



TRABALHO DE GRADUAÇÃO

**AUTOMAÇÃO DE REDES HIDRÁULICAS E
REUSO DE ÁGUA NO EDIFÍCIO NZEB COM
SISTEMAS SUPERVISÓRIOS**

**José Flávio Neto
Lucas Alcides de Assis Santos**

Brasília, Dezembro de 2015

UNIVERSIDADE DE BRASÍLIA

**FACULDADE DE TECNOLOGIA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E
AUTOMAÇÃO**

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

AUTOMAÇÃO DE REDES HIDRÁULICAS E REUSO DE ÁGUA NO EDIFÍCIO NZEB COM SISTEMAS SUPERVISÓRIOS

José Flávio Neto
Lucas Alcides de Assis Santos

Relatório submetido como requisito parcial para obtenção
do grau de Engenheiro de Controle e Automação.

Banca Examinadora

Prof. Marco A. E. Coelho, UnB/ENE
(Orientador)

Prof. José A. R. Vargas, UnB/ENE

Prof. Leonardo R. A. X. de Menezes, UnB/ENE

Brasília, Dezembro de 2015

FICHA CATALOGRÁFICA

JOSÉ, FLÁVIO NETO; LUCAS, ALCIDES DE ASSIS SANTOS	
Automação de redes hidráulicas e reuso de água no Edifício NZEB com sistemas supervisórios,	
[Distrito Federal] 2015.	
x, 93p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2015). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.	
1. Automação	2. Sistemas supervisórios
3. Redes Hidráulicas	4. Reaproveitamento de efluentes
I. Mecatrônica/FT/UnB	II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

NETO, J.F.; SANTOS, L.A.A., 2015. Automação de redes hidráulicas e reuso de água no Edifício NZEB com sistemas supervisórios. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-nº 16, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 93p.

CESSÃO DE DIREITOS

AUTOR: José Flávio Neto; Lucas Alcides de Assis Santos.

TÍTULO DO TRABALHO DE GRADUAÇÃO: Automação de redes hidráulicas e reuso de água no Edifício NZEB com sistemas supervisórios.

GRAU: Engenheiro

ANO: 2015

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

José Flávio Neto

Lucas Alcides de Assis Santos

AGRADECIMENTOS

Eu, José Flávio Neto, sou imensamente agradecido à Deus por toda força que me foi dada e pela possibilidade de vivenciar tal oportunidade acadêmica. Em segundo lugar, à toda minha família que sempre me deu base para alcançar meus objetivos, em especial os meus pais, Gleison e Suzana, e minha tia, Maria Lira, pelo imenso apoio moral e financeiro, sem os quais de maneira alguma eu teria concluído essa jornada tão árdua, porém gratificante.

Todos os docentes da Universidade de Brasília, assim como todos os seus funcionários contribuíram de maneira significativa nesta formação, assim demonstro, também, minha imensa gratidão à todos e minha promessa que jamais desistirei pela busca do conhecimento e pela aplicação do mesmo em nosso mundo em prol da humanidade.

Muito Obrigado!

“O único lugar onde sucesso vem antes do trabalho é no dicionário”.

Albert Einstein.

José Flávio.

Em primeiro lugar e acima de tudo, agradeço ao meu criador e sustentador, o Deus único. Em sequência, agradeço ao meu pai, Antônio, incentivador maior e inicial, que permaneceu firme em suas posições e sempre presente em minha vida. À minha irmã, Mariana, pela convivência e aprendizado conjunto, além do companheirismo em meu crescimento. À minha mãe, Lilian, pelo carinho e persistência maternos, dignos de admiração.

Agradeço pelo direcionamento e compreensão do meu orientador, Prof. Egito, que sempre provou-se calmo e apto a tomar decisões acertadas.

Há tanto a ser dito a tantos que me auxiliaram e ampararam em meu caminho pela universidade. Àqueles professores que fizeram a diferença, aos colegas que me ensinaram e comigo aprenderam, aos amigos que levarei para sempre. Muito obrigado!

Certamente todos aqueles que desejam meu bem contribuíram para que eu alcançasse esse momento, então, embora não haja espaço para citar todos, um agradecimento especial à turma 26, que me acolheu e da qual tornei-me parte.

Ao meu parceiro de trabalho, Neto, que comigo conquistou o último obstáculo desta empreitada.

À Beatriz, que tolerou a ausência temporária por um bem maior. Porque meu último pensamento sempre será seu.

Muito Obrigado!

“Todo ser que respira, louve ao Senhor”.

Salmos 150.6.

Lucas Alcides.

RESUMO

O foco do presente texto encontra-se em expor e explicar uma abordagem de sistemas de distribuição de recursos hídricos direcionada à redução de dispêndios desnecessários. Esse enfoque é obtido por meio da utilização de um sistema supervisorio para controle e aferição do volume hídrico demandado por um edifício de patamar residencial ou público. São apresentados detalhes para o desenvolvimento desse sistema, assim como embasamento para futuras aplicações hídricas no contexto de idealização de construções com custo energético virtualmente nulo (*net-zero energy building*). Além disso, a relevância do reaproveitamento de efluentes outrora desprezados é vital para este projeto. O resultado final deve se observar em uma apresentação de fácil implementação, baixo custo e aplicações eficazes e eficientes.

Palavras Chave: redução, sistema supervisorio, construções, reaproveitamento de efluentes, eficazes e eficientes.

ABSTRACT

The present text focus is found in exposing and explaining an approach of hydrous resources distribution systems directed to the reduction of unnecessary expenses. This focus is obtained by means of using a supervisory system to control and admeasurement of the hydrous volume demanded for residential or public level buildings. Details for the development of this system are shown, as well as basis for future applications on the context of idealizing buildings with virtually null energetic costs (*net-zero energy building*). Besides, the relevance of reusing effluents once forgotten is vital to this project. The final result must be observed in a presentation of easy implementing, low cost and effective and efficient application.

Keywords: reduction; supervisory system; buildings; effluent reuse; effective and efficient;

SUMÁRIO

1	CAPÍTULO 1 - INTRODUÇÃO	1
1.1	MOTIVAÇÃO	1
1.2	NORMAS	2
1.3	MÉTODOS DE TRATAMENTO	5
1.4	SISTEMAS SUPERVISÓRIOS	10
1.4.1	Introdução	10
1.4.2	Componentes Físicos	11
1.4.3	Componentes Lógicos	12
1.4.4	Modos de Comunicação	13
1.5	MODELAGEM	14
2	CAPÍTULO 2 – ELABORAÇÃO DO SISTEMA	15
2.1	AQUISIÇÃO DE DADOS	15
2.1.1	Sensor de Fluxo	15
2.1.2	Sensor de Nível e Capacidade	16
2.2	ATUADORES	18
2.2.1	Atuadores de Vazão	18
2.2.2	Atuadores de Volume	20
2.3	PROCESSAMENTO DE DADOS	21
2.3.1	Arduino e CLP	21
2.3.2	Supervisório	24
2.4	COMUNICAÇÃO ENTRE DISPOSITIVOS	26
3	CAPÍTULO 3 - SIMULAÇÃO	36
3.1	CIRCUITOS ELETRÔNICOS	36
3.2	COMUNICAÇÃO	38
3.3	EXECUÇÃO	41
4	CONSIDERAÇÕES FINAIS	59
	REFERENCIAS BIBLIOGRAFICAS	61
	ANEXOS	63

LISTA DE FIGURAS

Figura 1.1 - Formas potenciais de reuso de água.....	6
Figura 1.2 - Sistema de Tratamento de Água Cinza	9
Figura 1.3 - Sistema de supervisão e controle.....	12
Figura 2.1 - Sensor de Fluxo F-S201.....	16
Figura 2.2 - Sensor de Nível para Líquidos LA16M-40	17
Figura 2.3 - Válvula Solenoide de Duas Vias.....	19
Figura 2.4 - Válvula Solenoide de Três Vias.....	19
Figura 2.5 - Bomba Hidráulica Modelo	21
Figura 2.6 - Arduino Mega 2560	22
Figura 2.7 - Controlador Lógico Programável da Altus, série AL-2000	23
Figura 2.8 - Disposição de um Supervisório ScadaBR	24
Figura 2.9 - Ícones de um Supervisório ScadaBR	26
Figura 2.10 - Pirâmide da Automação Industrial	27
Figura 3.1 - Tela de Inicialização do Proteus.....	37
Figura 3.2 - Tela de Inicialização do ISIS	37
Figura 3.3 - Tela de Inicialização do ARES	38
Figura 3.4 - Tela de Inicialização do VSPE.....	39
Figura 3.5 - Criação de Dispositivo: VSPE	40
Figura 3.6 - Especificações do Dispositivo Criado: VSPE.....	40
Figura 3.7 - Passos Finais para Criação de Portas: VSPE	41
Figura 3.8 - Modelo de Rede Hidráulica	42
Figura 3.9 - Circuito elétrico no Proteus 8.....	43
Figura 3.10 - Inserção de Código para Arduino Virtual	45
Figura 3.11 - Localização do Código Hexadecimal para o Arduino Virtual.....	46
Figura 3.12 - Acesso ao ScadaBR	47
Figura 3.13 - Edição de Data Sources.....	47
Figura 3.14 - Edição de Data Sources - Opções.....	48
Figura 3.15 - Lista de Data Points	49
Figura 3.16 - Edição de Data Points.....	49
Figura 3.17 - Ícone de Acesso às Watch Lists.....	50
Figura 3.18 - Seleção de Watch List.....	50
Figura 3.19 - Ícone de Acesso às Representações Gráficas	51
Figura 3.20 - Seleção de Representação Gráfica	51
Figura 3.21 - Tela de Supervisão – Representação Gráfica	51

Figura 3.22 - Válvula – Representação ScadaBR.....	53
Figura 3.23 - Seleção de Componentes para Representação em Fluxograma.....	55
Figura 3.24 - Fluxograma – Válvula 3 Caminhos – Distribuição de Água Potável/Cinza.....	56
Figura 3.25 - Fluxograma – Válvulas – Cano1 e Cano2	56
Figura 3.26 - Fluxograma – Bomba Hidráulica	57
Figura 3.27 - Fluxograma – Válvula 3 Caminhos - Esgoto.....	57
Figura 3.28 - Fluxograma – Válvula Pluvial	58

LISTA DE TABELAS

Tabela 1.1 - Parâmetros de qualidade da água para reutilização segundo NBR 13969/1997 e SINDUSCON (2005)	5
Tabela 2.1 - Especificações de operação do Sensor de Nível iCOS LA16M-40	18
Tabela 2.2 - Soluções Populares para Ethernet Industrial	33
Tabela 2.3 - Comparação entre Modbus Over TCP/IP e PROFINET.....	34

LISTA DE SÍMBOLOS

Símbolos Latinos

A	Área	[m ²]
P	Pressão	[Pa]
T	Temperatura	[°C]
U	Tensão elétrica	[V]
I	Corrente elétrica	[A]
Pa	Potência Elétrica Aparente	[VA]
Pr	Potência Elétrica Real	[W]
SDT	Sólidos Dissolvidos Totais	[mg/L]

Símbolos Gregos

ω	Velocidade Angular	[rpm]
----------	--------------------	-------

Grupos Adimensionais

uT	Unidade de Turbidez
NMP	Número mais provável

Siglas

ABNT	Associação Brasileira de Normas Técnicas
ANEEL	Agência Nacional de Energia Elétrica
ASHRAE	American Society of Heating, Refrigerating, and Air-Conditioning Engineers
CLP	Controlador Lógico-Programável
DMAES	Departamento Municipal de Água, Esgoto e Saneamento
HMI	Human-Machine Interface
ICSP	In-Circuit Serial Programming
IDE	Integrated Development Environment
MRP	Material Requirements Planning
NBR	Norma Brasileira
NEMA	National Electrical Manufacturers Association
NZEB	<i>Net-zero Energy Building</i>
OCDE	Organização para Cooperação e Desenvolvimento Econômico
PARC	Palo Alto Research Center
SCADA	Supervisory Control and Data Acquisition

SQL	Structured Query Language
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
VSPE	Virtual Serial Ports Emulator

CAPÍTULO 1 - INTRODUÇÃO

Determinar o escopo de um sistema de automação é tarefa preliminar para seu projeto. Para as atribuições de um supervisor que atue sobre encanamentos e pontos de coleta de água em uma residência ou edifício de maior porte, é interessante compreender, primeiro, as possíveis razões para desenvolvimento de um trabalho nesse sentido, além de que benefícios essa ideia poderia trazer para o usuário e para aqueles dentro do alcance de seus efeitos. Após claramente elencadas as variáveis de saída para as quais tal sistema pretende oferecer ajustes, define-se que ferramentas serão utilizadas e qual será a interação entre elas, até que a imagem do seu funcionamento torne-se tangível para completa compreensão dos aspectos operacionais do supervisor.

1.1 MOTIVAÇÃO

As variáveis que sustentam a busca por opções viáveis para complementar a demanda por recursos hídricos são de simples compreensão: há uma alteração no sentido crescente para essa demanda, devido a condições que propiciaram aumento populacional e a mudanças climáticas de nível global. Embora a água seja um recurso renovável, a necessidade humana requer uma qualidade desse bem que não é comum a todos os estados ocupados pelo líquido em seu ciclo. Essa discrepância é ainda mais evidenciada quando há uso indiscriminado, tornando ainda maior a parcela hídrica que ocupa o estado impróprio para consumo.

Um dos dados que fundamenta a noção de quão limitada é a quantidade de água potável é o percentual de água que se encontra nos oceanos e, portanto, é salgada e imprópria para consumo: segundo Bloch (2002), 97% de toda a água da Terra. Já a água doce corresponde a apenas 0,62% de toda essa quantia. Essa grande diferença foi e tem sido ainda mais agravada em decorrência das diferentes ações humanas de interferência no ciclo hidrológico ao longo de nossa história, como construção de novas metrópoles, impermeabilização do solo, desmatamento, modificação do fluxo natural da água para uso em represas e etc.

A distribuição de água doce não é homogênea sobre o planeta, e embora o Brasil esteja situado em uma região favorável com disponibilidade significativa, a distribuição dentre suas regiões também é desigual. Quanto à demanda de cada região, é interessante a relação de oposição à supracitada presença, com maior disponibilidade na região Norte, que concentra apenas 5% da população brasileira, segundo pesquisa realizada pelo Departamento Nacional de Águas e Energia Elétrica, atual ANEEL (a proporção é de 68,5% para a região Norte, 3,3% para a região Nordeste, 6,0% para a região Sudeste, 6,5% para a região Sul e 15,7% para a região Centro-Oeste).

Em um contexto de desenvolvimento tecnológico e implementações voltadas à melhora da qualidade de vida da população, é natural que haja uma prolongação na expectativa de vida humana, assim como é esperado que mais recursos sejam consumidos pela população que se multiplica. Esse aumento se traduz em problemas de abastecimento e soluções paliativas que podem prejudicar mananciais e comprometer essas fontes, tornando essa situação em uma questão de longo prazo. Uma imediata aplicação de novas formas de gerenciamento dos recursos hídricos, integrada a métodos de reutilização e maior aproveitamento de toda a água de que já dispomos seria uma alternativa mais viável para uma visão holística e que compreende todo o ciclo de uso desse recurso natural renovável tão indispensável.

Outros dados que salientam a efetividade da ação individual para influir na economia de água podem ser fornecidos por órgãos que tratam diretamente desse assunto. Como amostra disso, o Departamento Municipal de Água, Esgoto e Saneamento (DMAES) de Ponte Nova/MG, indica que o simples gotejamento lento de uma torneira pode consumir cerca de 400 litros por mês, enquanto um gotejamento rápido pode consumir 1000L/mês e um vazamento contínuo pode gerar um desperdício de até 6500L/mês.

Ainda relativo a desperdícios individuais, torneiras com grande dispersão do jato d'água tornam-se causa comum. A água que é de fato usada é aquela cujo jato está concentrado num diâmetro específico, de forma que todo o volume que excede tal diâmetro, além de não ser utilizado, causa incômodo ao usuário. Assim sendo, o uso de produtos que contenham aeradores é bem-vindo no que se diz à redução do consumo e conseqüente o não desperdício. Esse pequeno instrumento introduz bolhas de ar no jato d'água, reduzindo a tensão superficial da água durante a vazão da torneira e diminuindo os respingos e o desperdício de água. A economia de água pode chegar a 50% e a eficiência do chuveiro e pia continua a mesma, de acordo com o Guia do Consumo Consciente de Água (Redação Ecod, 2015).

Certamente uma ação global de conscientização dos usuários tornaria o processo de gerenciamento da água mais acertado, mas enquanto tal atividade encontra-se fora do escopo das ações governamentais e dos gestores desse bem, há de se considerar a inserção da automação no controle mais efetivo do uso de água, atuando desde o uso consciente do recurso potável advindo da rede, assim como no correto manuseio do ferramental para a realização do reuso da água, verificando o correto tratamento para diferentes aplicações e os níveis de consumo e dados referentes à distribuição.

Para esse tipo de inserção, os conhecimentos e capacidades de avaliação de um sistema modelado por dispositivos característicos da engenharia provam-se fundamentais. A disposição acertada de funcionalidades que atinjam implementação de dificuldade reduzida, eficiência comprovada e custo minimizado caracterizaria um sistema ideal.

1.2 NORMAS

O reciclo de água exige definição de parâmetros e normas de comprovação da sua qualidade. Em um recente seminário cujo cerne era a gestão de água em situações de escassez, a ministra do Meio Ambiente, Izabella Teixeira, mencionou a possibilidade de uma revisitação às regulamentações sobre esse assunto, a fim de incentivar o consumo de água de reuso. Afirmou ainda que não está claro qual seria o formato dessa norma, se uma lei ou decreto, por exemplo. Essa menção esbarra em uma falta de normatização técnica específica para os sistemas de reutilização de água no Brasil. É comum que se adotem padrões referenciais internacionais ou normas produzidas por instituições privadas nessa conjuntura, o que dificulta a aplicação da prática de reciclo no país. O ideal seria uma melhor prescrição de diretrizes e dimensionamento, assim como existem para captação de água da chuva, presentes na Norma Brasileira (NBR 15.527 – Água da chuva – Aproveitamento de coberturas em áreas urbanas para fins não potáveis). Projetos que incluam essa premissa devem atender também às normas técnicas ABNT – NBR 5.626 e NBR 10.844, assim como devem delimitar claramente seu alcance, a população a ser atendida, a determinação da demanda e os estudos das séries históricas e sintéticas das precipitações da região (ABNT, 2007, p.2).

Ainda assim, são relevantes os avanços no município de Curitiba que regulamentaram por lei (Lei 10.785/03) medidas que visavam induzir a conservação da água por meio do uso racional, e de fontes alternativas de abastecimento de água nas novas edificações. A validação da referida lei aconteceu em 2006 (22 de abril), momento em que se determinou a obrigatoriedade para todas as novas edificações de inclusão de equipamentos com funcionalidade para captação de águas pluviais oriundas da cobertura da construção. No entanto, a obrigação do reuso das águas obtidas foi restrito aos prédios comerciais e industriais com área superior a cinco mil metros quadrados. A dificuldade de fiscalização e a já mencionada falta de normatização para propriedades de reciclo provaram-se empecilhos para a verdadeira aplicação dessa lei, de modo que apenas a partir de 2007, com o Decreto nº 212 (29 de março), foram estabelecidas exigências para cada tipo de uso das edificações e passou a vigorar de forma consistente essa iniciativa.

A legislação de Curitiba pode ser, em conjunto com as normas técnicas ASHRAE, uma base para iniciar um estudo de dimensionamento do nosso sistema de reuso. Algumas sugestões indispensáveis para o desenvolvimento de legislações nesse ínterim de pioneirismo, presentes em guias e publicações sobre o tema, são interessantes para compreensão dos pontos que buscamos para dimensionamento e definição do nosso sistema:

- Observação aos preceitos previamente estabelecidas nas legislações federais e estaduais, visto que não desejamos um confronto com legislações já existentes;

- Discussão do tema com os setores acadêmicos e representativos dos profissionais da área, para devido embasamento técnico e científico;
- Investigação dos planos diretores de drenagem urbana, gerenciamento de recursos hídricos e saneamento ambiental para uma adequação a eles;
- Capacitação, orientação e estruturação dos órgãos fiscalizadores responsáveis;
- Discussão com a sociedade para incentivar a participação desta na implantação e fiscalização dos sistemas.

(Ambiente Brasil, 2015, p.1).

O primeiro ponto nos remete à busca de legislações e normas já existentes, o segundo a buscar um contato com profissionais da área ambiental e o terceiro a considerar interferências com planos de gerência do Distrito Federal. Já o quarto e quinto pontos podem ser abstraídos como direcionamento aplicável apenas a órgãos cuja influência é suficiente para lançar mão de ferramentas sociais e contato a órgãos fiscalizadores deste entorno.

Em resposta à primeira recomendação, encontramos a norma NBR 13969:1997, que delimita níveis aceitáveis para diversos parâmetros de pureza da água, classificados de acordo com a finalidade de uso dessa água:

- Classe 1: lavagem de carros e outros usos que requerem contato direto do usuário com a água, com possível aspiração de aerossóis pelo operador.
- Classe 2: lavagens de pisos, calçadas e irrigação dos jardins, manutenção de lagos e canais para fins paisagísticos.
- Classe 3: reuso na descarga dos vasos sanitários;
- Classe 4: reuso nos pomares, cereais, forragens, pastagens para gado e outros cultivos através de escoamento superficial ou por sistema de irrigação pontual.

(Adaptado de ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 1997, p.22)

Nosso projeto engloba apenas as finalidades das classes 3 e 4, o que significa certa liberdade em relação aos parâmetros mencionados.

