

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS GUARAPUAVA
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET

CHRISTIAN WHELINTON SURKAMP

**PLATAFORMA DE GERENCIAMENTO DE CISTERNAS OPEN
HARDWARE E OPEN SOURCE**

TRABALHO DE CONCLUSÃO DE CURSO DE GRADUAÇÃO

GUARAPUAVA

2015

CHRISTIAN WHELINTON SURKAMP

**PLATAFORMA DE GERENCIAMENTO DE CISTERNAS OPEN
HARDWARE E OPEN SOURCE**

Trabalho de Conclusão de Curso de graduação apresentado à disciplina Trabalho de Conclusão de Curso II do Curso Superior de Tecnologia em Sistemas para Internet - TSI - da Universidade Tecnológica Federal do Paraná - UTFPR como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Me. Paulo Henrique Soares

Co-orientador: Prof. Esp. Maurício Barfknecht

GUARAPUAVA

2015

**ATA DE DEFESA DE MONOGRAFIA DE TRABALHO DE CONCLUSÃO DE CURSO DO
CURSO DE TSI**

No dia 1 de dezembro de 2015, às 16:00 horas, nas dependências da Universidade Tecnológica Federal do Paraná Câmpus Guarapuava, ocorreu a banca de **defesa da monografia** de Trabalho de Conclusão de Curso intitulada: **“Plataforma de Gerenciamento de Cisternas Open Hardware e Open Source”** do acadêmico **Christian Whelinton Surkamp** sob orientação do professor **Prof. Me. Paulo Henrique Soares** do Curso de Tecnologia em Sistemas para Internet.

Banca Avaliadora	
Membro	Nome
Orientador	Prof. Me. Paulo Henrique Soares
Coorientador	
Avaliador 1	Prof. Dr. Luciano Ogiboski
Avaliador 2	Prof. Esp. André George Lauer

Situação do Trabalho

Situação	<input checked="" type="checkbox"/> Aprovado <input type="checkbox"/> Aprovado com ressalvas <input type="checkbox"/> Reprovado <input type="checkbox"/> Não Compareceu
Encaminhamento do trabalho para biblioteca	<input checked="" type="checkbox"/> Pode ser encaminhado para biblioteca. <input type="checkbox"/> Manter sigilo para publicação ou geração de patente.

Guarapuava, 1 de dezembro de 2015.

RESUMO

SURKAMP, Christian. Plataforma de Gerenciamento de Cisternas Open Hardware e Open Source. 41 f. Trabalho de Conclusão de Curso de graduação – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2015.

Resumo.

A reutilização da água vem se tornando cada dia mais atrativa e necessária devido a vários fatores, como o seu custo e a dificuldade cada vez maior em consegui-la. Dado isto, o foco deste trabalho é o desenvolvimento de uma plataforma para o gerenciamento de cisternas, a fim de monitorar e otimizar a reutilização da água da chuva. Para tal, foi feito o uso de algumas tecnologias, como PHP, HTML5 e MySQL, a fim de processar, exibir e armazenar dados. Os dados foram coletados com o auxílio de uma plataforma de prototipagem eletrônica de hardware livre(arduino) e de sensores para nível de água, processando e armazenando os mesmos em um banco de dados SQL. Com isso, o objetivo foi gerar economia de recursos hídricos potáveis, buscando também uma economia financeira pela redução do consumo de água das companhias de saneamento.

Palavras-chave: arduino, automação, cisterna, controle

ABSTRACT

SURKAMP, Christian. Tanks Management Platform Open Hardware and Open Source. 41 f. Trabalho de Conclusão de Curso de graduação – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2015.

Abstract.

Reuse of water is becoming more attractive and necessary nowadays due to several factors such as the cost and the increasing difficulty to achieving it. Considering this necessity, the focus of this work is to develop a platform for managing cisterns to monitor and optimize rainwater reuse. To develop the platform it was used some technologies such as PHP, HTML5, and MySQL to process, display, and store data. The data was collected with the help of a free hardware electronics prototyping platform (Arduino) and sensors for water level, processing and storing them in a SQL database. Thus, the goal was to generate potable water savings, also looking for financial savings by reducing the water consumption from water utilities.

Keywords: arduino, automation, cistern, control

LISTA DE FIGURAS

Figura 1	– Válvula Solenoide	12
Figura 2	– Arduino IDE 1.0	17
Figura 3	– Arduino Uno R3	18
Figura 4	– Arduino Mega 2560 R3	19
Figura 5	– Ethernet Shield para Arduino	19
Figura 6	– Válvula Solenoide	20
Figura 7	– Conexão Arduino Mega 2560 com <i>Ethernet Shield</i>	23
Figura 8	– Ligação dos Sensores com o Arduino	25
Figura 9	– Modelo Entidade Relacionamento	26
Figura 10	– Página de login para acesso às páginas administrativas	28
Figura 11	– Visualização das informações referentes a nível das cisternas	28
Figura 12	– Visualização das informações referentes aos arduinos	29
Figura 13	– Visualização das informações referentes aos sensores de nível	29
Figura 14	– Visualização das informações referentes às válvulas	30
Figura 15	– Visualização das informações referentes às bombas	31
Figura 16	– Protótipo desenvolvido para a aplicação	32
Figura 17	– Protoboard e Jumpers Variados utilizados nas conexões	32

LISTA DE TABELAS

TABELA 1	–	Comparativo entre modelos analisados da plataforma arduino	23
----------	---	--	----

SUMÁRIO

1	INTRODUÇÃO	7
1.1	OBJETIVOS	8
1.1.1	Objetivo Geral	8
1.1.2	Objetivos Específicos	8
1.2	DIFERENCIAL TECNOLÓGICO	8
2	FUNDAMENTAÇÃO TEÓRICA	9
2.1	MICROCONTROLADORES	9
2.2	SENSORES	11
2.3	VÁLVULAS	11
2.3.1	Válvulas Solenoides	11
2.4	SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS	12
2.4.1	MySQL	12
2.5	LINGUAGEM DE PROGRAMAÇÃO WEB	13
2.5.1	PHP – Hypertext Preprocessor	13
3	ESTADO DA ARTE	14
4	METODOLOGIA	16
4.1	HARDWARE	17
4.1.1	Plataforma Arduino	18
4.1.2	Sensores	19
4.1.3	Válvulas	20
4.2	SISTEMA WEB	21
5	RESULTADOS	22
5.1	HARDWARE	22
5.1.1	arduino	22
5.1.1.1	<i>Ethernet Shield</i>	23
5.1.1.2	<i>MySQL Connector/Arduino</i>	23
5.1.2	sensores de nível	24
5.2	SISTEMA WEB	25
5.2.1	Banco de Dados	25
5.2.2	Desenvolvimento do Sistema Web	27
5.3	FUNCIONAMENTO DO PROTÓTIPO	29
6	CONSIDERAÇÕES FINAIS	33
	REFERÊNCIAS	34
7	APÊNDICE A – ALGORITMO DESENVOLVIDO PARA O ARDUINO	35

1 INTRODUÇÃO

É de conhecimento comum que a água é um recurso vital para a vida em nosso planeta, sendo ela extremamente necessária não só para o consumo dos seres vivos, como também nas fábricas, indústrias, agricultura e pecuária.

Segundo um relatório da UNESCO (2015), a demanda hídrica vem aumentando continuamente, enquanto a quantidade de água doce no planeta continua a mesma, causando preocupação na sociedade em geral, gerando uma série de estudos e incentivos na área, a fim de proporcionar um consumo sustentável e consciente.

São muitas as propostas de economia e consumo consciente de água, cada qual com sua particularidade e efetividade. Um dos principais métodos é o armazenamento da água da chuva, que poderá ser utilizada para inúmeros fins, os quais variam desde a lavagem de calçadas até a utilização em descargas de vasos sanitários, geralmente armazenando grandes quantidades por meio de cisternas.

Entretanto, quando se trata de usos constantes, ou pelo menos rotineiros, existe um agravante, a frequência e intensidade das chuvas, que são variáveis. Em certos períodos, onde a ocorrência de chuvas é menor, para que não haja falta deste recurso, é necessário completar o nível dos reservatórios com o auxílio dos serviços das companhias de saneamento e controlar quando este recurso deve ser utilizado. Em certos casos, este controle pode ser trabalhoso, seja por dificuldades em seu controle ou até mesmo no monitoramento.

Assim, observou-se a necessidade de implantar uma plataforma a fim de gerenciar cisternas, buscando otimizar o consumo de água, evitando gastos desnecessários com o recurso das companhias, mantendo a rede sempre abastecida e com baixo custo de implementação.

A plataforma deve controlar as bombas d'água, evitando que funcionem com o reservatório vazio, o que ocasionaria a queima das mesmas, controlando as entradas de água alternativas com base no monitoramento do nível do depósito, enviando os dados a um servidor que, por sua vez, irá processá-los e armazená-los. Também possibilitando que os dados sejam acessados em uma interface web, a qual também poderá controlar o funcionamento da cisterna,

controlando bombas e válvulas.

Com esta plataforma, o gerenciamento do reservatório ocorrerá de forma autônoma ou manual, tendo a facilidade de ser feito de maneira remota, sem que haja a necessidade de uma pessoa no local.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

O presente trabalho apresenta uma proposta de implementação de um sistema de gerenciamento remoto para cisternas, utilizando uma plataforma de prototipagem eletrônica de hardware livre, coletando, processando e armazenando todos os dados do sistema de forma autônoma.

