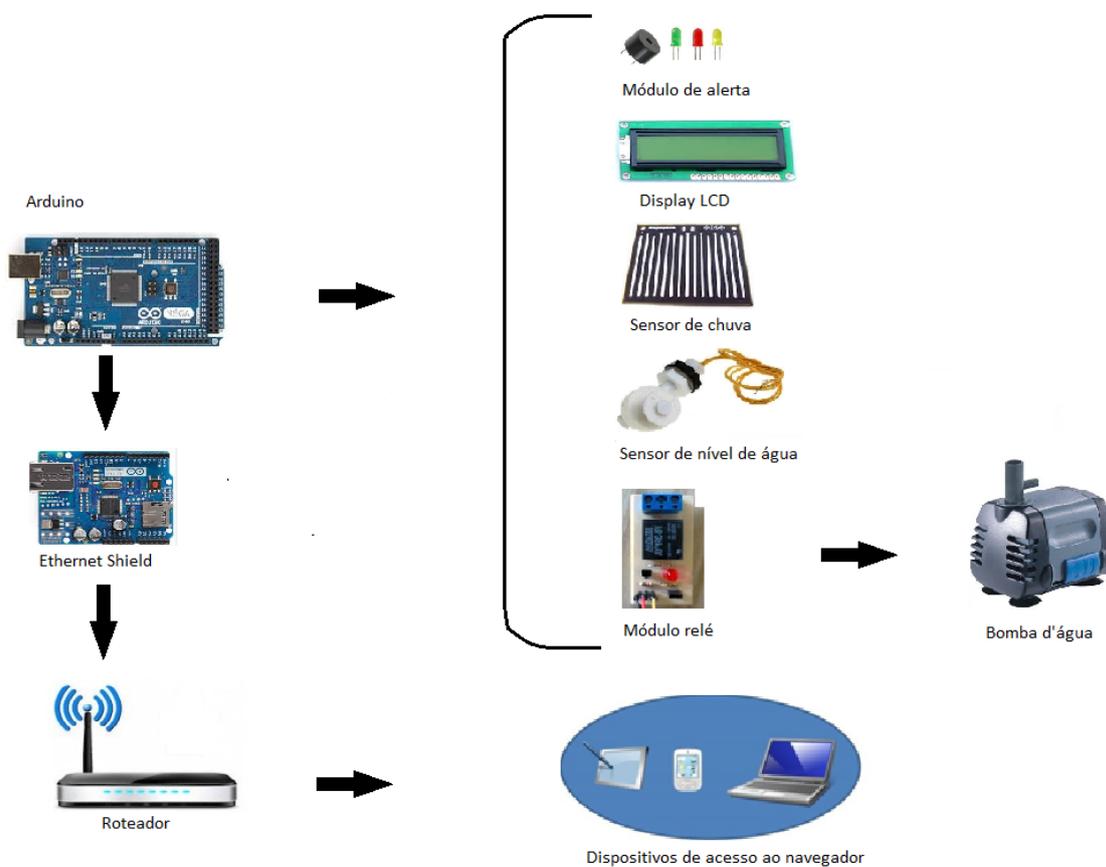


Figura 3.3 - Esquema de desenvolvimento do projeto

(Fonte: Autor)

3.2. Programação na IDE

A programação do projeto de captação de águas pluviais foi desenvolvida na IDE do Arduino com o propósito de atingir dois objetivos:

1. Gerar um código em linguagem C para o controle da placa de circuito.

2. Gerar um código HTML para que uma interface WEB seja disponibilizada em uma rede local.

Na IDE do Arduino foi criada uma única classe para desenvolver esses dois objetivos. Dessa forma, foi possível a linguagem de programação C interagir com a linguagem HTML

No código em linguagem de programação C da placa de circuito, é necessário primeiro incluir as bibliotecas e estabelecer quais portas digitais do Arduino serão utilizadas e armazená-las em variáveis para que seja possível manipular essas portas. A lista de código 1 abaixo mostra um trecho do código em que são incluídas essas bibliotecas e variáveis de tipo inteiro. Logo abaixo no código é também declarada as variáveis “tempChuva”, “vazao”, “intensidadePluvi” e “cont” para o cálculo da vazão do projeto e para armazenar o período de abastecimento.

```
#include <LiquidCrystal.h>

#include <SPI.h>

#include <Ethernet.h>

int LedVermelho = 24;

int LedAmarelo = 25;

int LedVerde = 23;

int SENSOR_CHUVA = 47;

int BUZZER = 51;

int SENSOR_CAIXA1 = 17;

int SENSOR_CAIXA2 = 16;

int SENSOR_RESERVATORIO1 = 15;

int SENSOR_RESERVATORIO2 = 14;

int BOMBA = 22;

double tempChuva = 0;

double vazao = 0;

double intensidadePluvi = 0;

double cont = 0;

LiquidCrystal lcd(40,41,42,43,44,45);
```

Lista de código 1

A biblioteca SPI é usada para comunicação entre o microcontrolador e outros periféricos como cartões SD e sensores. A Ethernet.h serve para que o Arduino se comunique com a Shield Ethernet e a LiquidCrystal.h permite com que o Arduino controle e exiba dados no display LCD. Para o display desse projeto, a função LiquidCrystal estabelece conexão com 6 portas digitais do Arduino que são: 40,41,42,43,44,45.

As variáveis “tempChuva”, ”vazao” e ”intensidadePluvi” são utilizadas para armazenar dados e com eles efetuar o cálculo da vazão do projeto e a variável “cont” apenas faz a contagem de tempo do abastecimento da caixa d’água. As demais variáveis são configuradas na função void setup() como pinos digitais para receber e enviar sinais de dispositivos e ler dados dos sensores de chuva e de níveis de água. Essa configuração é mostrada na lista de código 2 abaixo.

```
pinMode(LedVermelho,OUTPUT);  
  
pinMode(LedAmarelo,OUTPUT);  
  
pinMode(SENSOR_CAIXA1,INPUT);  
  
pinMode(SENSOR_CAIXA2,INPUT);  
  
pinMode(SENSOR_RESERVATORIO1,INPUT);  
  
pinMode(SENSOR_RESERVATORIO2,INPUT);  
  
pinMode(SENSOR_CHUVA,INPUT);  
  
pinMode(BOMBA,OUTPUT);  
  
pinMode(LedVerde,OUTPUT);  
  
pinMode(BUZZER,OUTPUT);  
  
lcd.begin(16, 2);
```

Lista de código 2

A função void setup() é executada apenas uma vez quando o programa é rodado e serve para inicializar variáveis, bibliotecas e funções como a pinMode() e begin(). A função pinMode() configura os pinos para que se comportem como uma entrada(INPUT) ou uma saída(OUTPUT). Os sensores de nível e de chuva são configurados como entrada. Isso significa que eles geram um sinal de entrada para que alguma operação seja realizada e gere um sinal de saída. A bomba d’água, os LEDs e o buzzer são configurados como sinal de saída pelo fato de responderem a um sinal de entrada dos sensores em um dado momento. Por ultimo a função begin() apenas serve para configuração de padrão usados em display LCD, Shield Ethernet e Serial monitor.

Depois da função void setup(), a IDE utiliza outra função chamada void loop() a qual é responsável por estar continuamente executando o programa em loops. Dentro da função void loop(), a cada loop realizado o programa executa a função verificaSensores(). Dentro dessa função, o estado dos sensores de níveis são verificados a cada loop e para cada valor do sinal obtido, o programa realiza uma série de operações.

A função verifica duas possibilidades: caso os dois sensores da caixa d'água estiverem em nível baixo(LOW), o sistema entende que a caixa está vazia. Nessa condição, o programa aciona os dispositivos do módulo de alerta e executa outra função chamada abastecerCaixa() onde a bomba é ligada para abastecer a caixa. Caso essa condição não seja satisfeita, o programa pula para a outra condição que verifica se os reservatórios estão cheios indicando que o sistema está estável e mostra o valor da vazão do projeto.

Além dos sensores de níveis, essa função verifica o estado da resistência do sensor de chuva. Se essa resistência estiver em nível baixo, significa que a água está em contato com essa resistência indicando que está chovendo. Para essa condição o programa usa a variável "tempChuva" para a contagem do tempo dessa chuva em segundos dentro de um laço de repetição. No momento em que o sensor retornar um valor de resistência em nível alto(HIGH), o programa interrompe essa contagem, converte esse valor em minutos e realiza o cálculo da vazão do projeto. Essa função é mostrada na lista de código 3 logo abaixo.

```
void verificaSensores(){
    if( digitalRead(SENSOR_CAIXA2)==LOW && digitalRead(SENSOR_CAIXA1)==LOW ){
        digitalWrite(LedAmarelo,HIGH);
        digitalWrite(LedVermelho,LOW);
        digitalWrite(LedVerde,LOW);
        digitalWrite(BUZZER,HIGH);
        lcd.clear();
        lcd.setCursor(0,0);
        lcd.print("Caixa vazia.");
        client.print("<tr><td><font color = 'red'>Caixa vazia</font></td></tr>");
        abastecerCaixa();
    }else{
        digitalWrite(LedVerde,HIGH);
        digitalWrite(LedAmarelo,LOW);
        digitalWrite(LedVermelho,LOW);
    }
```

```

digitalWrite(BOMBA,LOW);

digitalWrite(BUZZER,LOW);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Sistema estavel");

client.print("<tr><td><font color = 'green'>Sistema estavel</font></td></tr>");

if(vazao!=0){

  lcd.setCursor(0,1);

  lcd.print("Vazao:");

  lcd.print(vazao);

  lcd.print("L/min");

  client.print("<br><br>");

  client.print("<center><table border><tr><th>Pluviosidade Media Anual do Centro-Oeste(m)</th>");

  client.print("<th>Duracao do periodo de chuva(minutos)</th>");

  client.print("<th>Intensidade Pluviometrica(m/min)</th>");

  client.print("<th>Area de contribuicao do telhado(m.m)</th>");

  client.print("<th>Vazao do projeto(L/min)</th></tr>");

  client.print("<tr><td>");client.print("3.00");client.print("</td>");

  client.print("<td>");client.print(tempChuva);client.print("</td>");

  client.print("<td>");client.print(intensidadePluvi);client.print("</td>");

  client.print("<td>");client.print("0,2295");client.print("</td>");

  client.print("<td>");client.print(vazao);client.print("</td></tr></table></center>");

}

}

if(digitalRead(SENSOR_CHUVA)==LOW){

  tempChuva =0;

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Chuva detectada");

  client.print("<tr><td>Chuva detectada</td></tr>");

```

```

digitalWrite(LedAmarelo,HIGH);

digitalWrite(LedVerde,LOW);

for(;;){

  lcd.setCursor(0,1);

  delay(1000);

  tempChuva = tempChuva+1;

  lcd.print(tempChuva);

  if(digitalRead(SENSOR_CHUVA)==HIGH){

    tempChuva = tempChuva/60;

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Termino de chuva");

    lcd.setCursor(0,1);

    lcd.print("Duracao:");

    lcd.print(tempChuva);

    lcd.print("min");

    client.print("<tr><td>Termino de chuva.</td></tr>");

    client.print("<tr><td>Duracao:");

    client.print(tempChuva);

    client.print(" minutos</td></tr>");

    intensidadePluvi = 3/tempChuva;

    vazao = (intensidadePluvi*0,2295)*1000;

    delay(4000);

    break;

  }

}

}

}

```

Lista de código 3

A função `digitalRead()` lê o valor de um pino digital específico e verifica se é LOW ou HIGH. A função `digitalWrite()` manda um sinal alto(HIGH) correspondente a 5 Volts ou um sinal baixo(LOW) igual a 0 Volts para um pino digital. Esse comando é usado para ativar e desativar dispositivos conectados ao Arduino. O comando `break` interrompe imediatamente o laço de repetição fazendo com que o compilador execute os próximos comandos. Por último a função `delay()` espera 1 segundo para executar o próximo comando. Ela é necessária para que o valor armazenado na variável “tempChuva” seja dado em segundos para depois ser convertido em minutos. No cálculo da vazão, o valor 3 corresponde ao valor da pluviosidade média anual do Centro-Oeste convertida para metros e o valor 0,2295 foi obtido da área de contribuição do protótipo em m² (seção 2.3.1.1). O valor da vazão é multiplicado por 1000 para que a unidade seja convertida de m³/min para L/min.