Há também um manual do SINDUSCON (2005) que acrescenta um parâmetro aos já estabelecidos: cor. Sua classificação difere ligeiramente da proposta pela norma supracitada:

- Classe 1: descarga de vasos sanitários, lavagem de pisos, fins ornamentais, lavagem de roupas e veículos.
- Classe 2: lavagem de agregados, preparação de concretos, compactação de solos e controle de poeira.

A tabela 1.1 apresenta os padrões necessários para reuso segundo a NBR 13969/1997 e o manual do SINDUSCON (2005).

Tabela 1.1: Parâmetros de qualidade da água para reutilização segundo NBR 13969/1997 e SINDUSCON (2005)

Classes		Parâmetros				
NBR 13969/1997	Turbidez (uT)	pH	SDT (mg/L)	Cloro residual (mg/L)	Oxigênio dissolvido (mg/L)	Coliformes termotolerantes (NMP/100mL)
Classe 1	< 5	6,0 e 8,0	< 200	0,5 e 1,5	-	< 200
Classe 2	< 5	-	-	< 0,5	-	< 500
Classe 3	< 10	-	-	-	-	< 500
Classe 4	-	-	-	-	> 2,0	< 5000
SINDUSCON (2005)						
Classe 1	≤2	6,0 e 9,0	≤ 500	≤0,5	-	Não detectáveis
Classe 2	-	6,0 e 9,0	-	-	-	≤ 1000

Fonte: ABNT – Associação Brasileira de Normas Técnicas: NBR 13969/1997 e

SINDUSCON – Sindicato da Indústria de Construção Civil do Estado de São Paulo em “Manual de conservação e reuso da água em edificações” (2005)

Esse conjunto de normas deve ser satisfatório para delimitar um nível aceitável de resultados após a modificação do sistema de distribuição de água de um complexo de edifícios. Uma análise desses dois conjuntos de regras permite a utilização dos limites mais restritos a fim de atender a referências nacionais e internacionais.

As sugestões de desenvolvimento conjunto ao governo e a profissionais da área (segundo e terceiro pontos previamente indicados pelos manuais do Ambiente Brasil) possibilitam um encadeamento de aplicações futuras, de modo a formar parcerias que tornem o racionamento de água por uso de supervisor uma ferramenta nacional.

1.3 MÉTODOS DE TRATAMENTO

Por fatores de definição para uso de água cinza, Hespanhol (2003, p.413) observa que “As possibilidades e formas potenciais de reuso dependem, evidentemente, de características, condições e fatores locais, tais como decisão política, esquemas institucionais, disponibilidade técnica e fatores econômicos, sociais e culturais”. Em reforço às normas que parametrizam esse uso baseado no destino que se dará ao líquido proveniente de reciclo, o autor ainda avalia que fatores relacionados ao processo, como custos de implantação, de manutenção, de operação, a exigência de tratamento, a qualidade da água e

também parâmetros de segurança que venham a ser adotados, dependerão e estarão intimamente ligados à finalidade intencionada.

Das opções descritas por Hespagnol, demonstradas na Fig. 1.1, as mais pertinentes ao nosso intuito são os usos sequenciais à entrada de esgoto doméstico, restritos, para esse caso, em usos urbanos não potáveis.

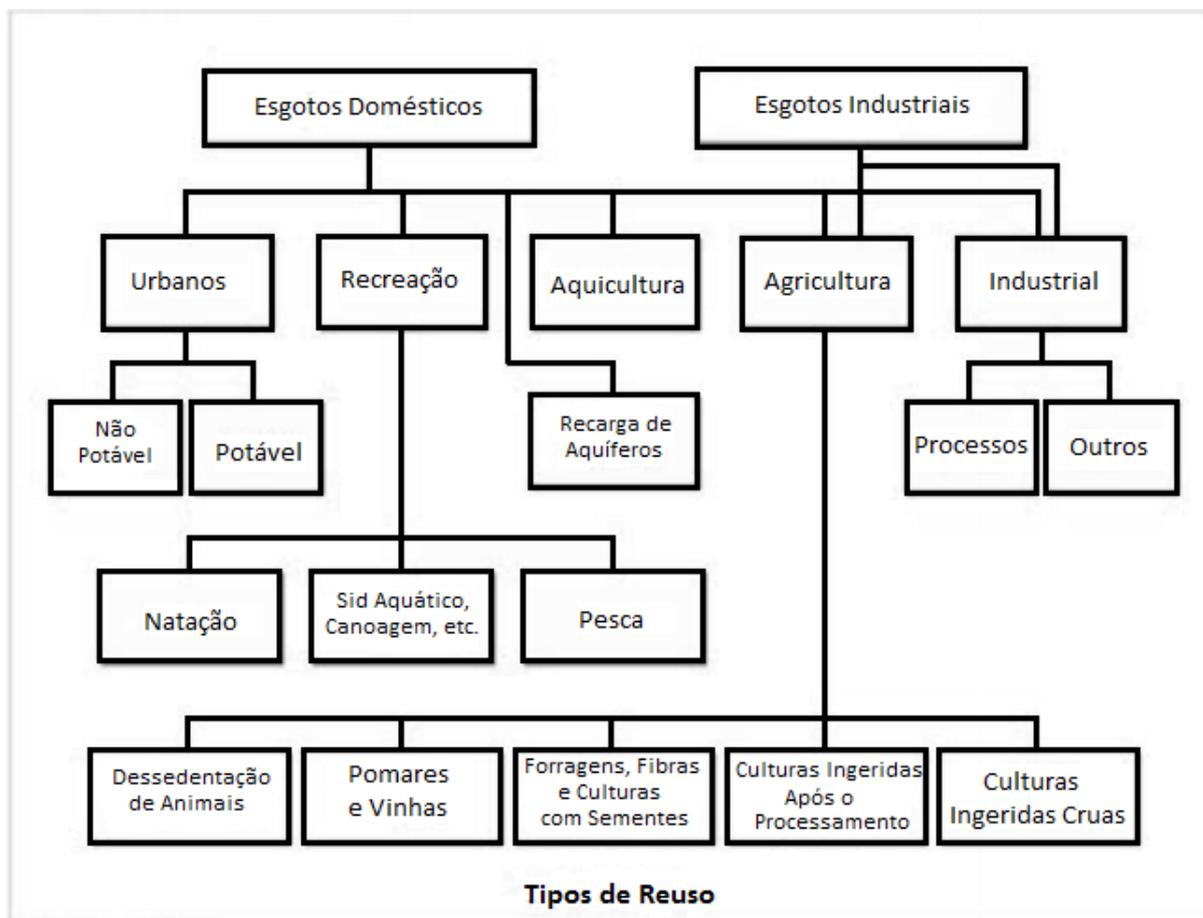


Figura 1.1. Formas potenciais de reuso de água
 FONTE: HESPANHOL (1997) apud HESPANHOL (2003)

A fase seguinte é determinar um método de tratamento efetivo para todos os pormenores que podem influir no funcionamento comum de um edifício em que se instale uma função de reuso de água. Em um enquadramento residencial, podemos mencionar, por exemplo, a exalação de mau odor decorrente de armazenamento de água com teores que a classificam como cinza, ou até a maior incidência de micro-organismos prejudiciais à vivência humana em corpos fluidos com partículas de proveniência alimentar. A diferenciação entre águas de lavagem e águas pluviais também é ponto vital, visto que as concentrações de detritos variam em consonância com a origem do líquido. Assim sendo, considerar-se-ão processos de tratamento distintos para esses corpos fluidos.

Dentre os diversos métodos existentes para purificação dos efluentes derivados das lavagens e enxagues que caracterizam a água cinza, há um consenso quanto à inevitabilidade do uso de não apenas uma etapa, mas de um módulo composto por processos complementares que atuem em propriedades específicas do efluente. Discorreremos aqui sobre duas alternativas.

A primeira opção considera materiais de baixo custo e implementação facilitada para uso pessoal, baseada em um esquema de 2010 (Karin Zan Huke, 2010 apud Johan Van Lengen, 2008). Sua ação se dá a partir do acoplamento de seis reservatórios, cada um com atribuição determinada. Esses recipientes podem ser caixas d'água, caixas de cimento impermeabilizadas ou qualquer outro depósito com capacidades impermeabilizantes similares, isolado do meio externo. Sugere-se por volume o valor de 200 litros de água por dia por pessoa que atue nesse sistema.

O primeiro receptáculo deve estar localizado em nível inferior ao solo, no qual será despejada a água cinza não tratada, para separação natural dos óleos resultantes da gordura e das lavagens de cozinha, além de possíveis óleos corporais enxaguados no banho. O duto de saída desse ambiente deverá se localizar em seu fundo e subir até o topo do ambiente seguinte, de dimensões similares. O processo previsto é a flutuação dos óleos, de densidade inferior à da água e não dissolúveis, para que a amostra subsequente esteja livre da presença desses compostos. Esse primeiro tanque deve possuir uma tampa de manutenção para remoção dos óleos excedentes a cada período apropriado.

A segunda caixa é o tanque de filtragem mineral anaeróbico, com disposição de pedriscos ao fundo (qualquer tipo de material rochoso inerte ocupa bem essa função) e de brita em uma quantidade pelo menos três vezes superior aos fragmentos de rocha inferior. Esse estágio propõe a alcalinização da água e retenção de resíduos sólidos, assim como a interrupção de micro-organismos dependentes de oxigênio. A manutenção recomendada para esse recipiente é a limpeza com jato d'água de grande pressão ao menos uma vez por ano.

A conexão entre o segundo e terceiro tanques deve ser feita de forma semelhante à já mencionada disposição do fundo para o topo. Em relação ao terceiro ambiente, deve atuar como um "tanque vivo", cuja disposição é formada por britas ao fundo (15% da capacidade do tanque), solo misturado e casca de arroz no centro (cerca de 60% da capacidade do tanque) e casca de arroz na camada superior (cerca de 15% da capacidade do tanque). O topo desse ambiente deverá ser aberto e o composto de cascas e solo atuará como meio para plantação de vegetais da família dos papiros, lírios ou juta. Essas espécies são recomendadas pela prolongação de suas raízes e alta capacidade de absorção de resíduos, tornando-as adequadas ao plantio nas condições criadas. Esse processo possui por expectativa a purificação, ou ao menos redução, dos componentes dissolvidos na água, nutrientes

buscados e removidos pelos vegetais aqui presentes. A manutenção deve ser realizada em períodos de um ano e meio, com a troca de todo o conteúdo desse tanque.

Mais uma vez, a conexão entre o terceiro e quarto ambientes é disposta como anteriormente. Esse tanque diferencia-se dos outros por não ser uma caixa d'água ou semelhante isolado. Em detrimento dessa opção, deve-se cavar um buraco no solo com certa de 50 cm de profundidade e dimensões de área de 1,5 m². Em seguida a superfície do buraco deve ser coberta por uma lona, de forma que se crie algo como um pequeno lago, com areia e pedras no fundo (certa de 3 cm de areia e 3 cm de pedras). Plantas aquáticas devem ser posicionadas no fundo do "lago" e plantas de superfície devem ser adicionadas à superfície do espelho d'água. Como esses vegetais se reproduzem de maneira acelerada, o excesso deverá ser removido constantemente. A manutenção nesse caso é relativa apenas à integridade da lona, que deve ser mantida sem rasgos e sem excesso de sujeira, e das pedras, que devem permanecer sem acúmulo de dejetos. O foco deste processo é introduzir oxigênio ao composto, acabando com organismos que sejam intolerantes a ele. As plantas aquáticas também executam uma absorção complementar ao tanque anterior, para redução de nutrientes suspensos na água.

Uma das bordas do lago deve ser acoplada com um filtro de tela e um declive para escoamento natural da água excedente. Essa conexão deve levar a um novo tanque anaeróbico de minerais, montado de forma igual ao segundo ambiente.

Por fim, utiliza-se a conexão inicialmente mencionada para depositar a água purificada em uma caixa d'água com função comum, apenas para armazenar e disponibilizar o líquido já tratado.

O dimensionamento sugerido pela autora (Karin Zan Huke, 2010) é o seguinte:

- 1 a 4 moradores: caixas de 200 litros
- 5 a 7 moradores: caixas de 300 litros
- 8 a 10 moradores: caixas de 400 litros
- 11 a 14 moradores: caixas de 500 litros

O acréscimo de grupos de três pessoas implica um aumento de 100 litros no dimensionamento. A autora ainda indica a possibilidade de não existir um volume de água reaproveitada para suprir as demandas de água cinza, o que exige um sistema de alimentação hídrica híbrida, que não seja exclusivamente dependente do volume reaproveitado para completar a demanda com o abastecimento comum.

Encerrada a primeira descrição de tratamento, consideremos uma segunda alternativa. Esse sistema experimental destina-se também a águas cinzas residenciais e foi concebido e testado pela Dra. Simone May (2009). Seu fluxo propõe entradas automatizadas de efluentes,

controladas pelo sistema de distribuição adaptado de uma residência e suas características o identificam como um sistema de tratamento biológico aeróbico. O pilar deste método é o constructo reator biológico de contato, também chamado de biodisco. Esse apetrecho é munido de discos rotativos que permitem o fornecimento artificial de oxigênio para o meio líquido no qual micro-organismos aeróbios predominam e são capazes de realizar remoção de carga orgânica. Associado a esse foco estão um tanque de equalização, reservatório de decantação, filtro de areia de pressão e sistema de desinfecção. O seguinte esquema (Fig. 1.2) é de propriedade da autora e representa de maneira mais intuitiva essa formação.

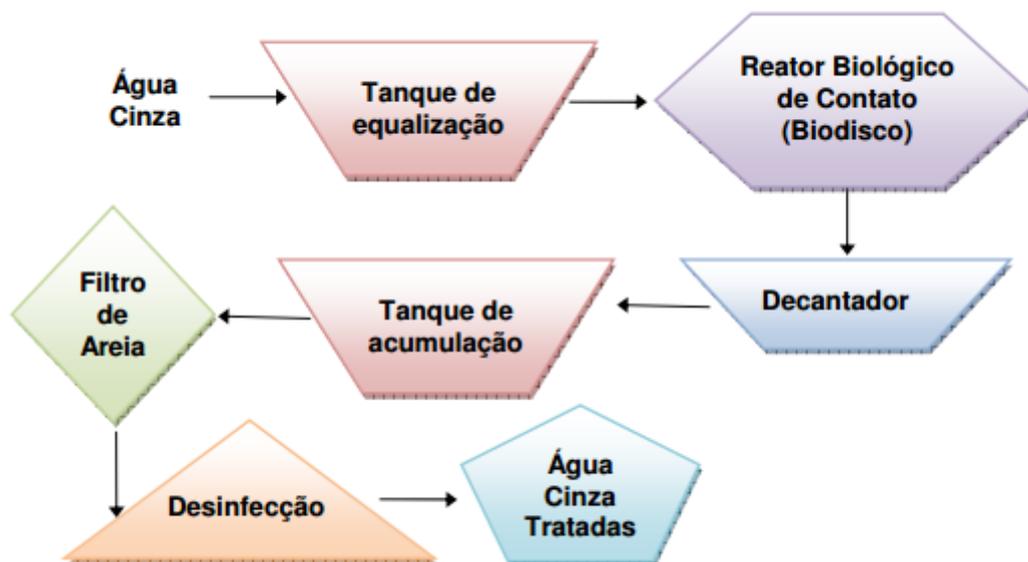


Figura 1.2. Sistema de Tratamento de Água Cinza

FONTE: May (2009)

A explanação desse esquema é relativamente simples: o efluente obtido a partir dos pontos de coleta será direcionado a um tanque de equalização em que um agitador evitará a sedimentação dos detritos presentes no fluido. Em seguida o efluente será conduzido ao reator para degradação da matéria orgânica. As águas cinzas não devem ser armazenadas por longos períodos para limitar a proliferação de organismos a níveis controláveis e reversíveis. Sugere-se, portanto, um máximo de 24 horas de estoque desses líquidos.

O momento seguinte é de sedimentação, em um decantador para depósito do lodo que ali se formara. O tanque de acumulação tem função de padronizar corpos de água com volumes fixos para passagem pelo seguinte filtro de areia, posicionado de maneira a atuar como um filtro rápido de pressão de escoamento descendente. Por fim, as águas tratadas eram desinfetadas com hipoclorito de sódio.

Um Reator Biológico de Contato consiste em um conjunto de rotores compostos por discos de plástico rígido de pequena espessura que trabalham parcialmente submersos (cerca de 40%). Os discos são montados lado a lado, num eixo horizontal, com afastamento

de 5 a 12 mm, formando grandes cilindros. O RBC é acoplado a um sistema de acionamento que faz os discos girarem lentamente a uma velocidade de 1,0 a 1,6 rpm (Mendonça, 2004). Seu princípio de funcionamento é a cultura de bactérias aeróbias nas superfícies dos discos, que, ao girarem, promovem a aeração e imersão das colônias de bactérias no efluente. A oxidação do carbono orgânico contido no fluido acaba por concentrar a reprodução das bactérias, que aderem cada vez mais aos discos.

O constante crescimento das colônias localizadas nas pás rotatórias prejudica a proliferação das bactérias localizadas na parte interna da película, que têm contato com o oxigênio do ar dificultado. As espessas camadas formadas pelas colônias nos discos, denominado biofilme, desprende-se, então, e segue para o ambiente decantador, formando um lodo ativo. Finalmente, o lodo sedimentado e a água tratada prossegue para o restante do tratamento.

Algumas vantagens que essa aparelhagem proporciona podem ser mencionadas:

- Tamanho compacto, quando comparado a outros métodos de tratamento confiáveis;
- Baixa poluição sonora resultante (apenas o ruído do motor);
- Contenção dos odores desagradáveis provenientes das bactérias que são removidas;
- Sua instalação permite posicionamento enclausurado, de maneira a esconder todo o processo e tornar sua aparência mais agradável;
- Baixo consumo de energia.

1.4 SISTEMAS SUPERVISÓRIOS

De acordo com Ana Paula Gonçalves da Silva e Marcelo Salvador, autores do documento “O que são sistemas supervisórios?”, este assunto é dividido em componentes físicos, componentes lógicos, e modos de comunicação. As descrições de cada um desses elementos foram feitas sob a luz deste documento de enorme valor de ensino e encontram-se localizadas nas subseções seguintes. Contudo, algumas modificações foram realizadas para facilitar a interpretação, assim como para adequação ao projeto atual deste sistema.

1.4.1 Introdução

Também chamados de Sistemas SCADA, do inglês *Supervisory Control and Data Acquisition*, sistemas supervisórios são capazes de monitorar grandezas físicas através de uma central de controle, que normalmente é um computador. As grandezas que podem ser monitoradas por esse sistema são geradas regularmente através de sensores de medição

acoplados ao sistema físico, que respondem em forma de sinais para a rede. Esses sinais são então traduzidos para valores perceptíveis no sistema de supervisão, que analisa a situação e gera uma reação apropriada a essa entrada.

No projeto, devido à eficiência e ao fator *open source*, o ScadaBR é o software adotado e que permite a identificação das tags, que são todas as variáveis numéricas ou alfanuméricas envolvidas na aplicação, podendo executar diferentes funções computacionais como operações matemáticas, lógicas ou representar pontos de entrada/saída de dados do processo que está sendo controlado. Neste último caso, essas tags correspondem às variáveis do processo real em si, como por exemplo o nível dos reservatórios ou mesmo o fluxo nos canos da rede hidráulica do projeto, realizando o link entre o sistema e o controlador. São esses valores os responsáveis pela apresentação dos dados ao usuário. O sistema pode também gerar alarmes, relatórios e estatísticas baseados em valores que o usuário predefine como valores críticos e que devem ser alertados. Todos esses dados vão para um banco de dados, que também fica a critério do usuário o modo e tempo de armazenagem.

1.4.2 Componentes Físicos

Os componentes físicos de um sistema de supervisão são, de forma resumida: sensores e atuadores, rede de comunicação, estações remotas (onde são realizadas aquisição e controle) e de monitoração central (sistema computacional SCADA). Os sensores são dispositivos conectados aos equipamentos controlados e monitorados pelos sistemas SCADA, que convertem parâmetros físicos, como por exemplo níveis dos reservatórios do projeto, para sinais analógicos e digitais compreensíveis pela estação remota. Os atuadores, papel desempenhado pelas válvulas do projeto, são utilizados para agir sobre o sistema quando uma dada situação é atingida por uma predefinição no sistema, ligando e desligando determinados equipamentos conforme necessidade do implementador.

O controle e aquisição de dados se iniciam em estações remotas PLCs (Programmable Logic Controllers) e RTUs (Remote Terminal Units), que realizam a leitura dos valores dos componentes a eles ligados. Sua representação no projeto simulado presente no capítulo 3 corresponde a um micro controlador Arduino Uno. Essas estações são utilizadas em instalações em que se deseja realizar monitoramento para desempenhar funcionalidades de leitura de entradas, realização de cálculos e controles, assim como atualização de saídas.

Os PLCs e as RTUs se diferem no que tange à flexibilidade de linguagem, de modo que os primeiros possuem uma maior flexibilidade na linguagem de programação e controle de entradas e saídas, enquanto os seguintes possuem uma arquitetura mais distribuída entre sua unidade de processamento central e os cartões de entradas e saídas, gerando uma maior precisão. A rede de comunicação pode ser implementada através de cabos como Ethernet, fibras ópticas, linhas dial-up, linhas dedicadas, etc.

As estações de monitoração central são responsáveis por recolher as informações que são geradas pelas estações remotas e responder conforme os eventos detectados. Tais estações podem ser centralizadas em um único computador ou distribuídas por uma rede de vários outros computadores, o que permite que as informações não fiquem presas a uma só central, sendo assim compartilhada.

A figura 1.3 exemplifica de forma didática a disposição de um sistema supervisório.