1.1.2 OBJETIVOS ESPECÍFICOS

- Construir um sensor de nível utilizando materiais básicos e de baixo custo a fim de verificar o nível de água do reservatório.
- Implementar um sistema de controle de fluxo de água com o auxílio de uma plataforma de prototipagem eletrônica de baixo custo de hardware livre.
- Implementar um sistema de monitoramento e supervisão web para gerenciamento do conjunto.

1.2 DIFERENCIAL TECNOLÓGICO

Este trabalho apresenta o desenvolvimento uma plataforma de gerenciamento de cisternas que possa funcionar de modo autônomo, fazendo uso de recursos open hardware e open source, garantindo assim um baixo custo, mas mantendo a eficiência do projeto, permitindo ainda que tudo possa ser monitorado e controlado remotamente em um ambiente web, garantindo assim que o sistema possa ser controlado de qualquer dispositivo que consiga acessar a interface web e que esteja autenticado no sistema.

2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção é realizado um estudo referente aos equipamentos, dispositivos e tecnologias relevantes para o entendimento do sistema proposto. Primeiramente serão apresentados os hardwares pesquisados para realizar a implementação do projeto.

2.1 MICROCONTROLADORES

Os microcontroladores são dispositivos eletrônicos que possuem um microprocessador, memórias para as funções de leitura e escrita, além de interfaces de entradas e saídas, podendo ser programados para funções bem específicas. Por seu tamanho diminuto e bom custo/benefício, podem ser encontrados em muitos lugares. Em outras palavras, os

Microcontroladores são computadores de propósito específico. Eles possuem tamanho reduzido, baixo custo e baixo consumo de energia. Devido a esses fatores há diversos segmentos, que os utilizam, tais como a indústria automobilística, de telecomunicações, de brinquedos, de eletrodomésticos, de eletroeletrônicos, bélica [...](SILVA, 2009, p. 17).

Uma forte característica dos microcontroladores, se deve ao fato deles poderem funcionar de maneira autônoma, incorporando seus vários recursos em uma mesma placa, montando os componentes em um circuito integrado, com dimensões reduzidas. Os componentes dos microcontroladores são descritos de maneira completa por Martins (2005, p. 16):

Tipicamente, um microcontrolador caracteriza-se por incorporar no mesmo encapsulamento um microprocessador (com a finalidade de interpretar as instruções de programa e processar dados), memória de programa (com a finalidade de armazenar de maneira permanente as instruções do programa), memória de dados (com a finalidade de memorizar os valores associados com as variáveis definidas no programa), uma série de pinos de entrada/saída (com a finalidade de realizar a comunicação do microcontrolador com o meio externo) e vários periféricos (tais como temporizadores, controladores de interrupção, temporizadores cão de guarda (WatchDog Timers – WDTs), comunicação serial, geradores de modulação por largura de pulso ou de PWM (Pulse Width Modulation), conversores analógico/digital etc.), fazendo com que o hardware final fique extremamente complexo.

Existem vários tipos de microcontroladores no mercado, cada qual com um objetivo específico. Segundo Martins (2005, p. 15), os que difere entre os vários modelos, são algumas características muito importantes, como a velocidade do processamento, a quantidade de memória interna disponível, ”a quantidade de pinos de [...] [entrada e saída], a forma de alimentação, os tipos e as quantidades de periféricos, a arquitetura e o conjunto de instruções disponibilizado nos circuitos internos”.

Os microcontroladores se dividem em famílias, onde os mais populares são da série PIC, fabricados pela empresa Microchip Technology Inc., a qual é popularizada por apresentar um plano “baseado na disseminação de uma ferramenta de auxílio à construção de programas – o MPLAB IDE. [...] [e, por possuírem] uma linguagem Assembly menos complexa em relação àquelas disponibilizadas por outros fabricantes” (MARTINS, 2005, p. 15). Como forte característica dessa família, temos o fato de possuírem várias opções de memória de programas, além de possuir ”opções de baixa tensão e inúmeros tipos de circuito osciladores, assim como várias opções de encapsulamento”(MARTINS, 2005, p. 17).

Outra família de microcontroladores amplamente citada em literaturas, a AVR, possui uma ampla aceitação no mercado, possuindo um grande representante, fabricado pela empresa ATMEL, o qual apresenta “bom desempenho frente ao número de instruções executadas por ciclo de clock.”(LIMA; SCHWARZ, 2009, p. 94).

Analisando estas duas famílias de microcontroladores, foi estudada a plataforma Arduino, a qual utiliza de um microcontrolador da família AVR. Segundo Carvalho (2011), o arduino tornou muito simples e intuitivo, o que antes exigia conhecimentos técnicos referentes a eletrônica e programação. São muitos os usos do arduino, pois esta plataforma permite o monitoramento de sensores, assim como o controle de algumas funções, como ligar e desligar equipamentos elétricos e abertura e fechamento de válvulas solenoides.

Dados estes fatos, temos ainda que

O Arduino oferece uma interface de hardware proporcionando todo o circuito necessário para funcionamento do microcontrolador e uma interface e ambiente de desenvolvimento em software para programação. Por ser uma plataforma de código aberto (open-source) há uma grande comunidade de desenvolvedores do mundo inteiro que publicam bibliotecas já com toda a programação pronta para se usar, com funções específicas, como, por exemplo, o controle de servo motores ou leitura de sensores analógicos(CARVALHO, 2011, p. 34).

2.2 SENSORES

Os sensores são dispositivos que, em interação com o ambiente, mudam de estado, podendo estes serem compostos por um ou vários componentes. Em outras palavras, "os sensores são dispositivos que agem sob a ação de uma grandeza física, conforme a mudança de comportamento dessa grandeza varia, esses dispositivos podem emitir sinais."(MORETTO, 2005, p. 41). De acordo com Marchesan (2012, p. 19), eles são capazes de interagir com inúmeras grandezas físicas, tais como temperatura, pressão e movimento, convertendo-as em sinais analógicos ou digitais.

A mudança de estado dos sensores podem ser captadas por meio de leituras, permitindo assim que alguma ação possa ser tomada de acordo com o seu evento. Os eventos gerados variam de acordo com o tipo de sensor utilizado, visto que existe uma vasta gama de sensores no mercado, cada qual com um propósito específico.

2.3 VÁLVULAS

As válvulas são dispositivos mecânicos que são usadas em sistemas hidráulicos e pneumáticos, com o intuito de guiar e controlar a vazão de um fluido. Elas podem ser divididas em dois grupos, as de posições finitas e as de posições infinitas. Dizemos que as válvulas de posições finitas são aquelas que apenas permitem ou bloqueiam a passagem dos fluídos, as quais tem função apenas de ligar e desligar atuadores. Já as válvulas de posições infinitas, além das funções de ligar e desligar, elas também conseguem controlar a vazão em qualquer valor entre a posição totalmente fechada e totalmente aberta(BOLTON, 2010).

2.3.1 VÁLVULAS SOLENOIDES

Válvulas solenoides são dispositivos eletromecânicos que são acionados com a passagem de uma corrente elétrica através de um solenoide, alterando assim o estado da válvula. Estas válvulas consistem de uma combinação de uma bobina de solenoide e uma válvula que irá controlar o fluxo dos fluídos. Quando eletricamente elas abrem, fecham ou direcionam o fluxo, caracterizando-se como válvulas do grupo com posições finitas(PETRUZELLA, 2013).

Abaixo podemos ver a estrutura de uma válvula solenoide:

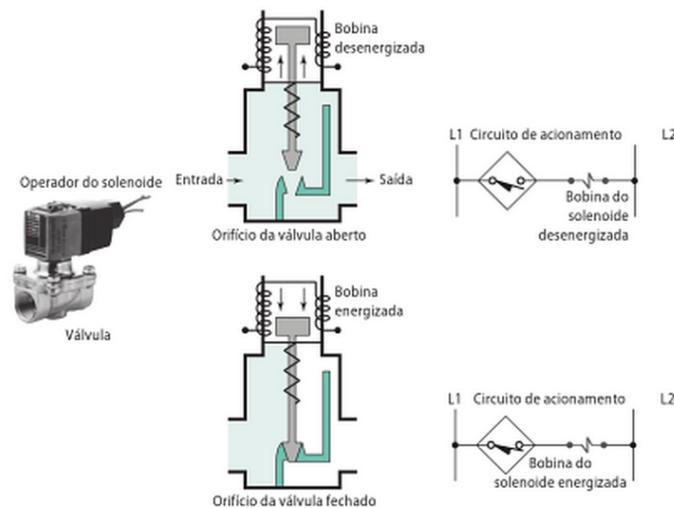


Figura 1: Válvula Solenoide

Fonte: <http://www.ascovalve.com/>

2.4 SISTEMA DE GERENCIAMENTO DE BANCO DE DADOS

Um sistema de gerenciamento de banco de dados, ou SGBD, consiste de um software que foi projetado para auxiliar na manutenção e utilização de vastos conjuntos de dados. São várias as vantagens da sua utilização, como independência dos dados e acesso eficiente a eles, manutenção da integridade e segurança dos dados, facilidade na administração dos dados, dentre outras (RAMAKRISHNAN; GEHRKE, 2008).

2.4.1 MYSQL

O MySQL é um sistema de gerenciamento de banco de dados relacional poderoso e rápido, o qual permite armazenamento, pesquisa, classificação e recuperação de dados de forma eficiente. O seu servidor consegue controlar o acesso aos dados assegurando que vários usuários consigam trabalhar ao mesmo tempo, fornecendo acesso rápido aos dados, assegurando ainda que somente usuários autorizados consigam acesso a ele (WELLING; THOMSON, 2005).

