UNIVERSIDADE DO SAGRADO CORAÇÃO

THIAGO DA SILVA PINTO

DESENVOLVIMENTO DE UM SISTEMA DE MONITORAMENTO DE CONSUMO DE ÁGUA UTILIZANDO ARDUINO

BAURU
2015

THIAGO DA SILVA PINTO

DESENVOLVIMENTO DE UM SISTEMA DE MONITORAMENTO DE CONSUMO DE ÁGUA UTILIZANDO ARDUINO

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração como parte dos requisitos para obtenção do Título de Bacharel em Ciência da Computação, sob orientação do Prof. Dr. Elvio Gilberto da Silva.

BAURU
2015

Pinto, Thiago da Silva

P6597d

Desenvolvimento de um sistema de monitoramento de consumo de água utilizando Arduino / Thiago da Silva Pinto. -- 2015.

49f. : il.

Orientador: Prof. Dr. Elvio Gilberto da Silva.

Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Universidade do Sagrado Coração – Bauru – SP.

1. Arduíno. 2. Consumo de Água. 3. Hidrômetro Digital. I. Silva, Elvio Gilberto da. II. Título.

THIAGO DA SILVA PINTO

DESENVOLVIMENTO DE UM SISTEMA DE MONITORAMENTO DE CONSUMO DE ÁGUA UTILIZANDO ARDUINO

Trabalho de Conclusão de Curso apresentado ao Centro de Ciências Exatas e Sociais Aplicadas da Universidade do Sagrado Coração como parte dos requisitos para obtenção do Título de Bacharel em Ciência da Computação, sob orientação do Prof. Dr. Elvio Gilberto da Silva.

Banca examinadora:

_____
Prof. Dr. Elvio Gilberto da Silva
Universidade do Sagrado Coração

_____
Prof. Me. Patrick Pedreira Silva
Universidade do Sagrado Coração

_____
Prof. Me. Henrique Pachioni Martins
Universidade do Sagrado Coração

Bauru, 01 de dezembro de 2015

RESUMO

Os recursos hídricos do planeta estão frequentemente em pauta nas mídias, bem como, o uso consciente da água, visando preservar esse bem tão precioso para a vida na Terra. Neste contexto, com o intuito de quantificar e apresentar dados referentes ao consumo de água em um determinado ponto de uma residência, de forma descomplicada e precisa, e ao mesmo tempo, objetivando a redução do consumo de água com a detecção precoce de vazamentos, este projeto consiste no desenvolvimento de um dispositivo baseado na plataforma Arduino, utilizando a linguagem Processing (Arduino), capaz de atuar como um hidrômetro digital, enviando assim os dados coletados para um banco de dados MySQL por meio da rede WiFi da residência. Para isso, propõe-se a criação de perfil de consumo, histórico e custo aproximado. Este trabalho também visa o desenvolvimento de uma aplicação web utilizando as linguagens PHP e HTML para apresentar os dados coletados aos usuários. Os resultados obtidos demonstraram que o sensor de fluxo empregado em conjunto com protótipo para a realização da coleta dos dados apresenta uma precisão bastante satisfatória para os objetivos, com cerca de apenas 3% de erro e, o software desenvolvido para a apresentação dos dados atingiu o seu objetivo de apresentar informações claras, precisas e detalhadas sobre o consumo de água em uma residência.

Palavras-chave: Arduino. Consumo de água. Hidrômetro Digital.

ABSTRACT

Water resources of the planet are often on the agenda in the media, as well as the conscious use of water, to preserve this as well as precious for life on Earth. In this context, in order to quantify and present data for the consumption of water at a given point of a residence, uncomplicated and precise, and at the same time, aiming to reduce water consumption with the early detection of leaks, this project is to develop a device based on the Arduino platform using the Processing language (Arduino), able to act as a digital meter, thus sending the collected data to a MySQL database through Wi-Fi network of the residence. For this, it is proposed the creation of consumption profile, history and approximate cost. This work also aims to develop a web application using PHP and HTML languages to present the collected data to users. The results showed that the flow sensor used in conjunction with prototype for the data collect presents a quite satisfactory accuracy for the goals, with only about 3% of error and the software developed for the presentation of the data reached its objective of presenting an information clear, accurate and detailed on water consumption in a residence.

Keywords: Arduino. Water consumption. Digital Water Meter

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

CSS	Cascading Style Sheets
HTML	Hiper Text Markup Language
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
ISAM	Indexed Sequential Access Method
PWM	Pulse Width Modulation
SGBD	Sistema de Gerenciamento de Banco de Dados
UML	Unified Modeling Language
USB	Universal Serial Bus
WWW	World Wide Web

SUMÁRIO

1 INTRODUÇÃO

A água é de fundamental importância para a existência de vida, segundo a matéria de Francisco (2015) publicada no site brasilescola.com, cerca de 70% da superfície terrestre é coberta por água, porém apenas 2,5% deste volume é de água doce.

A escassez de água no planeta e a necessidade de fazer uso consciente da mesma são assuntos amplamente abordados pela mídia e organizações mundiais que visam a vida e a saúde na Terra. De acordo com a publicação da Organização das Nações Unidas (ATÉ 2030 planeta..., 2015), o planeta enfrentará um déficit de água de 40% caso não seja melhorada a gestão desse recurso. Em 2014 foi possível acompanhar na mídia a crise hídrica em São Paulo, onde segundo a reportagem de Barifouse (2014) no site da BBC, o sistema Cantareira – principal conjunto de reservatórios da região metropolitana de São Paulo – atingiu os níveis mais baixos desde a sua construção em 1974, resultado de diversos fatores como o clima, falta de planejamento por parte do governo e o aumento da demanda, criando a crise mais impactante dos últimos 80 anos.

Para se evitar grandes crises, é necessário agir com o intuito de melhorar a gestão desse recurso. Fatores como a política, tecnologia, agricultura, natureza, crescimento demográfico e o clima são fortemente impactantes na gestão dos recursos hídricos, mas a população também possui um papel fundamental para melhorar esse cenário e um de seus principais pontos é fazer o uso consciente da água e evitar desperdícios.

Assim como na gestão financeira, para se melhorar a gestão de água dentro de uma residência faz-se necessário o monitoramento preciso de seu gasto. O presente trabalho visa possibilitar tal monitoramento, fornecendo informações do consumo de água em um ponto determinado da residência com registros de data e hora, sendo possível criar um perfil de consumo e, dessa forma, identificar aumentos de consumos inesperados que podem indicar um possível vazamento no sistema e assim providenciar o reparo com antecedência, por exemplo.

2 OBJETIVOS

A seguir são apresentados os objetivos, gerais e específicos, do presente trabalho.

2.1 OBJETIVOS GERAIS

Desenvolver um sistema de monitoramento de consumo de água residencial, em um determinado ponto de leitura, capaz de armazenar e apresentar os dados coletados por meio de interface web.

2.2 OBJETIVOS ESPECÍFICOS

a) realizar uma pesquisa exploratória a fim de proporcionar uma maior familiaridade com os assuntos abordados pelo presente trabalho;
b) desenvolver um software para Arduino capaz de coletar dados sobre o consumo de água em um determinado ponto de uma residência por meio de um sensor de fluxo;
c) transmitir os dados coletados via rede sem fio para um banco de dados MySQL sendo executado em um servidor na rede local;
d) desenvolver um software utilizando as linguagem de programação PHP e HTML para apresentar os dados coletados, criando um histórico de consumo;
e) realizar testes para verificar o funcionamento do protótipo, bem como a precisão da coleta dos dados e, a capacidade do software em apresentar os dados de modo claro e preciso.

3 REFERENCIAL TEÓRICO

Seguindo a proposta de Gil (2002), a seguir é apresentado o referencial teórico levantado com base nas pesquisas bibliográficas utilizadas visando o aprofundamento teórico das áreas abrangentes ao presente trabalho.

3.1 ARDUINO

Segundo McRoberts (2011), um Arduino é um pequeno computador onde é possível programá-lo para processar entradas e saídas entre o dispositivo e os componentes conectados a ele.

“O Arduino é o que chamamos de plataforma de computação física ou embarcada, ou seja, um sistema que pode interagir com seu ambiente por meio de hardware e software.”. (MCROBERTS, 2011, p. 22).

Gomes e Tavares (2013) definem o Arduino como sendo um microcontrolador de placa única, constituído fundamentalmente por um processador Atmel, um cristal oscilador e um regulador linear de 5 volts e um conjunto de software para programá-lo.

De acordo com Banzi (2011), o Arduino é formado por dois componentes principais, sendo o hardware, formado pela placa Arduino, e o software, formado pela integrated development environment (IDE) Arduino, necessária para escrever o código em uma linguagem específica chamada Processing, baseada na linguagem C. Essa IDE permite que os programas sejam escritos, compilados e enviados para o Arduino.

Uma das características mais interessantes do Arduino e que certamente foi a chave para sua popularização, é o fato de se tratar de um projeto opensource, ou seja, tanto o hardware quanto o software do Arduino são de fonte aberta, tendo seus códigos, esquemas e projetos disponíveis para serem utilizados livremente. Por esse motivo, é muito comum encontrar placas não oficiais do Arduino com funcionalidades especificas, porém, segundo Banzi (2011) algumas características essenciais são preservadas independentemente da versão da placa, sendo elas as entradas e saídas de sinal digital, entradas de sinal analógico e conectividade via Universal Serial Bus (USB).

Segundo Gomes e Tavares (2013), a plataforma Arduino está difundida por todo o mundo, sendo utilizada por pessoas de diversos segmentos, como artistas, designers e hobbystas, por exemplo, que buscam uma plataforma de protótipos eletrônicos de código aberto, baseada em hardware e software, flexíveis e fáceis de manipular.

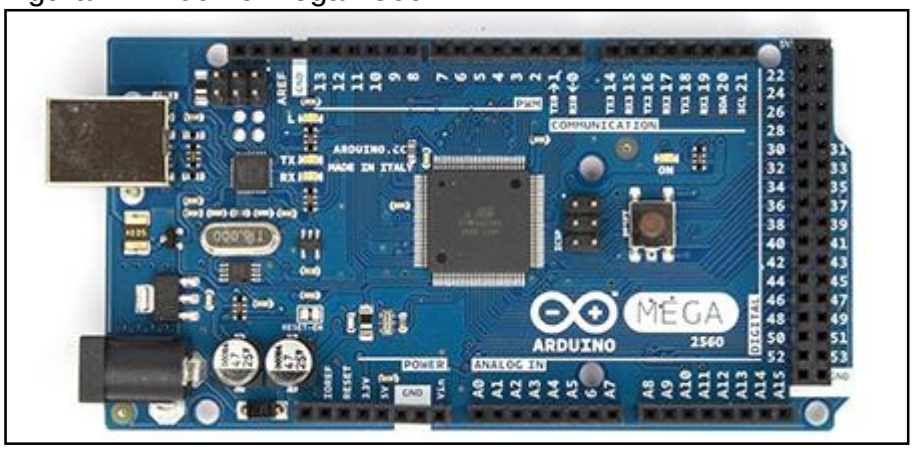
3.1.1 Hardware do Arduino

O Hardware do Arduino possui diversas versões oficiais e outras incontáveis não oficiais, que variam entre si principalmente na questão do controlador utilizado e consequentemente nos números de portas de entra e saída de sinais, tamanho, consumo de energia, capacidade de armazenamento e processamento.