A função `abastecerCaixa()` é chamada logo que o sensor de nível é ultrapassado pelo fluido e ativa a bomba d’água usando a função `digitalWrite()` dentro de um laço de repetição. Enquanto a condição `if(...)` não for satisfeita, o programa irá executar o laço de repetição mantendo a bomba ligada e incrementando a variável “cont” para obter o período de abastecimento. No momento em que os sensores de níveis do reservatório retornarem HIGH, o programa desativa a bomba e os dispositivos de alerta, converte o valor de “cont” para minutos e informa o usuário que a caixa d’água está cheia e o tempo de abastecimento . A lista 4 demonstra o código dessa operação.

```

lcd.setCursor(0,0);
lcd.print("Reabastecendo...");
client.println("<tr><td><font color = 'yellow'>Reabastecendo...</font></td></tr>");
cont = 0;
digitalWrite(BOMBA,HIGH);
for(;;){
  delay(1000);
  cont = cont +1;
  if( digitalRead(SENSOR_CAIXA1)==HIGH && digitalRead(SENSOR_CAIXA2)==HIGH){
    digitalWrite(BOMBA,LOW);
    digitalWrite(BUZZER,LOW);
    digitalWrite(LedAmarelo,LOW);
    digitalWrite(LedVermelho,LOW);
    digitalWrite(LedVerde,HIGH);
    cont = cont/60;
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Caixa cheia.");
    lcd.setCursor(0,1);
    lcd.print("tempo:");
    lcd.print(cont);
    lcd.print("minutos");
    client.println("<tr><td><font color = 'green'>Caixa cheia</font></td></tr>");
    client.println("<tr><td>Tempo de abastecimento:");
    client.println(cont);
    client.println(" minutos</td></tr>");
    delay(5000);
  }
}

```

```

        break;
    }
    ...
}

```

Lista de código 4

Essa função leva em conta outra condição: caso durante o abastecimento da caixa d'água o sensor do reservatório de chuva retornar LOW, o programa desliga a bomba mandando 0 Volts para o pino e informando o usuário que o reservatório está vazio. Quando essa condição for satisfeita o programa executa um laço de repetição e ali permanece em espera até que o sensor de chuva detecte um fluído em sua resistência. Esse código é mostrado na lista 5 logo abaixo.

```

if(digitalRead(SENSOR_RESERVATORIO1)==LOW){

    digitalWrite(BOMBA,LOW);

    digitalWrite(BUZZER,LOW);

    digitalWrite(LedAmarelo,LOW);

    digitalWrite(LedVermelho,HIGH);

    digitalWrite(LedVerde,LOW);

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Reservatorio");

    lcd.setCursor(0,1);

    lcd.print("vazio.");

    client.println("<tr><td><font color = 'red'>Reservatorio vazio</font></td></tr>");

    for(;;){

        if(digitalRead(SENSOR_CHUVA)==LOW){

            ...

        }

        ...

    }
}

```

Lista de código 5

Enquanto o programa permanecer no laço de repetição, a condição if(...) será testada até que o sensor de chuva detecte a presença de água retornando LOW. Quando essa condição for satisfeita, o programa informa ao usuário sobre a chuva e novamente

executa outro laço de repetição armazenando o tempo dessa chuva até que o sensor de nível superior do reservatório de chuva retorne HIGH. Assim que esse sensor estiver em nível alto, o programa aciona a bomba permanecendo em loop e reabastecendo a caixa d'água. Confira a lista de código 6.

```

if(digitalRead(SENSOR_CHUVA)==LOW){

    tempChuva=0;

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Chuva detectada.");

    client.print("<tr><td>Chuva detectada.</td></tr>");

    for(;;){

        delay(1000);

        tempChuva = tempChuva+1;

        lcd.setCursor(0,1);

        lcd.print(tempChuva);

        digitalWrite(LedAmarelo,HIGH);

        if( digitalRead(SENSOR_RESERVATORIO2)==HIGH){

            lcd.clear();

            lcd.setCursor(0,0);

            lcd.print("Reabastecendo...");

            client.println("<tr><td><font color = 'yellow'>Reabastecendo...</font></td></tr>");

            for(;;){

                digitalWrite(BOMBA,HIGH);

                ...

            }

            break;

        }

    }

    break;

}

```

Lista de código 6

A lista de código 7 mostra o caso em que quando os sensores de níveis da caixa d'água retornarem HIGH, o período da chuva é convertido para minutos e a bomba e os outros dispositivos são desligados. O programa informa o usuário que a caixa d'água está cheia e realiza o cálculo da vazão.

```

if( digitalRead(SENSOR_CAIXA1)==HIGH && digitalRead(SENSOR_CAIXA2)==HIGH ){

    tempChuva = tempChuva/60;

    digitalWrite(BOMBA,LOW);

    digitalWrite(BUZZER,LOW);

    digitalWrite(LedAmarelo,LOW);

    digitalWrite(LedVerde,HIGH);

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Caixa cheia.");

    lcd.setCursor(0,1);

    lcd.print("chuva:");

    lcd.print(tempChuva);

    lcd.print("Minutos");

    client.println("<tr><td><font color = 'green'>Caixa cheia</font></td></tr>");

    client.println("<tr><td>Tempo de chuva:"");

    client.println(tempChuva);

    client.println(" minutos</td></tr>");

    intensidadePluvi = 3/tempChuva;

    vazao = (intensidadePluvi*0,2295)*1000;

    delay(4000);

    break;

}

```

Lista de código 7

Na programação feita na IDE foi preciso desenvolver um código HTML interagindo com a linguagem C para o desenvolvimento de uma página WEB. Para que essa página seja disponibilizada na rede local, é preciso primeiro configurar a Shield Ethernet no código C conforme a lista 8 abaixo.

```
byte mac[] = {
```

```

0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

IPAddress ip(192,168,1,177); // Configura um IP

EthernetServer server(80); // Configura uma porta de acesso

EthernetClient client;

```

Lista de código 8

O byte mac define e armazena o endereço MAC do Arduino que é um endereço físico associado a uma interface de comunicação (página WEB) que conecta um dispositivo a uma rede local e o IPAddress(endereço IP) é a identificação do computador(dispositivo) nessa rede. O EthernetServer configura o servidor na porta 80 que é a porta padrão. O número da porta do servidor é simplesmente uma via para os dados, ela decide para onde os dados devem ir. O EthernetClient cria um cliente(navegador) que pode se conectar ao Arduino. Na função setup a comunicação do servidor e da Ethernet Shield são inicializadas pela função begin, especificando o endereço MAC e IP. A página WEB é criada na função loop e atualizada em tempo real. As atualizações são feitas durante a execução do programa ao testar as condições de estado do sistema nas funções verificaSensores e abastecerCaixa. Confira a lista 9.

```

void loop(){

  client = server.available();

  if (client) {

    boolean currentLineIsBlank = true;

    while (client.connected()) {

      if (client.available()) {

        char c = client.read();

        if (c == '\n' && currentLineIsBlank) {

          client.println("HTTP/1.1 200 OK");

          client.println("Content-Type: text/html");

          client.println("Connection: close");

          client.println("Refresh: 5");

          client.println();

          client.println("<center><h1><b>Sistema de captacao de aguas pluviais</b></h1><br>");

          client.println("<table border =><tr><th>Monitoramento</th></tr>");

          verificaSensores(); // Executa a função verificaSensores()

```

```

client.println("</table></center>");

client.println("<!DOCTYPE HTML>");

client.println("<html>");

break;

}

if (c == '\n')currentLineIsBlank = true;

else if (c != '\r')currentLineIsBlank = false;

}

}

delay(1);

client.stop();

}

delay(100);

}

```

Lista de código 9

Quando o cliente recebe o comando “server.available()” ele verifica se há dados disponíveis para leitura a partir do servidor. Logo depois, a condição if(client) verifica se o cliente (navegador) está conectado. Se isso for verdadeiro essa condição é executada.

Na condição if(client), é criada uma variável “currentLineIsBlank” que irá determinar se o fim dos dados dentro da página WEB foi atingido. Enquanto houver cliente conectado – while(client.connected()), o programa testa a condição if(client.available()) que retorna um número de bytes que o servidor escreveu no cliente. Caso esse valor seja maior que 0 o programa executa essa condição.

A variável “c” armazena a cada loop o próximo byte a ser recebido do servidor através do comando “client.read()”. Se esse byte for um caractere de nova linha(“\n”) e a variável “currentLineIsBlank” permanecer verdadeira, significa que o fim da solicitação HTTP foi atingida e portanto o programa pode executar o código HTML e enviar dados para a página chamando a função verificaSensores() e depois encerrar o loop com o comando break. Depois disso, se um caractere nova linha(“\n”) for lido, “currentLineIsBlank” permanece TRUE e continua a atualizar a página WEB. Porém se o programa ler um “\r”, significa que ainda há caracteres a serem lidos a partir do servidor e “currentLineIsBlank” recebe FALSE não permitindo o programa enviar dados para a página WEB.

O programa então espera um intervalo de tempo, permitindo ao navegador que receba os dados e interrompa o cliente com o comando stop() fazendo com que o cliente se desconecte do servidor. A figura 3.4 mostra a Interface WEB informando a área de contribuição do telhado, a pluviosidade média do Centro-Oeste e os dados (tempo de chuva e intensidade pluviométrica) que foram obtidos para o cálculo da vazão do projeto.

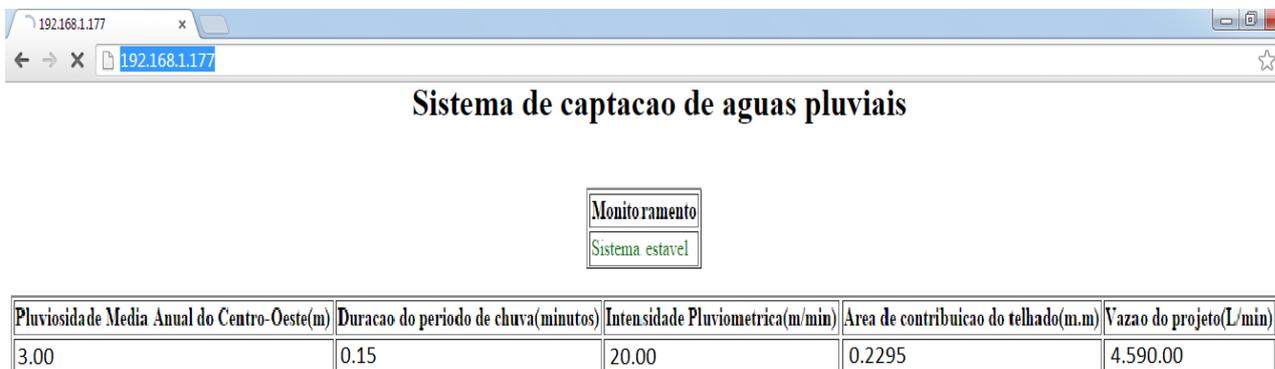


Figura 3.4-Interface WEB

(Fonte:Autor)

3.3.Placa de Circuito

O principal componente da placa de circuito é o Arduino que controla todos os dispositivos por meio de suas portas digitais e a sua alimentação vêm do computador. O Arduino faz a comunicação com o navegador por intermédio da Shield Ethernet que é conectada a um roteador. Ele também recebe o sinal dos sensores de níveis que estão acoplados nos reservatórios e do sensor de chuva para depois convertê-los em sinais digitais. O módulo relé é energizado pelo Arduino e assim a bomba d'água é ligada e desligada através dessa tensão que a bobina do relé recebe.

A construção da placa foi feita por etapas e aos poucos foi integrado todos os componentes necessários para o funcionamento do sistema: sensores de nível, sensor de chuva, Arduino, Shield Ethernet, display LCD, o módulo relé que aciona a bomba d'água e o módulo de alerta. A figura 3.5 demonstra a primeira versão do protótipo onde o circuito foi elaborado em uma placa de protoboard para realizar uma bateria de testes. Esse modelo foi melhorado com a construção de uma maquete que será mostrada na seção 3.4.



Figura 3.5-Esquema físico de desenvolvimento do projeto

(Fonte:Autor)

O circuito desenvolvido no protoboard foi usado apenas para testes até que o sistema pudesse ter a garantia de funcionamento. Logo depois esse circuito foi transportado para uma placa de fenolite (Figura 3.6) onde todos os componentes foram acoplados pelo método da soldagem exceto o Arduino Mega e a Shield Ethernet. Como não foi possível acoplar o microcontrolador e a Shield na placa, foram instalados conectores para fazer a ligação de cada dispositivo do circuito no Arduino.