MySQL está disponível sob licença dupla, onde ele pode ser usado sob uma licença *Open Source* (General Public License - GPL) gratuitamente, desde que os termos da mesma sejam cumpridos, e sob a opção de licença comercial, a qual deve ser utilizada em casos de distribuição de aplicação não-GPL (WELLING; THOMSON, 2005).

2.5 LINGUAGEM DE PROGRAMAÇÃO WEB

As linguagens de programação Web são utilizadas no desenvolvimento de sistemas, os quais deverão conter conteúdos dinâmicos, permitindo ainda que os mesmos possam ser acessados através de uma rede.

2.5.1 PHP – HYPERTEXT PREPROCESSOR

O PHP foi concebido em 1994 por Rasmus Lerdorf. Desde então, esta linguagem de programação foi adotada por outras pessoas que foram aperfeiçoando-o por três versões diferentes, chegando ao que temos hoje(WELLING; THOMSON, 2005). Esta linguagem faz uso de um conjunto de *scripts*, sendo voltada para aplicações Web, embutido no HTML. Os códigos são delimitados por *tags* iniciais e finais, permitindo ao programador alternar entre o uso de PHP e HTML. O seu código é aberto, dispensando a obtenção de licenças, o programa é gratuito, é multiplataforma, tem acesso a banco de dados e faz o processamento de imagens ao enviá-las para o navegador do usuário(HACKENHAAR et al., 2010).

A linguagem PHP apresenta muitas vantagens em sua utilização. Dentre elas, podemos citar o fato de que ela é compatível com a maioria dos sistemas operacionais presentes no mercado, assim como possui um bom suporte a ele em meio aos servidores web principais, além de que permite ainda uma integração com alguns dos principais bancos de dados presentes no mercado, como MySQL, Oracle, PostgreSQL, dentre outros(HACKENHAAR et al., 2010).

3 ESTADO DA ARTE

Existem trabalhos que tratam do controle de cisternas e de automação, cada qual optando por diferentes configurações de hardware e linguagens diversas. Estes trabalhos buscam otimizar algum processo em específico, seja monitoramento ou controle do processo. Dentre estes trabalhos, pode-se citar alguns, como:

Moretto (2005), que apresenta um “sistema didático de controle de nível com tanques acoplados”, o qual conta com um controlador lógico programável (CLP) para o controle do fluxo da água pelo sistema, utilizando sensores de nível, válvulas solenoides e um motor de indução, apresentando ainda uma interface gráfica para controle e supervisão do sistema.

Marchesan (2012), que desenvolveu o protótipo de um “sistema de monitoramento residencial utilizando a plataforma arduino”, o SiMRe - sistema de monitoramento residencial, o qual fez uso de diversos dispositivos que conferem boa segurança, juntamente com a utilização de ferramentas livres, efetuando o registro de eventos disparados pelo sistema em um banco de dados, permitindo o acesso ao sistema via internet e possibilitando monitoramento e controle do sistema.

Outro estudo relevante é o proposto por Batista (2014), que apresenta um “Sistema de Irrigação de Plantas em Ambientes Residenciais”, o qual consiste em um sistema de irrigação automático, que conta com uma interface web para monitoramento, fazendo uso de um arduino para as comunicações entre sensores e servidor, assim como para o controle dos demais componentes, como válvulas, sensores e bomba.

O trabalho proposto neste projeto tem como objetivo desenvolver um sistema eficiente e confiável, levando em consideração a necessidade de baixo custo de implementação e manutenção, a fim de que possa ser utilizado nas mais diversas cisternas ou qualquer outro tipo de reservatório de fluidos que necessitem de um sistema de gerenciamento.

Para isso, pretende-se utilizar sensores de altura e válvulas solenoides, estes que serão controlados por um arduino, que por sua vez, deverá realizar a integração do sistema com uma aplicação web, que fará o processamento das informações, armazenando-as em um banco de da-

dos relacional SQL, apresentando uma interface web para que todo o monitoramento e controle possa ser feito de maneira simples, segura e eficaz.

4 METODOLOGIA

A metodologia empregada para o desenvolvimento da plataforma se deu por etapas, também chamadas de fases. Cada uma teve um enfoque principal, visando o sucesso do trabalho.

Primeiramente, foi realizado um levantamento bibliográfico a respeito das tecnologias e equipamento que se almejava utilizar, verificando a viabilidade de cada uma.

Com base neste estudo, foram selecionadas algumas tecnologias:

- Arduino como plataforma open source de desenvolvimento;
- sensores;
- válvulas solenoides;
- SGBD: MySQL;
- Linguagem de programação Web – PHP.

Após o levantamento inicial, as etapas seguintes foram voltadas ao desenvolvimento do projeto.

Como etapa inicial do desenvolvimento, foi trabalhada a questão do desenvolvimento do sensor de nível, para aferir o nível da cisterna, visando o baixo custo do projeto e a utilização de materiais facilmente encontrados.

Em um segundo momento, utilizando a linguagem de programação C++ e o ambiente de desenvolvimento Arduino IDE (Figura 2), foram desenvolvidos os algoritmos necessários para o funcionamento do hardware, seguindo as necessidades do projeto proposto, efetuando ainda, a inserção dos dados coletados diretamente em um banco de dados relacional.

Como próxima etapa, desenvolveu-se o sistema web, o qual apresenta uma interface web, permitindo assim, que haja uma interação humana com o sistema, viabilizando o moni-

toramento remoto, assim como a definição dos parâmetros para funcionamento do sistema, de maneira fácil e intuitiva.

A etapa final do projeto, consistiu na fase de aplicação, onde as partes são agregadas, criando o sistema de gerenciamento de cisternas, o qual foi instalado em uma maquete para averiguar o seu funcionamento.

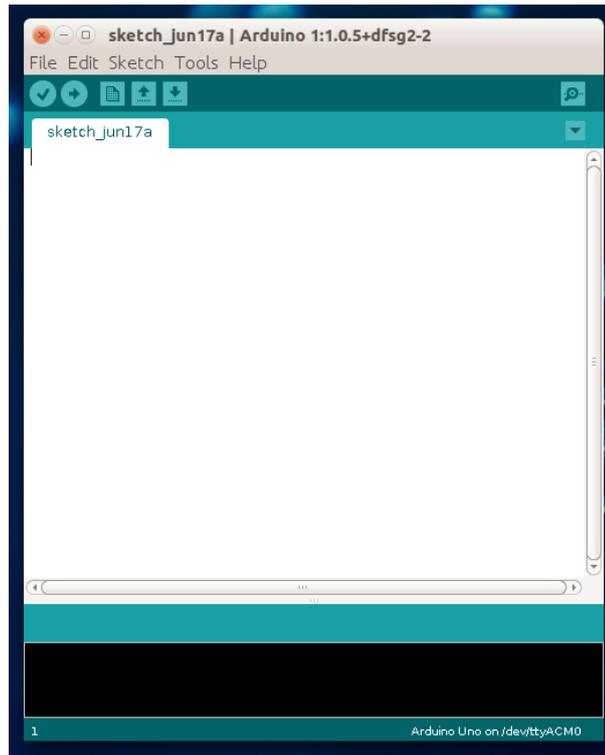


Figura 2: Arduino IDE 1.0

Fonte: O autor

4.1 HARDWARE

Para a decisão das tecnologias, foram estabelecidos alguns critérios, os quais visam facilitar a implantação do projeto e melhorar a sua viabilidade. Na escolha do microcontrolador para o sistema, optou-se por um que estivesse com uma plataforma de desenvolvimento agregada, o que diminuiria a necessidade de integração com muitos componentes facilitando o entendimento do projeto, bem como seu desenvolvimento. Podemos destacar ainda que, com a utilização de microcontroladores sem plataforma de desenvolvimento, elevaria consideravelmente o custo do projeto, fugindo a um de seus focos principais.

Com base nestas informações, optou-se pela plataforma de prototipagem eletrônica arduino, pois ela conta com vários dos componentes necessários integrados por meio de uma

placa de circuito impresso. Outro aspecto relevante para a escolha, se deu ao fato de que esta plataforma é facilmente integrada com sensores digitais e analógicos. As demais características são descritas nos próximos itens deste capítulo.

4.1.1 PLATAFORMA ARDUINO

De acordo com informações do site oficial, Arduino (ARDUINO...,) ”é uma plataforma *open-source* de prototipagem eletrônica baseada na flexibilidade, *hardware* e *software* fácil de usar”. Ela foi trabalhada de modo que seja totalmente autônoma, necessitando apenas de uma fonte de alimentação para que funcione.

Foram consideradas algumas características para que esta plataforma fosse considerada uma solução atraente. Dentre elas, pode-se destacar a existência de um ambiente de desenvolvimento de algoritmos - IDE (Figura 2) multiplataforma, assim como a existência de conectores em todas as portas de entrada e saída do microcontrolador, o que facilita a integração com outros componentes, além de que, por possuir uma saída de alimentação de 3,3 a 5 volts, os componentes podem ser ligados a ele sem que haja a necessidade de outra fonte de alimentação para os mesmos.

Pela utilização dos microcontroladores da série AVR, existem inúmeras versões desta plataforma, cada qual fazendo utilização de um microcontrolador específico. A seguir encontram-se as figuras 3 e 4, que mostram o Arduino nas versões UNO e MEGA 2560.