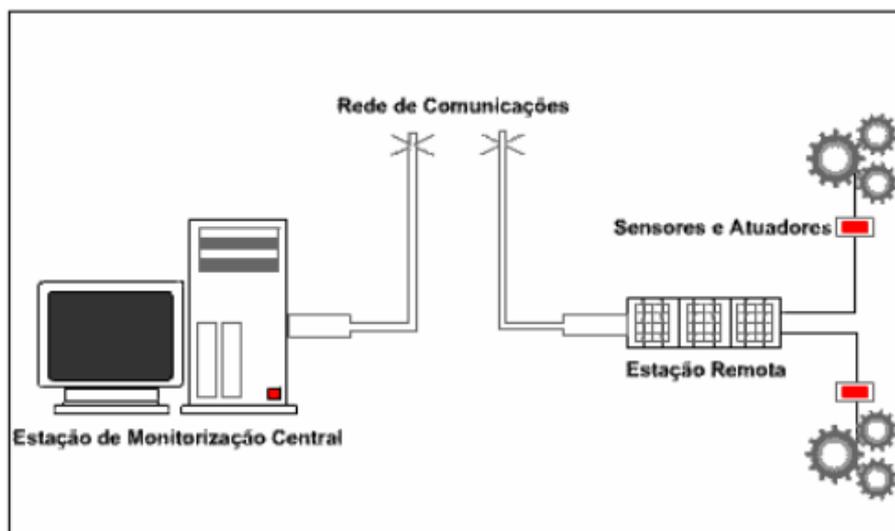


Figura 1.3. Sistema de supervisão e controle

FONTE: Da Silva & Salvador (2005)

1.4.3 Componentes Lógicos

Normalmente, os sistemas SCADA dividem tarefas em blocos ou módulos, que permitem maior ou menor flexibilidade e robustez.

As tarefas são geralmente divididas em:

- Núcleo de processamento;
- Comunicação com PLCs/RTUs;
- Gerenciamento de Alarmes;
- Históricos e Banco de Dados;
- Interface gráfica;
- Relatórios;
- Comunicação com outras estações SCADA;
- Comunicação com Sistemas Externos / Corporativos;
- Outros.

O funcionamento de um SCADA depende dos processos de comunicação entre o núcleo principal, o mestre, e o módulo localizado no local de aquisição de dados, que para o protocolo do projeto é visto como o escravo. O mestre é responsável por distribuir e coordenar o fluxo dessas informações para os outros escravos, até chegarem na forma esperada para o operador do sistema, na interface gráfica ou console de operação com o processo, geralmente acompanhadas de gráficos, animações, relatórios.

Os sistemas SCADA possuem diferentes softwares no mercado, dentre opções pagas e open sources. Algo em comum entre tais sistema é a constante inovação e aprimoramento, entregando confiança e exatidão aos usuários.

1.4.4 Modos de comunicação

A principal funcionalidade de qualquer sistema SCADA está ligada à troca de informações, que podem ser, basicamente:

- Comunicação com os PLCs/RTUs;
- Comunicação com outras estações SCADA;
- Comunicação com outros sistemas.

A comunicação entre o sistema central e os equipamentos de campo é realizada através de um protocolo em comum, no Projeto esse protocolo comum é o Modbus, que implementa a relação Mestre/Escravo. A comunicação por polling, ou Mestre/Escravo faz com que o Mestre tenha controle absoluto das comunicações, assim como seu nome sugere, efetuando sequencialmente o polling de cada estação remota dos escravos, que no modelo half-duplex só respondem ao mestre quando há um pedido. Este modelo half-duplex faz simplesmente o processo de coleta de dados, inexistência de colisões no tráfego da rede, facilidade na detecção de falhas de ligação e uso de estações remotas não inteligentes. Entretanto, é incapaz de comunicar situações ao mestre por iniciativa das estações remotas (escravos), que pode ser feito no modelo full-duplex.

Na comunicação por interrupção, quando o monitoramento dos valores de entrada pelo PLC ou RTU detecta alterações significativas ou valores que ultrapassem os limites estabelecidos pelo implementador, informações sobre os acontecimentos são enviadas para o Mestre. Tal ação evita a transferência de informações desnecessárias provocando assim uma diminuição do tráfego na rede, permitindo uma rápida detecção de informação urgente e a comunicação entre escravos. As desvantagens, no entanto, existem. O mestre consegue detectar as falhas na ligação apenas após determinado tempo, quando é efetuado um polling ao sistema, e são necessários outros métodos para obter os novos valores.

A comunicação entre outras estações SCADA pode ser realizada através de um protocolo desenvolvido pelo próprio fabricante do sistema SCADA, ou através de um protocolo conhecido via rede Ethernet TCP/IP, com linhas privadas ou discadas.

Através do uso de um navegador de Internet (Google Chrome, Mozilla, Internet explorer, etc.), é possível controlar, em tempo real, um equipamento situado em outro extremo do planeta. O navegador se comunica com o servidor web pelo protocolo http e, após o envio da *request* referente à operação que se pretende realizar, uma resposta é dada na forma de uma página HTML.

Uma grande vantagem do uso da Internet e de Browsers está na facilidade de manuseio do usuário devido ao costume digital que foi implantado ao longo dos anos. Além disso, a manutenção do sistema se faz necessária somente no servidor.

1.5 MODELAGEM

Para o início da concepção dos componentes de um sistema de automação, é necessário criar um modelo. Essa área costumeiramente lança mão de diagramas de blocos para representar suas entradas, processos e saídas. Dessa forma, a Fig. 1.4 consiste no diagrama de blocos do sistema a ser considerado. O detalhamento de cada segmento desse sistema será conteúdo do capítulo seguinte.

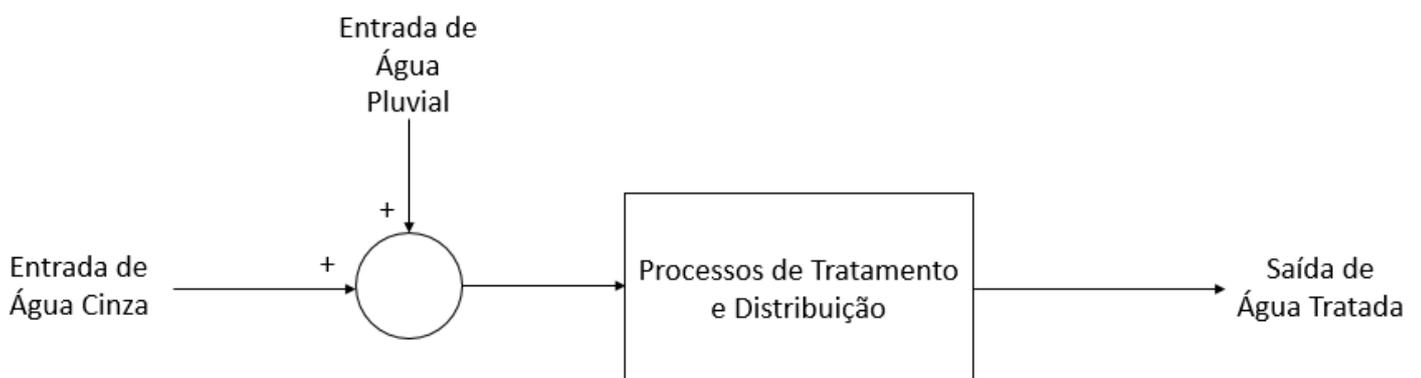


Figura 1.4. Diagrama de Blocos para o Sistema

CAPÍTULO 2 – ELABORAÇÃO DO SISTEMA

As etapas de seleção de componentes e estruturação do supervisor e de suas interações com um modelo de rede de distribuição de água em um prédio são elucidadas a seguir.

2.1 AQUISIÇÃO DE DADOS

Para a averiguação das variáveis que são de interesse para o controle de um sistema hídrico, alguns dispositivos são indispensáveis. Esses eletrônicos diversos são classificados conforme sua funcionalidade.

2.1.1 Sensor de Fluxo

O sensor de fluxo, fluxômetro ou, do inglês, flowmeter, é capaz de fornecer de maneira razoável a vazão de água através de seus terminais, sendo de grande importância em projetos em que se almeje um controle do volume demandado em determinado intervalo de tempo ou do próprio fluxo de água. No projeto realizado este será o sensor responsável por oferecer dados da quantidade volumétrica de água que flui na rede, que, após serem interpretados, resultarão em reação de interferência na rede hídrica.

Seu funcionamento é bem simples: em sua estrutura há um válvula em formato de cata-vento com um ímã acoplado que trabalha em conjunto com um sensor hall para enviar um sinal PWM. Através destes pulsos é possível mensurar a vazão de água, de maneira que cada pulso mede aproximadamente 2,25 ml.

O sensor de efeito hall presente no cata-vento do sensor de fluxo gera pulsos cada vez que detecta o campo magnético gerado por um ímã presente em uma extremidade do Sensor de Fluxo. Através da contagem desses pulsos e através de uma leitura inicial com um volume já conhecido de água é possível realizar a calibração deste sensor, isto é, determinar efetivamente a quantidade de água que atravessa o fluxômetro por giro da hélice nele presente.

O sensor de fluxo selecionado possui finalidade de demonstração em uma aplicação reduzida de uma hipótese virtual para detecção de vazamentos em uma rede. Dessa forma, não há necessidade de acurácia elevada. Para projetos mais exigentes, como a própria automação das redes hidráulicas de um edifício real, sensores de fluxo comerciais industriais são mais indicados. No entanto, é relevante notar que, ainda que disposto em uma simulação, seu funcionamento estima a real necessidade e atuação de um sensor correspondente em um projeto completamente executado.



Figura 2.1. Sensor de Fluxo F-S201

FONTE: Loja Laboratório de Garagem (2015)

As especificações técnicas do sensor em questão estão listadas a seguir:

- Menor tensão de trabalho: DC 4.5V
- Corrente máxima de operação: 15 mA (DC 5 V)
- Faixa de tensão de trabalho: DC 5 ~ 18 V
- Capacidade de carga: ≤ 10 mA (DC 5 V)
- Faixa de temperatura: de -25°C a $+80^{\circ}\text{C}$
- Faixa de umidade de operação: 35% ~ 90% RH (sem gelo)
- Pressão máxima permitida: 1.75Mpa
- Rosca externa: 1/2"

2.1.2 Sensor de Nível e Capacidade

Para obtenção de dados relativos à capacidade volumétrica dos reservatórios para os quais a água proveniente dos tanques de tratamento, é necessário um sensor de nível. O parâmetro por ele medido tem utilidade no sentido de definir se ainda há capacidade para absorção de efluentes pluviais, por exemplo, ou para interrupção do processo de purificação das águas cinzas, visto que purificar água sem espaço para depositá-la consistiria em desperdício.

Um dispositivo de medida de nível pode aferir duas variáveis: a posição da superfície do líquido sobre um ponto de referência e a altura hidrostática criada pelo líquido cuja superfície se deseja conhecer. Existem diversos tipos de medidores desse formato. Dentre os

mais comuns podem ser mencionadas as bóias (magnética, lateral e pêra), e os sensores condutivos, hidrostáticos e ultra-sônicos.

Existem diversas áreas em que há requisição de aferições de nível, de forma que a atuação em líquidos é apenas um dos ramos dentre todos. Para essa opção, um dispositivo amplamente utilizado é o LA16M-40. Seu funcionamento se dá através da disposição de um flutuador que gera um sinal magnético quando submetido ao empuxo de um corpo líquido que o eleve. Esse sinal pode acionar um atuador apropriado, como um relé ou CLP.



Figura 2.2. Sensor de Nível para Líquidos LA16M-40

FONTE: Hermini (2015)

As especificações técnicas do sensor supracitado estão dispostas no Quadro 2.1 e na Tab. 2.1.

Quadro 2.1: Especificações técnicas do Sensor de Nível iCOS LA16M-40 (2015)

Material	POM - Poliacetal (porca em PA)
Pressão máx. de trabalho	2×10^5 Pa
Temperatura de trabalho	-10 a 100°C
Cor	Branco
Densidade mín. do líquido (SG)	0,76
Peso	30g
Saída	Contato On/Off
Características elétricas	NA/NF - SPST
Conexão elétrica	Cabo 2 x 0,5mm ² x 40cm
Grau de proteção	IP66
Montagem	Lateral interna em furo de Ø16mm
Vedação	Arruela NBR (borracha nitrílica)
Espessura máx. parede reservatório	9mm
Raio mín. reservatório cilíndrico	150mm

FONTE: iCOS Sensores para Líquidos (2015)

Tabela 2.1: Especificações de operação do Sensor de Nível iCOS LA16M-40

TENSÃO TRABALHO [V]	POTÊNCIA MÁXIMA [VA ou W]	CORRENTE MÁXIMA [A]	CORRENTE DE PICO [A@20ms]
110 ac	20	0,2	0,5
220 ac	20	0,1	0,5
5V dc	2,5	0,5	1
12V dc	5	0,5	1
24V dc	10	0,5	1

FONTE: iCOS Sensores para Líquidos (2015)

O uso de sensores de nível também está associado à automação da vazão de caixas d'água de edifícios. Para aplicações a nível mais completo, em construções mais complexas ou compostas de diversos ambientes alimentados por um sistema de distribuição em comum (como uma fábrica), recomenda-se considerar a utilização da mesma aparelhagem já existente nos reservatórios de água existentes, visto que seu posicionamento já supre as necessidades daquele ambiente, o que evita um estudo desnecessário sobre que tipo de automação realizar sobre os reservatórios de água cinza e pluvial. No entanto, outra variável a ser considerada é a disponibilidade de líquido a ser tratado, de forma que instalações com uso reduzido de água não exigiriam tanques com volume considerável para armazenamento de águas cinzas e possibilitariam sensores menos robustos.

Em seguida, consideremos sua finalidade em cada reservatório. Os sensores de nível nos reservatórios podem ser sensores capacitivos, que alteram seu estado lógico quando a água entra em contato com o sensor. Assim, pode-se optar por fazer a instalação em pontos cruciais nos tanques, como, por exemplo, os que indicam máximo e mínimo de nível atingidos, para determinar quando serão acionados os atuadores apropriados. Não é necessário que o sensor seja instalado no ponto mais alto do reservatório, pois seria mais sensato utilizar uma margem de segurança de 10%, isto é, colocar o sensor a 10 cm da extremidade em um possível caso para qual o reservatório possua 1 m de altura. O mesmo vale para o sensor de nível inferior.

2.2 ATUADORES

A etapa seguinte é a definição de que dispositivos melhor se acoplam às saídas do sistema em questão, para atuarem sobre os parâmetros de interesse. Para o controle de distribuição de água, duas variáveis se destacam em importância: vazão e volume de água.

2.2.1 Atuadores de Vazão

Válvulas Solenoides são componentes que dependem de um acionamento elétrico para que o fluxo de fluido seja liberado ou interrompido. Geralmente, utiliza o processo de

indução elétrica, quando há energia elétrica atravessando seu rolamento, um campo magnético é criado acionando (empurrado) um eixo, de forma que este acione um interruptor da válvula, liberando ou bloqueando o fluxo de fluido.

Assim como no Projeto e em várias aplicações industriais, há locais com tubulações inacessíveis que necessitam de um controle que seja feito remotamente no controle de fluxo, e é neste contexto que a válvula solenoide é considerada ideal. No supervisorio usado para o controle da rede há dois tipos de válvulas solenoides, duas vias (Fig. 2.3) e de três vias ou caminhos (Fig. 2.4).



Figura 2.3. Válvula Solenoide de Duas Vias
FONTE: Klein Automação (2015)



Figura 2.4. Válvula Solenoide de Três Vias
FONTE: Refriparts (2015)

A válvula solenoide de 2 caminhos indica simplesmente a abertura ou fechamento de fluxo. O nome “2 vias ou caminhos” se deve ao fato de que ela possui um caminho de entrada e outro de saída, de maneira que esse caminho pode ser interrompido quando necessário. No projeto, essas válvulas funcionam como atuadores que dependem de valores de sensores de nível de reservatórios, acionamento por um botão local ou mesmo acionamento remoto

realizado via software ScadaBR. Válvulas como esta são representadas, no projeto, pela válvula de águas pluviais e pelas válvulas dos canos 1 e 2.

Já a válvula 3 caminhos ou 3 vias, continua dependendo de sinais de controle assim como a válvula anterior, porém seu funcionamento é diferente: quando o sinal é recebido, ela alterna a direção do fluxo entre duas posições. Como exemplo, basta imaginar uma tubulação de água que se ramifica em outros dois canos, porém esses dois canos nunca podem receber a água do cano central ao mesmo tempo, nessa situação é preferível a válvula de 3 vias: ora o fluxo é direcionado para um dos canos, ora para outro, dependendo apenas do comando. Esta válvula é aplicada na rede hidráulica do projeto tanto na região de descarte de esgoto, assim como na região que seleciona o tipo de abastecimento da rede, se cinza ou potável.

2.2.2 Atuadores de Volume

Para gerar modificação no valor do volume de um recipiente, duas opções existem: introduzir mais líquido ou remover líquido já presente. Ambas as opções requerem um dispositivo que seja capaz de deslocar corpos de água para dentro ou para fora dos reservatórios. A opção mais comum em sistemas de alimentação hídrica de edifícios é a bomba hidráulica.

“Uma bomba hidráulica é um dispositivo que adiciona energia aos líquidos, tomando energia mecânica de um eixo, de uma haste ou de um outro fluido: ar comprimido e vapor são os mais usuais. As formas de transmissão de energia podem ser aumento de pressão, aumento de velocidade ou aumento de elevação – ou qualquer combinação destas formas de energia. Como consequência, facilita-se o movimento do líquido.” (ANSI/HI 9.1 – 9.5, 2000).

A utilização de diversas bombas será necessária para deslocar os efluentes desde sua entrada no sistema até as estações de tratamento, para, em seguida, serem rearmazenados e disponibilizados para pronto uso. A atuação das bombas também está encadeada aos dados provenientes dos sensores de nível, em situações de envio de água a caixas de armazenamento, para noção do volume que ainda pode ser enviado sem que ocorra transbordamento. Há também situações de desarme das bombas, em que não haja, por exemplo, incidências pluviais, de maneira que acionar a bomba para direcionamento das águas daí provenientes seria ação sem resultado algum.

A figura 2.5 exemplifica uma bomba de nível industrial.

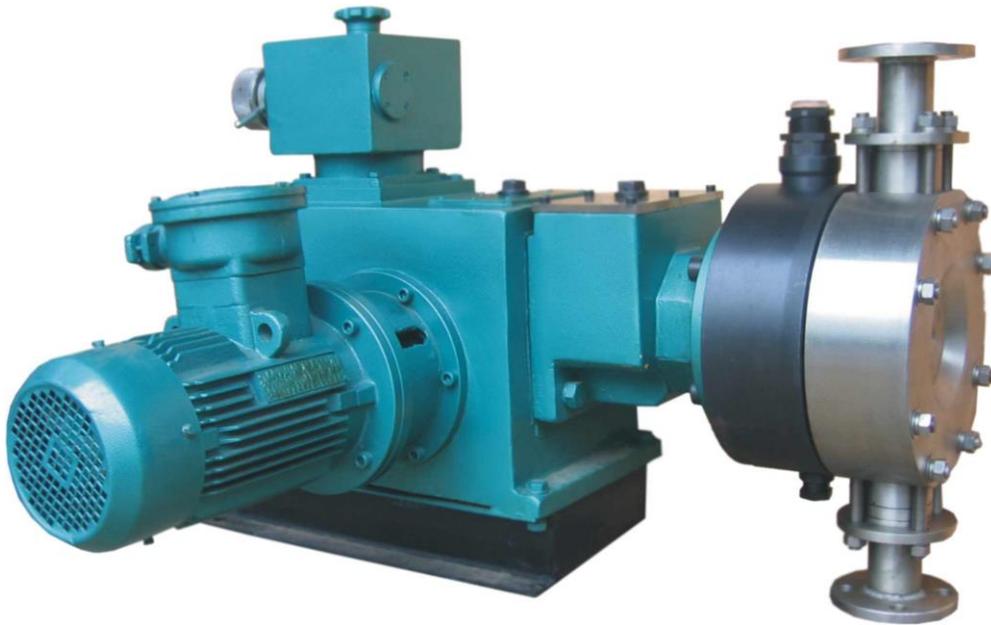


Figura 2.5. Bomba Hidráulica Modelo
FONTE: Soluções Industriais (2015)

Soluções para aplicações robustas podem ser associadas às estruturas já presentes nos edifícios que serão automatizados. Objetivos mais reduzidos podem considerar o uso de bombas de menor capacidade, cuja função natural relaciona-se a pára-brisas de veículo ou a aquários.

2.3 PROCESSAMENTO DE DADOS

2.3.1 Arduino e CLP

Para a etapa de processamento dos dados originários dos sensores e controle dos atuadores, é imperativa a presença de um recurso que preencha propósitos demonstrativos e reais. Por apresentar resultados satisfatórios a plantas de complexidade média, a aparelhagem de controle adotada para o módulo de apresentação foi o controlador Arduino.

Arduino é, conforme a descrição em seu site homônimo, “uma plataforma eletrônica de código aberto baseada em hardware e software fáceis de manusear. A intenção de sua idealização é que possa ser usado por qualquer pessoa realizando projetos interativos”.

O Arduino, basicamente, compõe-se em uma placa que associa diversas entradas e saídas analógicas ou digitais, interligando-as a um centro processador programável por um computador, quando conectado a ele por uma entrada USB (micro controlador). Sua linguagem de programação é bastante similar à linguagem C.