3.1.2 PlacaMega2560

Esta placa (Figura 1) é uma versão baseada no microcontrolador ATmega2560 e dispõe de 256 KB de memória flash – utilizada para o armazenamento do programa - possui 54 pinos de entrada/saída de sinal digital, sendo que 15 fornecem saída Pulse Width Modulation (PWM); 16 entradas analógicas; 4 portas seriais de hardware, além de portas de alimentação e todo os componentes necessários para operar o microcontrolador. (ARDUINO..., [c2015]).

Figura 1– Arduino Mega 2560



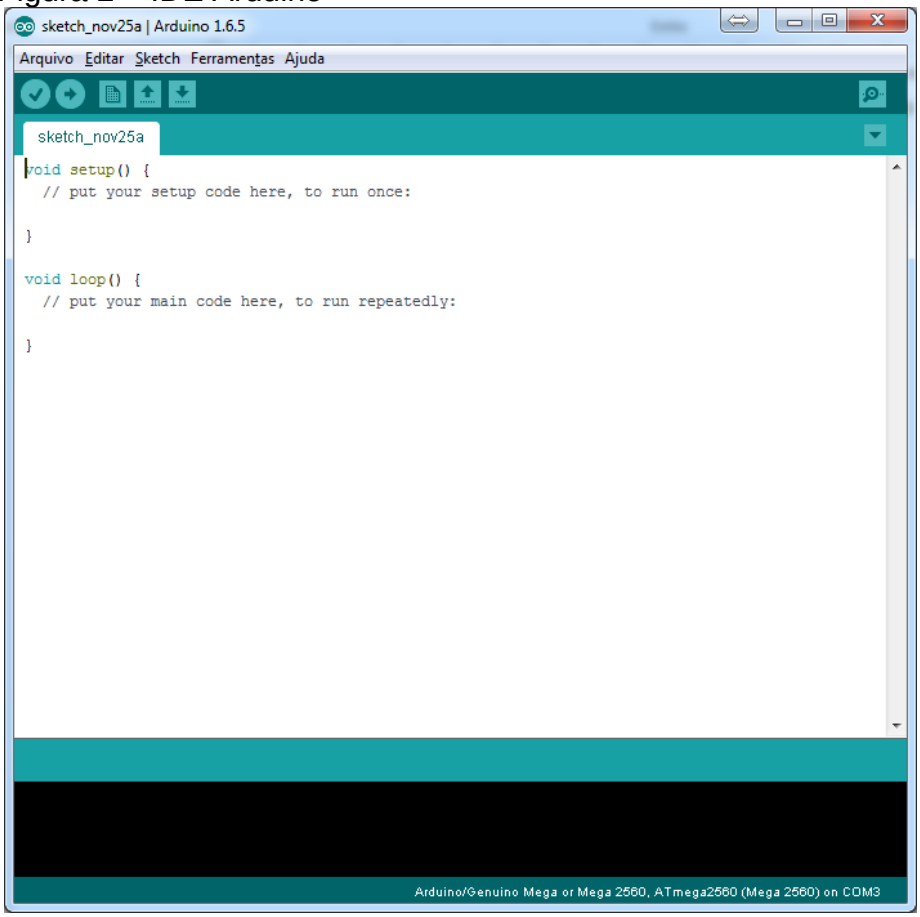
Fonte: Arduino... ([c2015]).

3.1.3 A IDE Arduino

A IDE Arduino (Figura 2), é um software multiplataforma desenvolvido em Java, disponível gratuitamente para download no site oficial do projeto (arduino.cc), criado para ser um ambiente de desenvolvimento de aplicações para o Arduino de fácil utilização, possibilitando que até mesmo pessoas não familiarizadas com a programação de softwares pudessem utilizá-lo.

A linguagem de programação utilizada para o desenvolvimento das aplicações é o Processing, segundo McRoberts (2011), um dialeto da linguagem C, podendo-se, inclusive, utilizar-se totalmente dessa até mesmo em IDEs mais robustas e completas como o Eclipse, porém sua grande vantagem é o emprego dos comandos e bibliotecas específicas para o controle do hardware, o que torna o desenvolvimento mais simples e produtivo.

Figura 2 – IDE Arduino



Fonte: Elaborada pelo autor.

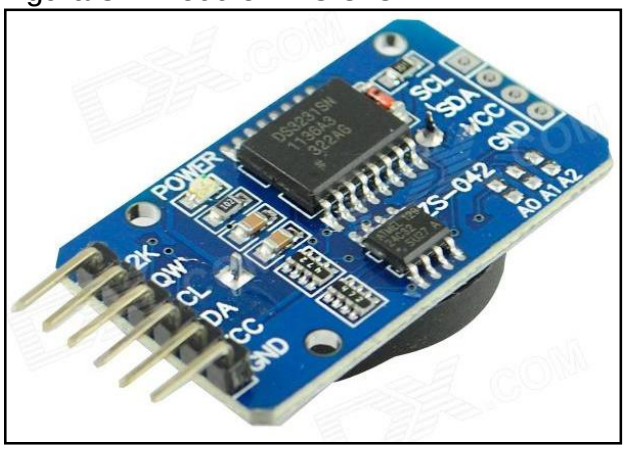
3.2 REAL TIME CLOCK

O módulo Real Time Clock (RTC) baseado no chip DS3231, Figura 3, é o responsável por fazer a contagem real do tempo e transmitir as informações de data e hora para o Arduino.

Segundo o site da Maxim Integrated, o chip DS3231 é um relógio de tempo real extremamente preciso que incorpora uma entrada de bateria para manter a contagem do tempo mesmo quando o fornecimento de energia principal é interrompido, além de realizar os registros em segundos, minutos, horas, dias, meses e anos com correções automáticas para os meses com menos de 31 dias e para os anos bissextos. (EXTREMELY…, [c2005]).

Essas características aliadas ao baixo custo tornam os módulos RTC baseados no chip DS3231 uma ótima escolha para implementar projetos que necessitam realizar registros de tempo real.

Figura 3 – Módulo RTC 3231



Fonte: DS3231 High Precision... ([c2015]).

3.3 SENSOR DE FLUXO

De acordo com Banzi (2011), sensores são componentes eletrônicos que permitem que um equipamento elétrico interaja com o mundo, convertendo informações antes incapazes de serem interpretadas por ele, como temperatura, umidade e intensidade luminosa, por exemplo, em sinais elétricos. A transformação do fluxo de água em sinais elétricos capazes de serem lidos pelo Arduino é feita por meio de um sensor de fluxo.

O sensor de fluxo FS300A, Figura 4, é constituído por uma carcaça plástica, um rotor[1] e um sensor de efeito Hall[2]. O rotor é acionado com o fluxo de água que passa por ele variando sua velocidade conforme o fluxo aumenta ou diminui. A cada volta do rotor o sensor de efeito Hall emite um sinal elétrico. (G3/4" WATER..., [c2015]).

Figura 4 – Sensor de fluxo FS300A G3/4”

Fonte: G3/4" Water...([c2015]).

3.4 MÓDULO WI-FI

A transmissão de dados entre o Arduino e outros dispositivos, como computadores, smartphones e outras placas Arduino pode ser feita por diversas tecnologias, como por exemplo o Bluetooth, USB, comunicação serial e Wi-Fi.

A utilização da tecnologia Wi-Fi pode ser considerada uma das mais convenientes para se realizar a transmissão de dados em uma residência, uma vez que essa já é uma tecnologia bastante difundida e dispensa a necessidade de se utilizar fios, facilitando a implementação de projetos.

Para realizar a transmissão de dados do Arduino por meio de uma rede Wi-Fi é necessário utilizar um módulo especifico, uma vez que as placas Arduino não dispõe de tal recurso. Um dos módulos disponíveis para realizar essa interface é o ESP8266-07, Figura 5, pertencente à família de módulos ESP8266 desenvolvidos pela empresa Espressif, que estão se popularizando devido ao seu tamanho

[1] Componente mecânico formado por pás em formato de turbina que gira em torno de seu próprio eixo.
[2] Componente eletrônico capaz de converter campo magnético em sinal elétrico

reduzido, facilidade de integração com outras soluções – como o Arduino - e principalmente seu baixo custo.

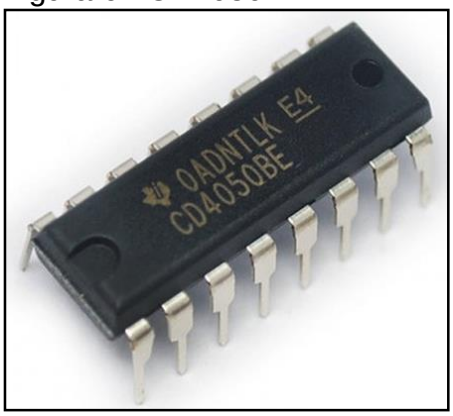
Figura 5– Módulo ESP8266-07

Fonte: ESP-07 ESP826... ([c2015]).

3.5 CD4050

O CD4050 (Figura 6) é um circuito integrado de buffer não inversor, que pode operar com tensões de alimentação de 3 a 15 volts, tendo como uma de suas utilidades a função de converter o nível de tensão de sinais lógicos, sendo possível, dessa forma, a conexão de dois dispositivos que operam com níveis de sinal lógico diferentes. (CD4049UBC . CD4050BC . HEX INVERTING BUFFER..., [c2002]).

Figura 6– CD4050



Fonte: Circuito....([c2014]).

3.6 BIBLIOTECAS

Bibliotecas são coleções de códigos que visam facilitar a conexão de sensores, displays, módulos, etc. ao Arduino. Apesar de ser possível criar as próprias bibliotecas, geralmente elas são desenvolvidas e disponibilizadas pelos próprios fabricantes dos periféricos ou por desenvolvedores independentes da comunidade do Arduino, garantindo assim, além da facilidade de uso, o seu correto funcionamento. (ARDUINO..., [c2015]).