Figura 3: Arduino Uno R3

Fonte: <http://www.arduino.cc/en/Main/ArduinoBoardUno>

Para o sistema proposto, será utilizada a versão MEGA 2560. Esta escolha baseou-se nas necessidades do projeto, em questões de número de portas de comunicação e hardware satisfatório para a aplicação, visto que a versão UNO, mesmo apresentando um custo relativamente menor, conta com um reduzido número de portas e uma memória relativamente pequena,

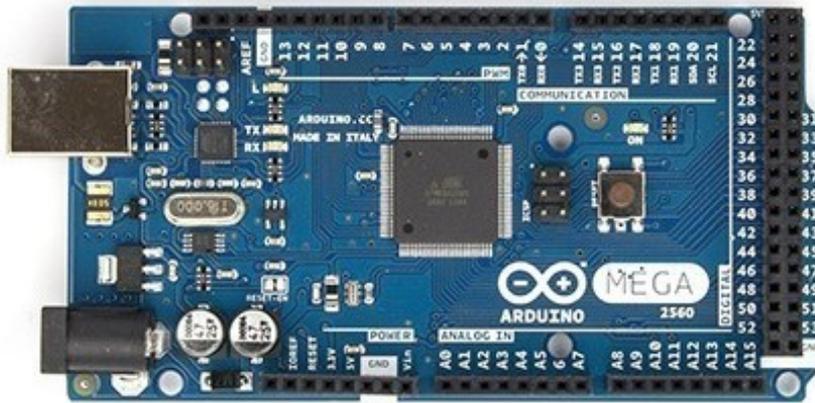


Figura 4: Arduino Mega 2560 R3

Fonte: <http://www.arduino.cc/en/Main/ArduinoBoardMega2560>

se levado em consideração que projetos maiores fazem utilização de grandes bibliotecas. A comunicação entre o arduino e rede ocorrerá com o auxílio de um módulo específico, chamado *Ethernet Shield*(Figura 5).

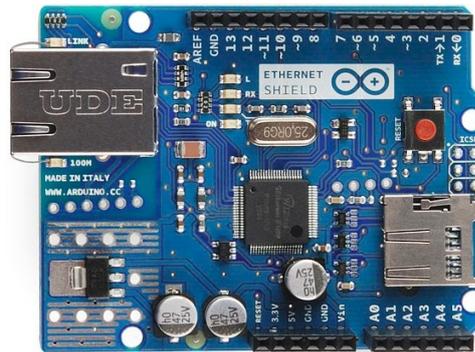


Figura 5: Ethernet Shield para Arduino

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

São denominados *Shields*, os inúmeros acessórios próprios para a plataforma arduino. Estes *Shields* são placas de circuito impresso, que tem por finalidade realizar outras tarefas, como no caso do *Ethernet Shield*, que tem por finalidade realizar a conexão do arduino com a rede. Um aspecto interessante sobre os *Shields*, se dá ao fato de que eles são projetados para "encaixar" sobre o arduino, o que reduz o espaço ocupado, aproveitando ainda a mesma fonte de alimentação.

4.1.2 SENSORES

Os sensores são peças indispensáveis em processos de monitoramento e controle remoto, ainda mais quando se trata de funcionamento autônomo, pois é baseado em suas leituras que as operações são realizadas.

Para o projeto em questão, visando o monitoramento de nível do fluido da cisterna, propõe-se a montagem de um sensor, fazendo uso de materiais básicos, fáceis de serem encontrados e ainda com um custo muito reduzido. Para isso, pretende-se utilizar eletrodos de cobre, em conjunto com um sensor de referência, os quais ficarão dispostos em uma das laterais do reservatório, partindo de sua base até sua superfície posterior, de maneira sequencial. Os sensores serão todos ligados ao arduino, que terá como um de seus papéis, realizar a leitura do acionamento de cada sensor.

A leitura dos sensores ocorre no momento em que os mesmos ficam submersos. Quando isso ocorre, os sensores entram em curto circuito, transmitindo sinal elétrico para o arduino, permitindo que o controle de nível do reservatório seja efetuado.

4.1.3 VÁLVULAS

As válvulas são peças muito importantes nos processos de controle, principalmente quando se trata de fluídos, pois é por meio delas que os fluxos são controlados. Para o seguinte projeto, optou-se por válvulas solenoides (Figura 6), as quais controlarão o fluxo das tomadas de água para os reservatórios.

Este tipo de válvula trabalha com correntes elétricas, possuindo um pistão interno que, ao receber a tensão desloca-se permitindo a passagem da água. No momento em que a tensão é interrompida, o pistão imediatamente é deslocado, bloqueando a passagem dos fluídos. Optou-se pela utilização de um modelo que trabalha com corrente AC 110V, a qual permite que seja controlada diretamente pela plataforma de prototipagem sem que haja a necessidade de uma fonte alternativa, fazendo uso de um relê para tal.



Figura 6: Válvula Solenoide

Fonte: <http://labdegaragem.com/forum/topics/v-lvula-solen-ide-d-vidas>

O controle das tomadas de água se dará com base nos dados obtidos por meio dos sensores de nível instalados nos reservatórios, de acordo com o que o sistema foi programado.

4.2 SISTEMA WEB

Para o desenvolvimento do sistema Web, após análise das várias soluções disponíveis no mercado, foram selecionadas algumas delas para o desenvolvimento desta etapa do projeto. Para realizar esta seleção foram levados em consideração alguns aspectos importantes, como o custo final do projeto para as implementações e que o sistema ganhasse algum aspecto robusto. Dado isso, optou-se pelas seguintes soluções de código livre:

Para servidor Web foi selecionado o Apache, pois o mesmo se trata de uma aplicação *open source*, apresentando uma vasta documentação disponível, fornecendo suporte a várias linguagens de programação.

Como linguagem de programação, foi escolhido o PHP, pois o mesmo possui módulos para o Apache, e tem suporte nativo aos principais bancos de dados, se trata de uma linguagem de código livre, sendo constantemente atualizada, e pelo conhecimento prévio do autor referente à linguagem.

Por fim, foi definido o sistema de gerenciamento de banco de dados, onde optou-se pelo *MySql*, por se tratar de um sistema de banco de dados com uma certa otimização para aplicações Web e pela sua boa integração com a linguagem de programação PHP, além da possibilidade de integração direta com a plataforma arduino, com o auxílio de um *ethernet shield* e de biblioteca específica.

5 RESULTADOS

Neste capítulo serão abordados os resultados do projeto proposto. Desta forma, na primeira seção são abordados os resultados da utilização do hardware. Na sequência, é apresentado o sistema web desenvolvido. Finalizando com a integração das partes, explicando todo o funcionamento, apresentando ainda uma visão geral do protótipo.

5.1 HARDWARE

5.1.1 ARDUINO

O arduino, neste trabalho, é utilizado como principal responsável pelo controle do processo, pois contém o programa do sistema. O arduino recebe os sinais dos sensores de nível, analisa-os, e com base nos resultados, toma as ações necessárias, comandando as válvulas e bomba d'água, além de gravar em um banco de dados remoto todas as informações inerentes à aplicação.

Foi previamente estabelecido que seria feita a utilização do arduino em sua versão UNO, pois o mesmo possui capacidade adequada de memória, suportando o programa com todas as suas bibliotecas, além de apresentar um custo relativamente baixo. Entretanto, o gargalo para a sua utilização foi o número de portas disponíveis para comunicação, pois a leitura de cada sensor deve ser efetuada por meio de uma porta de entrada analógica, estas que a versão UNO possui apenas 6, das quais, duas são utilizadas pelo *Ethernet Shield*, onde as restantes não são suficientes para as necessidades do projeto.

Visando contornar este empecilho, foi adotada uma nova versão da plataforma arduino, chamada arduino Mega 2560, a qual se sobrepõe em vários aspectos, se comparado à versão UNO, como pode ser observado na Tabela 1. Com esta nova versão, todas as necessidades do projeto foram satisfeitas. Entretanto, o custo do projeto se elevou, devido à diferença de valores entre as versões.

Uma das necessidades do projeto, foi deixar o arduino em rede, de modo que fosse

Tabela 1: Comparativo entre modelos analisados da plataforma arduino

	Arduino UNO	Arduino Mega 2560
Microcontrolador	ATmega328P	ATmega2560
Nº de portas de entrada e saída digital	14	54
Nº de portas de entrada analógica	6	16
<i>Flash memory</i>	32KB	256KB
CPU Speed	16 MHz	16 MHz

Fonte: <https://www.arduino.cc/en/Products/Compare>

possível o armazenamento os dados em um banco de dados remoto, para que os mesmos possam ser acessados por uma aplicação Web.

5.1.1.1 *ETHERNET SHIELD*

A fim de estabelecer esta conexão, foi feita a utilização de um *Shield* apropriado, chamado *Ethernet Shield*, o qual foi acoplado sobre o arduino(Figura 7). A utilização desta placa se fez atraente, pois a mesma encaixa-se perfeitamente sobre o microcontrolador, sem comprometer as portas de entradas e saída, além de que, a mesma é de fácil manuseio. Para tornar a lógica de programação simples, fez-se a utilização de uma biblioteca específica para esta placa, abstraindo toda a complexidade do código, garantindo a fácil utilização e o bom entendimento do algoritmo.