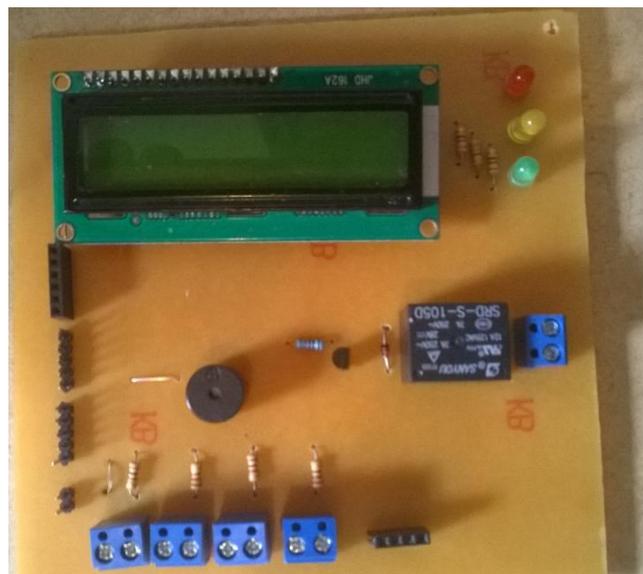


Figura 3.6 – Placa de circuito

(Fonte:Autor)

3.3.1.Sensor de nível de água

O sensor de nível de água possui uma bóia que se movimenta livremente ao redor de um eixo. Essa bóia funciona eletricamente como uma chave que fecha os dois contatos que saem do sensor quando o nível de água leva a bóia para cima e abre quando a água não toca a bóia fazendo com que ela permaneça embaixo. Um desses contatos é ligado em uma porta digital passando pela porta Gnd (Terra) usando um

resistor e o outro é ligado na porta 5 Volts do Arduino. A figura 3.7 demonstra o circuito do sensor.

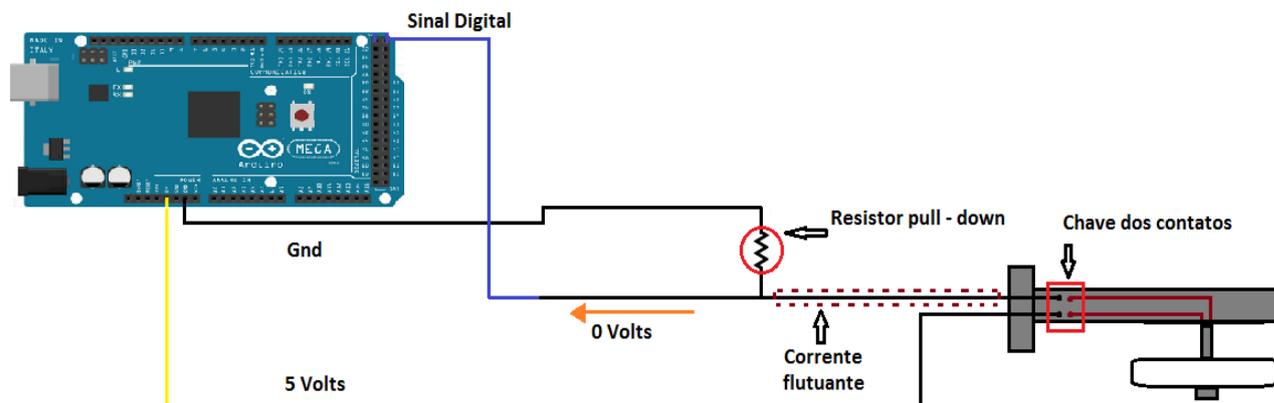


Figura 3.7-Circuito do sensor de nível de água (chave aberta)

(Fonte:Autor)

No circuito acima a chave permanece aberta, fazendo com que uma corrente de valor desconhecido permaneça fluando no circuito. O resistor de pull-down têm a função de gastar toda essa corrente garantindo que o sinal digital receba 0 Volts. Quando essa chave se fecha, é permitida a passagem da corrente de 5 Volts para o sinal digital e o resistor impede que esse valor de corrente chegue até o Gnd causando curto no circuito (Figura 3.8).

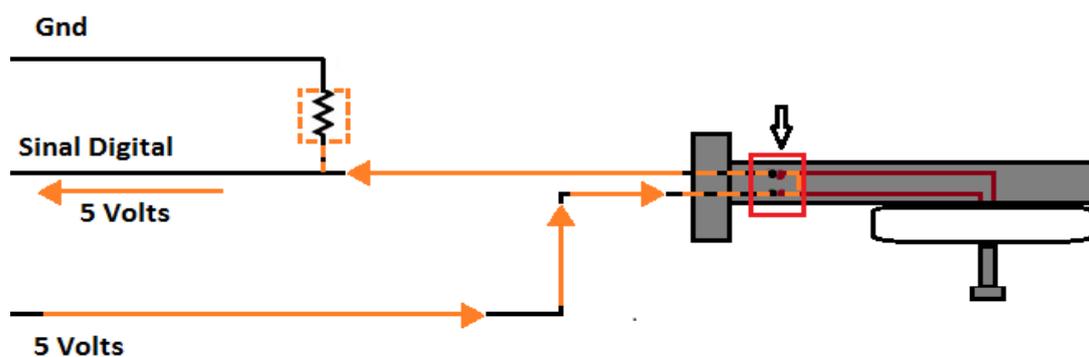


Figura 3.8 – Circuito do sensor de nível de água (chave fechada)

(Fonte:Autor)

Quando a bóia é atingida pela água ela sobe fechando os contatos e manda um sinal de 5 Volts (grandeza física) para o Arduino. Quando o Arduino converte essa grandeza para um sinal digital ele interpreta o valor 1 significando que o sensor está em nível alto (sinal digital HIGH). Caso o nível de água esteja baixo e não alcance o sensor, a bóia irá permanecer em nível baixo (sinal digital LOW) e os contatos em aberto fazendo com que o sensor envie 0 Volts para o Arduino que converte esse valor em sinal digital 0. Os sinais digitais dos sensores de níveis de água interpretados pelo

Arduino são sinais de entrada (INPUT) e servem de comando para que outras operações sejam feitas e gerem um sinal de saída como o sinal do módulo relé que aciona e desliga a bomba d'água dependendo do nível de água dos reservatórios. O momento em que os sensores de níveis retornam 5 e 0 Volts nos reservatórios é mostrado na figura 3.9.

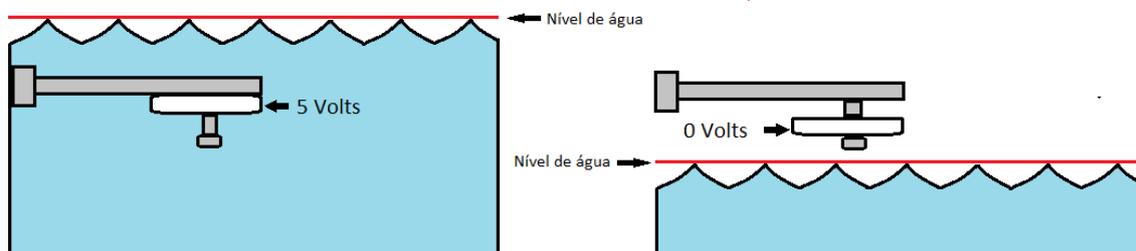


Figura 3.9-Tensão dos sensores de níveis de água

(Fonte:Autor)

Os sensores de nível são acoplados com o lado da bóia para dentro do reservatório com um ângulo horizontal em relação ao nível de água e os dois contatos que saem do sensor estão do lado de fora dos reservatórios e são ligados diretamente na placa de circuito. Tanto a caixa d'água da residência como o reservatório de água de chuva abaixo do solo possuem dois sensores, um superior que recebe o nível de água indicando que o nível está completo e outro inferior que detecta a passagem de água indicando que o reservatório precisa ser abastecido. Na figura 3.10, os sensores de nível foram instalados na caixa d'água com um ângulo horizontal, um na parte superior e outro na parte inferior da caixa que detectam a passagem de água nesses dois níveis.

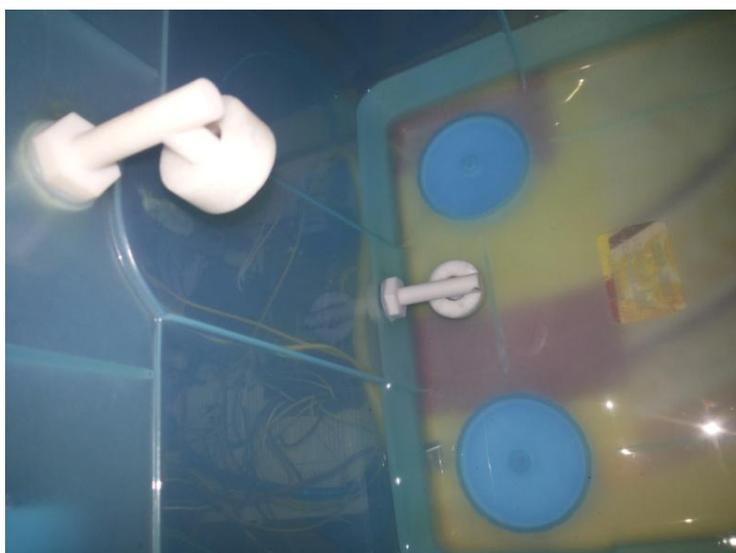


Figura 3.10-Sensores de níveis de água

(Fonte:Autor)

3.3.2.Sensor de chuva

O sensor de chuva é composto por uma placa revestida em todos os lados de duas malhas condutoras. No momento em que um fluido é detectado sobre a placa, a corrente começa a fluir de uma malha para outra e a resistência da placa cai para um nível mais baixo. Se a placa permanecer com uma resistência alta, significa que nenhum fluido foi detectado sobre as malhas e dessa forma não há movimentação de corrente elétrica. A figura 3.11 exemplifica o momento em que a resistência da placa cai para um nível mais baixo e a corrente começa a fluir de uma malha para outra.

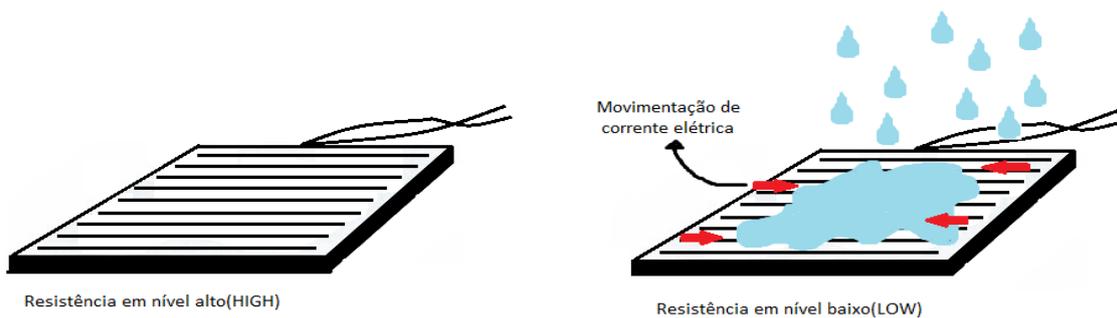


Figura 3.11-Resistência da placa do sensor de chuva

(Fonte:Autor)

Um comparador LM393 é usado para ler os dados do sensor e enviá-los para o Arduino. No momento em que uma movimentação de corrente é detectada sobre a placa do sensor, esse comparador envia um sinal em nível baixo na saída digital D0, caso contrário envia um sinal em nível alto. O Arduino converte esses valores como 0s e 1s respectivamente. O sensor de chuva possui duas saídas que se conectam ao LM393. Esse comparador possui 4 saídas incluindo a VCC(5Volts), Gnd(Terra), uma porta digital(D0) e outra analógica(A0) que não está sendo usada nesse projeto. A porta digital do Arduino que recebe os sinais da D0 é configurada no código da IDE como um sinal de entrada (INPUT). A porta digital do LM393 retorna 5 ou 0 Volts. Já a porta analógica possui uma tensão que varia de 0 a 5Volts e pode ser usada para ter valores mais precisos e determinar por exemplo quantas gotas caíram na placa. Para esse projeto foi necessário apenas usar a porta digital que determina quando o período chuvoso começa e termina. A figura 3.12 demonstra o circuito do sensor de chuva.