3.7 SOFTWARE

O termo software possui diversas definições, segundo Sommerville (2011) softwares não são apenas programas para computadores como muitas pessoas associam, mas sim todo o conjunto de documentações e configurações necessárias para que um programa opere de forma correta. Ainda nessa linha de definição, Goetsch (2004, p. 34, grifo do autor) cita:

> *Software* é o termo para os componentes não mecânicos de um sistema de computador. Tais componentes incluem programas de computador, documentação desses programas e os vários tipos de manuais e referências técnicas que acompanham o sistema. No entanto, quando a maioria das pessoas usa o termo *software*, estão se referindo estritamente a programas de computador. Um programa de computador é um conjunto de instruções especialmente codificado que orienta o computador na realização de todas as operações.

Já Agarwal (2010, p. 4) define como sendo “[...] um conjunto de instruções utilizadas para obter entradas e manipulá-las para produzir o resultado desejado em termos de funções e de desempenho, tal como determinado pelo utilizador do software.”

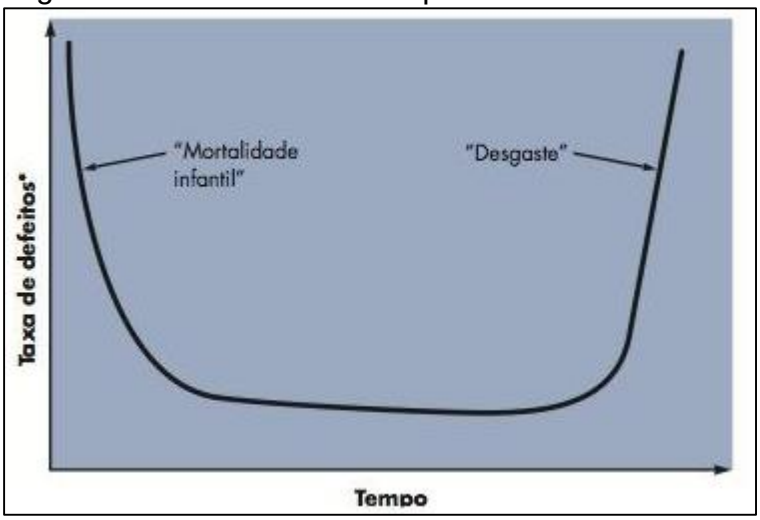
Segundo Pressman (2011, p. 29), “Software de computador é o produto que profissionais de software desenvolvem e ao qual dão suporte no longo prazo. Abrange programas executáveis em um computador de qualquer porte ou arquitetura [...].”

Pressman (2011) complementa a definição de software citando que o mesmo consiste em: Instruções, que fornecem características, funções e desempenho

desejado quando executadas; estruturas de dados que permitem a manipulação de informações de forma adequada pelos programas; e informações que descrevem a operação e o uso dos programas.

Outro ponto interessante levantado por Pressman (2011) sobre o software é sua diferença em relação aos elementos de hardware em função do tempo e o índice de defeitos. De acordo com ele, o hardware geralmente pode apresentar um alto índice de defeitos durante o início de sua vida, comumente recorrente a falhas no projeto ou na fabricação e posteriormente no fim de sua vida, uma vez que ele sofre desgastes cumulativos com o passar do tempo devido às intempéries ambientais como a poeira, vibração e a temperatura, por exemplo. A Figura 7 representa a curva de defeitos do hardware.

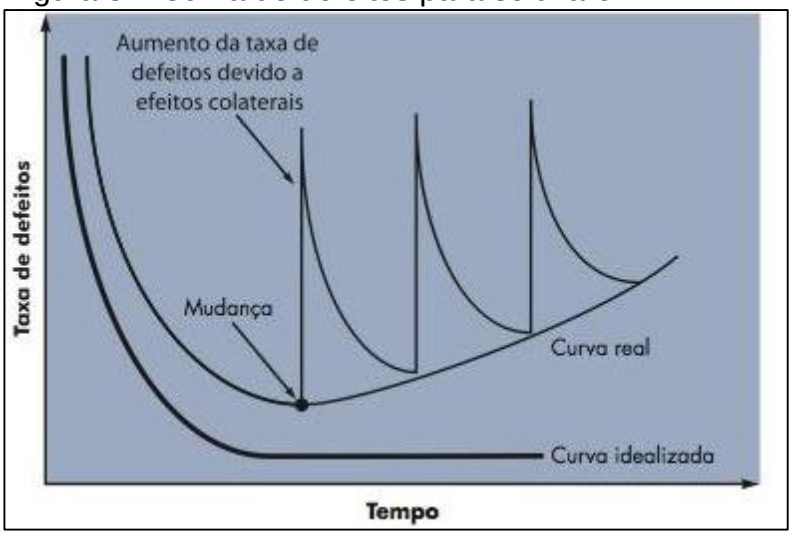
Figura 7 – Curva de defeitos para hardware



Fonte: Pressman (2011).

Já o software não se desgasta, uma vez que não está suscetível aos males ambientais e, portanto, poderia assumir a forma da “curva idealizada” representada na Figura 8. Mas devido às diversas mudanças que ocorrem durante a sua vida, quando é necessário adequar regras de negócios ou adicionar novas funcionalidades, por exemplo, novos defeitos surgem conforme representado pela “curva real” na Figura 8, e com isso conclui-se que apesar de não se desgastar, o software se deteriora. (PRESSMAN, 2011).

Figura 8 – Curva de defeitos para software



Fonte: Pressman (2011).

3.8 ENGENHARIA DE SOFTWARE

Hirama (2011) exemplifica a importância dos softwares afirmando que problemas decorrentes dos mesmos podem gerar desde grandes perdas a um cliente, devido a uma falha ocorrida em uma transação financeira, a até mesmo um acidente nuclear ocasionado por uma falha no sistema de ativação de um motor reserva em uma usina. Devido a essa importância e complexidade, um software deve ser cuidadosamente projetado, papel esse realizado pela engenharia de software, que surgiu para tratá-lo como um produto de engenharia a fim de fazer frente a essas necessidades.

> A Engenharia de Software abrange ferramentas de apoio para as atividades, métodos para orientar a realização das atividades, processo para definir as atividades e os produtos e a qualidade de processo e de produto de software. Dessa maneira o desenvolvimento de software pode obter produtos com qualidade e produtividade. (HIRAMA, 2011, p. 2).

Em seu livro, Wazlawick (2013) afirma que a engenharia de software não é algo simples de conceituar e praticar apesar de sua importância. O autor ainda cita a existência de diversas definições, porém o mesmo define como sendo um “processo de estudar, criar e otimizar os processos de trabalho para os desenvolvedores de software.”. (WAZLAWICK, 2013, p. 4).

De acordo com o Institute of Electrical and Electronics Engineers (IEEE) (1993 citado por Pressman, 2011, p. 39), a engenharia de software é: “(1) A aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software; isto é, a aplicação de engenharia ao software. (2) O estudo de abordagens como definido em (1).”

Porém, Pressman (2011) afirma que uma abordagem “sistemática, disciplinada e quantificável” conforme apresentado pela IEEE, pode não ser cabível para todas as equipes de desenvolvimento e, adaptabilidade e agilidade são características que também precisam estar presentes na engenharia de software. Além disso, é afirmado que o foco de qualquer abordagem de engenharia, não somente de software, deve ser no comprometimento organizacional com a qualidade.

Para Sommervile (2011, p. 5), a engenharia de software é “uma disciplina de engenharia relacionada com todos os aspectos da produção de software, desde os estágios iniciais de especificações do sistema até sua manutenção, depois que este entrar em operação.”

Frente aos conceitos apresentados, a engenharia de software é necessária quando se faz imperativo a qualidade de um produto de software como um todo, respeitando os prazos e orçamentos durante o desenvolvimento e satisfazendo os requisitos para o qual foi projetado quando o mesmo entrar em produção.

3.9 MODELAGEM DE SOFTWARE

Na engenharia de software as atribuições de modelos ajudam na compreensão do que será desenvolvido, podendo-se trabalhar com duas classes de modelos: de requisito, que ajudam a descrever o comportamento do software; e de projeto, que ajudam a representar as características importantes do software. (PRESSMAN, 2011).

> Um modelo de software captura uma visão de um sistema físico, é uma abstração do sistema com um certo propósito, como descrever aspectos estruturais ou comportamentais do software. Esse propósito determina o que deve ser incluído no modelo e o que é considerado irrelevante. Assim um modelo descreve completamente aqueles aspectos do sistema físico que são relevantes ao propósito do modelo, no nível apropriado de detalhe. (GUEDES, 2011, p. 21).

De acordo com Silva (2013), o objetivo da modelagem de software é auxiliar o mapeamento prévio e as definições do comportamento de determinados componentes do sistema. Ela permite refinar os detalhes do sistema, podendo-se utilizar para isso diversos tipos de diagramas, como o diagrama de classes e o diagrama de caso de uso.

3.9.1 Linguagem de modelagem unificada – UML

A UML – Unified Modeling Language ou Linguagem de Modelagem Unificada – é uma linguagem de modelagem de softwares baseados no paradigma de orientação a objetos, adotada internacionalmente como linguagem de modelagem padrão pelas empresas de engenharias de software. (GUEDES, 2011)

Guedes (2011) explica que a UML surgiu da união dos três métodos de modelagem orientada a objetos mais populares entre os profissionais da área até meados da década de 1990, sendo eles: o método de Booch, o método OMT (Object Modeling Technique) de Jacobson, e o método OOSE (Object-Oriented Software Engineering) de Rumbaugh.

> [...] a UML não é uma linguagem de programação, e sim uma linguagem de modelagem, uma notação, cujo objetivo é auxiliar os engenheiros de software a definirem as características do sistema, tais como seus requisitos, seu comportamento, sua estrutura lógica, a dinâmica de seus processos e até mesmo suas necessidades físicas em relação ao equipamento sobre o qual o sistema deverá ser implantado. (GUEDES, 2011, p. 19).

De acordo com Martinez ([c2015]), a UML auxilia a visualização das comunicações entre os objetos e permite que os desenvolvedores analisem os produtos de software em diagramas padronizados, sendo muito utilizado para a criação de modelos de sistemas.

3.9.1.1 Diagrama de caso de uso

Segundo Ribeiro (2013), o diagrama de caso de uso visa documentar o funcionamento do sistema do ponto de vista do usuário, descrevendo as principais

interações entre eles (sistema e usuário) e entre as suas próprias funcionalidades. Nesse tipo de diagrama, não é feito uma abordagem detalhada dos aspectos técnicos do software.