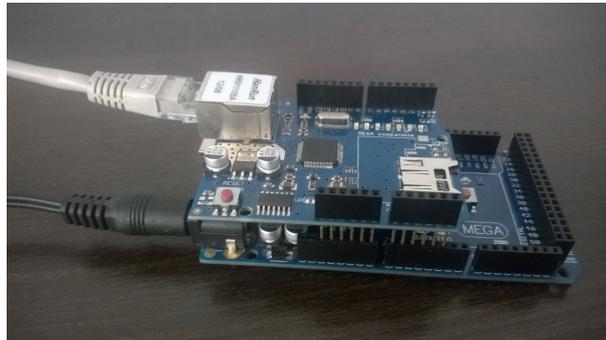


Figura 7: Conexão Arduino Mega 2560 com *Ethernet Shield*

Fonte: O autor

5.1.1.2 *MYSQL CONNECTOR/ARDUINO*

Visando a comunicação do arduino com o banco de dados, havia-se preestabelecido que a mesma ocorreria de maneira convencional, em que o microcontrolador comunica-se com um servidor, o qual processa e armazena as informações em um banco de dados. Entretanto, este processo era realizado em duas etapas, aumentando a sua complexidade, delegando ao servidor a responsabilidade de uma tarefa que pode ser considerada simples.

Objetivando-se uma abstração da complexidade encontrada nas comunicações, que se dava por etapas, foi realizado um levantamento de soluções alternativas para a comunicação arduino/Banco de Dados, pesquisa esta que resultou em uma solução que descarta a utilização de um serviço intermediário, esta que, por sua vez, se trata de uma biblioteca destinada exclusivamente para a comunicação direta de um projeto arduino, com um servidor MySQL.

Baseada em um protocolo cliente MySQL, esta biblioteca permitiu que os comandos SQL do arduino fossem enviados diretamente à base de dados MySQL, possibilitando o processamento da resposta da requisição, sem que fosse necessária a utilização de um intermediário, ou de alguma configuração externa especial.

Outro aspecto interessante do *MySQL Connector/Arduino*, se deve ao fato de a mesma possuir um manual, facilitando a sua utilização e entendimento, além de que ela está otimizada, a fim de reduzir o seu tamanho, visto que a memória disponível no arduino é bem reduzida.

5.1.2 SENSORES DE NÍVEL

Os sensores de nível ficam responsáveis pela detecção dos sinais relacionados ao nível, efetuando o envio destes dados para o arduino. Baseado nestas informações, o arduino pode atuar sobre o processo. Esta intervenção fica em função do programa de controle.

Para utilização neste trabalho, foram adotados eletrodos de cobre como sensores, os quais devem ficar submersos na água quando o nível da cisterna atinge a altura em que os mesmos estão dispostos.

Os sensores devem ficar dispostos aos pares, onde um deles se trata de um sensor de referência e o outro de um sensor de leitura. O primeiro deles, de referência, fica disposto logo abaixo do sensor de leitura e tem como função aplicar uma tensão de 5V no líquido do reservatório. O sensor de leitura fica disposto a poucos milímetros do seu sensor de referência. Quando ambos ficam submersos, acabam entrando em curto circuito, gerando sinal elétrico para o sensor de nível, o qual chega ao arduino, permitindo que o controle do nível seja realizado.

Para conectar os sensores ao arduino, foi necessário a utilização de um resistor no circuito. Isto se fez necessário para evitar que o arduino interprete erroneamente que o sensor foi "ativado", pois o mesmo evita que a tensão na porta de entrada fique flutuando, mantendo-a próximo a 0, quando o sensor estiver "desativado".

A Figura 8 apresenta o esquema de ligação dos sensores com o arduino, fazendo a utilização dos resistores.

Neste exemplo, as ligações em vermelho representam os sensores de referência, uma vez que estão ligadas à fonte de alimentação de 5V do arduino. As ligações em amarelo representam os sensores de leitura, os quais partem do reservatório de água, chegando às portas analógicas (pinos A6 e A7 utilizados no exemplo) de leitura do arduino, tendo ainda a ligação dos resistores em seu circuito conforme descrito.

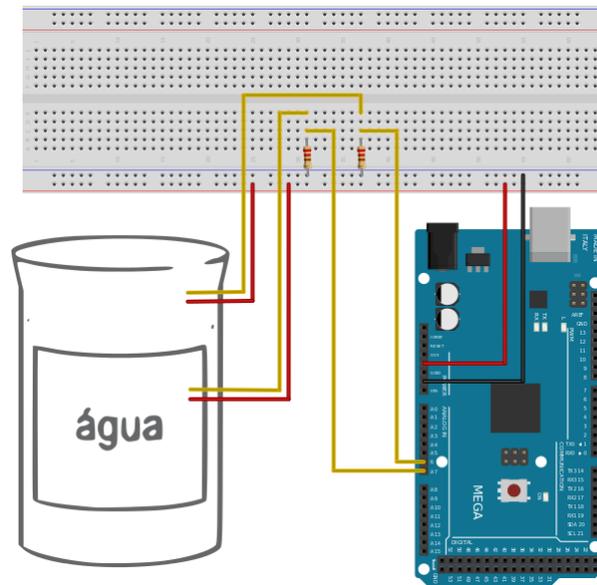


Figura 8: Ligação dos Sensores com o Arduino

Fonte: O autor

5.2 SISTEMA WEB

O sistema Web efetuará o armazenamento das informações enviadas pelo arduino, fornecendo ainda uma interface Web, a fim de garantir um acesso rápido e prático ao sistema de gerenciamento de cisternas, podendo este acesso ser efetuado de qualquer aparelho que possua acesso à internet.

A fim de facilitar o desenvolvimento e entendimento do projeto, o sistema Web foi subdividido em duas partes, em que cada qual teve um foco bem específico, uma delas abrangendo a persistência dos dados, e a outra a exibição dos mesmos aos usuários do sistema.

5.2.1 BANCO DE DADOS

Visando a persistência dos dados do sistema, elaborou-se um levantamento dos requisitos básicos para a composição do modelo conceitual do banco de dados, os quais são listados abaixo:

- Os usuários do sistema devem estar todos cadastrados;
- Deve-se saber o local em que cada arduino está instalado;
- Cada sensor deve estar ligado a um único arduino;
- A localização e um identificador para cada sensor;
- Cada válvula deve estar ligada a um único arduino;
- A localização e um identificador para cada válvula;
- Cada bomba deve estar ligada a um único arduino;
- A localização e um identificador para cada bomba;
- O modelo de cada arduino.

Com base nestas informações, foi possível elaborar o projeto lógico do banco, com o qual foram definidos os tipos das variáveis, assim como as chaves estrangeiras, conforme pode ser visualizado na Figura 9.

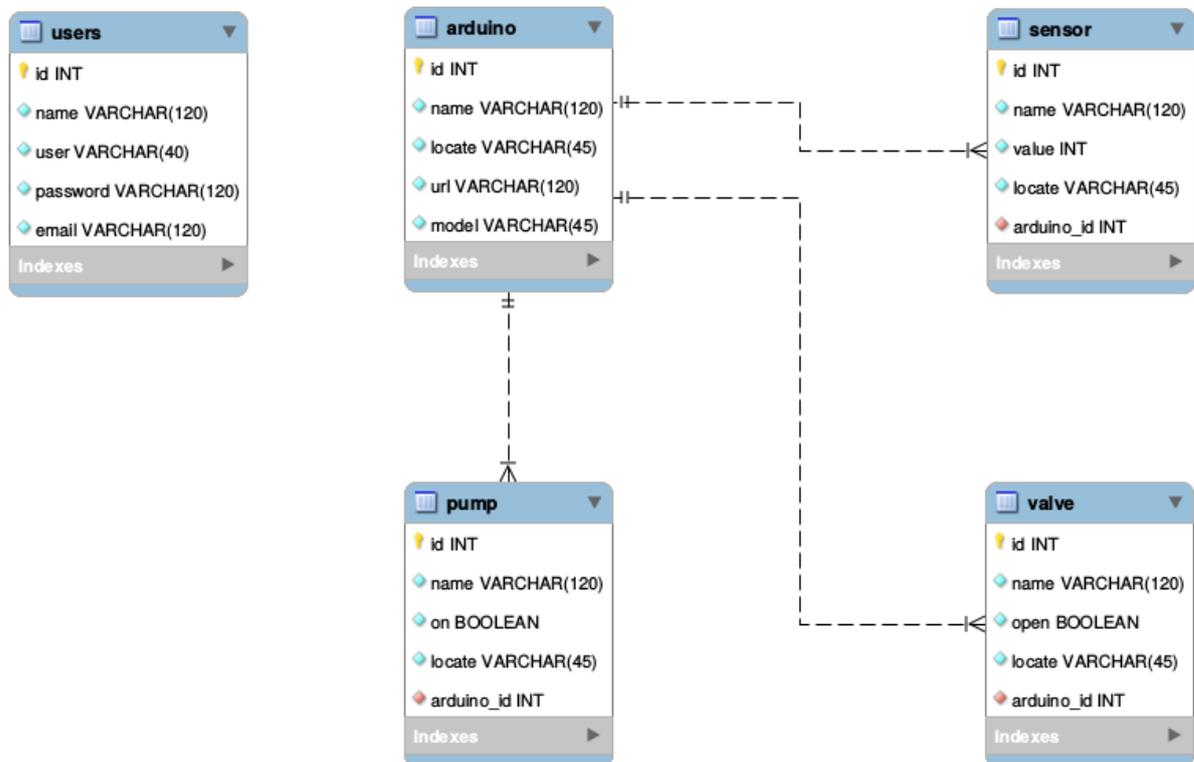


Figura 9: Modelo Entidade Relacionamento

Fonte: O autor

De posse destes dados, implementou-se o projeto físico, para o qual se fez uso do SGBD relacional MySQL realizando a criação das tabelas, conforme descrito brevemente abaixo:

A tabela "users" tem por finalidade efetuar o armazenamento das informações básicas necessárias a um usuário do sistema, sendo elas, um código identificador, nome, usuário para o sistema, senha de acesso e e-mail para contato.