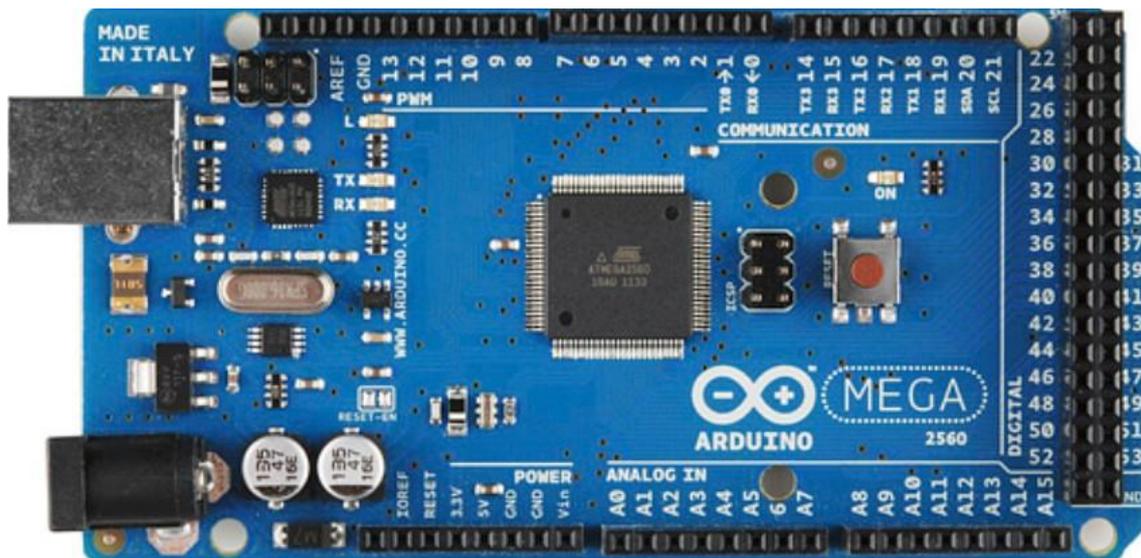


Figura 2.6. Arduino Mega 2560
FONTE: www.arduino.cc (2015)

Para implementações reais, dois Arduinos serão necessários devido a especificações de um atuador que será posteriormente detalhado de maneira melhor. Um deles será responsável por elencar toda a gama de sensores e atuadores, além de enviar todos os dados de interesse de armazenamento histórico e controle para o sistema supervisor, e outro pelo processamento dedicado do já referido atuador.

Como o desígnio do primeiro controlador é sustentar diversos componentes, uma versão um pouco mais robusta chamada Arduino Mega será considerada. Esse micro controlador é baseado em um ATmega1280, possui 54 pinos que podem ser utilizados como entrada ou saída, 16 entradas analógicas, 4 UARTs (portas de entrada serial), um oscilador de cristal de 16 MHz, uma conexão USB, uma entrada para alimentação, um header ICSP, e um botão de reinicialização. A grande quantidade de entradas e saídas disponíveis deve ser suficiente ao menos para nível de demonstração deste projeto.

Já o segundo Arduino não requer tanta disponibilidade de conexões, o que nos permite flexibilizar a pesquisa e selecionar uma alternativa menos custosa. A versão encontrada é o Arduino Uno, que se comporta da mesma forma que o já descrito Mega, assim como possui composição de itens semelhante, apenas com menor quantidade de portas e capacidade de processamento.

É oportuno notar que as aplicações a nível de edificações de grande porte (industrial) demandam ainda maior robustez em um apetrecho de controle, demanda essa que nos leva aos CLPs, ferramentas de grande préstimo no setor mecatrônico.

Um Controlador Lógico Programável, segundo a ABNT, é um equipamento eletrônico digital com hardware e software compatíveis com aplicações industriais. Uma definição um pouco mais completa é fornecida pela NEMA (National Electrical Manufacturers Association): é um aparelho eletrônico digital que utiliza uma memória programável para armazenar internamente instruções e para implementar funções específicas, tais como lógica, sequenciamento, temporização, contagem e aritmética, controlando, por meio de módulos de entradas e saídas, vários tipos de máquinas ou processos.

Os CLPs são integrantes essenciais às redes de controle a nível industrial. Seus canais de comunicação permitem conexão a diversos outros pontos de sua planta, como à interface de operação (ou interface homem-máquina), a computadores, a outros CLPs ou a equipamentos de campo remotos. Nosso interesse reside justamente nessa última capacidade: conexão remota a unidades de entrada e saída. Isso permitirá a canalização de dados dessas unidades até os servidores do supervisório para tratamento.



Figura 2.7. Controlador Lógico Programável da Altus, série AL-2000

FONTE: www.altus.com.br (2006)

Um fator auxiliar que reforça ainda mais a seleção de um CLP para efeito em construções de maior porte é a compatibilidade com os protocolos de comunicação disponíveis para Arduino. O protocolo deste projeto é o ModBus over TCP/IP e, assim como

já existe enorme suporte a desenvolvimento de propostas que associem Arduino e ScadaBr por meio deste protocolo, existem inúmeras concepções de CLPs com comunicação baseada nesse padrão de troca de informações, algumas inclusive em uso até o atual momento.

Por todos as razões mencionadas, o CLP torna-se a preferência para futuro uso em maior escala. Seu valor de dispêndio é elevado em comparação a um Arduino, entretanto sua funcionalidade de maior porte torna-o componente insubstituível para ambientes industriais.

2.3.2 Supervisório

Trata-se de um software open source do tipo SCADA, do ingles, Supervisory Control and Data Acquisition. Este software possui diversas finalidades, sendo útil para automação de processos de medição, ou seja, é possível ter acesso e controle de dispositivos físicos como: sensores, chaves, motores, etc. Entre outras funcionalidades o ScadaBR, possui um banco de dados onde é possível armazenar diversos dados de sensoriamento, visualizar históricos de sensores e variáveis, receber alarmes, ter a possibilidade de controlar os processos por meio de scripts.

Sua utilização requer um maior conhecimento a respeito das partes que o compõem. Dessa forma seguem enumeradas algumas características essenciais desse software SCADA.

1. Processo Controlado
2. Barramentos de Comunicação
3. Servidor de Aquisição de Dados e Controle Supervisório (SCADA)
4. Interface do Usuário ou HMI (*Human-Machine Interface*)

A figura 2.8 identifica esses componentes de forma mais lúdica.

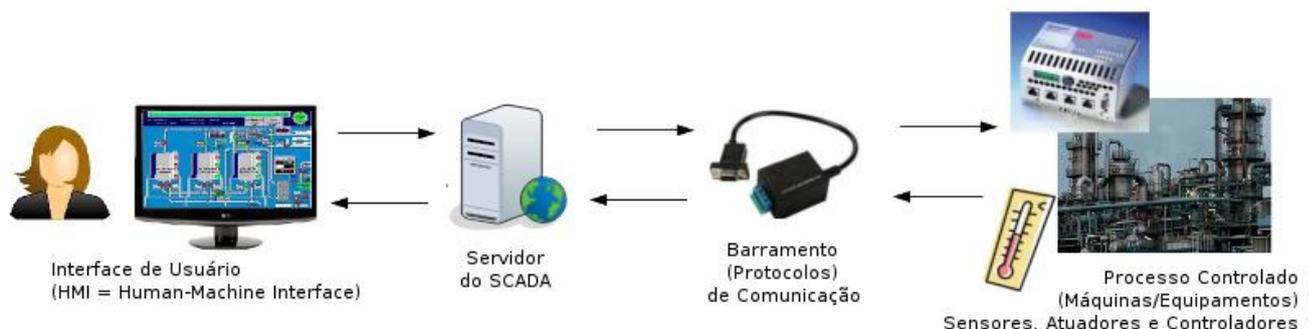


Figura 2.8. Disposição de um Supervisório ScadaBR

FONTE: CERTI-SCADABR (2015)

Para acesso aos seus componentes, o ScadaBR deve ser instalado em um computador, que será conhecido também como o "servidor da aplicação" ou "Servidor ScadaBR". Tanto as

telas "HMI" quanto as configurações do ScadaBR serão acessadas através de um navegador de Internet (Internet Explorer, Mozilla Firefox, Chrome etc.).

Sua execução é realizada partir de um software conhecido como "Servidor de Aplicação Java", cujo padrão é o Apache Tomcat, disponível no instalador do ScadaBR para Windows. Quando o servidor de aplicação está no ar, ele se torna um "endereço na Internet" com um link de acesso específico para ser acessado no navegador, independente de haver internet ou não. Este acesso pode ser realizado no mesmo computador onde o servidor foi instalado, ou mesmo em outros computadores, desde que estejam alocados na mesma rede.

Vários aspectos dentro do software precisam ser configurados de modo a produzir o correto funcionamento e aquisição de dados do sistema envolvido:

- 1) Datasources e Datapoints – como o próprio nome diz, são fontes de dados e pontos de dados respectivamente. Suas configurações determinam os equipamentos a serem controlados. O Datasource pode ser um barramento de comunicações, usando um protocolo do tipo ASCII, Modbus, SQL, OPC dentre outros. Dentro de cada Datasource, que no caso do projeto trata-se do Arduino Uno, pode haver vários Datapoints, que são os pontos de medição e controle. No projeto os Datapoints são representados por sinais de nível de reservatórios, fluxos de água representados por sensores, status de válvulas e etc. Além de leitura e sensoriamento, esses Datapoints podem também enviar sinais de controle quando necessários.
- 2) Telas ou representações gráficas – Local onde é possível monitorar dinamicamente as variáveis sobre uma planta ou figuras dos projetos. É possível realizar uploads de imagens para inserção de representações gráficas dinâmicas de sensores e medições nessa seção.
- 3) Tratadores de eventos - Habilita o envio de notificações em situações de alerta, ou ao tomar ações automaticamente com base em ocorrências no sistema pré definidas pelo usuário.
- 4) Relatórios – Seção designada para o acompanhamento e visualização do histórico do sistema, que inclui gráficos e estatísticas.
- 5) Scripts - Possibilidade de inserção de lógicas de programação para automação assim como para cálculos de variáveis derivadas de outras.

Na Figura 2.9 podem ser observados alguns dos ícones já mencionados. Embora a imagem também apresente diversos outros componentes, os mais importantes são os supracitados.



Figura 2.9. Ícones de um Supervisório ScadaBR

FONTE: CERTI-SCADABR (2015)

2.4 COMUNICAÇÃO ENTRE DISPOSITIVOS

Para a elaboração de um sistema supervisório, é primordial a compreensão de como se configura a arquitetura do sistema integrado, composto não apenas pelos sensores e atuadores, mas por tudo aquilo que está relacionado ao ciclo de vida do produto em questão.

O conceito mais amplo a ser investigado inicialmente é o de sistema. Há dois tipos de sistema para o contexto considerado: concentrado ou distribuído. Um sistema concentrado é composto de um único computador que gerencia todo o processo, enquanto um sistema distribuído (também chamado de rede de computadores) inclui diversas máquinas alocadas ao longo da planta para as quais a função de gerenciamento pode ser delegada. De semelhante forma, em um sistema distribuído, os terminais também realizam processamento de maneira a melhor distribuir a carga e não sobrecarregar um único componente. Para as especificações deste projeto, que possui dispositivos locais de processamento e equipamentos de controle supervisório remoto, um sistema distribuído se enquadra melhor. Algumas características de uma rede distribuída que tornam essa configuração a mais adequada são a habilitação de comunicação em tempo real vertical e horizontal, facultação de compartilhamento de recursos, substituição de controladores de grande porte e elevado custo por computadores (no nosso caso, uma única máquina com acesso ao servidor do supervisório e um micro controlador), além do aumento do desempenho da rede a partir do acoplamento de capacidades de processamento dos dispositivos acessíveis.

A etapa seguinte é a disposição dos diferentes níveis presentes em um sistema integrado. A figura 2.10 dispõe os níveis mencionados de forma piramidal para melhor compreensão de sua natureza sequencial.

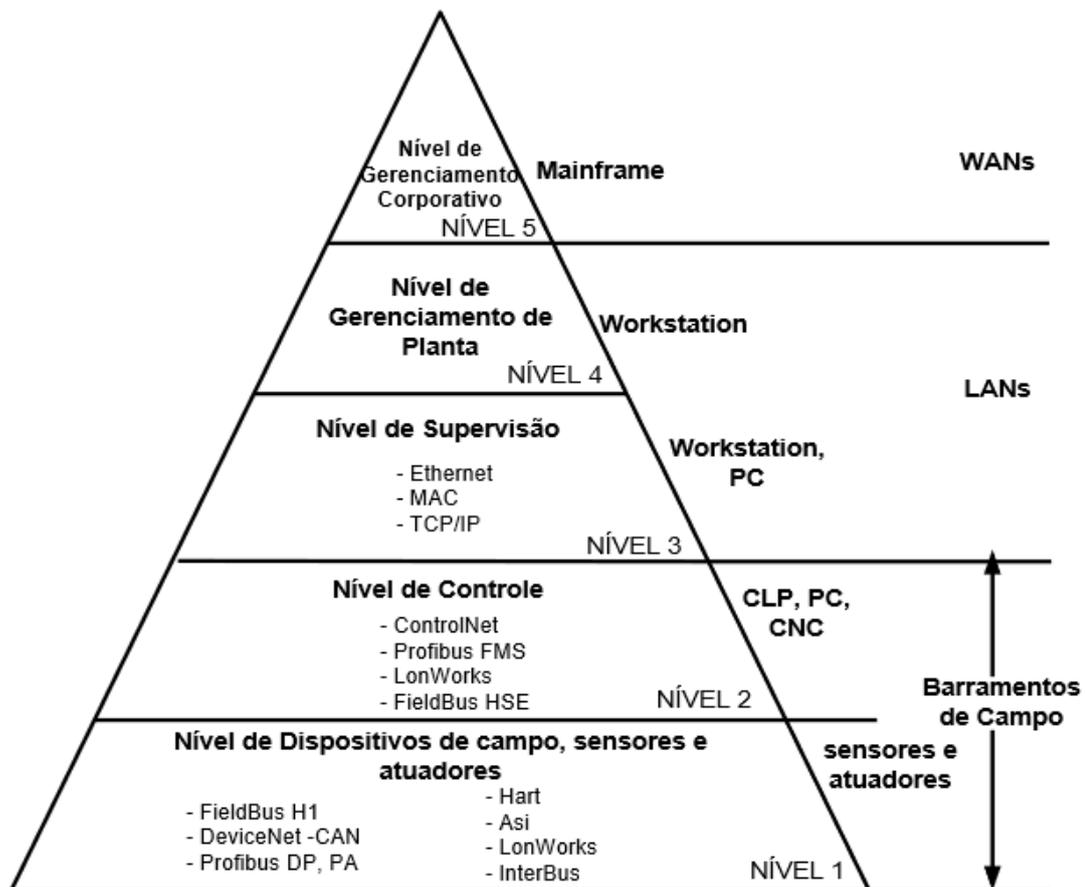


Figura 2.10. Pirâmide da Automação Industrial

FONTE: Portal o Setor Elétrico (2015)

Existem, por convenção, cinco níveis divididos de acordo com suas funções no processo produtivo:

- Nível 1: nele se encontram os dispositivos de campo, sensores e atuadores. A atuação dos controladores programáveis (CLP) é executada sobre os componentes deste nível.
- Nível 2 (ou nível de controle): aqui localizam-se os já mencionados controladores que atuam em dispositivos de campo (nível 1). É o nível que compreende a integração entre unidades inteligentes.
- Nível 3 (ou nível de supervisão): possui algum tipo de supervisão associada ao processo. Os concentradores de informações sobre o nível 1 e as interfaces homem-máquina se enquadram nesse nível.

- Nível 4: controle do processo produtivo da planta, constituído por bancos de dados, MRP, etc.
- Nível 5: programação e planejamento da produção realizando o controle e a logística dos suprimentos. Também inclui a administração dos recursos da empresa e possui softwares de gestões financeira e de vendas.

Ainda que nos interessem todos os níveis de controle desse sistema, concentrar-nos-emos nos níveis mais inferiores, 1 e 2, para seleção das opções que melhor complementam os objetivos e escopo deste projeto. Já que os dispositivos de campo já foram escolhidos e descritos, resta determinar que itens serão utilizados no segundo nível, mais especificamente que protocolo de comunicação.

Protocolo de comunicação é o termo que define um conjunto de regras que permite duas ou mais entidades de um sistema de comunicações transmitirem informação através de qualquer tipo de variação de uma quantidade física, ou seja, através de uma propriedade física mensurável de um fenômeno, corpo ou substância. De forma mais direta, segundo Torres (2014), protocolo é a “linguagem” usada pelos dispositivos de uma rede de modo que eles consigam se entender, isto é, trocar informações entre si. Temos ainda, segundo Forouzan (2006), a definição de que protocolo é um conjunto de regras que governa a comunicação de dados.

Assim como no campo da comunicação interpessoal, existem diversos protocolos, ou diversas “linguagens”, com o uso das quais dispositivos podem se comunicar. O início do desenvolvimento de tecnologias desse tipo pode ser atrelado às cercanias do ano de 1981, em que os primeiros *fieldbuses* começaram a surgir. Fieldbus é um termo genérico cunhado para descrever tecnologias de comunicação industrial. O objetivo era “resolver alguns problemas de usuários finais, que se referiam à heterogeneidade de seus dispositivos” (THOMESSE, 1999). A partir de então, várias companhias se empenharam em concatenar seus próprios protocolos.

“PROFIBUS foi um conceito baseado em uma ideia de controle distribuído e em sua forma original suportava uma comunicação vertical orientada em objetos de acordo com o modelo cliente-servidor no espírito da especificação MAP/MMS. FIP, por outro lado, foi projetada com um esquema de controle em tempo real central, mas estrito, e com um modelo produtor-consumidor (produtor-distribuidor-consumidor) recentemente desenvolvido para comunicação horizontal” (FELSER & SAUTER, 2002).

Ainda baseado nas concepções de Thomesse, existem dois pontos de vista para que tipo de problemas as tecnologias de *fieldbus* deveriam resolver: “um fieldbus é apenas uma rede para simplificar a conexão entre dispositivos, ou um fieldbus é a coluna espinhal de um sistema distribuído em tempo real”.

Embora as exigências de melhora e metas de otimização de eficiência fossem iguais para todos os desenvolvedores de *fieldbuses*, um grande número (mais de 50) de resultados surgiu. A França (FIP) e a Alemanha se engajaram em padrões nacionais e almejavam torná-los padrões internacionais. Essa situação encaminhou-se para um período chamado de “guerra dos *fielbuses*”. Foi impossível concordar em um padrão internacional por algo em torno de quinze anos. Finalmente, em 1999, a publicação da norma IEC 61158 encerrou a disputa e incluiu todos os *fieldbuses* populares em seu conteúdo. É interessante notar que esses protocolos, ainda que populares na época, já não correspondem aos mais utilizados na atualidade, restando apenas alguns desses em uma mistura de novos mais recentemente concebidos.

Desde o primeiro contato com os protocolos *fieldbus*, mais de trinta anos já se passaram e os requerimentos mudaram. Hodiernamente, é normal conectar todo tipo de equipamento à internet.

“A única barreira para acessar dispositivos de campo no mundo da automação, a partir da internet através de uma conexão de rede, são os *fieldbuses*. Dessa forma, a questão é: por que não é possível usar Ethernet também na tecnologia de automação?” (FELSER, 2005).

O esperado para transmissão de dados também se alterou, de modo que é comum transmitir quantidades de dados muito maiores que anteriormente. A taxa de transferência que *fieldbuses* oferecem é extremamente limitada em comparação a tecnologias Ethernet ou ATM (DECOTIGNIE, 2005).

No mundo corporativo, redes baseadas em Ethernet e em TCP/IP estão se tornando cada vez mais baratas, e, paralelamente, cada vez mais velozes. Isso é possibilitado pelo enorme emprego dessas tecnologias em todo tipo de sistemas, o que encoraja desenvolvedores e vendedores a investirem em soluções baseadas em Ethernet também para os ambientes industriais. Ao mesmo tempo, o setor de informática sofreu enorme expansão e não aparenta estar limitado por desaceleração alguma, inundando o mercado com aplicativos e protocolos voltados para conexões de computadores pessoais à internet.

Uma breve descrição de alguns dos protocolos mais utilizados nesse âmbito se segue:

- *HTTP – HyperText Transfer Protocol*: seu uso é comumente associado ao acesso de dados na *World Wide Web*. Permite a transferência de dados na forma de textos simples, hipertextos, áudios, vídeos entre muitas outras (FOROUZAN, 2006);
- *SMTP – Simple Mail Transfer Protocol*: é o mecanismo padrão de correio eletrônico da internet (FOROUZAN, 2006);

- *FTP – File Transfer Protocol*: esse protocolo é o mecanismo padrão oferecido pela internet para copiar um arquivo de um host para outro (FOROUZAN, 2006);
- *SNMP – Simple Network Management Protocol*: é um protocolo de gerência da internet (DANTAS, 2009);
- *DNS – Domain Name Server*: esse protocolo de aplicação tem por função identificar endereços OPs e manter uma tabela com os endereços dos caminhos de algumas redes na internet (DANTAS, 2009);
- *TCP – Transmission Control Protocol*: a característica desse protocolo é oferecer um serviço confiável entre aplicações (DANTAS, 2009);
- *UDP – User Datagram Protocol*: é conhecido pela característica de ser um protocolo otimista, ou seja, ele envia todos os seus pacotes, acreditando que eles chegarão sem problemas e em sequência ao destinatário (DANTAS, 2009);
- *IP – internet Protocol*: é o principal protocolo do nível de inter-rede na arquitetura TCP/IP (DANTAS, 2009);
- *ICMP – Internet Control Message Protocol*: esse protocolo tem por objetivo prover mensagens de controle na comunicação entre nós num ambiente de rede TCP/IP (DANTAS, 2009);
- *ARP – Address Resolution Protocol*: protocolo que mapeia um endereço IP no respectivo endereço MAC (FOROUZAN, 2006);
- *RARP – Reverse Resolution Protocol*: protocolo que mapeia um endereço MAC a um endereço IP (FOROUZAN, 2006);

Esses protocolos são geralmente utilizados para comunicação de dados a nível computacional, entre duas máquinas diferentes. No entanto, utilizar esses protocolos como padrão de comunicação entre dispositivos comuns ao ambiente industrial, ou até a dispositivos de automação simples, seria desnecessariamente trabalhoso. Visto que essa troca de informações é uma constante recorrente no campo da automação, alguns protocolos foram desenvolvidos especificamente para esse fim, com aplicação complementar aos já mencionados. Os principais podem ser considerados o ModBus, o ProfiBus, o Foundation FieldBus, e o Industrial Ethernet.