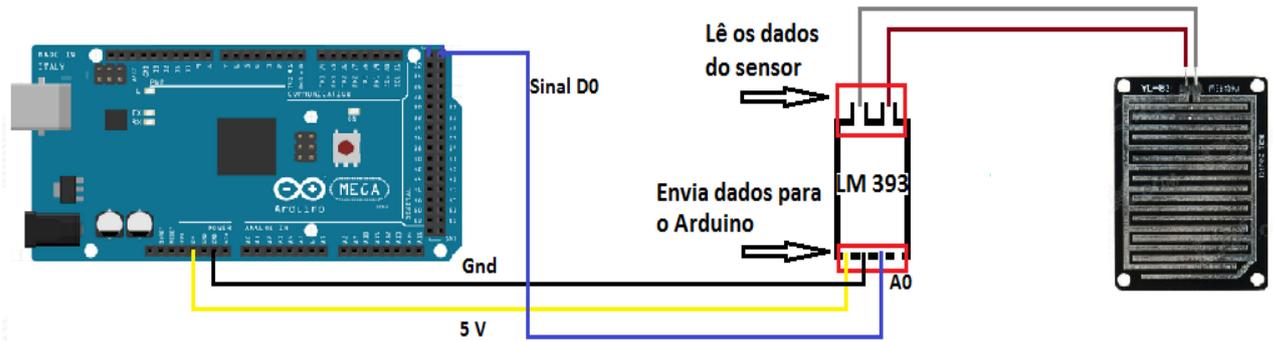


Figura 3.12-Ligação do sensor de chuva no Arduino Mega

(Fonte:Autor)

O sensor de chuva é instalado paralelo ao telhado da residência com uma declividade para que o fluído possa escorrer da placa evitando que fique possas de água em sua resistência. A água da chuva atinge a placa do sensor e escoar para a calha que direciona a água por um condutor até chegar a cisterna. O sensor de chuva marca o começo e o fim do período chuvoso e junto com a área de contribuição do telhado (seção 2.3.1.1) servem de variáveis para o cálculo da intensidade pluviométrica e da vazão do projeto (seções 2.3.1.2 e 2.4.2). A figura 3.13 demonstra o modo de instalação do sensor de chuva.

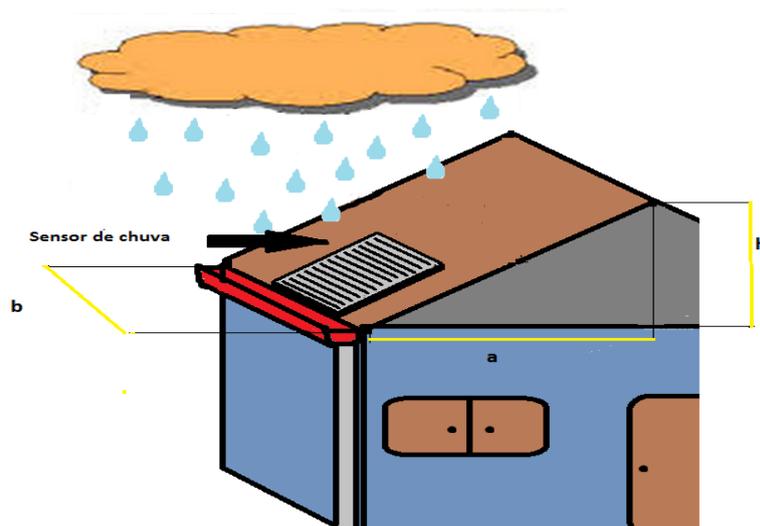


Figura 3.13-Modo de instalação do sensor de chuva no telhado

(Fonte:Autor)

3.3.3.Acionamento da bomba d'água

A bomba d'água possui uma tensão de 220 Volts e é ligada pelo módulo relé que aciona um contato quando sua bobina é energizada. Um sinal de 5 Volts é emitido nas espiras da bobina do relé fazendo com que o contato comum se desloque do normalmente fechado (NF) para o normalmente aberto (NA) e assim aciona a bomba

d'água que possui uma tensão de 220 Volts. No momento em que a bomba deve ser desligada, a passagem de tensão na bobina é interrompida e assim o contato comum se desloca do normalmente aberto para o normalmente fechado.

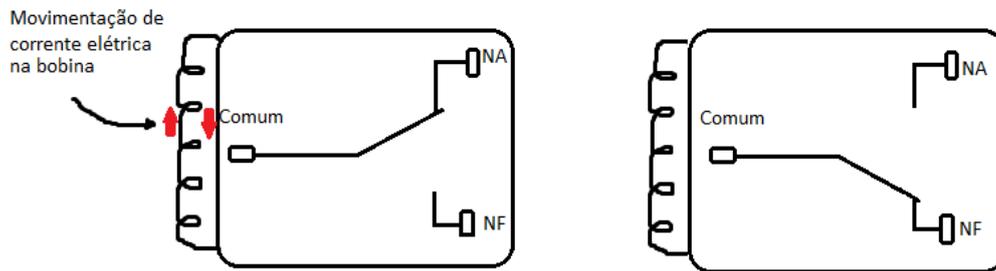


Figura 3.14-Contatos NA e NF do relé

(Fonte:Autor)

O contato comum é ligado na rede elétrica e o normalmente aberto é conectado na tomada da bomba. Assim, quando a bobina for energizada e o contato comum passar do NF para o NA, esse contato já estará conectado com a bomba d'água permitindo a passagem de corrente elétrica. A tomada que se conecta a bomba possui duas saídas: uma é o NA e a outra vai direto para a tomada da rede elétrica. O Arduino envia uma tensão através da porta digital que passa por uma resistência "R" chegando ao transistor. Os sinais de 5 Volts e Gnd são ligados nas extremidades do relé. A figura 3.15 demonstra a ligação do módulo relé no Arduino e na bomba.

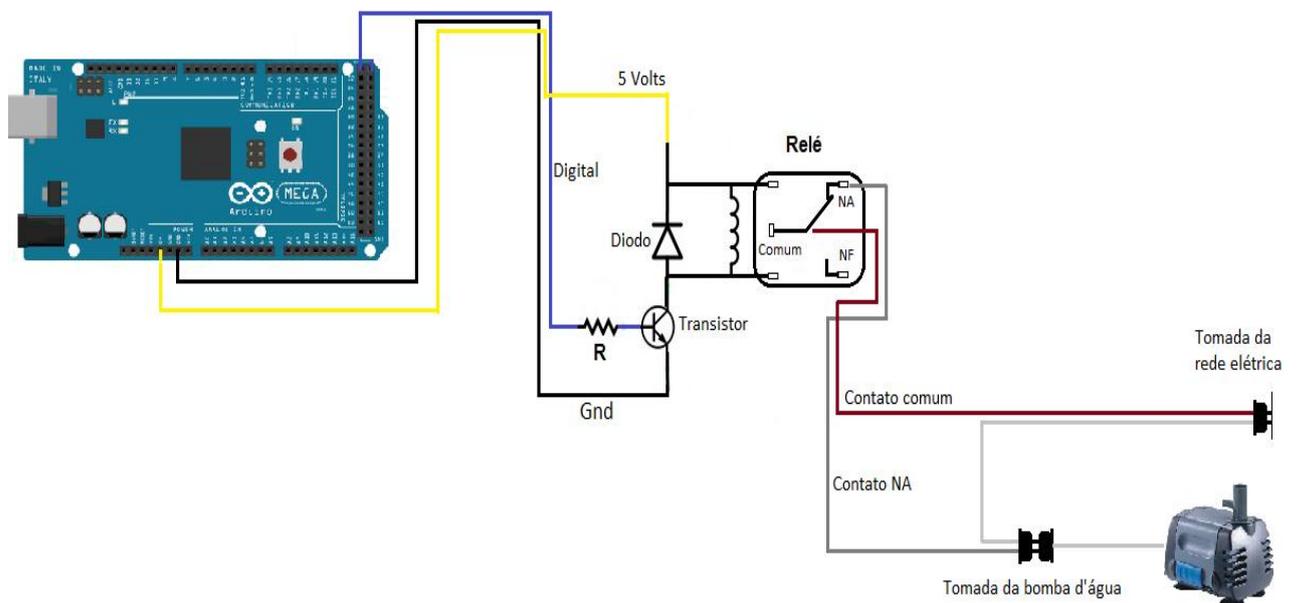


Figura 3.15-Ligação do módulo relé no Arduino e na bomba d'água

(Fonte:Autor)

O módulo relé liga e desliga a bomba d'água nos seguintes casos:

1. Quando o programa verifica que os dois sensores de níveis da caixa d'água da residência estão em nível baixo, o Arduino envia uma tensão de 5 Volts para o transistor. No circuito, o transistor possui a função de enviar essa tensão para a bobina do relé. Desse modo o relé consegue controlar a bomba d'água que possui uma tensão de 220 Volts através do transistor que energiza a bobina apenas com uma tensão de 5 Volts (Figura 3.16 – a). No momento em que o programa consta que as bóias dos sensores estão em nível alto novamente o Arduino interrompe a passagem de corrente pela porta digital. Quando a passagem de tensão é interrompida, o campo magnético dessa tensão que foi gerada ainda permanece na bobina. Como o diodo está polarizado no sentido inverso do campo magnético que ainda permanece na bobina, ele irá conduzir uma corrente nesse sentido absorvendo esse campo para evitar que o dispositivo seja acionado em um momento oportuno (Figura 3.16 – b).

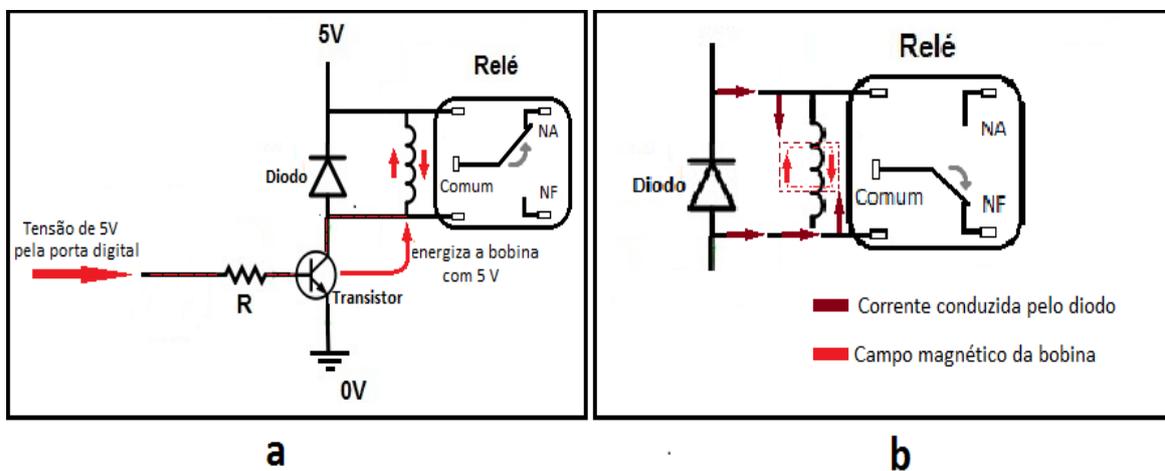


Figura 3.16-Ativação / desativação da bomba d'água

(Fonte:Autor)

2. Durante o abastecimento da caixa d'água, é possível que o reservatório de água de chuva fique vazio deixando de abastecer a caixa. Nesse caso os sensores de níveis retornam 0 Volts para o Arduino que converte em sinal digital 0. O programa verifica o momento em que o sensor inferior retorna um sinal digital em nível baixo (Lista 5). Caso essa condição seja atendida o Arduino deixa de alimentar a porta digital do relé fazendo que o transistor não alimente mais a bobina. O diodo então absorve o campo magnético das espiras e a bomba é desligada. Quando o sensor de chuva detectar alguma presença de água, o programa fica em loop armazenando o tempo da chuva para posteriormente calcular a vazão do projeto. O sistema espera o reservatório ser abastecido durante o período chuvoso. A cada loop o programa irá verificar se o sensor superior do reservatório de chuva está retornando um sinal digital em nível alto (Lista 6). Se isso for verdadeiro o Arduino recebe uma tensão de 5 Volts dos sensores. Nesse momento a porta digital do relé é alimentada com 5 Volts pelo Arduino e essa tensão chega até o transistor do circuito que alimenta a bobina fazendo com que a bomba seja ativada e termine de abastecer a caixa d'água da

residência. A figura 3.17 demonstra o funcionamento físico, especificando o momento em que a bomba é desligada e ligada novamente.