O autor ainda cita uma lista com os quatro componentes básicos do diagrama de caso de uso:

a) cenário: sequência de eventos que acontecem quando um usuário interage com o sistema;
b) ator: usuário do sistema, ou melhor, um tipo de usuário;
c) use case: é uma tarefa ou uma funcionalidade realizada pelo ator (usuário);
d) comunicação: é o que liga um ator com um caso de uso.

Pressman (2011) cita que, apesar de possuir limitações, assim como em todas as modelagens, a modelagem baseada em cenários é apropriada para grande parte das situações existentes e, quando desenvolvida corretamente, pode gerar grandes benefícios como ferramenta de modelagem.

3.9.1.2 Diagrama de classes

O diagrama de classes pode ser considerado um dos mais importantes e utilizados da UML – linguagem de modelagem que permite representar um sistema de forma padronizada – e sua principal função é permitir a visualização das classes bem como seus métodos correspondentes, atributos e relacionamentos. Ela também demonstra como as classes transmitem e dividem as informações entre si, apresentando uma visão estática de como estão organizadas definindo apenas as suas estruturas lógicas. (SILVA, 2013).

Segundo Pressman (2011), o diagrama de classes é utilizado para representar os objetos, suas características e as formas como eles se interagem no sistema, sendo uma ferramenta para auxiliar na elaboração do projeto e no desenvolvimento do banco de dados e das classes de objetos de um sistema.

3.9.1.3 Diagrama de atividades

O diagrama de atividades definido pela UML visa complementar o caso de uso por meio de representações gráficas do fluxo de interações de um determinado cenário, demonstrando as ações e decisões que ocorrem quando uma dada função é executada. (PRESSMAN, 2011).

> O diagrama de atividade preocupa-se em descrever os passos a serem percorridos para a conclusão de uma atividade específica, podendo esta ser representada por um método com certo grau de complexidade, um algoritmo, ou mesmo por um processo completo. O diagrama de atividade concentra-se na representação do fluxo de controle de uma atividade. (GUEDES, 2011, p. 36).

Segundo Pressman (2011), a representação do diagrama de atividades é similar ao fluxograma – diagrama com a finalidade de representar processos ou fluxos – utilizando retângulos com cantos arredondados para representar funções, setas para indicar o fluxo através do sistema, losangos representando as tomadas de decisões e linhas horizontais cheias indicando atividades paralelas em andamento.

3.10 TESTE DE SOFTWARE

De acordo com Hetzel (1988), teste de software é um processo para avaliação dos atributos ou capacidades de um determinado programa ou sistema bem como determinar a capacidade deste em atender os requisitos para o qual foi desenvolvido.

Para Myers (2004) testar um software é um processo realizado para provar que os erros não estão presentes e que o programa executa suas funções corretamente e faz exatamente aquilo para o que foi projetado.

> [...] um processo, ou uma série de processos, desenvolvido para assegurar que o código de computador faz o que foi projetado para fazer e que não faz nada não intencional. O *software* deve ser previsível e consistente, não oferecendo surpresas aos usuários. (MYERS, 2004, p. 8).

Apesar das definições acima demonstrarem que o objetivo dos testes de softwares é comprovar a não existência de erros e a capacidade de atender os seus objetivos, os testes são executados de forma a induzir os programas a falharem para que então eles possam ser corrigidos. Myers (2004, p. 11) cita: “testar é o processo de executar um programa com a intenção de encontrar erros”.

A contradição entre o objetivo de provar que um software não possui erros com a ideia de induzir erros pode gerar confusão no momento de realizar os testes, de acordo com Mosley (1993), quando o objetivo de um teste é demonstrar que um programa não possui erros, o nosso subconsciente atua buscando este objetivo trabalhando com entradas de dados que possuem baixa probabilidade de provocar erros no software, quando deveria fazer justamente o contrário.

> [...] se nosso objetivo é mostrar que o programa possui erros, nós selecionamos os testes que possuem uma maior probabilidade de encontrarmos erros, focando nas partes mais críticas do programa com o objetivo de encontrar ainda mais erros. (MOSLEY, 1993, p. 68).

De acordo com Halili (2008), algumas empresas possuem a cultura de não realizar a etapa de testes de software por possuírem custos elevados, atrasar a entrega do produto por demandar muito tempo e ainda acreditarem que tal processo não ajuda com o desenvolvimento, podendo gerar intrigas entre as equipes de programação e de testes. Porém, lançar um produto de software pulando esta etapa certamente acarretará em prejuízos ainda maiores, já que é inevitável desenvolver um sistema com falhas. Segundo Humphrey (1989), uma pesquisa realizada com 13 mil programadores indicou que cada profissional gera de 100 a 150 erros a cada mil linhas de códigos escrita.

3.10.1 Tipos de teste

Para se testar um software é possível aplicar diversas técnicas já existentes, sendo algumas bem conhecidas e aceitas e outras nem tanto. Entre as mais conhecidas e utilizadas então: Teste Funcional; Teste Não Funcional; e Teste de Regressão.

3.10.1.1 Teste funcional

Teste funcional ou teste baseado em requisitos ou especificações, como também é conhecido, tem como objetivo identificar os erros do software por meio da utilização do mesmo, sem o acesso a seu código-fonte, tendo conhecimento apenas da maneira como o sistema deverá responder. (JESUS, 2013).

De acordo com Desikan (2006), esse tipo de teste ajuda a averiguar o comportamento do sistema, testando suas características e funcionalidades, tendo como resultado apenas o veredicto de que atende ou não aos requisitos.

> [...] seu objetivo é ser completamente indiferente sobre o comportamento interno e da estrutura do programa. Em vez disso, se concentrar em encontrar circunstâncias em que o programa não se comporta de acordo com as suas especificações. (Myers, 2004, p. 13).

Pressman (2001, p. 460) aponta cinco categorias de erros que podem ser encontradas pelo teste Black-box (outra denominação para o teste funcional):

> [...] o teste black-box tenta encontrar erros nas seguintes categorias: (1) funções incorretas, (2) erros de interface, (3) erros nas estruturas de dados ou acesso à bases de dados externas, (4) erros de comportamento ou desempenho, e (5) erros de inicialização e encerramento.

Tian (2005, p. 75), aponta que “o teste funcional verifica o correto manuseio das funções externas fornecidas pelo *software*, através da observação do comportamento externo do programa durante sua execução”.

Como esse teste de software baseia-se fundamentalmente nas descrições fornecidas sobre o seu funcionamento, estas devem possuir a maior quantidade de detalhes e informações possíveis, citando os tipos de dados que cada campo pode aceitar e os resultados esperados, por exemplo.

Segundo Veenendaal (2008), as técnicas aplicadas durante o teste funcional geralmente são baseadas em requisitos, embora as técnicas baseadas em conhecimentos também possam ser utilizadas, ou seja, além das informações fornecidas nas descrições do sistema, situações comuns ao dia a dia podem ser aplicadas a fim de simular a inserção de dados incorretos que os usuários do sistema podem vir a inserir erroneamente.

Jesus (2013, p. 37) cita como situações comuns do dia a dia:

> (1) intervalo de datas, onde a data inicial deve ser menor ou igual à data final, (2) preenchimento de campos numéricos com quantidade de caracteres insuficiente, como por exemplo, telefone ou CEP (Código de Endereçamento Postal), (3) validação de CPF (Cadastro de Pessoa Física), para evitar fraudes ou erros de digitação, (4) fornecer tipo de dado incompatível, por exemplo, digitar letras em campos numéricos, (5) campos de preenchimento obrigatório preenchidos com espaços, entre outras.

Uma das principais vantagens do teste funcional é a sua independência com relação ao testador e a equipe de desenvolvimento, uma vez que o teste é feito seguindo as perspectivas do usuário final, e o responsável por realizar os testes não precisa ter os mesmos conhecimentos dos desenvolvedores. (AGARWAL, 2009).

3.10.1.2 Teste não funcional

Testes não funcionais são destinados a verificar os fatores de qualidade do software, tais como confiabilidade, escalabilidade, desempenho e etc., seu foco está em qualificar o sistema e não em encontrar defeitos. A sua aplicação é considerada complexa devido ao grande volume de dados envolvidos e seus resultados devem ser documentados em termos qualitativos e quantitativos. (DESIKAN, 2006).

> Testes não funcionais referem-se a testes que avaliam os atributos do software, tais como a sua velocidade de execução, o tempo de inicialização e encerramento do software, segurança, e às vezes o espaço de memória necessário quando o aplicativo está em execução. (JONES; BONSIGNOUR, 2011, p. 598).

De acordo com Jones e Bonsignour (2011), a principal diferença entre o teste funcional e o não funcional é que o teste não funcional está relacionado diretamente com o desempenho do sistema no computador, enquanto o teste funcional diz respeito a seus recursos.

Segundo Jesus (2013, p.38) “os tipos de testes não funcionais mais conhecidos e utilizados são testes de desempenho, testes de carga, testes de estresse, testes de usabilidade, teste de manutenção, testes de confiabilidade e testes de portabilidade”.

3.10.1.3 Teste de regressão

Sempre que um software é modificado e é submetido a novos testes, estes são chamados de teste de regressão e é considerada parte essencial de todo processo de desenvolvimento de software. (AMMANN; OFFUT, 2008).

Segundo B'Far (2005, p. 799), o "teste de regressão é absolutamente crucial para assegurar que novos erros não sejam introduzidos em módulos que já foram testados e aprovados".

De acordo com Jesus (2013), alterações realizadas em uma determinada parte do sistema podem causar efeitos inesperados em outra, aparentemente não relacionadas entres si e que já havia sido liberado em testes anteriores. O teste de regressão prevê essa possibilidade e visa justamente identificar esses novos problemas antes do produto ser disponibilizado.

3.11 BANCO DE DADOS

De acordo com Korth (1994 citado por Rezende, c2015), um banco de dados "é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico".

Setzer e Silva (2005) explica que a expressão Banco de Dados originou-se do termo em inglês Databanks que posteriormente foi substituída por Databases (Base de Dados).

Segundo Date (2004, p. 10), "Um banco de dados é uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa.".

3.12 MYSQL

Segundo pesquisa realizada em setembro de 2015 pelo site db-engines (DB-ENGINES Ranking, [c2015]), o MySQL é o segundo sistema de gerenciamento de banco de dados mais popular no mundo.

> O MySQL é um servidor e gerenciador de banco de dados (SGBD) relacional, de licença dupla (sendo uma delas de software livre),

> projetado inicialmente para trabalhar com aplicações de pequeno e médio porte, mas hoje atendendo a aplicações de grande porte e com mais vantagens do que seus concorrentes. Possui todas as características que um banco de dados de grande porte precisa, sendo reconhecido por algumas entidades como o banco de dados open source com maior capacidade para concorrer com programas similares de código fechado, tais como SQL Server (da Microsoft) e Oracle. (MILANI, 2006, p. 22).