O ModBus é um protocolo desenvolvido e publicado pela Modicon Industrial Automation Systems em 1979 para uso do seu CLP. Mesmo sendo um dos protocolos mais antigos ainda é largamente utilizado para aquisição de sinais de instrumentos e para comandar atuadores usando uma porta serial. Um dos motivos para tal é que a original

Modicon, atualmente parte do grupo Schneider Electric, colocou as especificações e normas que definem o ModBus em domínio público, o que o torna uma das soluções mais baratas em automação industrial. Sua topologia se caracteriza por um sistema de Mestre/Escravo (Monomestre).

O ProfiBus (Process Field Bus) é um protocolo que foi concebido a partir de 1987, em uma iniciativa conjunta de diversos fabricantes, usuários e do governo alemão. Consiste em um padrão aberto de barramento de campo para uma larga faixa de aplicações em automação de fabricação e processos. Seu destaque fica por conta da atuação em diversos níveis da linha industrial: chão de fábrica, processo e gerência. Sua topologia, assim como o protocolo ModBus, se dá por um sistema de Mestre/Escravo, mas com opção do número de mestres (Monomestre ou Multimestre).

O Foundation FieldBus corresponde a um padrão aberto que engloba diversas tecnologias aplicadas no controle de processos e automação industrial, tais como processamento distribuído, diagnóstico avançado e redundância. Seu sistema difere dos outros por ser heterogêneo e distribuído, composto por softwares de configuração e supervisão, equipamentos de campo, e interfaces supervisoras e de comunicação, com alimentação realizada pela própria rede que interconecta todos esses componentes. Essa diferença também se propaga para a função dos equipamentos de campo, que executam a aplicação de controle e supervisão do usuário que foi distribuída pela rede, enquanto outras tecnologias são baseadas na execução dependente de um controlador central.

Por fim, o Industrial Ethernet é um protocolo que se desenvolveu a partir da rede Ethernet, criação da Xerox nos anos 70 em seu Centro de Pesquisa Palo Alto (PARC). Seu uso é extremamente popular e difundido em redes corporativas, mas sua aplicação a usos de automação não é recomendável, ou ao menos não corresponde a seu propósito quando criado, consequência do seu modelo não determinístico, que permite colisões de dados na rede e tempos de resposta imprecisos. Para um sistema sob esse tipo de controle e que priorize informações recebidas em tempos específicos, saídas com flutuação podem comprometer medidas de desempenho. Além disso, suas características de segurança na transmissão de dados não são ideais para redes de automação. No entanto, por ser a tecnologia de rede mais dispersa no meio, que permite grande escala de produção e compatibilidade, esse protocolo é reconhecido como uma alternativa bastante atrativa para esse tipo de interconexão. Sua topologia é do tipo estrela com par trançado. Como sua evolução está atrelada ao crescimento do setor informático (do qual a rede Ethernet faz parte), suas vantagens incluem grandes passos em adaptação para qualquer sistema operacional, diversidade de serviços disponíveis, acesso a informações da planta através de redes públicas e privadas e implementação de métodos de segurança, velocidade e confiabilidade que suplantam os obstáculos iniciais dessa “linguagem”. Porém, no contexto industrial atual,

não há motivo óbvio ou razão técnica para substituição dos barramentos de campo tradicionais, que ainda funcionam e alcançam as expectativas de seus usuários. Em complemento, o maior impedimento para a universalização desse protocolo, seu modelo não determinístico, permanece presente e encarece o uso para situações em que esse parâmetro é tido como meta, segundo ERIKSSON, COESTER e HENNIG (2006).

Para evitar o comportamento aleatório de um algoritmo Ethernet comum, muitas soluções foram visionadas. Algumas atuam em redes com dispositivos usuários do Ethernet padrão, e outras não. A partir dessa concepção, o Ethernet em tempo real pode ser dividido em duas categorias de modificação: “modificações que alteram compatibilidade” e “modificações que mantêm compatibilidade” (DECOTIGNIE, 2005).

As diferentes abordagens são descritas por Felser em três categorias:

- Acima da camada de TCP/IP: a pilha TCP/IP permanece inalterada, o protocolo de tempo real é definido na camada superior;
- Acima da camada de Ethernet: O Ethernet padrão é utilizado mas os protocolos de TCP/UDP/IP são substituídos por protocolos próprios;
- Ethernet modificada: o protocolo de Ethernet é modificado.

A diferenciação prática entre modificações que alteram ou não a compatibilidade do protocolo se encontra descrita pelo elenco de prós e contras que cada um proporciona. Enquanto alterar a compatibilidade proporcionaria menores custos de hardware, essa vantagem só é real se encontrado um meio de utilizar equipamentos que não dependam do Ethernet padrão, visto que a comunicação entre nós modificados e nós padrões tornar-se-ia impossibilitada. Assim, plantas já existentes e com dispositivos baseados em Ethernet padrão não são boas opções para abordagens de Ethernet com compatibilidade alterada.

Mesmo quer haja uma variada gama de soluções e protocolos para adequar o gigante Ethernet às plantas industriais, essa situação não deve se converter em nova disputa pela internacionalização de uma norma. Como a imagem dos desenvolvedores da área já se encontra relativamente desgastada pelo período de confronto relativo aos *fieldbuses*, a comissão responsável pela padronização desses protocolos (IEC) busca por uma solução multifocal, em contraponto a uma normatização única. Assim, a norma IEC 61784-2 (IEC, 2009) e o autor Felser listam as soluções contidas na Tab. 2.2 como possibilidades.

Tabela 2.2: Soluções Populares para Ethernet Industrial

Nome	Tipo de realização	Velocidade
P-NET	Acima da UDP/IP, mantém compatibilidade	Baixa
Modbus over TCP	Acima da TCP/IP, mantém compatibilidade	Baixa
Modbus-RTS	Acima da TCP/IP, mantém compatibilidade	Média
Vnet/IP	Acima da TCP/IP, mantém compatibilidade	Média
Ethernet/IP	Acima da TCP/IP, mantém compatibilidade	Alta
PROFINET CBA	Acima da Ethernet	Média
TCnet	Acima da Ethernet	Alta
EPA	Acima da Ethernet	Não Determinística
ETHERNET Powerlink	Acima da Ethernet	Alta
PROFINET IO	Ethernet modificada	Alta
EtherCAT	Ethernet modificada	Alta
SERCOS	Ethernet modificada	Alta

FONTE: IEC 61784-2 (2009)

Para o nosso caso, o protocolo escolhido foi o ModBus over TCP/IP. O primeiro fator que pesou nessa designação foi sua natureza pública e gratuita, que muito agrada a estudantes cujo capital de iniciativa é limitado. Além disso, diversas aplicações para comunicação entre Arduino e outros sistemas já estavam implementadas em ModBus e disponibilizadas para acesso e estudo, o que nos inclinou a uma abordagem nesse sentido. Em seguida, considera-se ainda a compatibilidade mantida e a ausência de necessidade de velocidades de comunicação elevadas, posto que a planta em questão almeja reduzir custos a longo prazo, de forma que a atuação imediata de uma válvula não é fator impactante sob a óptica de influência relativamente pequena do intervalo entre a recepção de informação que engatilha um atuador e a atuação em si. Para fins de suporte documentado, as ponderações de Kasberger (*Advantages of Industrial Ethernet – Comparison of Modbus over TCP/IP and PROFINET*, 2011) estão presentes na Tab. 2.3.

Tabela 2.3: Comparação entre Modbus Over TCP/IP e PROFINET

	Modbus over TCP/IP	PROFINET CBA	PROFINET IO
Requerimentos de hardware	Ethernet padrão	Ethernet padrão	Ethernet modificada
Classe de baixa velocidade (100ms)	Sim	Sim	Sim
Controle do processo (10ms)	Sim (com RTPS)	Sim	Sim
Controle de movimento (1ms)	Não	Não	Sim
Endereçamento	Endereço IP + Identificador Unitário	IP ou endereço MAC	
Espaço de endereçamento	16 bits	32767 slots com 32767 subslots	
Acesso de dados	Bit ou word	Bit, byte ou word	
Extensão de aplicação	Controle humano	Controle de processos	Controle de Movimento possibilitado
Segurança	Não coberto pelos protocolos		
Complexidade	Simple	Avançada	Avançada

FONTE: Kasberger (2011)

No projeto, é utilizado o esquema de transmissão RS-485 associado ao protocolo Modbus. As normas RS-485 e RS-422 definem esquemas de transmissão de dados balanceados que oferecem soluções robustas para transmitir dados em longas distâncias em ambientes ruidosos. Estas normas não definem qual o protocolo a ser utilizado para a comunicação dos dados, e são adotadas como especificação da camada física de diversos protocolos, como, por exemplo, Modbus, Profibus, DIN-Measurement-Bus e muitos outros. Todos os aparelhos que possuem comunicação serial por barramento utilizam o padrão RS-485, devido às vantagens que ele apresenta em ambientes industriais, por ser amplamente difundido, e bem aceito em todas as partes do globo.

A norma TIA/EIA-485, conhecida popularmente como RS-485, descreve uma interface de comunicação operando em linhas diferenciais capaz de se comunicar com 32 “unidades de carga”. Normalmente, um dispositivo transmissor/receptor corresponde a uma “unidade de carga”, o que faz com que seja possível comunicar com até 32 dispositivos. Entretanto, existem dispositivos que consomem frações de unidade de carga, o que aumenta o máximo

número de dispositivos a serem interligados. O meio físico mais utilizado é um par trançado. Através deste único par de fios, cada dispositivo transmite e recebe dados. Cada dispositivo aciona o seu transmissor apenas no instante em que necessita transmitir, mantendo-o desligado no resto do tempo de modo a permitir que outros dispositivos transmitam dados. Em um determinado instante de tempo, somente um dispositivo pode transmitir, o que caracteriza esta rede como half-duplex. Uma rede RS-485 pode também utilizar 2 pares trançados, operando no modo full-duplex, totalmente compatível com RS-422.

O valor da ligação em rede para os projetos de engenharia pode ser reforçado pela viabilização do compartilhamento de recursos e bases de dados, que passaram a ser únicos, o que conferiu mais segurança aos usuários da informação quanto ao acesso a esses arquivos. Além disso, a visualização de uma planta em sistema supervisorio ou para sistemas de gerenciamento da produção tornou-se não apenas possível, mas desejada pela facilidade de interpretação e controle proporcionadas por essa ferramenta. As vantagens dessa implementação englobam o aperfeiçoamento do processo de produção, com redução de dispêndio de tempo, insumos e mão de obra, visto que os complexos industriais interligados ao ambiente corporativo permitem ação direta para realização de correções, aperfeiçoamentos e comandos para otimização de todos esses fatores.

CAPÍTULO 3 - SIMULAÇÃO

A realização de um sistema de controle só pode ser realmente avaliada quando seus resultados são de alguma forma mensuráveis. Enquanto em nível idealizado, inúmeras possibilidades permanecem sem exploração, incluindo falhas e erros que levem toda a execução de seus atributos a ser perdida. Para compreender e determinar o padrão operacional de um sistema, o mais recomendado é a utilização de softwares de simulação que permitam a visualização de suas sequências de comandos e reações.

3.1 CIRCUITOS ELETRÔNICOS

O Proteus atualmente é considerado um dos melhores simuladores de circuitos eletrônicos existentes. Embora em inglês, ele possui uma interface de grande facilidade para interação com o usuário e é altamente recomendado para a criação de projetos eletrônicos, tanto para simulação e criação de circuitos quanto para elaborar placas de circuito impresso. Para tais finalidades há no Proteus duas interfaces: ARES e ISIS. O ARES é utilizado para a elaboração de placas de circuito impresso com nível de dificuldade considerado fácil e simples. Já o ISIS é o simulador de circuitos, possui uma vasta biblioteca de componentes e é o mais indicado para tal finalidade.

A versão 8 representa mais de três anos de desenvolvimento contínuo e inclui melhorias em cada área do conjunto de software. Um grande trabalho sobre a estrutura do aplicativo em conjunto com o introdução de um banco de dados comum que possibilita um fluxo de trabalho mais tranquilo para os usuários enquanto os novos conjuntos de recursos economizam tempo e esforço no ciclo de vida do projeto.

A importância da realização de uma simulação para montagem do circuito que corresponda à futura placa a ser manufaturada reside na possibilidade de alterações e adaptações do plano de componentes, visto que pode haver indisponibilidade ou incompatibilidade de preços no momento da montagem real do circuito. Dessa forma, o design utilizado nesse contexto é apenas uma proposta inicial, o que previne margem para inclusão ou alteração de etapas desenvolvidas. No entanto, é verossímil a utilidade desse circuito na compreensão do comportamento das entradas e saídas do sistema, em um estudo de previsibilidade essencial ao controle proposto.

As figuras 3.1, 3.2 e 3.3 correspondem às principais interfaces iniciais do Proteus, ISIS e ARES.

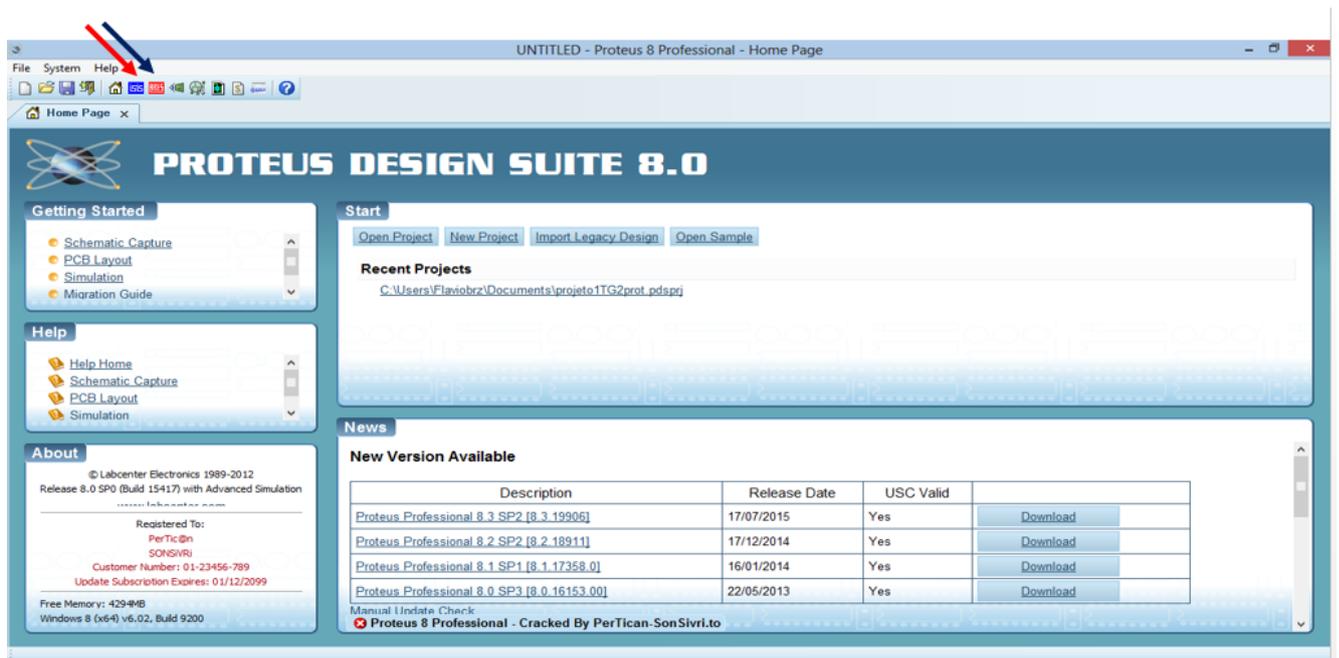


Figura 3.1. Tela de Inicialização do Proteus

É possível ver nas setas indicativas na interface de inicialização do Proteus os ícones para acesso ao ISIS e ao ARES.

Um clique no ícone do ISIS e a seguinte interface surgirá (Fig. 3.2).

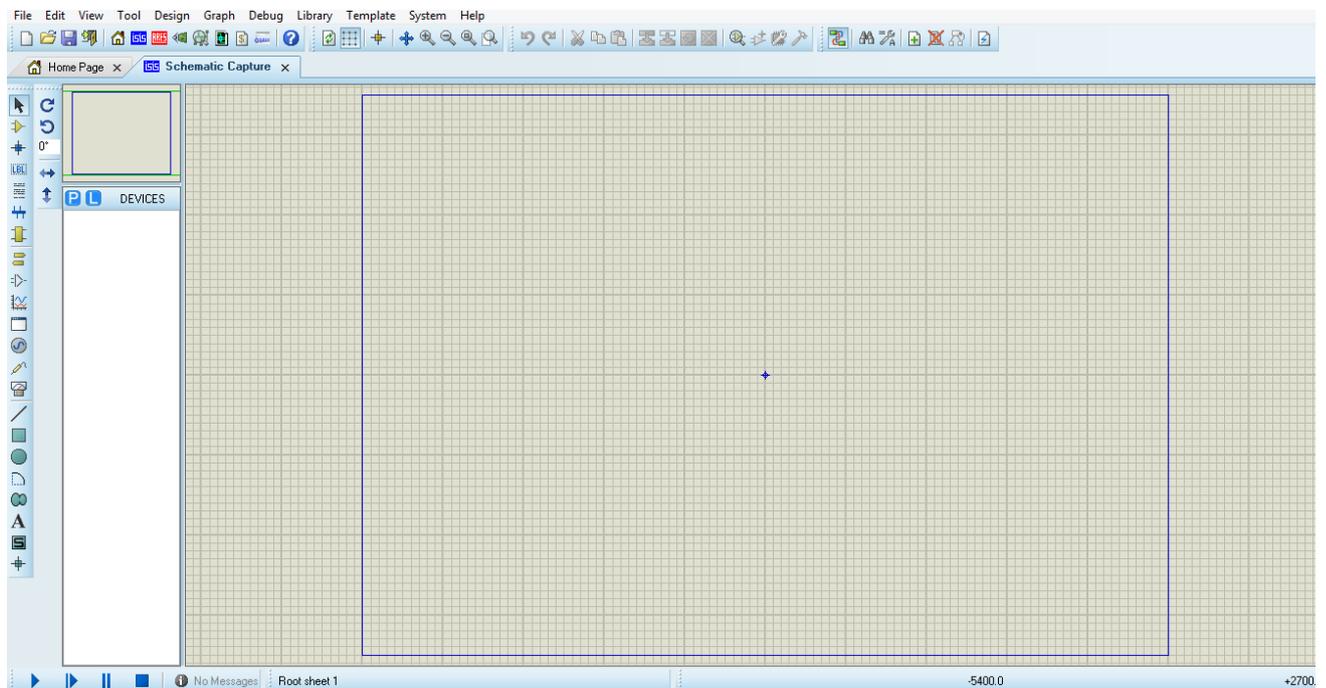


Figura 3.2. Tela de Inicialização do ISIS

Nesta interface é possível selecionar na régua vertical à esquerda diversas funções, inclusive selecionar componentes para inserção, inserção de fontes de alimentação e ground, adição de textos e etc. Para simular o circuito criado, basta clicar no símbolo “Play” localizado no canto inferior esquerdo.

Retornando à interface inicial do Proteus, clicar no ícone ARES resultará na seguinte imagem (Fig. 3.3).

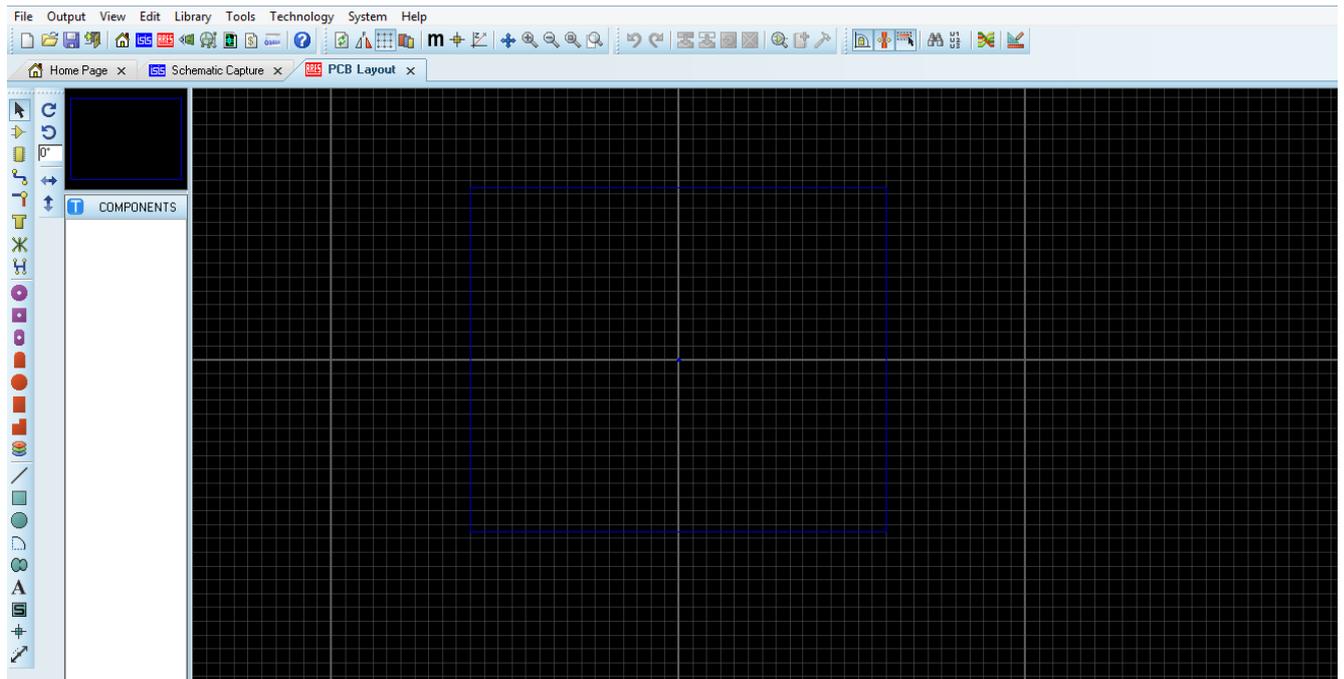


Figura 3.3. Tela de Inicialização do ARES

Nesta interface é possível criar montagens para impressão visando a elaboração de circuitos reais.