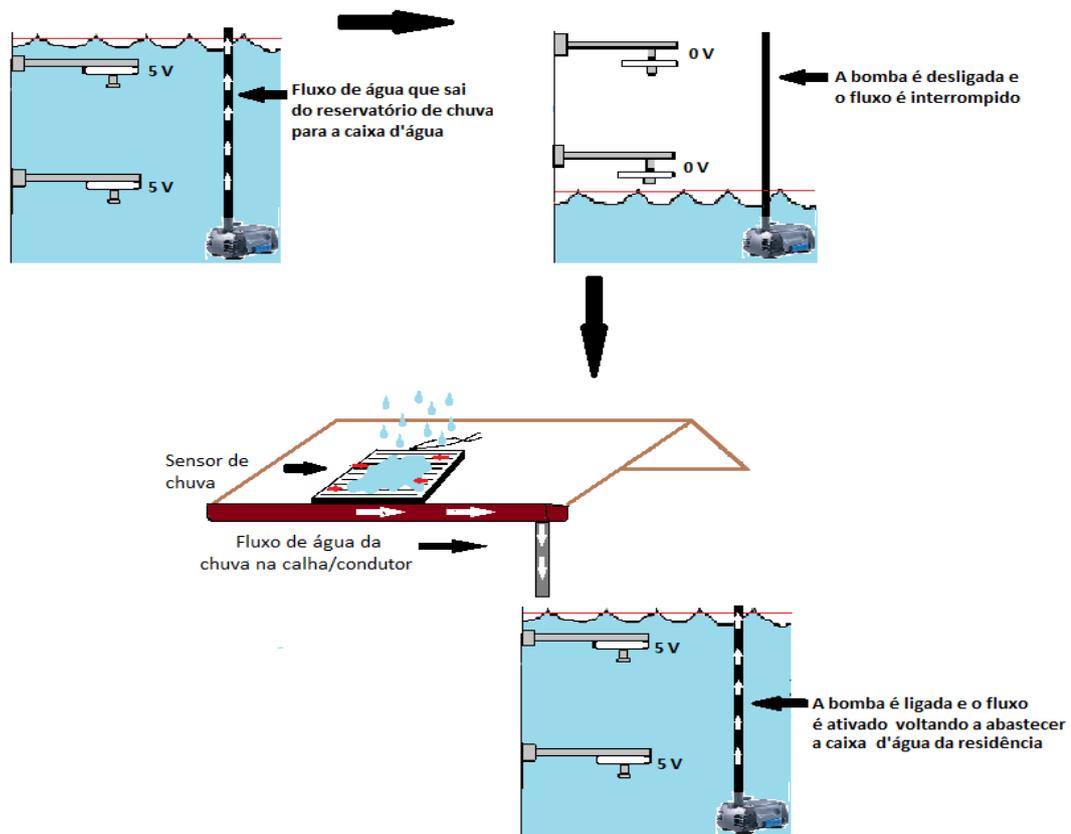


Figura 3.17-Desativação da bomba d'água durante o abastecimento

(Fonte:Autor)

3.3.4.Módulo de alerta

O módulo de alerta é composto por um Buzzer e 3 LEDs: amarelo, verde e vermelho. Sem esses dispositivos o sistema funcionaria normalmente, porém eles servem para alertar o usuário sobre o momento em que o período de chuva se inicia e termina e o momento em que o sistema liga a bomba para abastecer a caixa d'água e desliga quando a caixa está cheia. Para os LEDs, cada cor é ligada em um caso específico:

1. O LED amarelo é ligado quando os sensores de níveis da caixa d'água retornam 0 Volts indicando o momento em que a caixa d'água precisa ser abastecida e desligado quando retornam 5 Volts para o Arduino informando o usuário que o período de abastecimento se concluiu. Ele também é ligado quando o sensor de chuva detecta um período chuvoso e permanece até que esse período acabe.
2. O LED verde permanece ligado em todo tempo que o sistema estiver estável, sem alterações e desligado caso um período chuvoso seja detectado pelo sensor ou a caixa d'água esteja vazia.
3. O LED vermelho é ligado no momento em que os sensores do reservatório de chuva retornam 0 Volts durante o abastecimento e permanece até que um

período chuvoso seja detectado e assim abasteça o reservatório até o nível do sensor superior.

O LED é um dispositivo eletrônico que permite a passagem de corrente elétrica para transformar essa energia em luz. Essa transformação é possível pela ação de um chip semicondutor que é o elemento mais importante do LED. O LED é considerado um componente bipolar pois possui dois terminais: anodo (pólo negativo) e catodo (pólo positivo). A figura 3.18 mostra um esquema de um LED simples que permite visualizar os terminais do anodo e do catodo e o local onde é instalado o chip semicondutor.

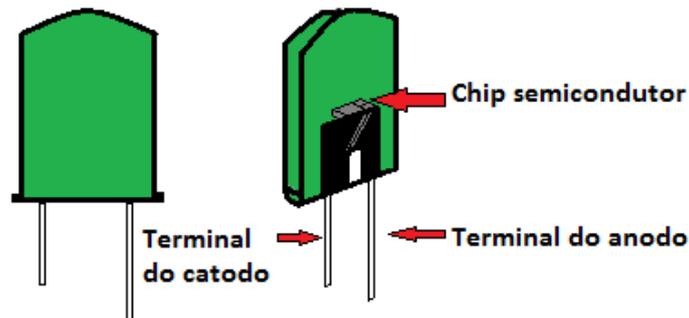


Figura 3.18-Componentes do LED

(Fonte:Autor)

O buzzer é um dispositivo alto-falante que possui um oscilador interno para produzir sons quando alguma corrente elétrica passa pelos seus contatos e é utilizado em circuitos eletrônicos para a sinalização de alguma alteração de comportamento do circuito. Assim como os LEDs eles possuem um terminal anodo e catodo Para esse circuito ele é acionado no instante em que a bomba é ligada para abastecer a caixa d'água e permanece ligado até que a caixa esteja cheia ou o reservatório de chuva fique vazio durante o abastecimento.

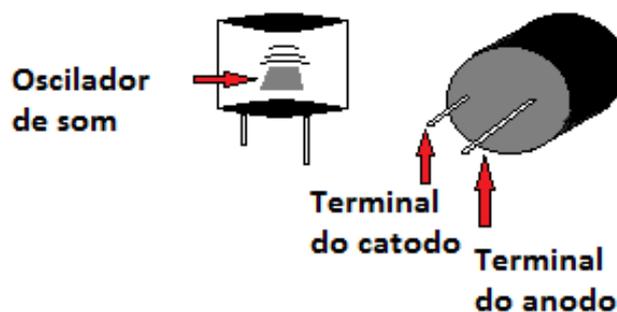


Figura 3.19-Componentes do Buzzer

(Fonte:Autor)

O catodo do LED se conecta a porta Gnd do Arduino e o anodo se conecta a uma porta digital passando por um resistor. O catodo do buzzer também se conecta a porta

Gnd e o anodo a uma porta digital. O esquema de ligação do módulo de alerta é demonstrado na figura 3.20.

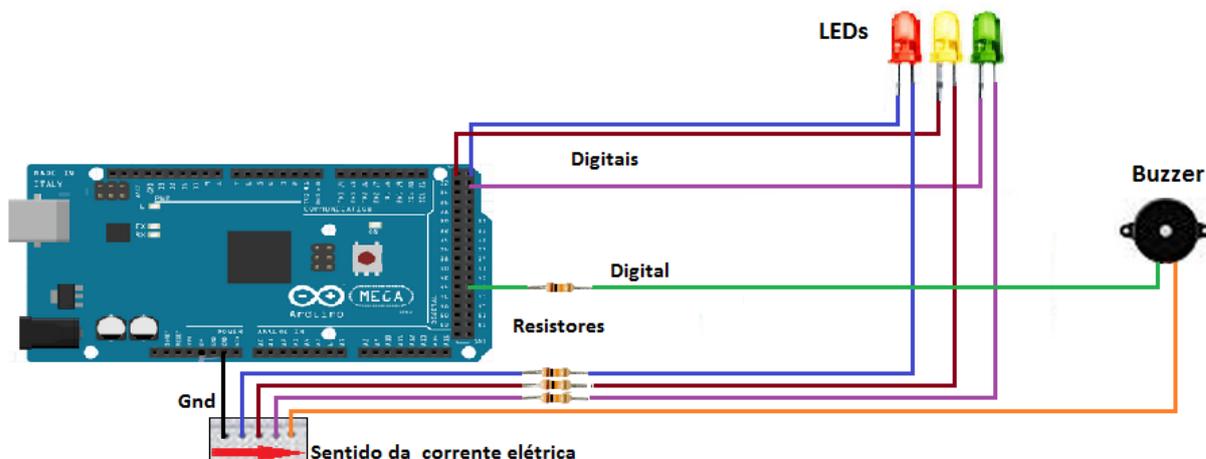


Figura 3.20-Esquema de ligação do módulo de alerta

(Fonte:Autor)

3.3.5.Display LCD

O display LCD é preenchido por cristais líquidos feitos para exibir informações por via eletrônica. A tela do display possui várias camadas sobrepostas para que as informações desejadas sejam visualizadas. A figura 3.21 mostra todas as camadas sobrepostas do LCD.

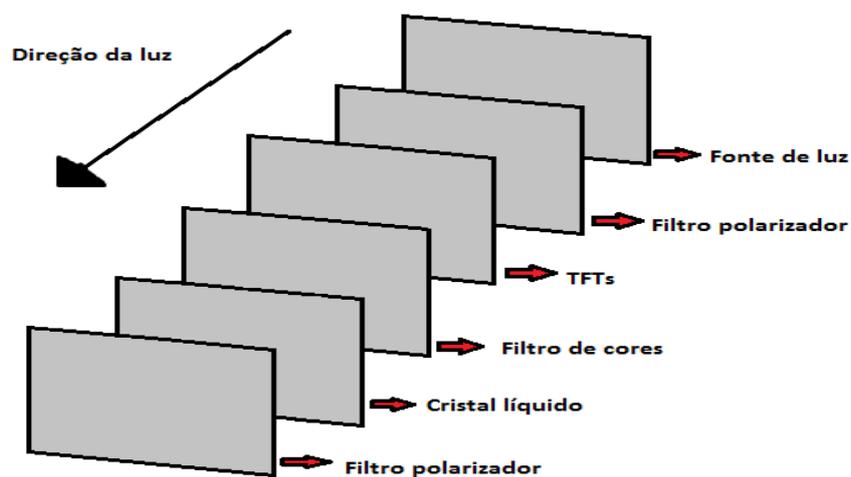


Figura 3.21-Camadas do display LCD

(Fonte:Autor)

A luz vem da fonte e passa por um filtro polarizador que permite a passagem de luz polarizada em uma direção específica. A luz polarizada é aquela que se propaga em um único plano. Logo depois recebe uma qualidade melhor na camada TFT que

aperfeiçoa o contraste e endereçamento de pixels. Na camada de cores o display realiza um controle de tensão para exibir os diversos tipos de cores possíveis. No instante em que a luz chega na camada de cristal, a variação de campo magnético coloca os cristais na posição desejada formando as informações visualizadas na tela e sofrendo um último ajuste no segundo filtro polarizador. Essas informações são escritas na tela por meio da função `print()` usada na IDE.