De acordo com Maxfield (2002), o MySQL surgiu apenas como uma ferramenta para atender as necessidades de seus desenvolvedores – David Axmark, Allan Larsson e Michel “Monty” Widenius – quando precisaram de uma interface para trabalhar com as rotinas ISAM utilizadas em suas aplicações, mas com os excelentes resultados obtidos, seus desenvolvedores deram continuidade ao projeto, hoje mundialmente conhecido.

Segundo Milani (2006), o MySQL além de ser extremamente rápido e confiável, é suportado pela maioria dos sistemas operacionais existentes no mercado, sendo indicado para ser utilizado em sistema de pequeno, médio e grande porte em todas as áreas de negócio.

3.13 HTML

De acordo com Rocha (1996), o HTML (Hiper Text Markup Language) não é uma linguagem de programação e sim uma linguagem de marcação de texto utilizada para a publicação de hipertextos na World Wide Web (WWW) de forma que sejam preservadas as suas estruturas nas diversas plataformas onde podem ser apresentados. As marcações que definem a forma como o texto será mostrado são feitas por meio da inclusão de pequenos códigos denominados Tags.

> Hipertexto é a tecnologia que permite marcar palavras ou imagens para que elas sirvam de ligação para outros documentos. Isso permite que palavras ou imagens marcadas – geralmente sublinhadas e de cor diferente – mantenham vínculo com outro documento ou página da Web, ou seja, ao clicar com o mouse sobre elas um novo documento será demonstrado pelo *browser*. Dessa maneira, o hipertexto torna uma página da Web mais interativa permitindo que o usuário “navegue” pelos documentos e, ao mesmo tempo, ele se transforma na base da organização da Web, dispensando a forma antiga de organização baseada em menus. (FURGERI, 1999, p. 34).

3.14 PHP

O PHP é uma linguagem de programação web utilizada por milhões de sites no mundo inteiro. A sua principal diferença em relação a outras linguagens web, é a sua capacidade de transformar totalmente sites que possuem apenas páginas estáticas. (NIEDERAUER, 2011).

> PHP é uma linguagem de script embutida no HTML. Muito da sua sintaxe é emprestada de C, Java e Perl com algumas características específicas do PHP juntas. O objetivo da linguagem é permitir que desenvolvedores web escrevam páginas geradas dinamicamente de forma rápida.(INFORMAÇÕES Gerais, [c2015]).

De acordo com Niederauer (2011), uma das grandes vantagens do PHP é o fato dele ser gratuito e possuir código-fonte aberto, com toda a sua documentação disponível em seu site oficial (www.php.net). Outra característica importante explicada pelo autor é o fato do PHP ser baseado no servidor, que, diferentemente das linguagens baseadas no cliente - onde todo o processamento é feito no próprio computador que faz a solicitação - o processamento do código PHP fica por conta do servidor, apresentando apenas os resultados ao cliente.

4 TRABALHOS CORRELATOS

Com o desenvolvimento do levantamento bibliográfico, foi possível observar um trabalho correlacionado que também serviu para auxiliar no desenvolvimento do presente.

Rocha et al. (2014), desenvolveram o trabalho intitulado “Sistema De Monitoramento De Consumo De Água Doméstico Com A Utilização De Um Hidrômetro Digital”, onde é apresentado o projeto de um hidrômetro digital desenvolvido com Arduino para o monitorar o consumo de água em uma residência.

O sistema apresentado realiza a leitura do consumo de água por meio de um sensor de fluxo e armazena os dados em uma memória EEPROM para posterior transferência, por meio da tecnologia Bluetooth, para um dispositivo equipado com o sistema operacional Android, onde é executada uma aplicação para apresentar os dados coletados.

O seu principal diferencial com relação ao presente trabalho é o emprego da tecnologia Bluetooth, onde os próprios autores recomendam o emprego do Wi-Fi para trabalhos futuros, e o fato de não sincronizar e concentrar os dados automaticamente em um único banco de dados para que seja possível acessar por diversos dispositivos.

5 METODOLOGIA

5.1 TIPO DE PESQUISA

Assim como colocado por Gil (2002), a fim de proporcionar uma maior familiaridade com o assunto proposto pelo presente trabalho, inicialmente foi realizada uma pesquisa do tipo exploratória, podendo desta forma, o pesquisador aproximar-se de um assunto ainda pouco conhecido por ele, sanado dúvidas e problemas que inevitavelmente surgiriam durante o desenvolvimento do trabalho.

5.2 DESENVOLVIMENTO DO PROTÓTIPO

Após a realização do levantamento bibliográfico, foi iniciado o desenvolvimento do protótipo responsável pela aquisição dos dados referente ao consumo de água em um determinado ponto de uma residência. Para a sua montagem foram utilizados os componentes apresentados na Figura 9.

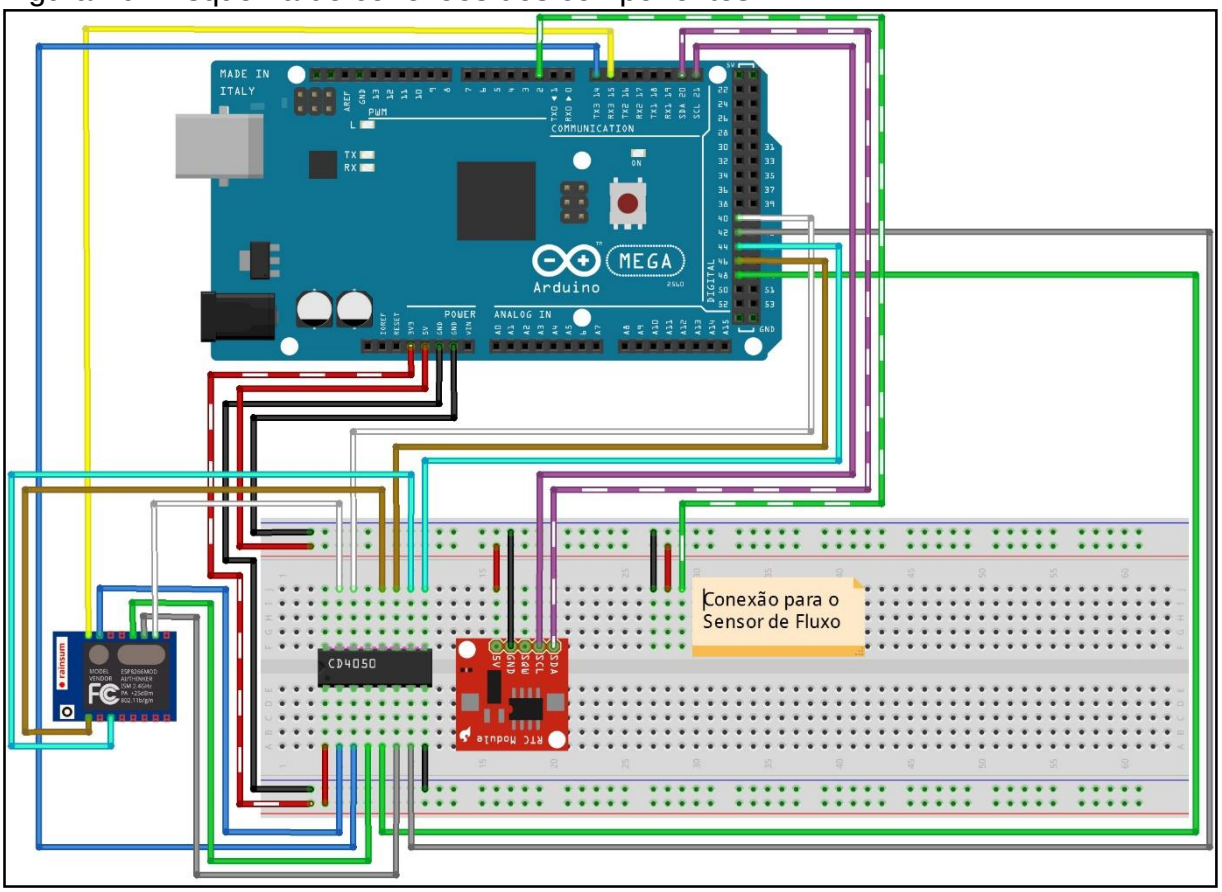
Figura 9- Lista de componentes do protótipo

Item	Descrição	Quantidade
1	Arduino Mega 2560	1
2	Módulo ESP8266-07	1
3	Módulo RTC DS-3231	1
4	Buffer CD4050BE	1
5	Matriz de Contatos Protoboard de 840 pontos	1
6	Sensor de Fluxo FS300A ¾”	1

Fonte: Elaborada pelo autor.

Inicialmente foi desenvolvido um esquema de conexões dos componentes (Figura 10), objetivando facilitar a montagem, evitar que algum componente possa apresentar mau funcionamento ou até mesmo venha a ser danificado devido a ligações incorretas e promover um meio para ilustrar, de modo claro, a forma como os componentes são interligados, uma vez que a montagem de protótipos utilizando matriz de contatos do tipo protoboard tende a ser confusa para análises visuais.