3.2 COMUNICAÇÃO

O software Virtual Serial Ports Emulator corresponde a um emulador de portas seriais e é destinado a ajudar engenheiros de softwares e projetistas a criar/debugar/testar aplicações que fazem uso desse tipo de conexão. Ele é capaz de criar vários dispositivos virtuais para transmitir/receber dados. Ao contrário de portas seriais regulares, esses dispositivos virtuais possuem capacidades especiais. Uma delas é a possibilidade de um mesmo dispositivo ser aberto mais de uma vez por diferentes aplicações, o que pode ser de grande uso em vários casos. Com o VSPE é possível compartilhar portas seriais de dados físicas para diversas aplicações, expor portas seriais para a rede local (via protocolo TCP),

criar pares de portas seriais para dispositivos e conectar dispositivos virtuais para comunicação.

No projeto atual, o software VSPE é útil para a conexão virtual entre o circuito montado e simulado através do software Proteus 8 com o software supervisor SCADABR. Sua utilização é indispensável nesse caso, visto que não existem circuitos e conexões físicas para fins demonstrativos. Em futuras utilizações, como em implementações reais em edifícios, tal software não se faz necessário, posto que a comunicação do módulo será feita fisicamente através das portas seriais da máquina local em que o sistema SCADA estiver inserido.

Ao abrir o Software VSPE, a imagem disposta na Fig 3.4 é lançada.

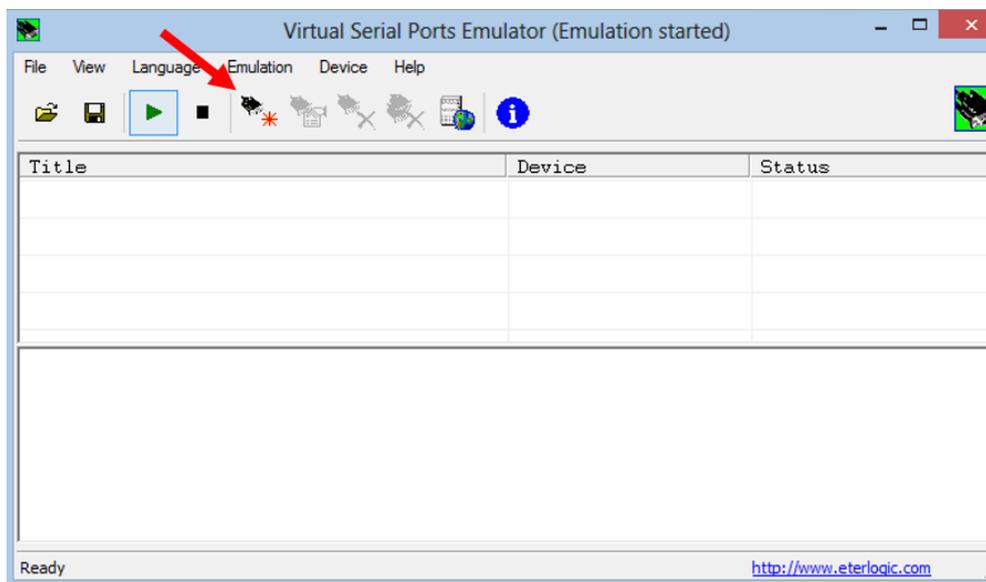


Figura 3.4. Tela de Inicialização do VSPE

Clicando no ícone “Create new device” mostrado pela seta indicativa na figura acima, surgirão diversas opções para criação de portas virtuais. Como no projeto atual a intenção é a conexão direta entre o circuito virtual e o supervisor, a melhor escolha para tal necessidade é a criação de um par de portas virtuais conectadas diretamente. Na lista de opções que aparecem em “Device type”, a opção correspondente é a definida por “Pair”. A imagem que indica essa escolha está contida na Fig. 3.5.

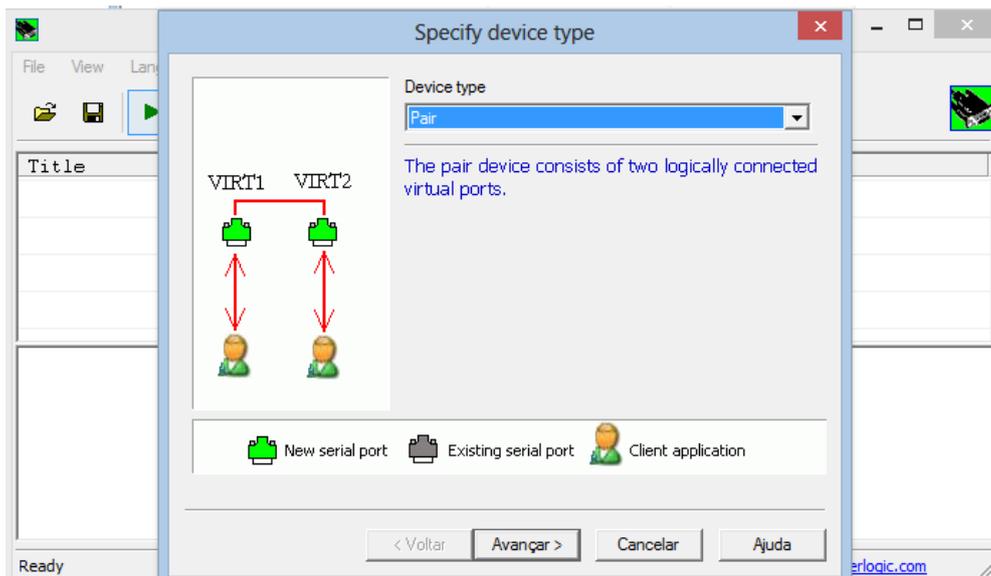


Figura 3.5. Criação de Dispositivo: VSPE

Após seleção da opção para criação do par de portas e clique na opção avançar, abre-se a janela denotada pela Fig. 3.5 para criação das duas portas virtuais.

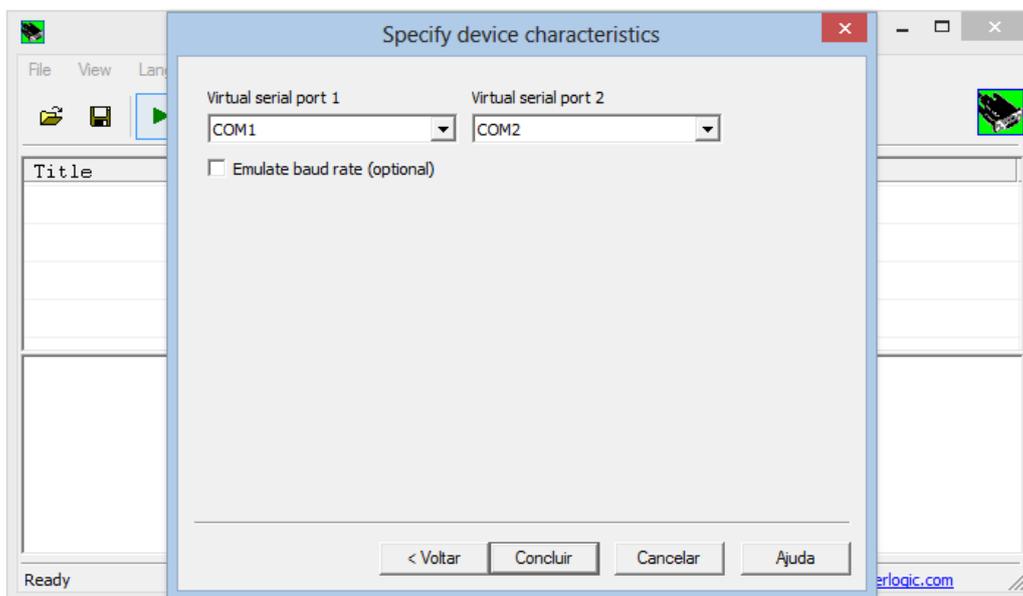


Figura 3.6. Especificações do Dispositivo Criado: VSPE

A escolha de quais portas seriais serão a primeira e a segunda é flexível, desde que elas sejam diferentes. O detalhamento atual considera as portas COM 1 e COM 2. Terminada a criação das portas seriais virtuais, tem-se o seguinte status (Fig. 3.7).

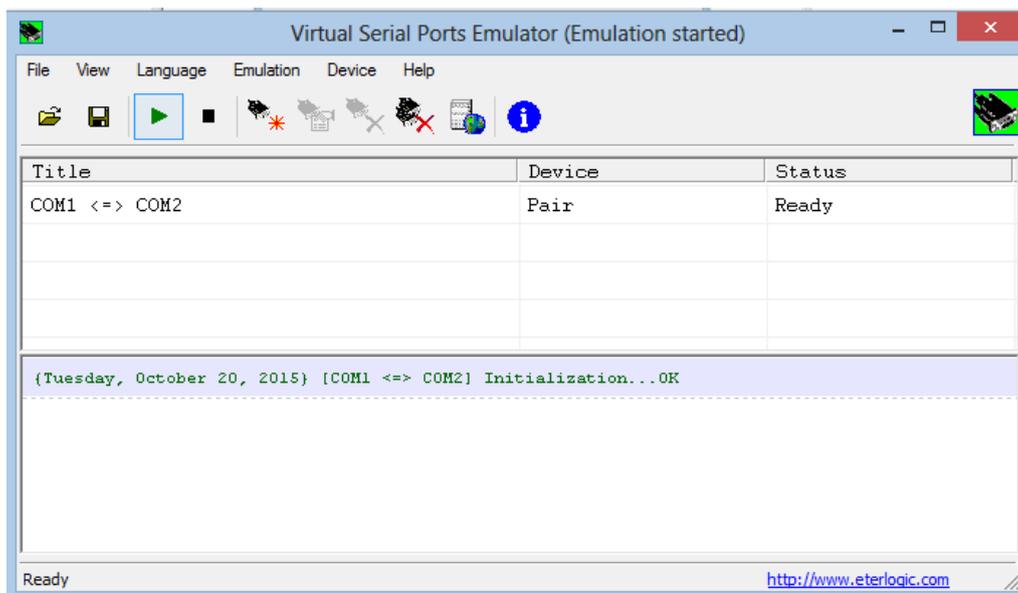


Figura 3.7. Passos Finais para Criação de Portas: VSPE

As portas já estão prontas para uso, o que pode ser verificado pelo status “Ready” da tela apresentada na Fig 3.7. Resta clicar em “Start simulation” para que se inicie a simulação. É importante a lembrança de que os dispositivos a serem conectados devem ser configurados de modo que cada um se conecte às portas seriais que foram criadas, não importando a ordem de conexão. A qualquer momento, quando necessário, a simulação pode ser interrompida, mas para isso é necessário que nenhum dos dispositivos esteja usando uma das portas.

3.3 EXECUÇÃO

A simulação do módulo de automação das redes hidráulicas é possível graças à integração dos seguintes softwares: Isis (Proteus), ScadaBR, AutoCAD 2016, Virtual Serial Ports Emulator, Arduino IDE.

Por se tratar de um circuito virtual, foram utilizados, no ISIS, potenciômetros para simular os sensores de nível de caixa e de fluxo de água. Este potenciômetro varia entre 0 e 1023 para seus valores analógicos, onde 0 equivaleria ao valor mínimo do sensor (nível de água vazio ou quantidade de fluxo zero) e 1023 ao valor máximo (representa que o sensor de fluxo está lendo o maior valor possível ou que o nível de água no reservatório está no máximo).

Para as válvulas foram utilizadas variáveis do tipo booleanas, valor 0 ou 1, que representa fluxo aberto ou fluxo fechado, ou então bomba ligada ou desligada, caso seja a bomba o atuador em questão.

(Arduino) na pasta *Libraries* do computador em que foi compilado o código. A Figura 3.9 corresponde ao circuito de simulação, já com essas modificações .

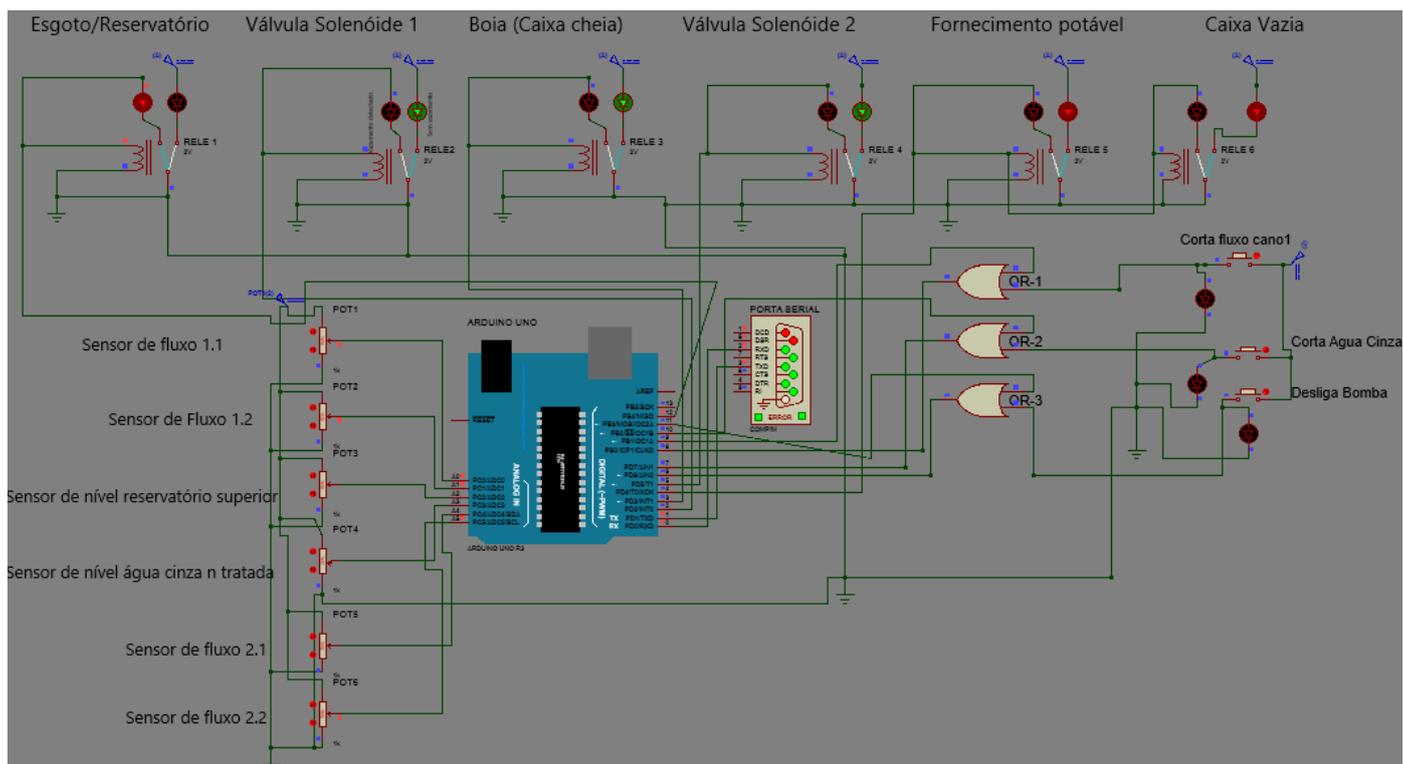


Figura 3.9. Cicuito elétrico no Proteus 8

As subseções descritas no circuito são melhor detalhadas a seguir:

- “Esgoto/Reservatório”: o Led vermelho indica que a válvula de 3 caminhos da seção Esgoto reservatório está aberta para o reservatório de “águas cinzas + pluviais não tratadas” e fechada para a rede de esgotos. Quando o “Sensor de nível de água cinza não tratada” acusa reservatório cheio, o Led torna-se, então, verde indicando que a coleta de água cinzas da rede é agora direcionada para a rede de esgotos. Além disso, a válvula presente no cano coletor de águas pluviais também é fechada.
- “Válvula Solenoide 1”: O Led verde nesta subseção indica que a válvula do cano 1 está aberta, permitindo o livre fluxo de água nesta seção, cuja atualização depende da leitura dos sensores de fluxo 1.1 e 1.2, representados pelos potenciômetros 1 e 2 do circuito acima. A válvula é fechada e o led se torna vermelho quando é detectada diferença entre as leituras dos sensores de fluxo 1.1 e 1.2, simulando, assim, um vazamento.
- “Boia (Caixa cheia)”: O Led verde representa que o reservatório superior de águas não está cheio. Caso ele esteja cheio, o Led vermelho se acende e o verde apaga. A atualização de estado desses Leds baseia-se na leitura do

potenciômetro 3, ou “Sensor de nível reservatório superior”, que simula o nível de água neste reservatório.

- “Válvula Solenoide 2”: Assim como na subseção “Válvula Solenoide 1”, o Led verde nesta seção indica que a válvula do cano 2 está aberta, permitindo o livre fluxo de água nesta seção, cuja atualização depende da leitura dos sensores de fluxo 2.1 e 2.2, representados pelos potenciômetros 5 e 6 do circuito acima. A válvula é fechada e o Led se torna vermelho quando é detectada diferença entre as leituras dos sensores de fluxo 2.1 e 2.2, simulando, assim, um vazamento.
- “Fornecimento potável”: O Led nesta subseção depende, assim como a seção “Boia (Caixa cheia)”, da leitura do potenciômetro 3 ou “Sensor de nível reservatório superior”, que simula o nível de água no reservatório superior de águas cinzas tratadas. Caso o nível deste sensor seja considerado vazio ou a uma porcentagem próxima do seu limite inferior, o Led se torna verde, acusando que o fornecimento para a rede será feito pela água potável fornecida pela rede, fechando a válvula que libera o fornecimento vindo do reservatório superior de água cinza tratada. O Led é vermelho quando o sensor de nível não acusa nível baixo no reservatório superior.
- “Caixa vazia”: Depende exatamente dos mesmos sinais que a subseção “Fornecimento potável”, onde o Led verde acusa que a o Reservatório superior está vazio ou próximo de seu limite inferior, e o Led vermelho indica que o reservatório possui volume suficiente para continuar abastecendo a rede com água cinza tratada.

Na seção de controle Local, há três botões com funcionalidades diferentes:

- “Corta fluxo cano1”: caso seja necessário que o fluxo de água seja cessado no cano 1 sem ser preciso que a leitura dos sensores de fluxo atuem ou mesmo o acionamento remoto, basta pressionar este botão. A porta OR-1 recebe sinais de entrada advindos do botão “Corta fluxo cano 1” e do acionamento remoto via software vindo da porta digital 9 do arduino. Caso um desses dois sinais sejam ativados, ou “1”, a válvula 1 é fechada.
- “Corta água cinza”: caso seja necessário que o abastecimento da rede proveniente do reservatório superior de águas cinzas tratadas seja cessado, passando, assim, o fornecimento a ser feito pela água potável da rede, sem ser preciso que a leitura do sensor de nível atue ou mesmo o acionamento remoto, basta pressionar este botão. A porta OR-2 recebe sinais de entrada advindos do botão “Corta água cinza” e do acionamento remoto via software vindo da

porta digital 10 do arduino. Caso um desses dois sinais sejam ativados, isto é, “1”, a válvula de 3 caminhos desta seção é fechada para o reservatório superior de águas cinzas, sendo aberta para o reservatório de água potável da rede.

- “Desliga bomba”: caso seja necessário que o bombeamento de água para o reservatório superior de água cinza tratada seja cessado, sem ser preciso que a leitura do sensor de nível atue ou mesmo o acionamento remoto, basta pressionar este botão. A porta OR-3 recebe sinais de entrada advindos do botão “Desliga bomba” e do acionamento remoto via software ScadaBR vindo da porta digital 11 do arduino. Caso um desses dois sinais sejam ativos, isto é, “1”, a bomba é desligada.

Para o circuito confeccionado no ISIS do Proteus, foi utilizado um Arduino Uno com o intuito de realizar o controle das variáveis. O código do programa encontra-se na sessão de anexos (Anexo 1) deste trabalho. Para realizar o download do código para o arduino, é necessário adquirir o código hexadecimal gerado pela IDE do Arduíno no ato da compilação e então inseri-lo no campo “Program File” que aparece quando clica-se duas vezes na figura do arduino virtual.

A Figura 3.10 demonstra a tela que deve surgir.