O display usado nesse projeto possui 16 pinos de saída. Os pinos 1, 5 e 16 são ligados no terra (Gnd) e os pinos 2 e 15 são ligados no 5 Volts. O pino 3 permite um ajuste na intensidade da luz emitida. Isso é possível variando-se a tensão nesses pinos por meio de uma de 10 Kiloohm. Os pinos 4, 6, 11, 12, 13 e 14 são pinos conectados as portas digitais do Arduino e o restante (7, 8, 9, 10) não são usados. A figura 3.22 mostra a função usada para escrever informações no display e como os pinos digitais são configurados na IDE para se conectarem as portas digitais do Arduino. A função `LiquidCrystal` seta as portas do Arduino em que os pinos irão se conectar. A figura 3.23 demonstra como os pinos do display foram conectados.

```
LiquidCrystal lcd(40,41,42,43,44,45); ← Configura os pinos
:
:
lcd.print("Sistema estavel"); ← Escreve informações no LCD
```

Figura 3.22-Configuração dos pinos digitais do LCD

(Fonte:Autor)

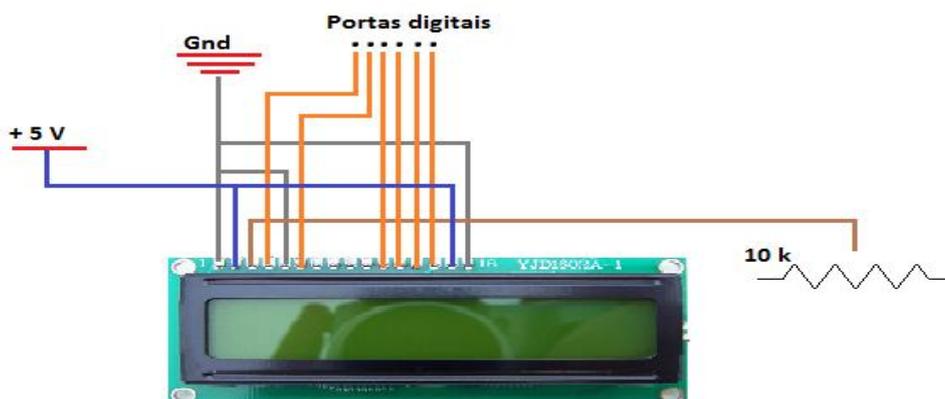


Figura 3.23-Ligação do display LCD

(Fonte:Autor)

3.3.6.Conexão da rede local

A página WEB é acessada por meio de um IP dentro de uma rede local. Esse IP é configurado no código da IDE do Arduino usando o `IPAddress` (Lista 8). A rede local é conhecida como Wi-Fi e para ter acesso a ela o usuário deve estar dentro do raio de

atuação dessa rede e passar por um processo de autenticação. O processo de autenticação é baseado em uma senha criada pelo próprio usuário evitando que outros dispositivos não autorizados que estão dentro do raio de ação dessa rede se conectem a ela. Essa rede é criada da seguinte forma: o Arduino mega utiliza a Shield Ethernet para estabelecer uma conexão com o roteador e então envia os dados para a rede local. O roteador usado nesse projeto é um modelo Technicolor TD5130. O servidor é o próprio Arduino e o cliente é o navegador. O cliente então envia uma solicitação para o servidor e este envia uma resposta. A figura 3.24 demonstra o esquema de conexão entre o Arduino mega, a Shield Ethernet e o roteador para que a rede seja criada e acessada por meio de dispositivos eletrônicos.

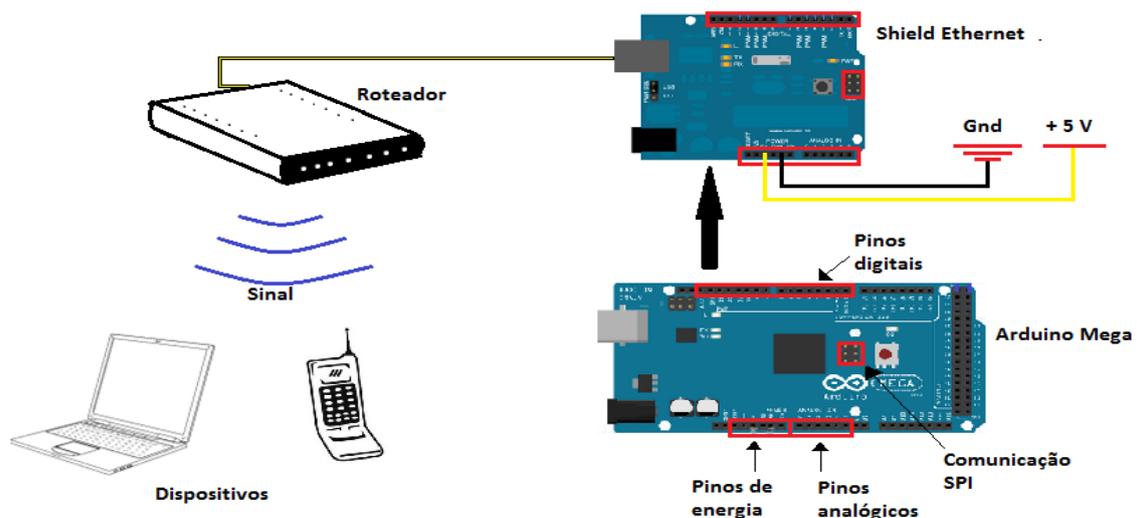


Figura 3.24-Esquema de conexão da rede

(Fonte:Autor)

A Shield Ethernet é conectada nos pinos digitais (0 a 13), analógicos (A0, A1, A2, A3, A4 e A5), de energia (Gnd, vin, 3.3V e 5 V) sendo que os pinos Gnd e 5 V da Shield alimentam o circuito da placa e aos pinos de comunicação SPI do Arduino mega que permitem a comunicação do microcontrolador com diversos dispositivos formando uma rede. Para se conectar com o roteador a Shield usa um cabo de rede RJ45. O roteador então gera um sinal com alcance limitado e qualquer dispositivo que estiver dentro da área de atuação desse sinal poderá se conectar a rede local da residência e acompanhar o monitoramento do sistema pela página WEB em tempo real via Wi-Fi. O SPI permite que os dados sejam transmitidos e recebidos simultaneamente.

3.4.Protótipo

O protótipo foi desenvolvido em madeira MDF e cano de PVC. Esse protótipo têm o objetivo de simular o processo de coleta de água da chuva através do telhado, calhas e condutores para uma cisterna abaixo do solo. Para o reservatório inferior foi usado um galão de água e para o superior uma caixa com dimensões menores (ver figura 3.25).



Figura 3.25 – Protótipo do projeto

(Fonte:Autor)

Para o cálculo da área de contribuição do protótipo foi obtida as medidas em centímetros do telhado: 0,54 cm de profundidade, 0,27 cm de altura e 0,29 cm da metade do comprimento. A figura 3.26 mostra essas medidas com mais clareza.

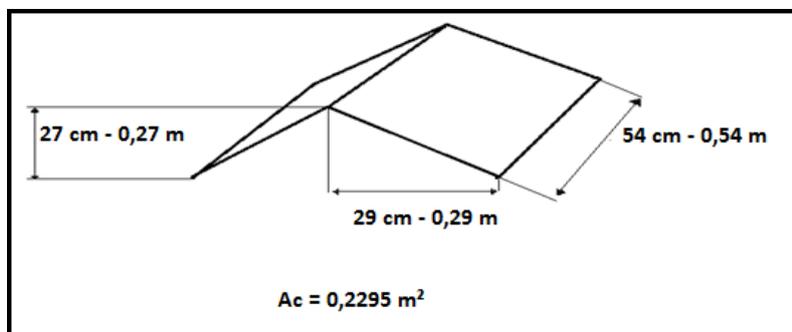


Figura 3.26 – Medidas da área de contribuição do protótipo

(Fonte:Autor)

A calha foi instalada na beira do telhado para coletar a água e nas suas extremidades foi instalado um cano que representa um condutor vertical que direciona a água coletada para o reservatório inferior. A figura 3.27 demonstra a instalação da calha e do condutor no protótipo.



Figura 3.27 – Instalação da calha e do condutor no protótipo

(Fonte:Autor)

Capítulo 4 – Testes realizados

Neste capítulo serão relatados os testes realizados com os dispositivos do sistema. Os testes são organizados em cenários e são divididos em 2 partes: sensor de chuva e sensores de níveis de água e módulo relé (acionamento da bomba elétrica).

Tabela 4.1 – Cenário de testes

	Dispositivos	Tensão aplicada	Pinos usados no Arduino
Cenário 1	Sensor de chuva	5 Volts - 0 Volts	Digital 47
Cenário 2	Sensores de níveis de água e Módulo relé	5 Volts - 0 Volts	Digitais 14,15,16 e 17 - Sensores de níveis Digital 22 - Módulo relé

(Fonte:Autor)

Cada cenário de teste descreve qual dispositivo foi testado, qual o meio utilizado para o teste, o processo que levou a obter os resultados, as tensões aplicadas dentre outras informações usadas durante os testes.

4.1.Cenário 1

Este cenário descreve os testes realizados com o sensor de chuva. Todos os dados foram obtidos por um osciloscópio *Agilent Technologies* modelo DSO-X 3024A. Este aparelho permite a visualização de uma onda de frequência aplicada em um determinado nível de tensão. Para obter o sinal é necessário que uma entrada de tensão do osciloscópio se conecte a porta digital do Arduino por onde passa o sinal do sensor de chuva e a outra a um sinal terra qualquer da placa de circuito.

O sensor de chuva possui uma placa resistiva composta por duas malhas. Os testes realizados comprovaram que a resistência dessa placa diminui caso um fluido seja aplicado constantemente sobre sua superfície. A queda da resistência se dá pelo fato de ocorrer movimentação de corrente elétrica entre as malhas da placa enquanto se aplica o fluido. Nesse caso foi obtido no osciloscópio um sinal digital constante em nível baixo na faixa de 0 Volts. Caso não ocorra movimentação de corrente elétrica, ou seja, se não houver nenhum fluido sendo aplicado constantemente na placa, não haverá queda de resistência. Para essa situação foi comprovado no osciloscópio que o sensor retorna um sinal com valor de tensão maior comparada ao outro caso. Esse sinal digital permanece constante na faixa de 5 Volts. No osciloscópio, a referência “T” marca a tensão de 5 Volts atingida pelo sinal alto gerado pelo sensor. Logo abaixo está o eixo terra que é o 0 Volts onde o sinal em nível baixo permanece (Figura 4.1). Esses sinais em níveis baixos e altos são respectivamente sinais digitais 0s e 1s convertidos pelo Arduino.



Figura 4.1 – Sinal do sensor de chuva no osciloscópio

(Fonte:Autor)

Dentre as etapas de testes, o osciloscópio obteve vários momentos de queda e aumento de sinal do sensor de chuva. Para isso foi simulado vários períodos de chuva onde havia um acréscimo do sinal e conseqüentemente da tensão entre eles. Quando o período chuvoso era detectado novamente o sinal caía para o eixo terra, entrando na faixa de 0 Volts. O osciloscópio gerou um sinal de onda com uma frequência de 321.20mHz.

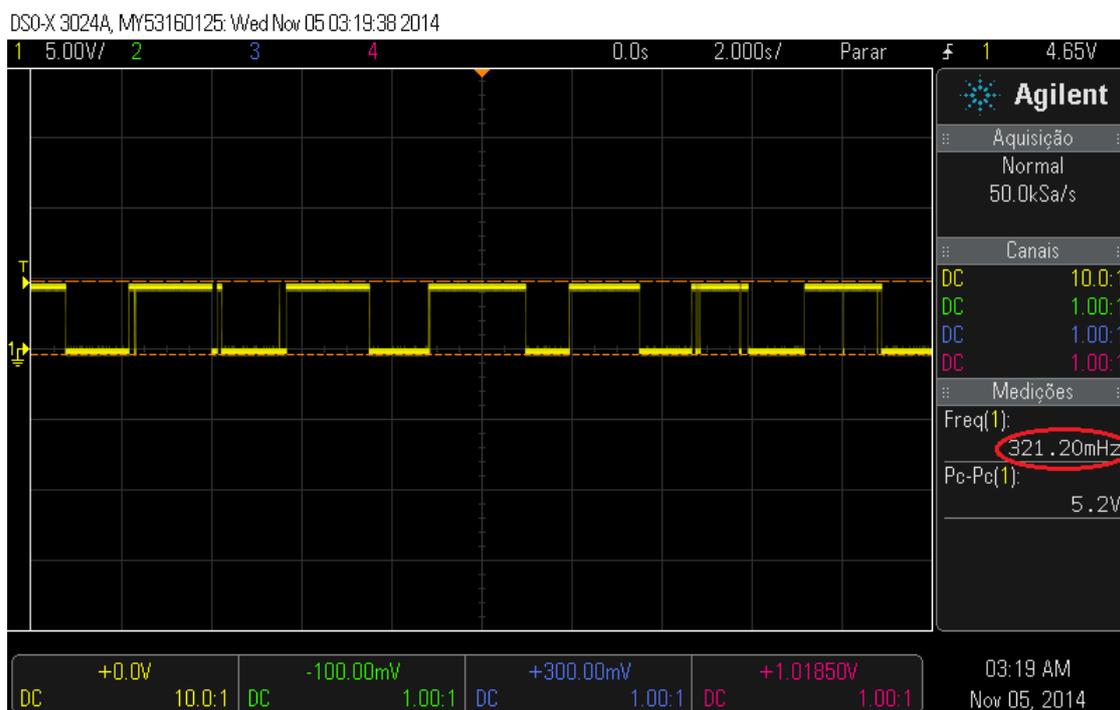


Figura 4.2 – Queda e aumento do sinal do sensor de chuva no osciloscópio

(Fonte:Autor)

4.2.Cenário 2

O cenário 2 descreve os testes realizados com os sensores de níveis de água e o módulo relé que aciona a bomba dependendo do valor de retorno desses sensores. Os testes foram feitos pelo Arduino Mega analisando o sinal de saída digital dos dois atuadores pelo serial monitor do microcontrolador.