Figura 10– Esquema de conexões dos componentes

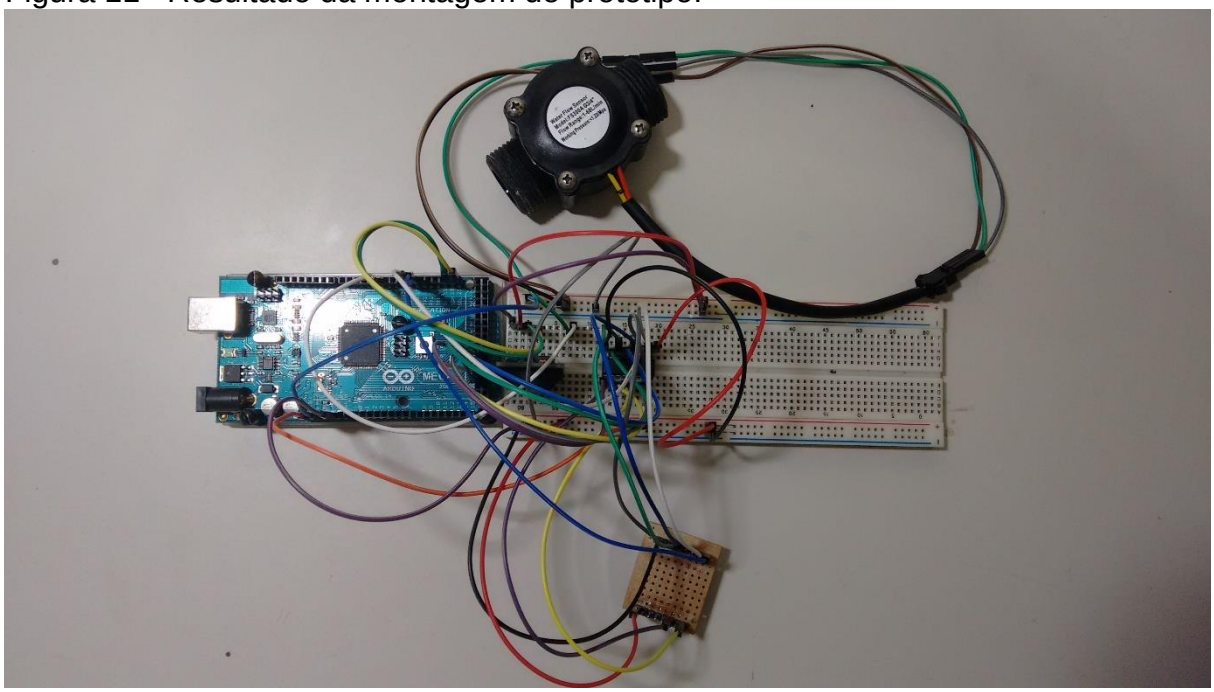


Fonte: Elaborada pelo autor.

No diagrama apresentado na Figura 10 é possível notar o uso do componente CD4050 para realizar a conexão entre o Arduino e o módulo Wi-Fi Esp8266-07 (componente representado na parte inferior esquerda), tal componente se faz necessário, pois o Arduino emite sinais lógicos com níveis de tensão em 5 volts, superior aos 3,3 volts suportado pelo módulo ESP. Ao lado direito do componente CD4050 é possível observar a representação do módulo RTC DS-3231. Outro ponto relevante no diagrama é a “Conexão para o Sensor de Fluxo”, uma vez que não foi possível representar o sensor de fluxo propriamente dito, apenas as conexões referentes ao mesmo foram demonstradas, sendo a linha verde tracejada a conexão referente aos sinais de pulsos enviados pelo sensor para o Arduino.

Após a elaboração do diagrama, deu-se início à montagem do protótipo e o resultado pode ser observado na Figura 11. As cores dos fios utilizados para realizar as conexões bem como a disposição dos componentes não foram necessariamente as mesmas apresentas pelo diagrama anterior, em face das necessidades de adequações, entretanto, todas as conexão foram seguidas.

Figura 11– Resultado da montagem do protótipo.



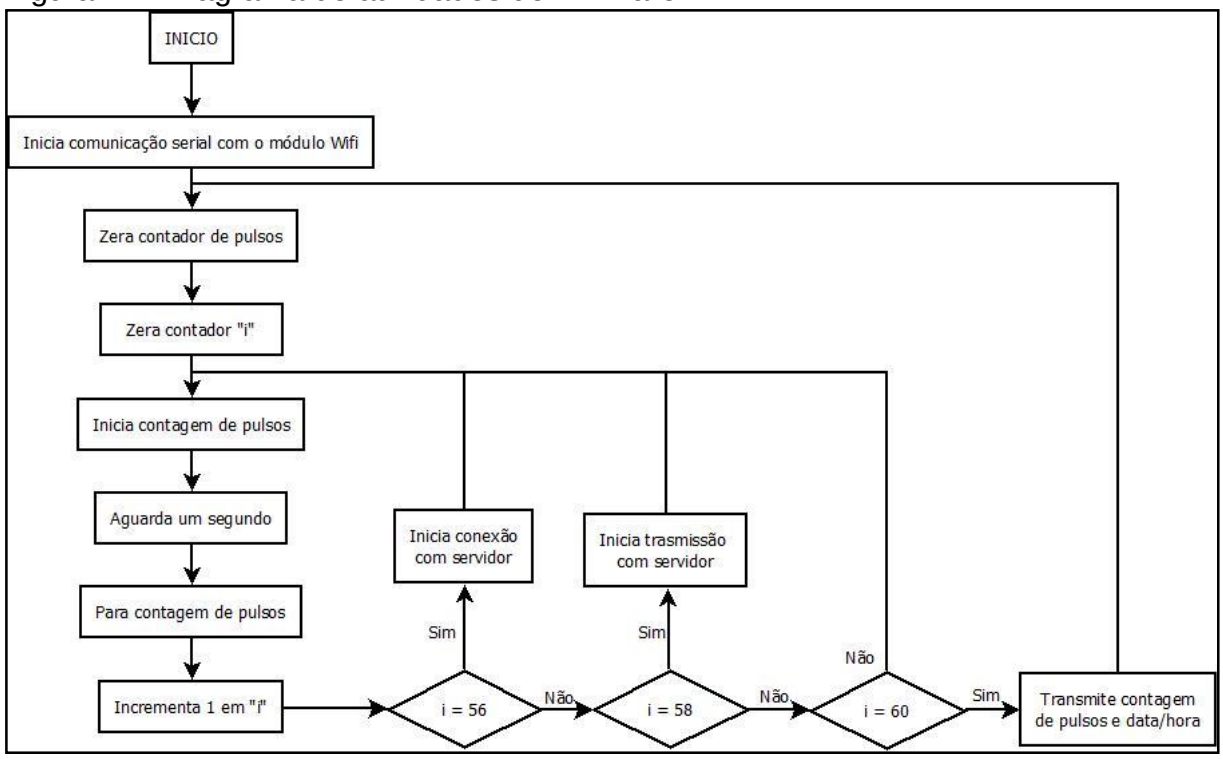
Fonte: Elaborada pelo autor.

5.3 DESENVOLVIMENTO DO FIRMWARE

Após a montagem do hardware do protótipo, deu-se início ao desenvolvimento do firmware (programa executado pelo Arduino).

Objetivando ilustrar o funcionamento do firmware e facilitar o seu desenvolvimento, o primeiro passo dessa etapa foi a elaboração de um diagrama de atividades, visto na Figura 12.

Figura 12– Diagrama de atividades do firmware



Fonte: Elaborado pelo autor.

O firmware consiste em realizar a contagem dos pulsos emitidos pelo sensor de fluxo durante o intervalo de tempo de um minuto para, então, enviar a contagem realizada ao banco de dados, juntamente com a data e hora da aquisição dos dados. Após o envio dos dados, a contagem dos pulsos é zerada e todo o processo é reiniciado, realizando os registros das informações a cada intervalo de tempo, ininterruptamente até que o protótipo seja desligado.

Como é possível observar no diagrama da Figura 12, o processo para enviar os dados ao servidor foi dividido em três etapas, sendo elas: o início da conexão com o servidor; início da transmissão e a transmissão dos dados propriamente dita. Tal manobra foi necessária devido ao tempo demandado pelo módulo ESP8266 para realizar cada uma dessas etapas, que caso fossem realizadas simultaneamente, exigiria cerca de seis segundos – conforme testes realizados – sendo que durante esse tempo o Arduino estaria parado e impossibilitado de realizar a contagem dos pulsos que eventualmente poderiam estar ocorrendo, acarretando na redução da precisão de leitura do protótipo.

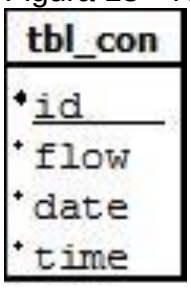
Também é possível observar que a informação referente ao consumo de água enviada ao servidor é somente a contagem dos pulsos captada pelo sensor de fluxo e não a quantidade de água propriamente dita, tal conversão será realizada

posteriormente, em tempo de execução, no software em PHP que irá gerar e apresentar os relatórios.

5.4 BANCO DE DADOS

Para o armazenamento dos dados coletados pelo protótipo, foi desenvolvido uma única tabela, conforme Figura 13, utilizando-se o MySQL por se tratar de um banco de dados gratuito e conceituado, principalmente em aplicações web.

Figura 13– Tabela tbl_con



Fonte: Elaborada pelo autor.

A tabela tbl_con (Figura 13), é formada por quatro campos, sendo eles:

a) **id**: campo numérico, auto incrementado, utilizado como índice da tabela;
b) **flow**: campo numérico, utilizado para armazenar a contagem dos pulsos realizada pelo protótipo;
c) **date**: campo utilizado para armazenar a data da coleta dos dados pelo protótipo;
d) **time**: campo utilizado para armazenar a hora da coleta dos dados pelo protótipo.

5.5 DESENVOLVIMENTO DO SOFTWARE

O desenvolvimento dos softwares necessários para a inserção e apresentação dos dados coletados pelo protótipo foram desenvolvidos

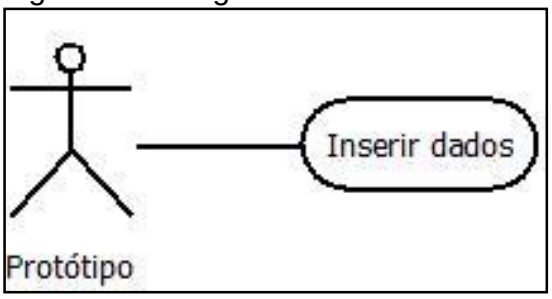
essencialmente em PHP porém, para melhor elaboração do layout, também foram utilizadas linguagens auxiliares como CSS[3] e JavaScript[4].

Os softwares foram divididos em dois módulos para melhor organização e elaboração, sendo o primeiro deles o módulo de inserção, responsável por receber os dados vindos do protótipo e inseri-los na tabela do banco de dados, e o módulo de consultas, o qual gera e apresenta os relatórios para o usuário.

5.5.1 MÓDULO DE INSERÇÃO

Como mencionado anteriormente, o módulo de inserção é o responsável por receber os dados – contagem de pulsos, data e hora – enviados pelo protótipo e inseri-los na tabela do banco de dados. Esse processo é feito de forma transparente para os usuários, sendo assim, não possui nenhuma interface gráfica e a sua única interação é com o protótipo, assim como pode ser observado no diagrama de caso de uso apresentado na Figura 14.

Figura 14– Diagrama caso de uso do módulo de inserção



Fonte: Elaborada pelo autor.

No diagrama de atividades apresentado na Figura 15 são apresentadas todas as etapas realizadas pelo software do módulo de inserção.

[3] Linguagem de folha de estilos utilizada para definir a apresentação dos elementos utilizados na linguagem HTML, por exemplo.

[4] Linguagem de scripts utilizada principalmente para tornar páginas Web mais dinâmicas.

Figura 15– Diagrama de atividades do módulo de inserção

Fonte: Elaborada pelo autor.