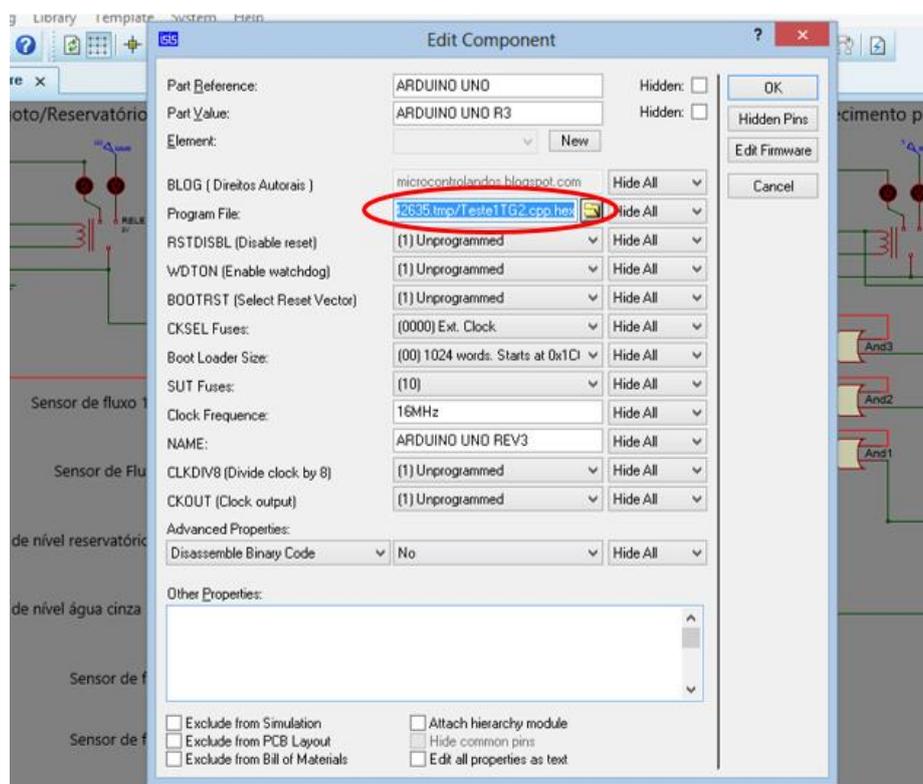
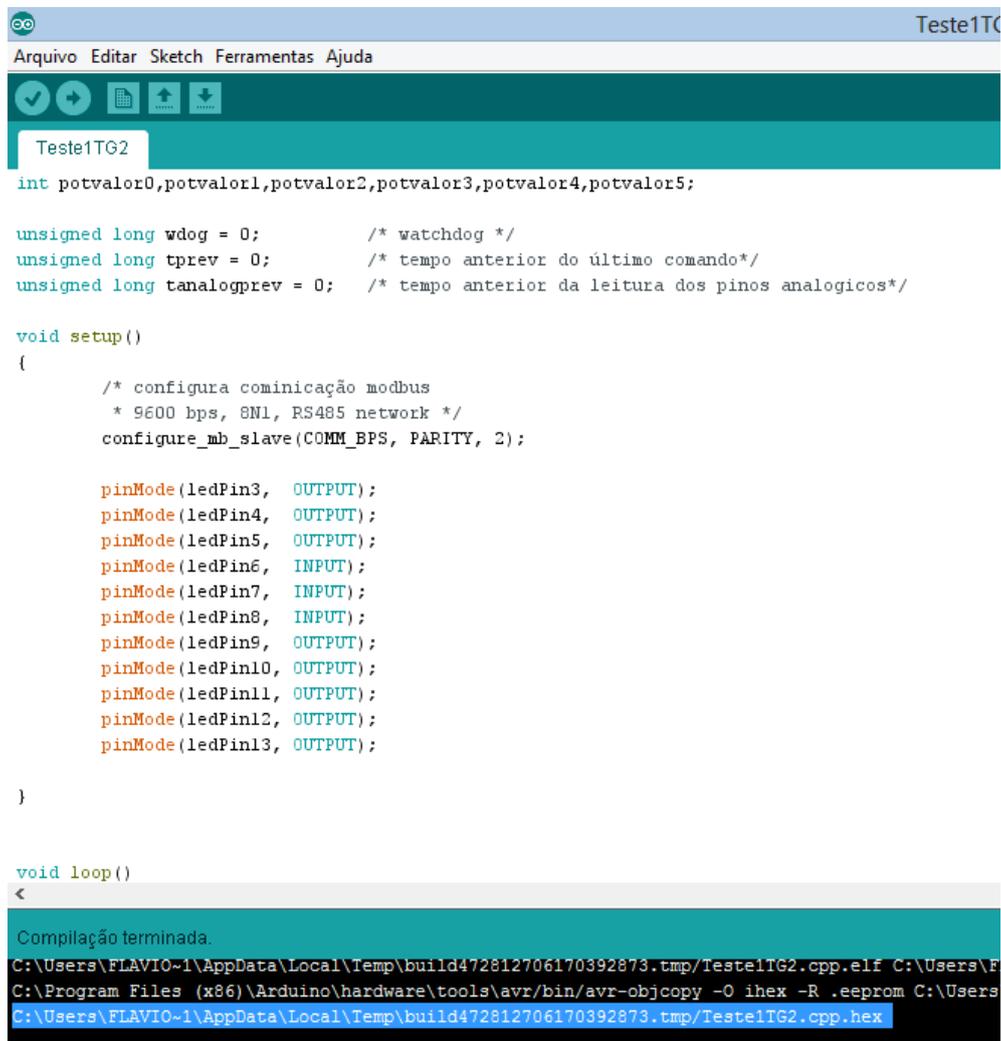


Figura 3.10. Inserção de Código para Arduino Virtual

O código hexadecimal gerado pelo arduino para carregamento no homônimo virtual do Proteus 8 encontra-se no caminho especificado pela área denotada pela Fig. 3.11.



```
Teste1TG2
Arquivo Editar Sketch Ferramentas Ajuda
Teste1TG2
int potvalor0,potvalor1,potvalor2,potvalor3,potvalor4,potvalor5;

unsigned long wdog = 0;          /* watchdog */
unsigned long tprev = 0;        /* tempo anterior do último comando*/
unsigned long tanalogprev = 0;  /* tempo anterior da leitura dos pinos analogicos*/

void setup()
{
  /* configura comunicação modbus
   * 9600 bps, 8N1, RS485 network */
  configure_mb_slave(COMM_BPS, PARITY, 2);

  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);
  pinMode(ledPin5, OUTPUT);
  pinMode(ledPin6, INPUT);
  pinMode(ledPin7, INPUT);
  pinMode(ledPin8, INPUT);
  pinMode(ledPin9, OUTPUT);
  pinMode(ledPin10, OUTPUT);
  pinMode(ledPin11, OUTPUT);
  pinMode(ledPin12, OUTPUT);
  pinMode(ledPin13, OUTPUT);
}

void loop()
<
Compilação terminada.
C:\Users\FLAVIO-1\AppData\Local\Temp\build472812706170392873.tmp\Teste1TG2.cpp.elf C:\Users\F
C:\Program Files (x86)\Arduino\hardware\tools\avr\bin\avr-objcopy -O ihex -R .eeprom C:\Users
C:\Users\FLAVIO-1\AppData\Local\Temp\build472812706170392873.tmp\Teste1TG2.cpp.hex
```

Figura 3.11. Localização do Código Hexadecimal para o Arduino Virtual

Feito isso, o código já estará carregado no Arduino virtual e estará pronto para a simulação. Como o circuito virtual será integrado com o ScadaBR, é necessário realizar uma comunicação virtual entre os dois softwares através do simulador de portas virtuais VSPE, que é explanado na seção de Softwares utilizados. Uma porta será utilizada para a comunicação com o circuito confeccionado no ISIS do Proteus, e outra para a comunicação com o ScadaBR.

A etapa seguinte é a integração do sistema, modelado pelo circuito, com o supervisor ScadaBR. Embora suas vias de comunicação estejam abertas, há algumas configurações a serem realizadas.

Para acesso ao ScadaBR, é preciso digitar, na barra de endereços de um navegador, **http://localhost:8080/ScadaBR/login.htm**. A tela da Fig. 3.12 aparecerá, para a qual o login de acesso é “admin” e senha “admin” (as credenciais de acesso podem ser modificadas).

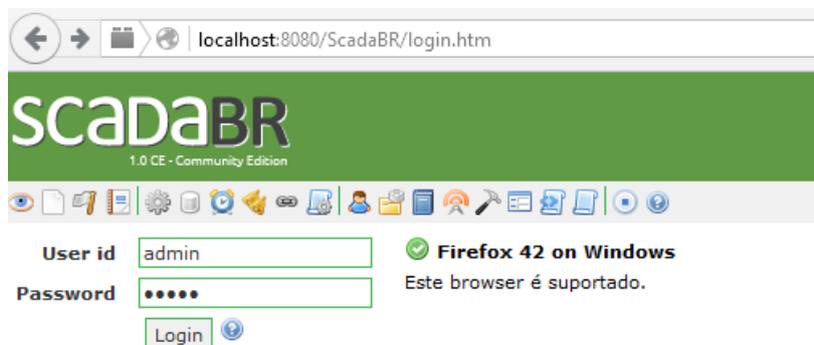


Figura 3.12. Acesso ao ScadaBR

Na barra de opções, em “Data sources” é possível verificar o status do Data Source criado, assim como editá-lo na opção “Editar Data source” conforme a Fig. 3.13 salienta.



Figura 3.13. Edição de Data Sources

Note que é possível adicionar novos Data Sources. No trabalho atual, assim como visto na seção “Tipo” da imagem acima, é utilizado o protocolo de comunicação Modbus Serial.

Dentro da opção “Editar Data Source”, podemos adicionar novos Data Points, que são pontos de aquisição de dados que relacionam registradores do Supervisório com pontos de aquisição de dados das portas do Arduino virtual, além disso, podemos editar a velocidade de aquisição de dados, que neste caso é de 9600, o baud rate, o intervalo de aquisição de dados, neste caso de 5 segundos, porta de comunicação, COM4 para este trabalho, dentre outros aspectos conforme a Fig. 3.14.

Figura 3.14. Edição de Data Sources - Opções

Abaixo, ainda na seção de Data Sources, é possível adicionar Data Points para aquisição de dados.

Plantas mais complexas geralmente exigem sistemas com conjuntos maiores de Data Points, para fins de controle e averiguação de um maior número de variáveis. Sistemas simplificados, no entanto, podem lançar mão de Data Points que representem variáveis agrupadas, de modo a reduzir a dispersão de valores gerados. Um exemplo seria uma aplicação de medição de temperaturas, em que diversos sensores são utilizados e o valor médio de suas aferições é o utilizado como entrada em um sistema. A utilização desse número de sensores é necessária para captação da temperatura ambiente, e não pontual, de forma que a nível físico eles são necessários. Para uma representação, no entanto, basta utilizar um Data Point para todos os sensores e, por uma rotina gerada por um software acessório, ou pelo próprio código do processador utilizado, selecionar o valor médio para ser enviado ao supervisor.

A Figura 3.15 lista todos os Data Points presentes no projeto assim como seus nomes, seu tipo varia entre binário e numérico, ou, em outras palavras, digital e analógico. No círculo vermelho na figura estão presentes as opções para adição de novos Data Points e a opção de habilitar todos os Data Points de uma única vez. Para saber se o Data Point está ativo basta olhar o seu Status, de modo que verde significa ativo para leitura e escrita em registradores e vermelho inativo.

Nome	Tipo de dado	Status	Escravo	Faixa	Offset (baseado em 0)	
AguaCinzaSujaCheio	Binário		1	Registrador holding	10/0	
Caixa cheia	Binário		1	Registrador holding	1/0	
Caixa vazia	Binário		1	Registrador holding	2/0	
Comando de Bomba	Binário		1	Registrador holding	4/0	
Comando Remoto AC	Binário		1	Registrador holding	8/0	
Comando Remoto Bomba	Binário		1	Registrador holding	9/0	
Comando Remoto Cano1	Binário		1	Registrador holding	7/0	
Corta Água Cinza	Binário		1	Registrador holding	5/0	
Fluxo 1.1	Numérico		1	Registrador holding	11	
Fluxo 1.2	Numérico		1	Registrador holding	12	
Fluxo 2.1	Numérico		1	Registrador holding	15	
Fluxo 2.2	Numérico		1	Registrador holding	16	
Fornecimento Cano1	Binário		1	Registrador holding	6/0	
Nível Caixa	Numérico		1	Registrador holding	13	
Sensor caixa Cheia	Numérico		1	Registrador holding	14	
Vazamento 1	Binário		1	Registrador holding	0/0	
Vazamento 2	Binário		1	Registrador holding	3/0	

Figura 3.15. Lista de Data Points

Essa tela também inclui a opção de adicionar ou editar novos data points presentes dentro da seção de edição de data sources. Para editar cada um dos Data points, basta clicar na opção mostrada pelo círculo azul. Edições como tipo de registradores, nome e número offset dos registradores são possíveis, conforme a Fig. 3.16 demonstra.

Detalhes do data point

Nome

Export ID (XID)

Id do escravo

Faixa do registro

Tipo de dados modbus

Offset (baseado em 0)

Bit

Número de registradores

Codificação de caracteres

Configurável

Multiplicador

Aditivo

Figura 3.16. Edição de Data Points

Para uma visão geral dos estados e valores das variáveis de interesse presentes no sistema, é possível acessar uma “Watch List”. Para isso, no menu de seleção do ScadaBR, basta selecionar a opção “Watch List” conforme o círculo na Fig. 3.17 mostra.



Figura 3.17. Ícone de Acesso às Watch Lists

Clicando nesta opção é possível acompanhar em tempo real, com atualização a cada 5 segundos, o valor de todos os sensores, assim como criar mais de uma Watch list com os sensores que o usuário deseje observar. Para este projeto, há diversas Watch Lists, cujas funcionalidades são relacionadas a todos os sensores, aos sensores relacionados à bomba, aos sensores relacionados ao cano 1 e aos sensores relacionados ao reservatório superior de águas cinzas tratadas. Para observar cada uma dessas basta que haja seleção na seção indicada na Fig. 3.18. O valor de cada sensor, assim como o horário de atualização são mostrados em cada Watch List.

Watch list	Value	Time	Actions
TESTE 1 - Fluxo 1.1	553.0		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Fluxo 1.2	553.0		<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Fluxo 2.1	512.0	13:28:42	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Fluxo 2.2	553.0	13:28:43	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Sensor caixa Cheia	389.0	13:28:47	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Nível Caixa	635.0	13:28:44	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Vazamento 2	0	13:28:44	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Vazamento 1	0	13:28:45	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
Nov 06 20:21 - Vazamento 1 foi 1 por mais de 2 minuto(s) <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> Out 22 10:46 - Vazamento 1 foi 1 por mais de 2 minuto(s) <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>			
TESTE 1 - Caixa cheia	0	13:28:49	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Comando de Bomba	0	13:28:46	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Fornecimento Cano1	0	13:28:50	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Corta Água Cinza	0	13:28:51	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Comando Remoto Bomba	0	13:28:53	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Comando Remoto AC	0	13:28:53	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
TESTE 1 - Comando Remoto Cano1	0	13:28:55	<input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>

Figura 3.18. Seleção de Watch List

Com uso do desenho realizado no Autocad e de recursos do ScadaBR, uma representação gráfica dinâmica foi preparada para acompanhamento em tempo real dos sensores e atuadores do sistema. Para ter acesso a essa representação, basta clicar na opção “Representação gráfica” no menu do ScadaBR conforme a indicação na Fig. 3.19.

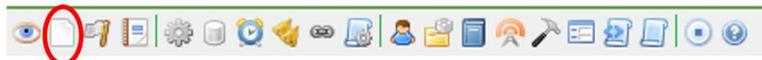


Figura 3.19. Ícone de Acesso às Representações Gráficas

Na janela que se abre, é possível adicionar uma nova representação gráfica assim como selecionar uma das duas representações gráficas criadas para este projeto, Planta Geral ou Cano1. Essa opção está representada na Fig. 3.20.



Figura 3.20. Seleção de Representação Gráfica

Surge, então, a seguinte imagem atualizada em tempo real (Fig. 3.21).

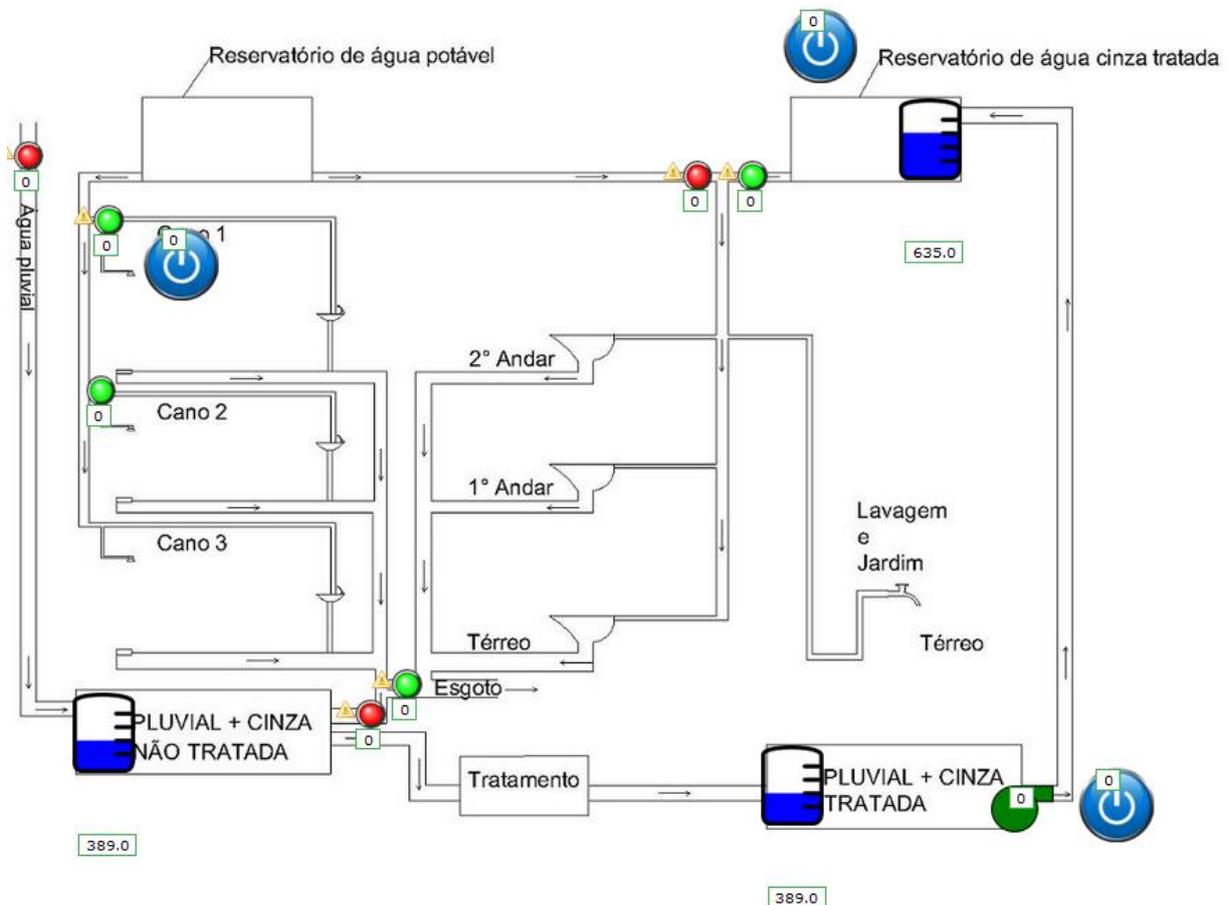


Figura 3.21. Tela de Supervisão – Representação Gráfica

Todos os recursos do ScadaBR utilizados acima são dinâmicos e configuráveis, atualizando seus estados em tempo real de acordo com as leituras feitas no circuito. Os Leds vermelhos e verdes, representam respectivamente, fluxo bloqueado e fluxo livre. Eles fazem a simulação do papel exercido pelas válvulas controladas. Os botões On/Off representam a possibilidade de o usuário atuar em válvulas específicas remotamente através do Software supervisor ScadaBR. Caso o valor inserido em cada botão seja “0”, o fluxo é liberado, ou a bomba liga para o bombeamento de água cinza para o reservatório superior. Caso o valor do botão seja “1”, o fluxo das válvulas controladas por eles é cortado, ou então a bomba é desligada, caso o botão seja o controlador desta.

Os reservatórios de “Água pluvial + Cinza não tratada”, “Pluvial + Cinza Tratada”, “Água cinza tratada”, são representados no circuito por potenciômetros que podem constantemente ser alterados durante a simulação do circuito, simulando o nível de cada um desses reservatórios e respectivamente suas atuações em válvulas dependente dos valores de suas leituras. A atuação que cada sensor de nível faz em cada válvula será explanado mais à frente.

Em relação ao padrão de operação das válvulas, representadas pelos LEDs, baseado em sua lógica de atuação, temos as seguintes rotinas:

- Válvula de água pluvial: esta válvula evita que águas pluviais sejam coletadas caso o reservatório de “Água cinza + pluvial” acuse nível máximo permitido atingido. Esse sensor é simulado através de um potenciômetro e quando 95% de sua capacidade é atingida, a válvula recebe o comando de fechar. Já quando o nível permanece abaixo dos 95%, ela recebe o comando automático para liberar o fluxo para captação de água pluvial. Nesta simulação, o comando remoto via software não foi implementado para tal válvula, porém é claro que não há dificuldade para implementação dele caso seja necessário.
- Válvula Cano 1: responsável por cortar o fluxo de água caso seja detectado um vazamento no Cano 1. O vazamento é detectado através da leitura de dois sensores de fluxo, dispostos um antes e outro após da válvula. Caso o fluxo que passa através de um sensor não seja o mesmo fluxo identificado pelo outro, será enviado um alerta de vazamento e conseqüentemente a válvula cortará o fluxo. Além da capacidade da leitura dos sensores de fluxo atuarem sobre a válvula, ela é também controlada remotamente via software (Supervisor), ou mesmo no quadro elétrico do circuito através de um botão de comando. Para se ter uma melhor visão dos sensores de fluxo, suas leituras e atuação da válvula 2, basta selecionar na representação gráfica a opção “Cano 1” para que a seguinte imagem seja apresentada (Fig. 3.22).