De maneira semelhante, criou-se a tabela "arduino", na qual devem ser dispostos os dados referentes a cada arduino utilizado no projeto, uma vez que, pelas restrições de cada plataforma, ou até mesmo pela distância entre reservatórios, faz-se necessária a utilização de mais de um microcontrolador. Para este, são armazenadas algumas informações referente ao arduino, como o seu código identificador, um nome, localização, url e modelo.

Como ponto trivial para o projeto, produziu-se a tabela "sensor", visto que, baseado em seus valores, todas as ações são tomadas no gerenciamento dos reservatórios. Como o trabalho propõe a utilização de apenas um tipo de sensor, fez-se desnecessária a utilização de um campo "tipo" nesta tabela, gravando apenas os dados referentes ao seu nome, localização, código identificador de arduino, para o qual envia suas leituras, um código identificador e, mais importante, o valor observado.

A tabela "valve" é o local em que são armazenados os dados inerentes às válvulas de tomadas d'água dos reservatório. Para as mesmas, armazenam-se as informações de localização, nome, código identificador, arduino que esta conectado e valor, que diz respeito ao seu estado de abertura/fechamento.

Concebeu-se ainda uma tabela chamada "pump", a fim de gravar os dados referentes às bombas d'água conectadas ao sistema, persistindo os dados inerentes ao seu nome, estado de funcionamento, localização, código identificador e arduino ao qual está conectada.

5.2.2 DESENVOLVIMENTO DO SISTEMA WEB

O sistema Web permite a visualização dos dados referentes ao estado de cada parte que compõe o sistema, permitindo ainda a realização de algumas interações no mesmo, como a definição dos parâmetros, que regulam a abertura e fechamento das válvulas.

Para ter acesso às páginas de monitoramento e gerenciamento, o usuário deve estar autenticado no sistema, a qual é feita pela utilização de um usuário e uma senha, conforme pode ser visualizado na Figura 10. O sistema de login é um sistema básico de segurança, que tem por finalidade evitar o acesso de pessoas não autorizadas ao sistema.



Figura 10: Página de login para acesso às páginas administrativas
 Fonte: O autor

Com a autenticação do usuário, são habilitados os demais itens do menu, assim como as demais páginas do sistema. Com a utilização desta área de navegação, através do link "Reservatórios", é possível visualizar uma página com as informações referentes às cisternas que estão sendo gerenciadas pelo sistema, apresentando de forma gráfica as informações inerentes à nível, conforme pode ser observado na Figura 11.

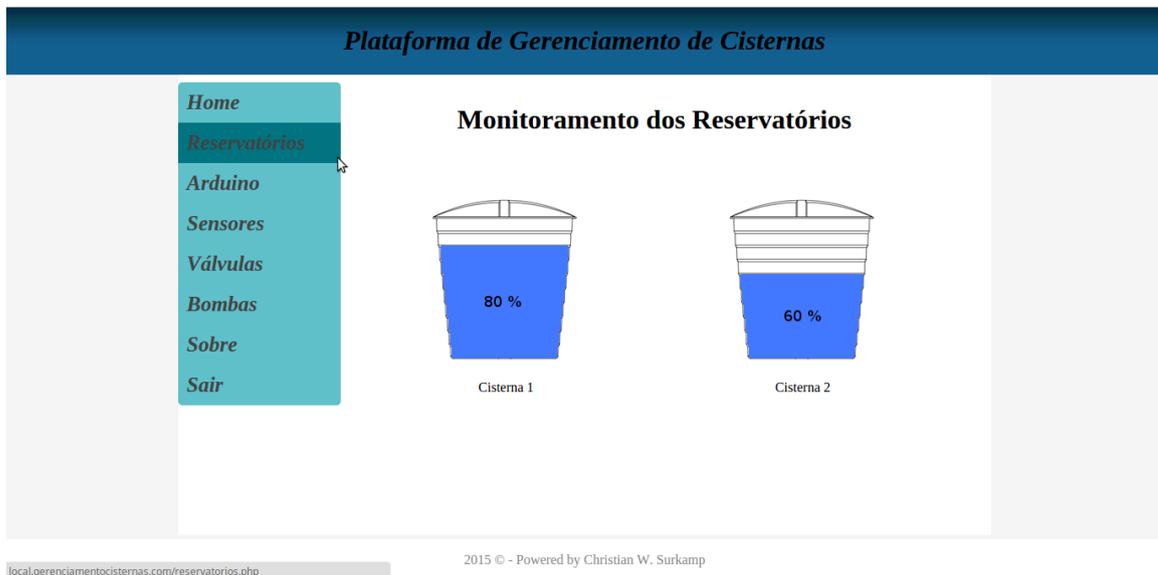


Figura 11: Visualização das informações referentes a nível das cisternas
 Fonte: O autor

Para facilitar o entendimento de todo o esquema, assim como a identificação de todos os componentes, suas ligações e seus respectivos locais de instalação, foram adicionadas

páginas de listagem, as quais podem ser observadas nas Figuras 12, 13, 14 e 15.

Plataforma de Gerenciamento de Cisternas

Home
Reservatórios
Arduino
Sensores
Válvulas
Bombas
Sobre
Sair

Listagem de Arduino

ID Arduino	Nome	Local	Url	Modelo
1	Arduino Mega	Cisterna 1	http://local.arduino.com.br	Arduino Mega 2560
2	Arduino Mega 2	Cisterna 2	http://local.arduino.com.br/2	Arduino Mega 2560

2015 © - Powered by Christian W. Surkamp

local.gerenciamentocisternas.com/arduino.php

Figura 12: Visualização das informações referentes aos arduinos

Fonte: O autor

Plataforma de Gerenciamento de Cisternas

Home
Reservatórios
Arduino
Sensores
Válvulas
Bombas
Sobre
Sair

Listagem de Sensores

ID Sensor	Nome	Valor	Localização	id_arduino
1	Sensor 1	1	Cisterna 1	1
2	Sensor 2	1	Cisterna 1	1
3	Sensor 3	1	Cisterna 1	1
4	Sensor 4	1	Cisterna 1	1
5	Sensor 5	0	Cisterna 1	1
6	Sensor 1	1	Cisterna 2	2
7	Sensor 2	1	Cisterna 2	2
8	Sensor 3	1	Cisterna 2	2
9	Sensor 4	0	Cisterna 2	2
10	Sensor 5	0	Cisterna 2	2

2015 © - Powered by Christian W. Surkamp

local.gerenciamentocisternas.com/sensores.php

Figura 13: Visualização das informações referentes aos sensores de nível

Fonte: O autor

5.3 FUNCIONAMENTO DO PROTÓTIPO

De acordo o diagrama da Figura 8, pode-se observar o esquema de ligação dos sensores de nível aos reservatórios. Cada reservatório deve possuir uma tomada d'água, oriunda de uma fonte alternativa, que não as águas das chuvas. Os sensores são dispostos na vertical, em uma das laterais do reservatório, dispostos de uma maneira que, cada sensor represente 20% do

Plataforma de Gerenciamento de Cisternas

Listagem de Válvulas

ID Válvula	Nome	Estado	Localização	id_arduino
1	Valvula 1	Fechada	Cisterna 1	1
2	Valvula 2	Fechada	Cisterna 2	2
3	Valvula 3	Fechada	Cisterna 1	1
4	Valvula 4	Aberta	Cisterna 2	2

local.gerenciamentocisternas.com/valvulas.php

2015 © - Powered by Christian W. Surkamp

Figura 14: Visualização das informações referentes às válvulas

Fonte: O autor

volume total do reservatório. As entradas d'água de cada reservatório é independente. A Figura 16 apresenta o protótipo desenvolvido para demonstrar o funcionamento do projeto

O controle dos níveis dos tanques se dá pelo algoritmo implementado no arduino, podendo ser monitorado através do sistema supervisor Web. Foi programado para que, quando o nível não atingir os sensores que representam 60% da capacidade do reservatório, o arduino deve enviar um comando para as devidas válvulas solenoides, para que as mesmas permitam a passagem de água pelo seu orifício interno. Foi estabelecido também que, no momento em que o nível do reservatório atinja os sensores que representam 80% da capacidade da cisterna, as entradas de água devem ser interrompidas, a fim de evitar que ocorram desperdícios. A fim de representar as válvulas solenóides, foi feita a utilização de leds coloridos no protótipo, uma vez que os comandos enviados para válvulas solenóides e para um led, pelo arduino, são os mesmos.

As bombas são ativadas de acordo com as necessidades de utilização de água, parte esta que foge ao escopo do projeto. A ação que o sistema pode tomar sobre as bombas, é impedir o funcionamento das mesmas, quando os níveis das cisternas se encontrarem abaixo de 20%, representado pelo sensor mais baixo do reservatório. De maneira semelhante às válvulas, para a representação das bombas no protótipo, também foram utilizados alguns leds coloridos, conforme pode ser observado na Figura 16, uma vez que o comando enviado pelo arduino para ligar um led e uma bomba, são os mesmos.