5.5.2 MÓDULO DE CONSULTAS

O módulo de consultas é o software responsável por realizar as consultas na tabela do banco de dados, executar os cálculos necessários para converter os pulsos registrados em litros ou metros cúbicos de água e apresentar esses dados ao usuário por meio de gráfico.

Visando tornar o uso do software uma tarefa descomplicada e acessível para todos os tipos de usuários de computador, optou-se por reduzir o máximo possível de interações entre o software e o usuário, tendo assim, apenas uma tela onde são apresentados, de forma automática, todos os dados relevantes, cabendo ao usuário apenas selecionar as datas de interesse.

Os dados apresentados nessa tela são referentes ao consumo acumulado, o consumo do último mês[5] e a média mensal, além de um gráfico com o comparativo

[5] Tendo como referência a data atual do momento da execução do servidor que hospeda a aplicação.

mês a mês de dois anos diferentes, um gráfico para detalhar o consumo mensal e outro para detalhar o consumo diário.

6 TESTES E RESULTADOS

Nesse capítulo são apresentados os testes e os resultados obtidos durante o desenvolvimento do sistema de monitoramento de consumo de água.

6.1 TESTES DO PROTÓTIPO E MÓDULO DE INSERÇÃO

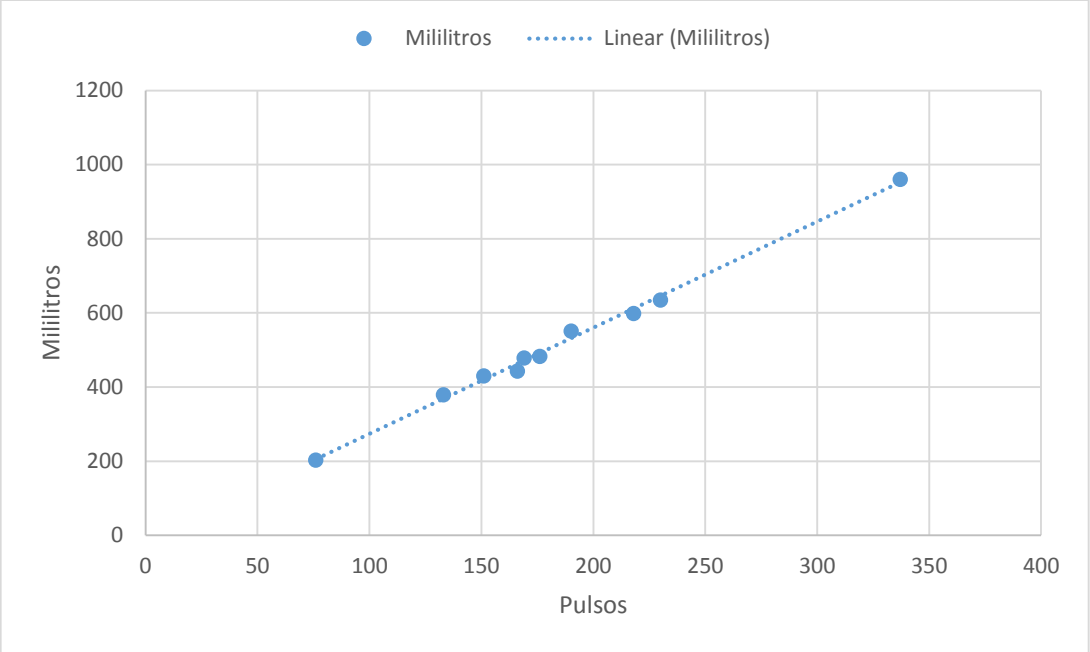
Essa etapa teve como objetivo realizar os testes de funcionamento do protótipo e do módulo de inserção do software de forma simultânea, uma vez que ambos trabalham em conjunto. Os testes realizados também visavam determinar o coeficiente necessário para converter os pulsos enviados pelo sensor de fluxo em litros.

A realização dos testes deu-se com os seguintes passos:

a) o módulo de inserção e o banco de dados foi disponibilizado em um servidor web rodando em um computador na rede local;
b) o módulo ESP8266 foi configurado para se conectar na mesma rede local do servidor web.
c) o sensor de fluxo do protótipo foi conectado em uma torneira residencial de forma que a água por ele passada fosse armazenada em um recipiente;
d) o protótipo foi ligado;
e) abriu-se a torneira durante um intervalo de tempo aleatório;
f) após a torneira ser fechada foi verificada a quantidade de água armazenada no recipiente com o uso de uma proveta graduada em intervalos de 1 ml e a soma dos pulsos registrados no banco de dados durante esse período;
g) o recipiente foi esvaziado e o processo recomeçado a partir do passo “e”.

O processo foi realizado dez vezes e os dados obtidos foram sintetizados e apresentados no gráfico da Figura 16, onde é possível observar que o número de pulsos está diretamente proporcional ao volume de água e muito próximo da linha de tendência, representando uma baixa oscilação da leitura do sensor.

Figura 16– Gráfico do volume de água em função da soma dos pulsos



Fonte: Elaborada pelo autor.

Com esses dados foi calculado o coeficiente referente a cada leitura, dividindo-se a soma dos pulsos pelo volume obtido multiplicado por 1000[6]. Posteriormente foi calculada a média aritmética dos resultados encontrados, afim de determinar o coeficiente que será adotado para a conversão dos pulsos em litros no módulo de consultas do software, e o valor encontrado foi 360, o que representa, uma variação de aproximadamente 3%[7] apenas.

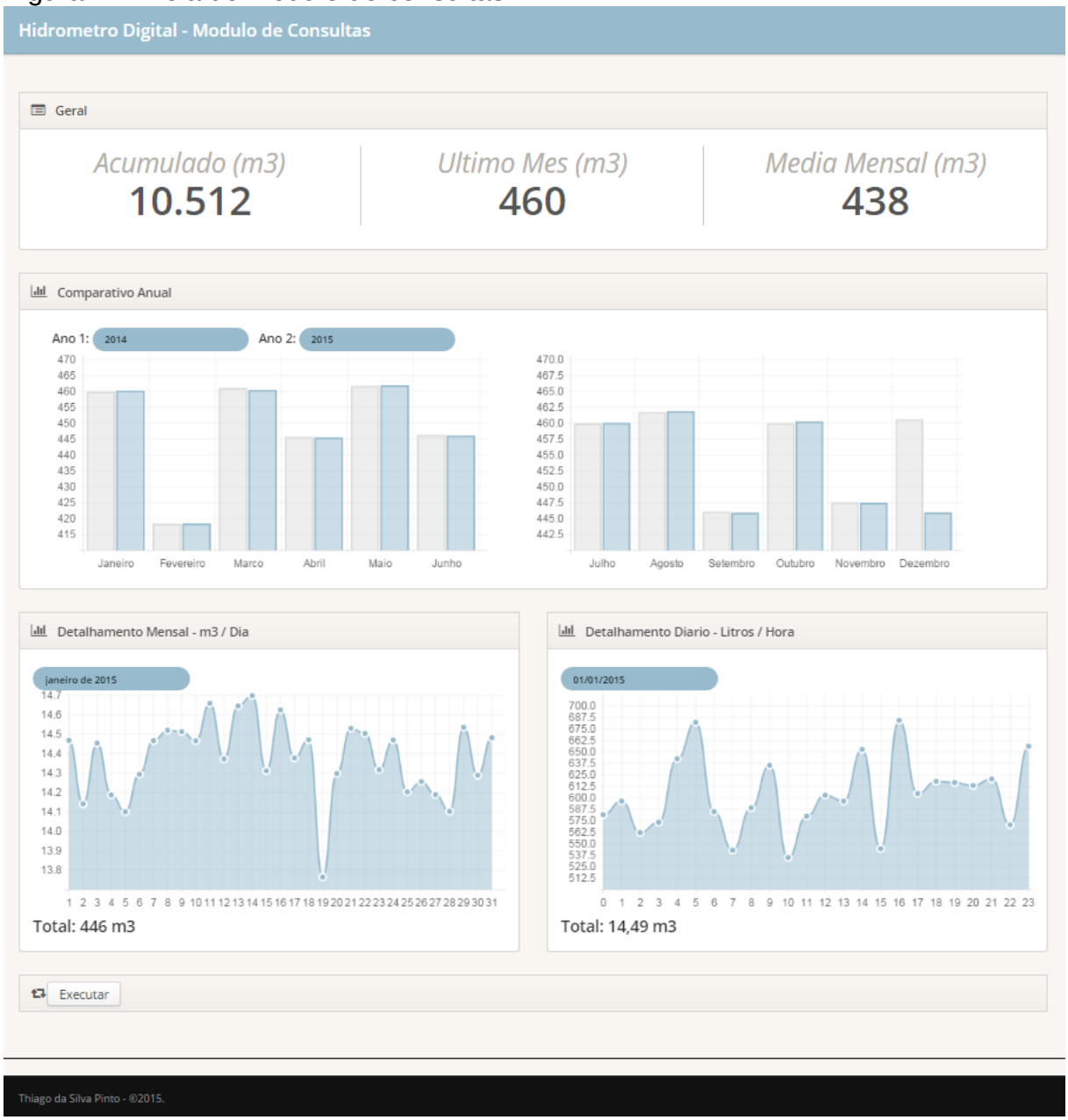
6.2 TESTES DO MÓDULO DE CONSULTAS

Para a realização dos testes no módulo de consulta, inicialmente foi necessário popular a tabela com dados de consumo gerados de forma aleatória, simulando a utilização do protótipo por um período de 24 meses, uma vez que tal módulo visa apresentar dados comparativos de longo prazo como, por exemplo, o gráfico de anual, tornando a aquisição de dados reais para fins de testes inviável.

[6] $\text{Coeficiente} = \frac{\sum Pulsos}{Volume(ml)} * 1000$

[7] Coeficiente de variação encontrado com base no desvio padrão dos resultados obtidos na equação anterior.

Figura 17– Tela do módulo de consultas



Fonte: Elaborada pelo autor.

A Figura 17 representa a tela gerada pelo módulo de consulta após a tabela ter recebidos os dados aleatórios referentes aos anos de 2014 e 2015.

Conforme pode ser observado, a parte superior da tela, na área denominada “Geral”, apresenta os dados referentes ao consumo acumulado, o consumo do último mês e a média mensal.

Na área central da tela, denominada “Consumo Mensal”, são apresentados dois gráficos (primeiro e segundo semestre) com o consumo mensal, em metros

cúbicos, de dois anos distintos escolhidos pelo usuário, afim de se comparar as diferenças de consumo ocorridas em um mesmo período em anos diferentes.