Como foi descrito na seção 3.3.1, os sensores de níveis são sensores digitais que retornam apenas dois valores de tensão: 5 e 0 Volts. O Arduino ao ler essas grandezas converte essas tensões em sinais digitais 1 e 0. O módulo aciona a bomba d'água em dois casos (seção 3.3.3) e o processo é o mesmo em ambos. Enquanto o nível da água manter a bóia do sensor voltada para cima, sua tensão permanecerá em 5 Volts e em contrapartida a tensão na bobina do relé será de 0 Volts. Ambas as tensões se mantêm constantes durante um determinado período. A partir do momento em que a tensão do sensor cai para 0 Volts, é corrido um período “ ΔT ” até o instante em que uma tensão de 5Volts é aplicada na bobina do relé ligando a bomba d'água. A variável “ ΔT ” é o período de tempo desde que o sinal digital do sensor caiu para o nível 0 até o momento em que o Arduino leva para executar o comando no código na IDE que envia a tensão de 5 Volts para o circuito do relé ligando a bomba. Nota-se no gráfico abaixo que essa variável representa o tempo em que a tensão do sensor e do relé permanece no mesmo nível. De forma similar esse período é decorrido no instante em que os sensores retornam uma tensão de 5 Volts até o momento em que o Arduino executa o comando para interromper a passagem de corrente no circuito do relé desligando a bomba. Esse período não foi medido mais foi estimado com um valor igual a menos de 1 segundo.

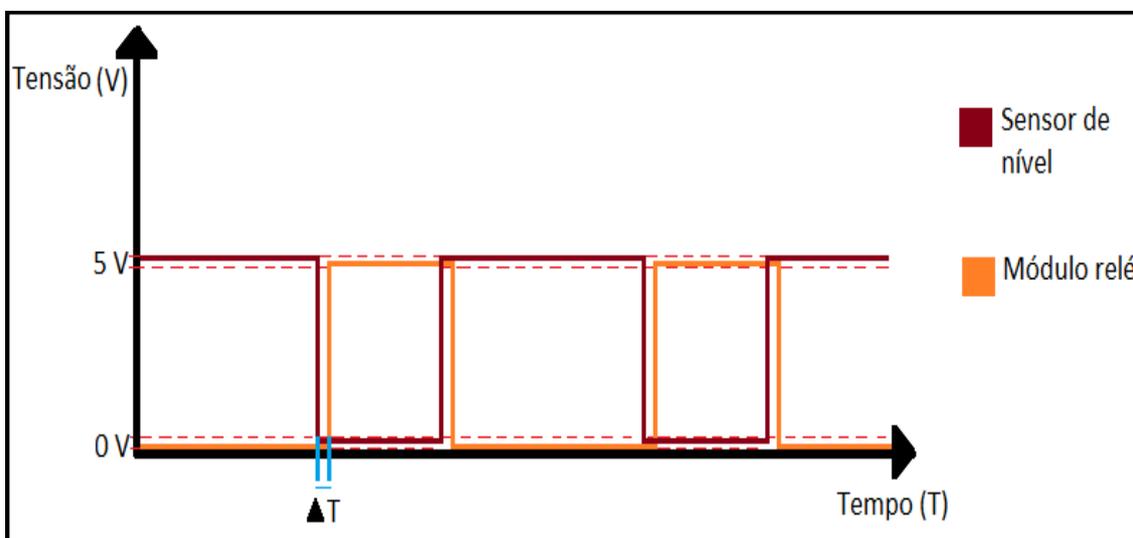


Figura 4.3 – Gráfico do sinal digital dos sensores de níveis e do módulo relé

(Fonte: Autor)

Capítulo 5 – Conclusão

Neste capítulo será abordado todas as conclusões referentes ao desenvolvimento do projeto, quais são as aplicações possíveis e as sugestões de melhorias que esse projeto pode ter para atingir maiores proporções e aplicações.

5.1. Conclusões do Projeto.

A utilização desse projeto pode propiciar uma economia no uso da água potável se for aplicado em uma residência levando proporções maiores. O fator econômico depende de algumas variáveis como a quantidade de pessoas que irão consumir essa água, o volume do reservatório de água pluvial e a demanda de chuva na região.

Com o uso do Arduino no projeto foi possível propiciar escalabilidade ao sistema pois o microcontrolador dá a ele a capacidade de acrescentar novas funcionalidades ao ambiente de desenvolvimento se houver maior demanda de informações. Isso é possível pois o Arduino têm a facilidade de controlar diversos dispositivos através de suas portas digitais e analógicas, basta configurá-las no código-fonte. Várias tecnologias são adaptáveis ao Arduino, como a Shield Ethernet que estabelece uma conexão local para que as informações possam ser transmitidas para a interface WEB. Outra característica que foi encontrada no Arduino através dos testes realizados foi sua precisão de leitura de dados dos sensores e tempo de resposta para uma dada operação. As informações são mantidas constantemente em tempo real e a precisão com que o sistema opera é notável, pois todos os dispositivos responderam de uma forma eficaz aos comandos da IDE do Arduino.

5.2. Aplicações do sistema

O sistema de captação de águas pluviais possui várias aplicações além da citada neste trabalho pois além de ter seu funcionamento em residências particulares, esses sistemas podem ser instalados em grandes indústrias com altos índices de produção e em sistemas de drenagens de asfaltos.

Caso grandes indústrias aplique esse sistema, elas estarão reduzindo seu custo de produção e manutenção pelo fato de que grandes partes de seus processos e atividades necessite do consumo de água, principalmente as indústrias de agricultura. A água da chuva poderia ser usada para esses fins pois eles não necessitam de água potável. Nesse caso a proporção das cisternas seriam maiores para atender a grande demanda de uma indústria e o bombeamento de água precisaria ser mais potente, tendo uma capacidade maior de fluxo de água para atender a demanda de produção mais rapidamente.

Em sistemas de drenagens de pavimentos, as águas que ocasionariam as enchentes são escoadas para um reservatório abaixo do solo e logo depois são descartadas através de tubulações em cursos de águas naturais, como os oceanos, lagos ou em alguns casos, lançadas sobre terrenos que infiltram essa água no subsolo. O caminho com que a água percorre com esse sistema é totalmente diferente do original.

Nesses sistemas de drenagem, as cisternas de armazenamento e as tubulações para direcionarem as águas aos seus cursos naturais necessitariam de um raio maior de circunferência pelo fato de ter maior quantidade de água podendo vir de chuvas fortes com longos períodos de duração e maior espaço para o seu manejo.

5.3.Sugestões de melhoria

Para que o sistema tenha funcionalidades a mais, em vez de monitorá-lo somente por uma rede local na intranet, é possível desenvolver um monitoramento através de uma rede global na internet tornando possível ao usuário ter acesso as informações do sistema em qualquer lugar do mundo através de redes Wi-Fi e 3G.

Seria possível também o sistema possuir um banco de dados que se conectasse a estações meteorológicas para armazenar o valor das precipitações de pluviosidade média anual na determinada região. Com isso o sistema teria valores mais exatos da vazão do projeto.

Referências

Arduino. *Arduino Ethernet Shield*, 2014. Acessado em 10/2014. Disponível em:

<http://arduino.cc/en/Main/ArduinoEthernetShield>

Arduino. *Arduino Mega*, 2014. Acessado em 10/2014. Disponível em:

<http://arduino.cc/en/Main/arduinoBoardMega>

Argolo, Isabelle Galeão, Edward Menezes, Tonis Magno, Robson Rodrigues, Urbano Augusto e Murilo Plinio. *Artigo Sensores*, 2010. Acessado em 10/2014. Disponível em:

<http://www.ebah.com.br/content/ABAAABa3lAK/artigo-sensores>

Alessandro Silva. *Ethernet Shield, saiba mais desta placa muito conhecida*, 2013. Acessado em 10/2014. Disponível em:

<http://www.oarduino.com/placa-arduino-ethernet-shield/>

Archibald Joseph Macintyre. *Instalações Hidráulicas Prediais e Industriais*. 4ªEd. Rio de Janeiro: Editora LTC, 2010.

Banzi. *Primeiros passos com o arduino*. 1ªEd. São Paulo: Editora Novatec Ltda, 2010.

Diego Marcelo dos Santos. *Funcionamento dos Relês*, 2014. Acessado em 10/2014. Disponível em :

<http://www.infoescola.com/electronica/rele/>

Edison Crepani, José Medeiros, Alessandro Ferraz. *Instituto Nacional de Pesquisas Espaciais*. Ministério da Ciência e Tecnologia, 20014. Acessado em 09/2014. Disponível em:

http://www.dsr.inpe.br/dsr/simeao/Publicacoes/Intensidade_Pluviom%E9trica.pdf

Enciclopédia Tigre. *Dimensionamento de Calhas de Telhado*, 2010. Acessado em 09/2014. Disponível em:

<http://www.tigre.com.br/enciclopedia/artigo/83/Dimensionamento+de+Calhas+de+Telhado>

Felipe Martini. *Potencial de Economia de água potável por meio do uso de água de chuva em São Miguel do Oeste – SC*, 2009. Universidade Federal de Santa Catarina. Acessado em 11/2014. Disponível em:

http://www.labee.ufsc.br/sites/default/files/publicacoes/tccs/TCC_Felipe_Martini.pdf

George Coulouris, Jean Dollimore & Tim Kindberg. *Sistemas Distribuídos – Conceitos e Projetos*. 4ªEd. São Paulo: Editora Bookman, 2007.

Gérard Moss- Globo Ecologia. *Mundo enfrenta crise de água doce*, 2013. Acessado em 09/2014. Disponível em:

<http://redeglobo.globo.com/globoecologia/noticia/2013/05/mundo-enfrenta-cri-se-de-agua-doce.html>

Helber Henrique Lorenzete & Dr. Ricardo Luiz. *Estudo de vantagens da captação de água de chuva para uso doméstico*, 2011. Acessado em 08/2014. Disponível em:

<http://rmai.com.br/v4/Read/657/estudo-de-vantagens-da-captacao-de-agua-de-chuva-para-uso-domestico.aspx>

Instalações Prediais de águas Pluviais (NBR 10844). ABNT. Rio de Janeiro, Dezembro de 1989. Acessado em 08/2014 . Disponível em:

<file:///C:/Users/Usuario/Downloads/NBR+10844.pdf>

JACEE. *Minicurso Arduino*, 2012. Equipe de robótica UFES. Acessado em 10/2014. Disponível em:

http://www.inf.ufes.br/~erus/arquivos/ERUS_minicurso%20arduino.pdf

João Mamede Filho. *Manual de Equipamentos Elétricos*. 3ªEd. Rio de Janeiro: Editora LTC, 2005.

João Manoel Gomes. *Controle em Malha Aberta*, 2000. Acessado em 10/2014. Disponível em:

<http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node5.html>

João Manoel Gomes. *Controle em Malha Fechada*, 2000. Acessado em 10/2014. Disponível em:

<http://www.ece.ufrgs.br/~jmgomes/pid/Apostila/apostila/node6.html>

Katsuhiko Ogata. *Engenharia de Controle Moderno*. 3ªEd. Rio de Janeiro: Editora LTC, 2000.

Leandro Oriente. *HTML Básico*, 2014. Acessado em 10/2014. Disponível em:

<http://www.linhadecodigo.com.br/artigo/81/html-basico.aspx>

Leonardo Weierbacher. *Estudo de Captação e Aproveitamento de água da chuva na indústria moveleira bento móveis de alvorada – RS*, 2008. Acessado em 09/2014. Disponível em:

<http://www.pliniotomaz.com.br/downloads/07leonardo.pdf>

Lívia Cunha. *Relés e Contatores*, 2009. Acessado em 10/2014. Disponível em:

<http://www.osetoreletrico.com.br/web/a-revista/edicoes/169-reles-e-contatores.html>

Manual de construções e instalações Versão 1.1. *Projeto experimental do Filtro de água de chuva de baixo custo*, 2012. Acessado em 08/2014. Disponível em:

<http://www.sempresustentavel.com.br/hidrica/minicisterna/filtro-de-agua-de-chuva.htm>

Marcelo Wendling. *Sensores*, 2010. Universidade Estadual Paulista. Acessado em 10/2014. Disponível em:

<http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>

Mariana Segala. *Água: a escassez na abundância*, 2014. Acessado em 08/2014. Disponível em:

<http://planetasustentavel.abril.com.br/noticia/ambiente/populacao-falta-agua-recursos-hidricos-graves-problemas-economicos-politicos-723513.shtml>

Matheus Fabiani. *Rugosidade das superfícies*, 2014. Acessado em 11/2014. Disponível em:

<http://www.ebah.com.br/content/ABAAfXcgAB/rugosidade>

Michael McRoberts. *Arduino Básico*. Editora Novatec Ltda, 2011.