Cabe enfatizar que, a lógica desenvolvida para o funcionamento dos tanques é a mesma,

ID Bomba	Nome	Situação	Localização	id_arduino
1	Bomba 1	Em funcionamento	Cisterna 1	1
2	Bomba 2	Desligada	Cisterna 2	2

Figura 15: Visualização das informações referentes às bombas

Fonte: O autor

entretanto, cada um dos reservatórios opera de maneira independente.

Os valores utilizados como parâmetros de funcionamento do protótipo foram preestabelecidos para os apresentados. Entretanto, os mesmos podem ser configurados pelos usuários do sistema pela interface Web.

A ligação entre as partes ocorreu de maneira simples, com a utilização de jumpers variados, com o auxílio de uma protoboard a fim de facilitar as ligações(Figura 17).

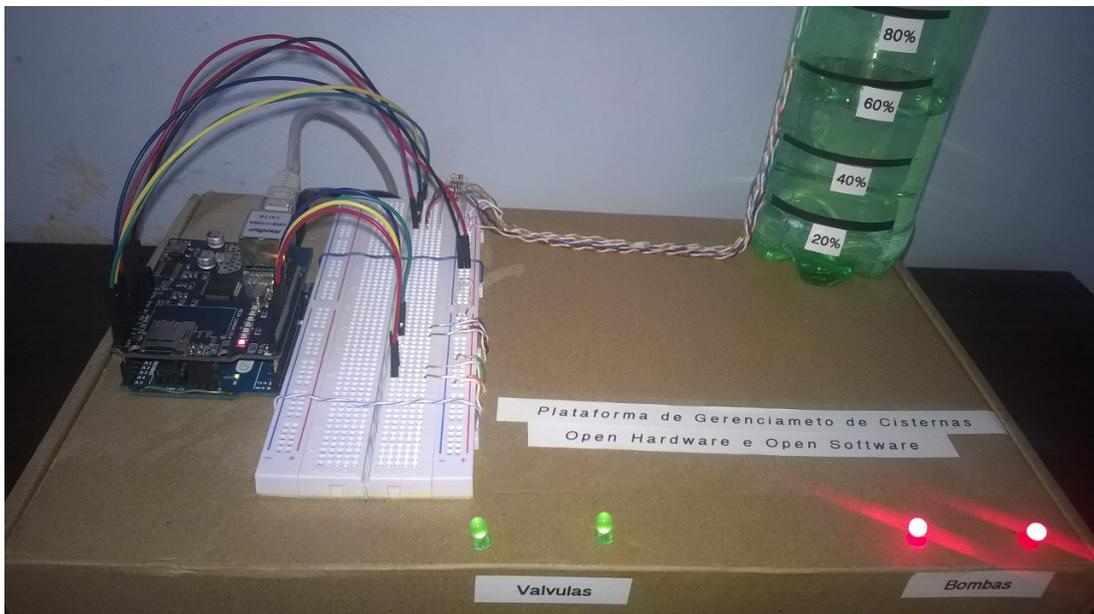


Figura 16: Protótipo desenvolvido para a aplicação
Fonte: O autor

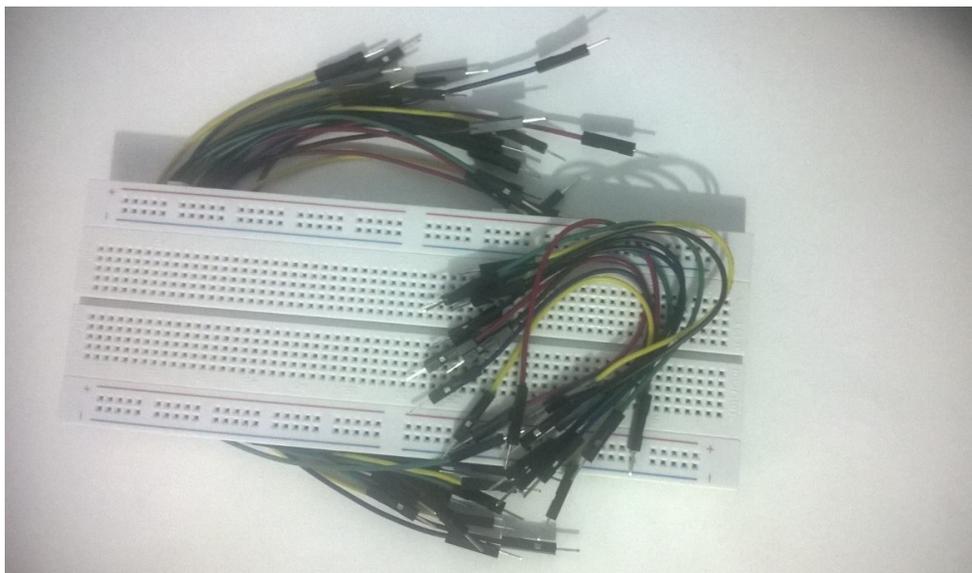


Figura 17: Protoboard e Jumpers Variados utilizados nas conexões
Fonte: O autor

6 CONSIDERAÇÕES FINAIS

O presente trabalho apresentou o projeto de uma aplicação que visa otimizar o aproveitamento de água de cisternas, baseando-se na necessidade de economia com recursos hídricos, visando a diminuição do desperdício deste recurso.

Como grande desafio para este projeto, foi a necessidade de manter o baixo custo do mesmo, a fim de evitar desperdícios de recursos, prezando sempre pela qualidade e robustez da plataforma de gerenciamento.

Desta maneira, fazendo uso do referencial teórico levantado, buscando soluções para as dificuldades encontradas em conjunto com os orientadores, foi possível obter êxito na montagem de um sensor de baixo custo, mas que se mostrou eficiente para o seu propósito.

Cabe ressaltar que foi muito satisfatório trabalhar com programação, aplicada à automação, uma vez que foi possível aplicar os conceitos abordados no decorrer do curso, para a resolução de um problema recorrente em vários locais nos dias de hoje.

Como sugestão de trabalhos futuros, em termos de sistema Web, sugere-se o desenvolvimento de um sistema alternativo, voltado ao acesso por dispositivos móveis, melhorando a renderização e disposição das informações. Outro aspecto interessante seria a implementação de um histórico de eventos, possibilitando a geração de relatórios baseados nos volumes registrados, assim como as aberturas e fechamentos das válvulas, juntamente com o seu tempo ativo.

Em termos de algoritmo desenvolvido para o arduino, sugere-se a melhoria do mesmo, tornando-o mais dinâmico para possíveis expansões do projeto, uma vez que seu código fonte é disponibilizado como Apêndice A..

REFERÊNCIAS

- ARDUINO – home page. Acessado em 13/06/2015. Disponível em: <<http://www.arduino.cc>>.
- BATISTA, H. R. **Sistema de Irrigação de Plantas em Ambientes Residenciais**. Monografia (Trabalho de Conclusão de Curso) — Sistemas da Informação - Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2014.
- BOLTON, W. **Mecatrônica: Uma abordagem multidisciplinar**. Bookman, 2010. ISBN 9788577807123. Disponível em: <<https://books.google.com.br/books?id=u2Zzjhq-LOgC>>.
- CARVALHO, M. F. P. R. de (Ed.). **AUTOMAÇÃO E CONTROLE RESIDENCIAL VIA INTERNET UTILIZANDO ARDUINO**. Rio de Janeiro: [s.n.], 2011.
- HACKENHAAR, J.; ZANELLA, R.; CARDOSO, T. Um comparativo entre php e jsp: definindo a melhor aplicação para o desenvolvimento de projetos web. **Revista iTEC**, v. 1, p. 32 – 36, 2010.
- LIMA, C. B. de; SCHWARZ, L. Kit didático para trabalho com os microcontroladores avr – kit atmega++. **Revista Ilha Digital**, v. 1, p. 93 – 99, 2009.
- MARCHESAN, M. **Sistema de Monitoramento Residencial Utilizando a Plataforma Arduino**. Monografia (Trabalho de Conclusão de Curso) — Tecnologia em Redes de Computadores - Universidade Federal de Santa Maria, Santa Maria, 2012.
- MARTINS, N. A. **Sistemas Microcontrolados: Uma Abordagem com o Microcontrolador PIC 16F84**. São Paulo: Novatec, 2005. ISBN 85-7522-074-8.
- MORETTO, D. A. **SISTEMA DIDÁTICO DE CONTROLE DE NÍVEL COM TANQUES ACOPLADOS**. Monografia (Trabalho de Conclusão de Curso) — Bacharel em Informática - Universidade do Planalto Catarinense, Lages, 2005.
- PETRUZELLA, F. **Motores Elétricos e Acionamentos: Série Tekne**. [s.n.], 2013. (Tekne). ISBN 9788580552584. Disponível em: <<https://books.google.com.br/books?id=4xw4AgAAQBAJ>>.
- RAMAKRISHNAN, R.; GEHRKE, J. **Sistemas de gerenciamento de banco de dados - 3.ed.:** McGraw Hill Brasil, 2008. ISBN 9788563308771. Disponível em: <<https://books.google.com.br/books?id=COUJpkH5v38C>>.
- SILVA, D. F. d. **Sistema de comunicação Bluetooth utilizando microcontrolador**. Monografia (Trabalho de Conclusão de Curso) — Escola Politécnica de Pernambuco – Universidade de Pernambuco, 2009.
- UNESCO, U. N. W. W. A. P. **The United Nations World Water Development Report 2015: Water for a Sustainable World**. Paris: [s.n.], 2015.
- WELLING, L.; THOMSON, L. **PHP e MySQL - Desenvolvimento Web**. São Paulo: Elsevier, 2005.