O gráfico apresentado na área denominada “Detalhamento Mensal” destina-se a apresentar o consumo detalhado de um determinado mês, informando a quantidade de água, em metros cúbicos, consumidos por dia e o total daquele mês.

O último gráfico, apresentado na área denominada “Detalhamento Diário”, apresenta o consumo de água, em litros, ocorrido durante um determinado dia.

7 CONSIDERAÇÕES FINAIS

Após o desenvolvimento e a realização dos testes do protótipo e do software, foi possível observar que o sensor de fluxo FS300A, apesar de seu baixo custo, se mostrou bastante preciso para a finalidade a qual foi aplicado, desde que se determine o valor do coeficiente correto para transformar o número de pulsos em litros.

Também foi observado que o software atingiu seu objetivo de apresentar informações de forma precisa e detalhada sobre o consumo de água em uma residência, apenas coletando dados sobre o fluxo de água. Entretanto, para se verificar a eficácia do sistema no auxílio da redução do consumo de água, seria necessário realizar um estudo em residências reais, tomando as ações cabíveis, para diminuir os consumo, frente aos dados apresentados, uma vez que o sistema faz apenas o monitoramento.

Como propostas para trabalhos futuros, tem-se a possibilidade de implementar recursos de inteligência artificial ao software, para que seja possível identificar variações no padrão de consumo, identificando possíveis vazamentos de forma automatizada, bem como a implementação de sistemas de alertas via e-mail ou mensagem SMS.

Também é possível o desenvolvimento de um aplicativo específico para dispositivos móveis, equipados com Android, por exemplo, visando facilitar ainda mais o monitoramento do consumo de água na residência.

REFERÊNCIAS

AGARWAL, B. B.; TAYAL, S. P.; GUPTA, M. **Software Engineering & Testing**. Sudbury: Jones and Bartlett Publishers, 2009.

ARDUINO Mega 2560. **Arduino.cc**, [c2015]. Disponível em: <http://arduino.cc/en/Main/arduinoBoardMega2560>. Acesso em: 27jul. 2015.

ATÉ 2030 planeta pode enfrentar déficit de água de até 40%, alerta relatório da ONU.**Nacoesunidas.org,** [c2015]. Disponível em:<http://nacoesunidas.org/ate-2030-planeta-pode-enfrentar-deficit-de-agua-de-ate-40-alerta-relatorio-da-onu/>. Acesso em: 25 ago. 2015.

B'FAR, R. **Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML**. Cambridge: Cambridge University Press, 2005.

BANZI, M. **Primeiros passos com Arduino**. São Paulo: Novatec, 2011.

BARIFOUSE, R.**Maior crise hídrica de São Paulo expõe lentidão do governo e sistema frágil.** BBC.com, 2014. Disponível em: <http://www.bbc.com/portuguese/noticias/2014/03/140321_seca_saopaulo_rb >. Acesso em: 25 ago. 2015.

CD4049UBC - CD4050BC HEX INVERTING BUFFER - Hex Non-Inverting Buffer. **Fairchildsemi.com**, [c2002]. Disponível em: <https://www.fairchildsemi.com/datasheets/CD/CD4049UBC.pdf >. Acesso em: 09 nov. 2015.

CIRCUITO integrado CD4050 –Buffer. **Baudaeletronica**, [c2014]. Disponível em: <https://www.baudaeletronica.com.br/circuito-integrado-cd4050-buffer.html>. Acesso em: 09nov. 2015.

DATE, C. J. **Introdução a sistemas de bancos de dados**. 8 ed. Rio de Janeiro: Elsevier, 2004.

DB-ENGINES Ranking. **db-engines.com**, [c2015]. Disponível em: < http://db-engines.com/en/ranking>. Acesso em: 10 set. 2015.

DESIKAN, S; RAMESH, G. **Software Testing: Principles and Practices**. Delhi: Pearson Education India, 2006.

DS3231 HIGH PRECISION Real-Time Clock Module - Blue (3.3~5.5V). **DX.com**, [c2015]. Disponível em: <http://www.dx.com/p/ds3231-high-precision-real-time-clock-module-blue-3-3-5-5v-222910#.VemZmvlViko>. Acesso em: 04 set. 2015.

EXTREMELY Accurate I2C-Integrated RTC/TCXO/Crystal. **Alldatasheet.com**, [c2005]. Disponível em: <http://pdf1.alldatasheet.com/datasheet-pdf/view/112132/DALLAS/DS3231.html>. Acesso em: 02 set. 2015.

ESP-07 ESP8266 Uart Serial to Wi-Fi Wireless Module with Built-in Antenna for Arduino / Raspberry Pi. **DX.com**, [c2015]. Disponível em: <http://www.dx.com/p/esp-07-esp8266-uart-serial-to-wi-fi-wireless-module-with-built-in-antenna-for-arduino-raspberry-pi-375400#.VemcV_lVikp>. Acesso em: 04 set. 2015.

FRANCISCO, W. C.; **Água**. brasilescola.com, c2015. Disponível em: <http://www.brasilescola.com/geografia/agua.htm>. Acesso em: 25 ago. 2015.

FURGERI, S. **Utilização de catálogos XML para o desenvolvimento do comércio eletrônico**. 1999. 97 f. Dissertação (Mestrado em Informática) – Pontifícia Universidade Católica, Instituto de Informática, Campinas, 1999.

G3/4" WATER Flow Sensor. **Seeedstudio.com**, [c2015]. Disponível em: <http://www.seeedstudio.com/depot/g34-water-flow-sensor-p-1083.html?cPath=144_151>. Acesso em: 09 set. 2015.

GIL, A. C. **Como elaborar projetos de pesquisa.**4. ed. São Paulo: Atlas, 2002.

GOMES, E. V. L.; TAVARES, L.A. **Uma solução com Arduino para controlar e monitorar processos industriais**. 2013. 6 f. Tese (Doutorado) - Curso de Pós-graduação em Engenharia de Sistemas Eletroeletrônicos, Automação e Controle Industrial, Instituto Nacional de Telecomunicações, Porto Alegre, 2013.

GUEDES, G. T. A. **UML 2**: Uma Abordagem Prática. 2. ed. São Paulo: Novatec, 2011.

HETZEL, W. C. **The Complete Guide to Software Testing**. 2th ed. Michigan: QED Inforíation Sciences, 1988.

HIRAMA, K. **Engenharia de Software:** qualidade e produtividade com tecnologia. Rio de Janeiro: Elsevier, 2011.

HUMPHREY, W. S. **Managing the Software Process.** Software Engineering Institute of Carnegie Mellon University, USA, 1989. 494 p.

INFORMAÇÕES Gerais. **Secure.php.net**, [c2015]. Disponível em: <https://secure.php.net/manual/pt_BR/faq.general.php#faq.general>. Acesso em: 13 out. 2015.

JESUS, R. C.**UTILIZAÇÃO DE FRAMEWORKS PARA AUTOMATIZAÇÃO DE TESTES DE APLICATIVOS NA PLATAFORMA ANDROID**. 2013. 112 f. Trabalho de Conclusão de Curso (Graduação em Bacharel em Ciência da Computação) – Universidade Sagrado Coração, Bauru, 2013.

JONES, C.; BONSIGNOUR, O.**The Economics of Software Quality.** New York: Addison-Wesley Professional, 2011.

Korth, H.F. **Sistemas de Bancos de Dados.** 2. ed.Makron Books,1994.

MARTINEZ, M. **UML.** infoescola.com, [c2015]. Disponível em: <http://http://www.infoescola.com/engenharia-de-software/uml/>. Acesso em: 28out. 2015.

MAXFIELD, W. **Aprendendo MySQL & PHP**. São Paulo: Pearson Education do Brasil, 2002.

MCROBERTS, M. **Arduino básico**. São Paulo: Novatec Editora, 2011.

MILANI, A. **MySQL:** Guia do Programador. São Paulo: Novate Editora Ltda, 2006.

MOSLEY, D. J. **The Handbook of Mis Application Software Testing: Methods, Techniques, and Tools for Assuring Quality Through Testing**. Michigan: Prentice Hall, 1993.

MYERS, G. J. **The art of software testing**. 2th ed. New Jersey: John Wiley & Sons, 2004.

NIEDERAUER, J. **Desenvolvendo Websites com PHP**. 2. ed. São Paulo: Novatec Editora Ltda., 2011.

PRESSMAN, R. S. **Engenharia de Software:** Uma Abordagem Profissional. 7. ed. São Paulo: Mc Graw Hill, 2011.

PRESSMAN, R. S. **Software Engineering: A Practitioner's Approach**. 5th ed. Boston: McGraw-Hill, 2001.

REZENDE, R.**Conceitos Fundamentais de Banco de Dados.**devmedia.com.br, [c2015]. Disponível em: <http://www.devmedia.com.br/conceitos-fundamentais-de-banco-de-dados/1649>. Acesso em: 28out. 2015.

RIBEIRO, L. **O que é UML e Diagramas de Caso de Uso**:Introdução Prática à UML. 2013. Disponível em: <http://www.devmedia.com.br/o-que-e-uml-e-diagramas-de-caso-de-uso-introducao-pratica-a-uml/23408#ixzz3Jl41NvXh>. Acesso em: 21out. 2015.

ROCHA, C. B. et al. **SISTEMA DE MONITORAMENTO DE CONSUMO DE ÁGUA DOMÉSTICO COM A UTILIZAÇÃO DE UM HIDRÔMETRO DIGITAL**. 2014. 43f.Relatório Final(Oficina de Integração 3) – Engenharia de Computação, Departamento Acadêmicode Informática, Universidade Tecnológica Federal do Paraná. Curitiba, 2014.

ROCHA, H. **Como criar a sua Home Page HTML**. Rio de Janeiro: Infobook, 1996.

SETZER, V. W.; SILVA, F. S. C. da. **Bancos de dados:** aprenda o que são, melhore seu conhecimento, construa os seus. São Paulo: Edgard Blücher, 2005.

SILVA, P. C. B. **Utilizando UML**: Diagrama de Classes. 2013. Disponível em: <http://www.devmedia.com.br/artigo-sql-magazine-63-utilizando-uml-diagrama-de-classes/12251>. Acesso em: 28out. 2015.

SOMMERVILLE, I. **Software Engineering**. 9th ed. Boston: Pearson Education, 2011.

TIAN, J. **Software Quality Engineering: Testing, Quality Assurance, and Quantifiable Improvement**. Hoboken: John Wiley & Sons, 2005.

VEENENDAAL, E. V.**Foundations of Software Testing: ISTQB Certification**. 2th ed. London: Cengage Learning EMEA, 2008.