Nilza Ramos e Ariovaldo Luchiar. *Monitoramento Ambiental*, 2014. Embrapa – Agência Embrapa de Informação Tecnológica(ageitec). Acessado em 11/2014. Disponível em:

http://www.agencia.cnptia.embrapa.br/gestor/cana-de-acucar/arvore/CONTAG01_73_711200516719.html

Norman S. Nise. *Engenharia de Sistemas de Controle*. 6ªEd. Rio de Janeiro: Editora LTC, 2012.

Prefeitura de São Paulo (Desenvolvimento Urbano). *Manual de Drenagem e Manejo de Águas Pluviais – Gerenciamento do Sistema de Drenagem Urbana*, 2012. Volume 1. São Paulo. Acessado em 09/2014. Disponível em:

http://www.prefeitura.sp.gov.br/cidade/secretarias/upload/desenvolvimento_urbano/arquivos/manual-drenagem_v1.pdf

Redação: Revista Atualidades Vestibular. *O mundo com sede*, 2014. Acessado em 08/2014. Disponível em:

http://planetasustentavel.abril.com.br/noticia/desenvolvimento/conteudo_261013.shtml

Renata Branco. *Usos do relé eletromagnético*, 2011. Acessado em 10/2014. Disponível em:

<http://www.manutencaoesuprimentos.com.br/conteudo/3753-usos-do-rele-eletromagnetico/>

Richard C. Dorf & Robert H. Bishop. *Sistemas de controle Modernos*. 12ªEd. Rio de Janeiro: Editora LTC, 2013.

Rodolfo Alves Pena. *O problema das enchentes*, 2014. Acessado em 09/2014. Disponível em:

<http://www.brasilecola.com/geografia/enchentes.htm>

São José do Cerrito. *Materiais e ferramentas para construção de uma caixa de água em ferro cimento*, 2003. Acessado em 08/2014 e 09/2014. Disponível em:

http://ieham.org/html/docs/Caixa_%C1gua_Ferrocimento.pdf

Simon Monk. *Programação com Arduino – Começando com Sketches*. Editora Bookman, 2013.

SNatural. *Água de chuva – Captação e armazenamento*, 1989-2011. Acessado em 09/2011. Disponível em:

<http://www.snatural.com.br/Agua-Chuva-Captacao-Armazenamento-B.html>

Sperancetta, Juliano Vizzotto Alves e Cláudio Kruger. *Captação de águas pluviais para abastecimento residencial*. Artigo Da Vinci – v.1. Edição: Curitiba, 2004.

Yuri Pacievitch. *HTML*, 2014. Acessado em 10/2014. Disponível em:

<http://www.infoescola.com/informatica/html/>

Anexo – Código Fonte do Sistema

```
#include <LiquidCrystal.h>

#include <SPI.h>

#include <Ethernet.h>

int LedVermelho = 24;

int LedAmarelo = 25;

int LedVerde = 23;

int SENSOR_CHUVA = 47;

int BUZZER = 51;

int SENSOR_CAIXA1 = 17;

int SENSOR_CAIXA2 = 16;

int SENSOR_RESERVATORIO1 = 15;

int SENSOR_RESERVATORIO2 = 14;

int BOMBA = 22;

double tempChuva = 0;

double vazao = 0;

double intensidadePluvi = 0;

double cont = 0;

LiquidCrystal lcd(40,41,42,43,44,45);

byte mac[] = {

    0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };

IPAddress ip(192,168,1,177);

EthernetServer server(80);

EthernetClient client;
```

```
void setup(){

  pinMode(LedVermelho,OUTPUT);

  pinMode(LedAmarelo,OUTPUT);

  pinMode(SENSOR_CAIXA1,INPUT);

  pinMode(SENSOR_CAIXA2,INPUT);

  pinMode(SENSOR_RESERVATORIO1,INPUT);

  pinMode(SENSOR_RESERVATORIO2,INPUT);

  pinMode(SENSOR_CHUVA,INPUT);

  pinMode(BOMBA,OUTPUT);

  pinMode(LedVerde,OUTPUT);

  pinMode(BUZZER,OUTPUT);

  lcd.begin(16, 2);

  Ethernet.begin(mac, ip);

  server.begin();

  delay(100);

}

void abastecerCaixa(){

  lcd.setCursor(0,0);

  lcd.print("Reabastecendo...");

  client.println("<tr><td><font color = 'yellow'>Reabastecendo...</font></td></tr>");

  cont = 0;

  digitalWrite(BOMBA,HIGH);

  for(;;){

    delay(1000);
```

```
cont = cont +1;

if( digitalRead(SENSOR_CAIXA1)==HIGH && digitalRead(SENSOR_CAIXA2)==HIGH ){

    digitalWrite(BOMBA,LOW);

    digitalWrite(BUZZER,LOW);

    digitalWrite(LedAmarelo,LOW);

    digitalWrite(LedVermelho,LOW);

    digitalWrite(LedVerde,HIGH);

    cont = cont/60;

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Caixa cheia.");

    lcd.setCursor(0,1);

    lcd.print("tempo:");

    lcd.print(cont);

    lcd.print(" minutos");

    client.println("<tr><td><font color = 'green'>Caixa cheia</font></td></tr>");

    client.println("<tr><td>Tempo de abastecimento:");

    client.println(cont);

    client.println(" minutos</td></tr>");

    delay(5000);

    break;

}

if(digitalRead(SENSOR_RESERVATORIO1)==LOW){

    digitalWrite(BOMBA,LOW);

    digitalWrite(BUZZER,LOW);

    digitalWrite(LedAmarelo,LOW);

    digitalWrite(LedVermelho,HIGH);
```

```

digitalWrite(LedVerde,LOW);

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Reservatorio");

lcd.setCursor(0,1);

lcd.print("vazio.");

client.println("<tr><td><font color = 'red'>Reservatorio vazio</font></td></tr>");

for(;;){

  if(digitalRead(SENSOR_CHUVA)==LOW){

    tempChuva=0;

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Chuva detectada.");

    client.print("<tr><td>Chuva detectada.</td></tr>");

    for(;;){

      delay(1000);

      tempChuva = tempChuva+1;

      lcd.setCursor(0,1);

      lcd.print(tempChuva);

      digitalWrite(LedAmarelo,HIGH);

      if( digitalRead(SENSOR_RESERVATORIO2)==HIGH){

        lcd.clear();

        lcd.setCursor(0,0);

        lcd.print("Reabastecendo...");

        client.println("<tr><td><font color = 'yellow'>Reabastecendo...</font></td></tr>");

        for(;;){

          digitalWrite(BOMBA,HIGH);

```

```
digitalWrite(BUZZER,HIGH);

digitalWrite(LedVerde,LOW);

digitalWrite(LedVermelho,LOW);

if( digitalRead(SENSOR_CAIXA1)==HIGH && digitalRead(SENSOR_CAIXA2)==HIGH ){

    tempChuva = tempChuva/60;

    digitalWrite(BOMBA,LOW);

    digitalWrite(BUZZER,LOW);

    digitalWrite(LedAmarelo,LOW);

    digitalWrite(LedVerde,HIGH);

    lcd.clear();

    lcd.setCursor(0,0);

    lcd.print("Caixa cheia.");

    lcd.setCursor(0,1);

    lcd.print("chuva:");

    lcd.print(tempChuva);

    lcd.print("Minutos");

    client.println("<tr><td><font color = 'green'>Caixa cheia</font></td></tr>");

    client.println("<tr><td>Tempo de chuva:");

    client.println(tempChuva);

    client.println(" minutos</td></tr>");

    intensidadePluvi = 3/tempChuva;

    vazao = (intensidadePluvi*0,2295)*1000;

    delay(4000);

    break;

}

}

break;
```

```
    }  
  }  
  break;  
}  
}  
break;  
}  
}  
}  
  
void verificaSensores(){  
  if( digitalRead(SENSOR_CAIXA2)==LOW && digitalRead(SENSOR_CAIXA1)==LOW ){  
    digitalWrite(LedAmarelo,HIGH);  
    digitalWrite(LedVermelho,LOW);  
    digitalWrite(LedVerde,LOW);  
    digitalWrite(BUZZER,HIGH);  
    lcd.clear();  
    lcd.setCursor(0,0);  
    lcd.print("Caixa vazia.");  
    client.print("<tr><td><font color = 'red'>Caixa vazia</font></td></tr>");  
    abastecerCaixa();  
  }else{  
    digitalWrite(LedVerde,HIGH);  
    digitalWrite(LedAmarelo,LOW);  
    digitalWrite(LedVermelho,LOW);  
    digitalWrite(BOMBA,LOW);  
    digitalWrite(BUZZER,LOW);  
    lcd.clear();  
  }  
}
```

```

lcd.setCursor(0,0);

lcd.print("Sistema estavel");

client.print("<tr><td><font color = 'green'>Sistema estavel</font></td></tr>");

if(vazao!=0){

  lcd.setCursor(0,1);

  lcd.print("Vazao:");

  lcd.print(vazao);

  lcd.print("L/min");

  client.print("<br><br>");

  client.print("<center><table border><tr><th>Pluviosidade Media Anual do Centro-
Oeste(m)</th>");

  client.print("<th>Duracao do periodo de chuva(minutos)</th>");
  client.print("<th>Intensidade Pluviometrica(m/min)</th>");
  client.print("<th>Area de contribuicao do telhado(m.m)</th>");
  client.print("<th>Vazao do projeto(L/min)</th></tr>");
  client.print("<tr><td>");client.print("3.00");client.print("</td>");
  client.print("<td>");client.print(tempChuva);client.print("</td>");
  client.print("<td>");client.print(intensidadePluvi);client.print("</td>");
  client.print("<td>");client.print("0,2295");client.print("</td>");
  client.print("<td>");client.print(vazao);client.print("</td></tr></table></center>");

}

}

if(digitalRead(SENSOR_CHUVA)==LOW){

  tempChuva =0;

  lcd.clear();

  lcd.setCursor(0,0);

  lcd.print("Chuva detectada");

  client.print("<center><table border><tr><td>Chuva detectada.</td></tr>");

```

```
digitalWrite(LedAmarelo,HIGH);  
digitalWrite(LedVerde,LOW);  
for(;;){  
    lcd.setCursor(0,1);  
    delay(1000);  
    tempChuva = tempChuva+1;  
    lcd.print(tempChuva);  
    if(digitalRead(SENSOR_CHUVA)==HIGH){  
        tempChuva = tempChuva/60;  
        lcd.clear();  
        lcd.setCursor(0,0);  
        lcd.print("Termino de chuva");  
        lcd.setCursor(0,1);  
        lcd.print("Duracao:");  
        lcd.print(tempChuva);  
        lcd.print("min");  
        client.print("<tr><td>Termino de chuva.</td></tr>");  
        client.print("<tr><td>Duracao:");  
        client.print(tempChuva);  
        client.print(" minutos</td></tr></table></center>");  
        intensidadePluvi = 3/tempChuva;  
        vazao = (intensidadePluvi*0,2295)*1000;  
        delay(4000);  
  
        break;  
    }  
}
```

```

    }
}
void loop(){
    client = server.available();
    if (client) {
        boolean currentLineIsBlank = true;
        while (client.connected()) {
            if (client.available()) {
                char c = client.read();
                if (c == '\n' && currentLineIsBlank) {
                    client.println("HTTP/1.1 200 OK");
                    client.println("Content-Type: text/html");
                    client.println("Connection: close");
                    client.println("Refresh: 5");
                    client.println();
                    client.println("<center><h1><b>Sistema de captacao de aguas pluviais</b></h1><br>");
                    client.println("<table border =><tr><th>Monitoramento</th></tr>");
                    verificaSensores();
                    client.println("</table></center>");
                    client.println("<!DOCTYPE HTML>");
                    client.println("<html>");
                    break;
                }
                if (c == '\n')currentLineIsBlank = true;
                else if (c != '\r')currentLineIsBlank = false;
            }
        }
    }
}

```

```
    delay(1);  
    client.stop();  
  }  
  delay(100);  
}
```