Apêndice A – Algoritmo desenvolvido para o Arduino

```
#include "SPI.h"
#include "Ethernet.h"
#include "sha1.h"
#include "mysql.h"

//constantes para definição de portas de conexão dos componentes
const int SensorAgua1 = 2;
const int SensorAgua2 = 3;
const int SensorAgua3 = 4;
const int SensorAgua4 = 5;
const int pump1 = 4;
const int pump2 = 3;
const int valve1 = 6;
const int valve2 = 5;

//variáveis para leitura dos sensores
int leituraSensor1 = 0;
int leituraSensor2 = 0;
int leituraSensor3 = 0;
int leituraSensor4 = 0;

//variáveis de controle
int statusSensor1 = 0;
int statusSensor2 = 0;
int statusSensor3 = 0;
int statusSensor4 = 0;
int statusValve1 = 0;
int statusValve2 = 0;
int statusPump1 = 0;
int statusPump2 = 0;
int nivelCisterna = 0;

//parâmetros iniciais para funcionamento do sistema
int nivelFuncionamentoBombas = 20;
int nivelAberturaValvulas = 40;
int nivelFechamentoValvulas = 60;

byte mac_addr[] = { 0xDE, 0xBE, 0xAD, 0xEF, 0xFE, 0xED };
//endereço mac do ethernet shield
IPAddress ip(192,168,1,13); //ip do arduino
IPAddress server_addr(192, 168, 1, 2); //ip do servidor MySQL

Connector my_conn;

//dados de acesso ao banco de dados
char user[] = "root";
char password[] = "surkamp";
```

```

//variáveis para atualização dos dados no banco
char UPDATE_VALVE1[] =
  "UPDATE sistema_de_monitoramento.valve SET open = %d WHERE
id_valve = 1";
char UPDATE_VALVE2[] =
  "UPDATE sistema_de_monitoramento.valve SET open = %d WHERE
id_valve = 2";

char UPDATE_PUMP1[] =
  "UPDATE sistema_de_monitoramento.pump SET active = %d WHERE
id_pump = 1";
char UPDATE_PUMP2[] =
  "UPDATE sistema_de_monitoramento.pump SET active = %d WHERE
id_pump = 2";

char UPDATE_SENSOR1[] =
  "UPDATE sistema_de_monitoramento.sensor SET value = %d WHERE
id_sensor = 1";
char UPDATE_SENSOR2[] =
  "UPDATE sistema_de_monitoramento.sensor SET value = %d WHERE
id_sensor = 2";
char UPDATE_SENSOR3[] =
  "UPDATE sistema_de_monitoramento.sensor SET value = %d WHERE
id_sensor = 3";
char UPDATE_SENSOR4[] =
  "UPDATE sistema_de_monitoramento.sensor SET value = %d WHERE
id_sensor = 4";

char UPDATE_CISTERN1[] =
  "UPDATE sistema_de_monitoramento.cistern SET level = %d WHERE
id_cistern = 1";

void setup(){
  Ethernet.begin(mac_addr);
  Serial.begin(115200);
  Serial.print("Arduino IP ");
  Serial.println(Ethernet.localIP());
  Serial.println("Conecting...");

  if (my_conn.mysql_connect(server_addr, 3306, user, password))
    Serial.println("Connect Success!");
  else
    Serial.println("Connect failed!");

  pinMode(pump1, OUTPUT);
  pinMode(pump2, OUTPUT);
  pinMode(valve1, OUTPUT);
  pinMode(valve2, OUTPUT);
  delay(500);
}

```

```

void loop(){
    //realiza leitura dos sensores
    leituraSensor1 = analogRead(SensorAgua1);
    leituraSensor2 = analogRead(SensorAgua2);
    leituraSensor3 = analogRead(SensorAgua3);
    leituraSensor4 = analogRead(SensorAgua4);

    calculaNivel(); //verifica nível da cisterna

    verificaParametros(); //atualiza parâmetros de funcionamento do
sistema

    verificaBombas(); //controla o funcionamento das bombas

    verificaValvulas(); //controla o funcionamento das válvulas

    atualizaBanco(); //atualiza o estado dos componentes no banco de
dados

    delay(2000);
}

```

```

void atualizaBanco(){
    //variáveis utilizadas na preparação das queries
    char letterChar[10];
    char queryValve1[100];
    char queryValve2[100];
    char queryPump1[100];
    char queryPump2[100];
    char querySensor1[100];
    char querySensor2[100];
    char querySensor3[100];
    char querySensor4[100];
    char queryCistern1[100];

    //preparando queries para atualização do banco
    dtostrf(statusValve1, 1, 1, letterChar);
    sprintf(queryValve1, UPDATE_VALVE1, statusValve1);

    dtostrf(statusValve2, 1, 1, letterChar);
    sprintf(queryValve2, UPDATE_VALVE2, statusValve2);

    dtostrf(statusPump1, 1, 1, letterChar);
    sprintf(queryPump1, UPDATE_PUMP1, statusPump1);

    dtostrf(statusPump2, 1, 1, letterChar);
    sprintf(queryPump2, UPDATE_PUMP2, statusPump2);

    dtostrf(statusSensor1, 1, 1, letterChar);
    sprintf(querySensor1, UPDATE_SENSOR1, statusSensor1);
}

```

```

dtostrf(statusSensor2, 1, 1, letterChar);
sprintf(querySensor2, UPDATE_SENSOR2, statusSensor2);

dtostrf(statusSensor3, 1, 1, letterChar);
sprintf(querySensor3, UPDATE_SENSOR3, statusSensor3);

dtostrf(statusSensor4, 1, 1, letterChar);
sprintf(querySensor4, UPDATE_SENSOR4, statusSensor4);

dtostrf(nivelCisterna, 1, 1, letterChar);
sprintf(queryCistern1, UPDATE_CISTERN1, nivelCisterna);

if (my_conn.is_connected()) {
    //atualiza os dados no banco remoto
    my_conn.cmd_query(queryValve1);
    my_conn.cmd_query(queryValve2);
    my_conn.cmd_query(queryPump1);
    my_conn.cmd_query(queryPump2);
    my_conn.cmd_query(querySensor1);
    my_conn.cmd_query(querySensor2);
    my_conn.cmd_query(querySensor3);
    my_conn.cmd_query(querySensor4);
    my_conn.cmd_query(queryCistern1);
    Serial.println("Updating database OK!");
} else {
    my_conn.disconnect();
    Serial.println("Connecting...");
    if (my_conn.mysql_connect(server_addr, 3306, user, password))
    {
        delay(500);
        Serial.println("Success!");
    } else {
        Serial.println("Connect failed!");
    }
}
}

void calculaNivel(){
    if (leituraSensor1 > 50)
        statusSensor1 = 1;
    else
        statusSensor1 = 0;

    if (leituraSensor2 > 50)
        statusSensor2 = 1;
    else
        statusSensor2 = 0;

    if (leituraSensor3 > 50)
        statusSensor3 = 1;
    else
        statusSensor3 = 0;
}

```

```

if (leituraSensor4 > 50)
    statusSensor4 = 1;
else
    statusSensor4 = 0;

//verifica o nível propriamente dito
if (statusSensor4 == 1)
    nivelCisterna = 80;
else if (statusSensor3 == 1)
    nivelCisterna = 60;
else if (statusSensor2 == 1)
    nivelCisterna = 40;
else if (statusSensor1 == 1)
    nivelCisterna = 20;
else
    nivelCisterna = 00;
}

//impede o funcionamento das bombas em caso de reservatório vazio
void verificaBombas(){
    if(nivelCisterna >= nivelFuncionamentoBombas){
        digitalWrite(pump1, HIGH);
        digitalWrite(pump2, HIGH);
        statusPump1 = 1;
        statusPump2 = 1;
    } else {
        digitalWrite(pump1, LOW);
        digitalWrite(pump2, LOW);
        statusPump1 = 0;
        statusPump2 = 0;
    }
}

void verificaValvulas(){
    if(nivelCisterna <= nivelAberturaValvulas){
        digitalWrite(valve1, HIGH);
        digitalWrite(valve2, HIGH);
        statusValve1 = 1;
        statusValve2 = 1;
    }
}

if (nivelCisterna >= nivelFechamentoValvulas){
    digitalWrite(valve1, LOW);
    digitalWrite(valve2, LOW);
    statusValve1 = 0;
    statusValve2 = 0;
}
}

```

```

//atualização dos parâmetros de funcionamento do sistema
void verificaParametros(){
  int cont = 0;
  char select_values[] = "SELECT value FROM
sistema_de_monitoramento.operatingParameter";

  if (my_conn.is_connected()) {

    my_conn.cmd_query(select_values);
    my_conn.get_columns();

    row_values *row = NULL;
    do {
      row = my_conn.get_next_row();

      if (row != NULL) {
        if(cont == 0){
          nivelFuncionamentoBombas = atoi(row->values[0]);
        } else if(cont == 1){
          nivelAberturaValvulas = atoi(row->values[0]);
        } else if(cont == 2){
          nivelFechamentoValvulas = atoi(row->values[0]);
        }
      }
      my_conn.free_row_buffer();
      cont = cont + 1;
    } while (row != NULL);
    my_conn.free_columns_buffer();
    delay(2000);
  } else {
    my_conn.disconnect();
    Serial.println("Connecting...");
    if (my_conn.mysql_connect(server_addr, 3306, user, password))
  {
    delay(500);
    Serial.println("Success!");
  } else {
    Serial.println("Connect failed!");
  }
  }
}
}

```