Figura 3.22. Válvula – Representação ScadaBR

Nesta representação, nota-se que ambos os sensores de fluxo, 1.1 e 1.2, acusam o mesmo valor de fluxo, o que representa uma situação sem vazamento, deixando a válvula no estado “0” ou aberto.

- Válvula Cano 2: assim como a válvula do cano 1, ela é responsável por cortar o fluxo de água caso seja detectado um vazamento no Cano 2. O vazamento é detectado através da leitura de dois sensores de fluxo dispostos um antes e outro após da válvula. Caso o fluxo que passa através de um sensor não seja o mesmo fluxo identificado pelo outro, será enviado um alerta de vazamento e consequentemente a válvula cortará o fluxo. Diferente da válvula do cano 1, esta válvula não possui o modo de acionamento remoto apenas para simplificar a demonstração, porém a sua implementação é pouco complexa, caso seja necessária no projeto real.
- Válvulas “Reservatório de água cinza tratada”: as duas válvulas (Led’s) presentes nesta seção podem ser exemplificadas pelo funcionamento de uma válvula de 3 caminhos. Caso o reservatório de água cinza tratada esteja abastecido, isto indica que não é necessário fazer o uso de água potável para fins de descarga e jardim, e, com isso, o único caminho para a água é sair do reservatório de águas cinzas tratadas em direção ao seu devido uso. Porém, caso tal reservatório seja detectado como nível baixo ou vazio, o sistema de descargas e jardinagem não podem ter seu uso comprometido. Quando é lido nível insuficiente do reservatório de água cinza, imediatamente a válvula muda seu fluxo, vindo agora da caixa de água potável que é fornecida pela rede de abastecimento.

O funcionamento desta válvula evita que haja possíveis misturas de água potável com águas cinzas.

- Válvulas “Esgoto”: as suas válvulas nesta seção são também a simulação de uma válvula de 3 caminhos. Estas válvulas atuam dependendo da leitura do nível de água cinza recolhida da rede no reservatório de “Água pluvial + cinza

não tratada”. Caso o nível deste reservatório seja considerado cheio, o caminho das águas cinzas da rede vão diretamente para a rede de esgoto evitando o transbordamento ou outros problemas na rede. Porém, se o reservatório ainda possuir capacidade de armazenamento, a coleta de água cinzas vai diretamente para o reservatório fazendo que a saída para a rede de esgoto seja bloqueada.

A descrição da finalidade de cada reservatório segue:

- Reservatório de água potável: para este reservatório, não é utilizado o sistema do nível de água em razão de assumir que tal fornecimento vem da rede local de abastecimento, e que portanto, não cessará.
- Reservatório de “Água pluvial + cinza não tratadas”: este reservatório coleta águas de chuva, de chuveiros e de pias de banheiro, visto que a água da pia de cozinha é descartada na rede de esgoto. Há um sensor para constante leitura do nível deste reservatório. Em nossa simulação, tal sensor de nível é simulado por um potenciômetro, que, quando atinge um valor acima de 95% é dado com cheio, fazendo assim com que sinais sejam emitidos para válvulas específicas, a fim de que as devidas providências sejam tomadas.
- Reservatório inferior de “água pluvial + cinza tratada”: este reservatório faz o tratamento das águas coletadas no reservatório anterior ao reservatório de “águas pluviais + cinza não tratadas” para que as devidas propriedades para o reuso destas águas estejam em conformidade às normas. Há também um sensor de nível simulado através do mesmo potenciômetro do reservatório anterior para a facilidade do entendimento, ressaltando que é possível fazer com que tal sensor seja independente. Este reservatório inferior possui uma bomba controlada responsável pelo bombeamento para o reservatório superior, responsável por realizar a distribuição desta água para os devidos fins.
- Reservatório superior de “águas cinzas tratadas”: após o tratamento de toda água coletada, a água é bombeada para este reservatório para que seja distribuída aos determinados fins. Caso o sensor de nível deste reservatório acuse que ele está cheio, a bomba inferior é desligada evitando o transbordamento de líquido. Por outro lado, caso este seja considerado vazio, um sinal é enviado para a válvula de 3 caminhos na saída para que o fornecimento seja cessado deste reservatório e passe a ser obtido de água potável da rede, para que o sistema não fique ineficiente. Este reservatório é também simulado como um potenciômetro.

Por fim, a descrição da bomba hidráulica controlada presente no sistema:

- Bomba do reservatório inferior de águas cinzas e pluviais tratadas: responsável pelo bombeamento de águas cinzas tratadas para o reservatório superior. Trata-se de uma bomba que pode ser controlada automaticamente (dependente do nível do reservatório superior), remotamente (via software SCADA) ou mesmo localmente por um botão. Esta bomba assume a coloração verde caso esteja ligada e vermelha, caso contrário.

Fluxogramas são grandes aliados para o fácil entendimento do funcionamento de um sistema. Para a rede em questão, foram montados fluxogramas de estados independentes para a bomba e válvulas da rede. Para simplificação, as seleções em azul e vermelho abaixo, representam, em cada uma delas, uma única válvula de 3 caminhos (Fig. 3.23).

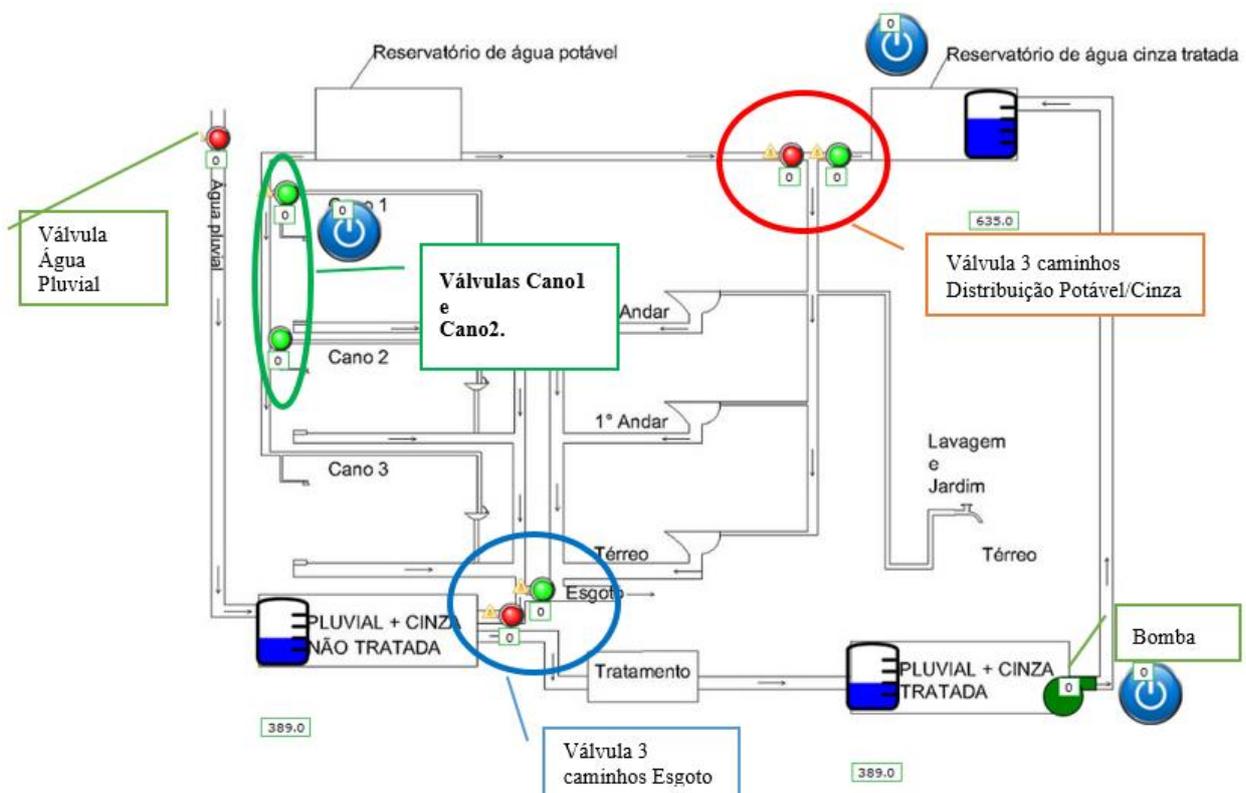


Figura 3.23. Seleção de Componentes para Representação em Fluxograma

A figura 3.24 corresponde ao fluxograma da válvula de distribuição de água potável/cinza.

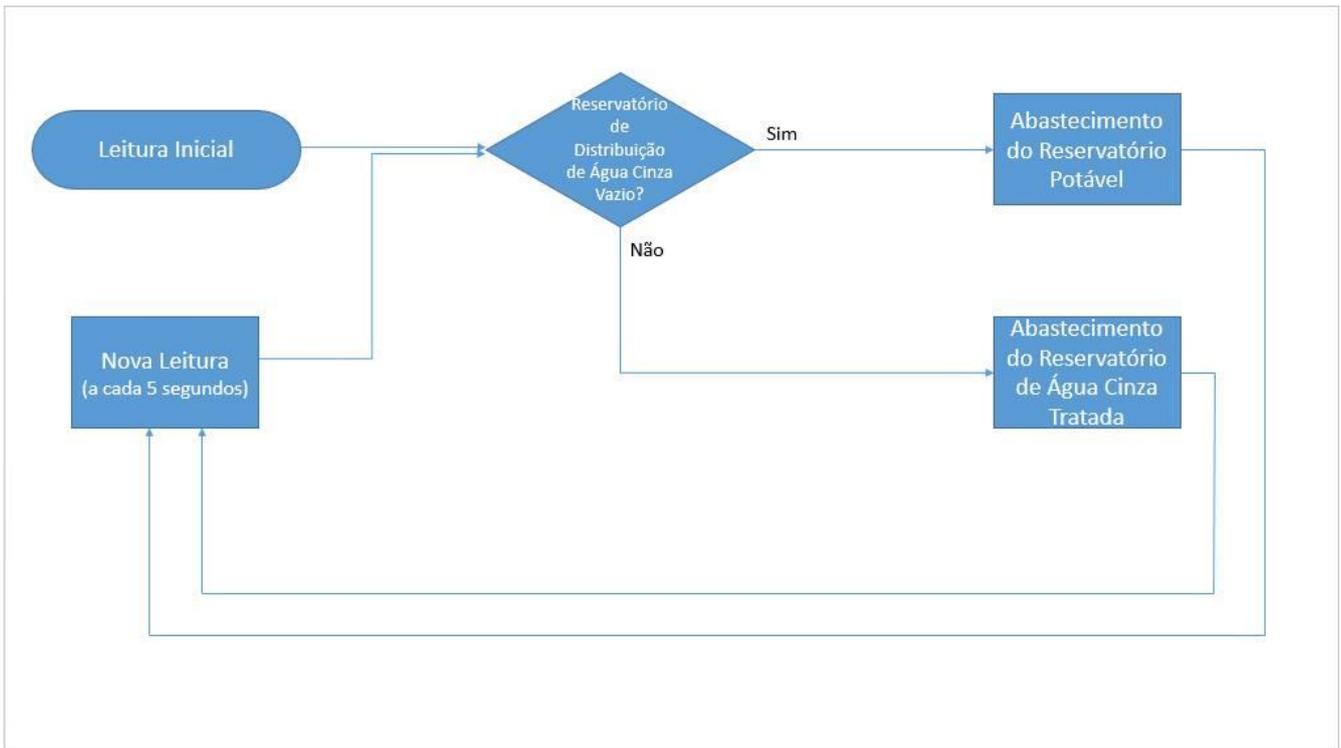


Figura 3.24. Fluxograma – Válvula 3 Caminhos – Distribuição de Água Potável/Cinza

A figura 3.25 denota o fluxograma das válvulas do Cano1 e Cano2.

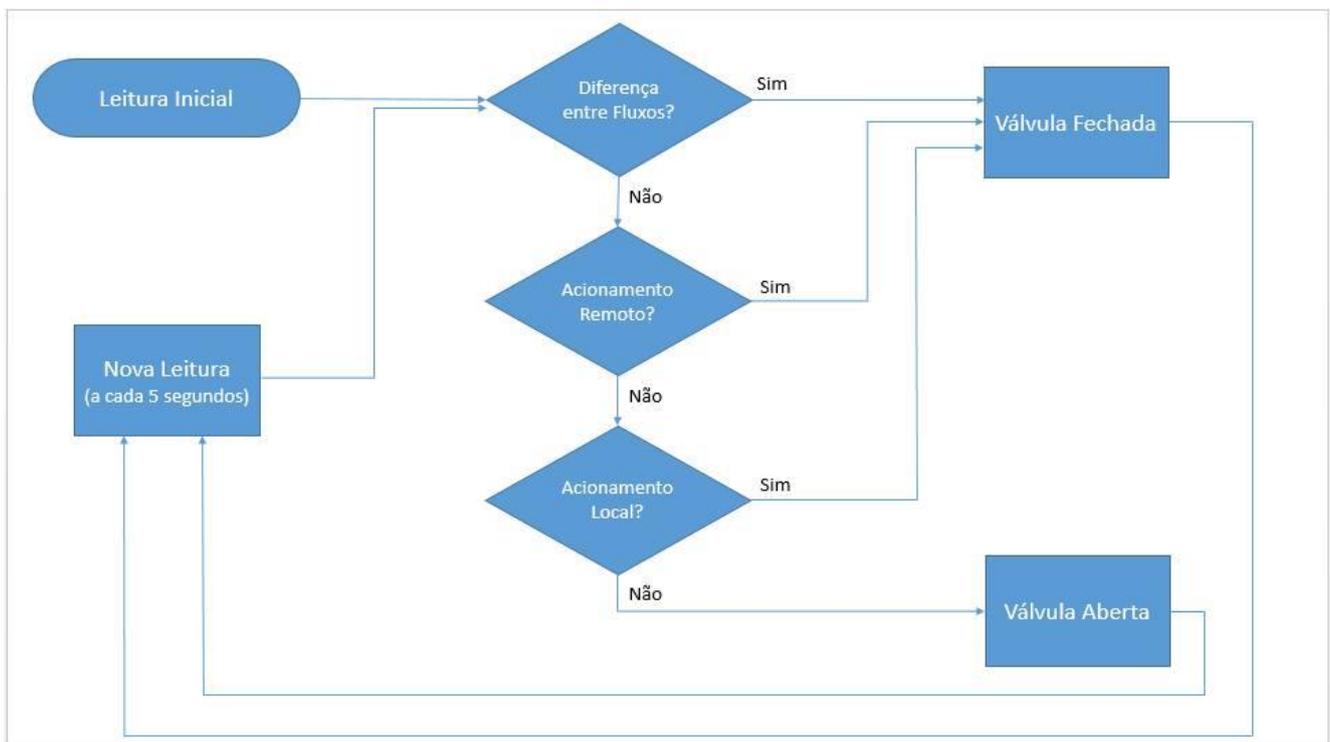


Figura 3.25. Fluxograma – Válvulas – Cano1 e Cano2

Já a figura 3.26 corresponde ao fluxograma da bomba hidráulica.

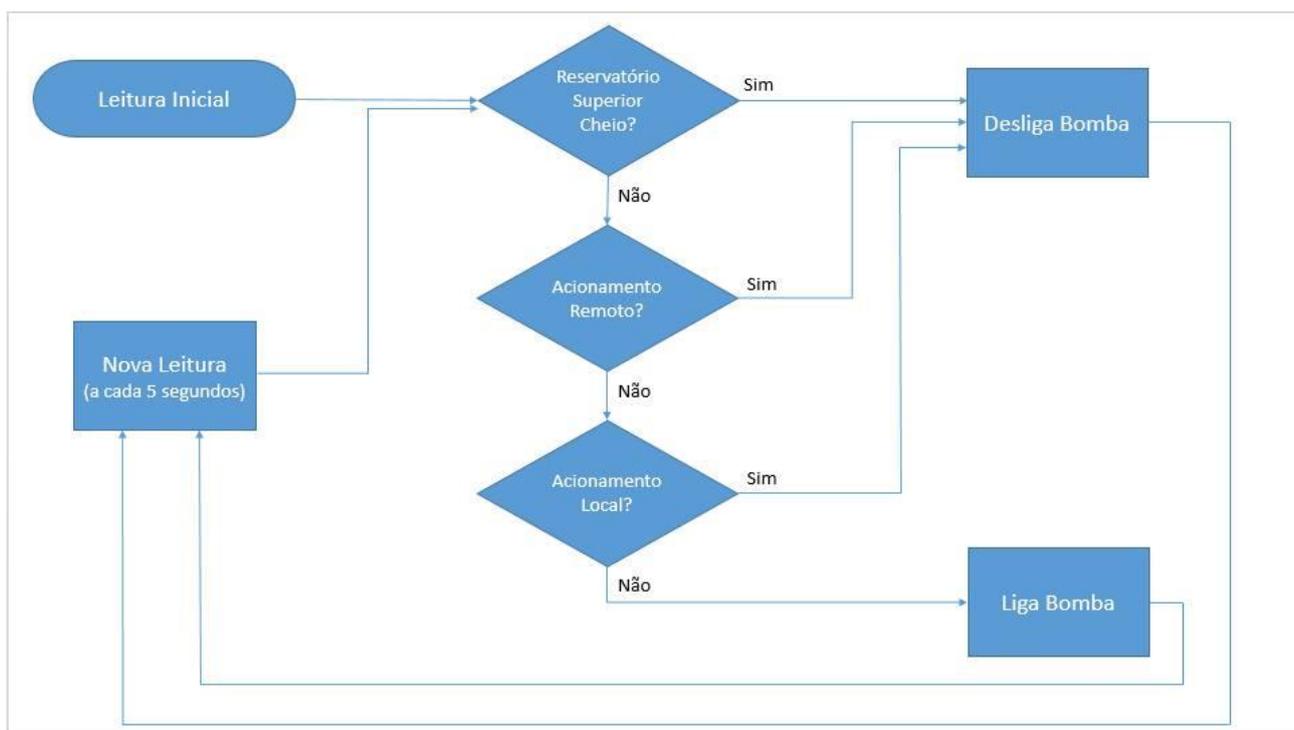


Figura 3.26. Fluxograma – Bomba Hidráulica

De semelhante modo, a figura 3.27 corresponde ao fluxograma da válvula de esgoto.

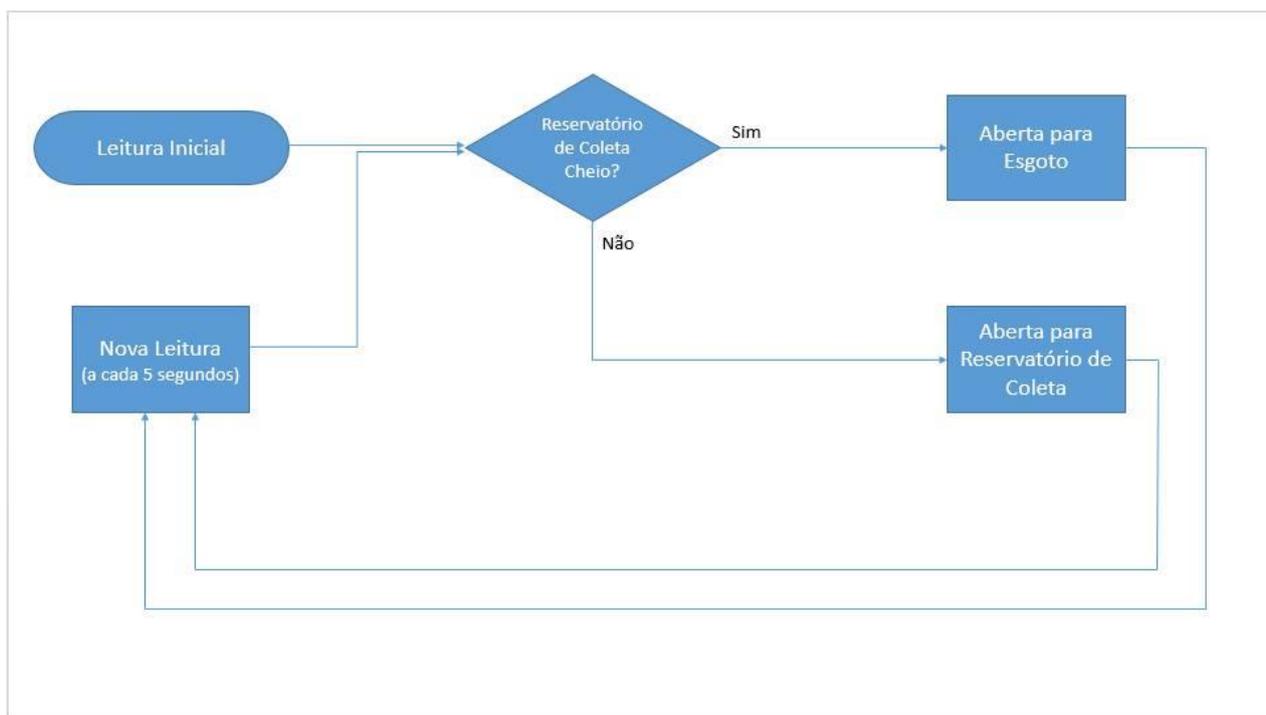


Figura 3.27. Fluxograma – Válvula 3 Caminhos - Esgoto

Por fim, o último fluxograma é representado na a figura 3.28, pertencente à válvula pluvial.

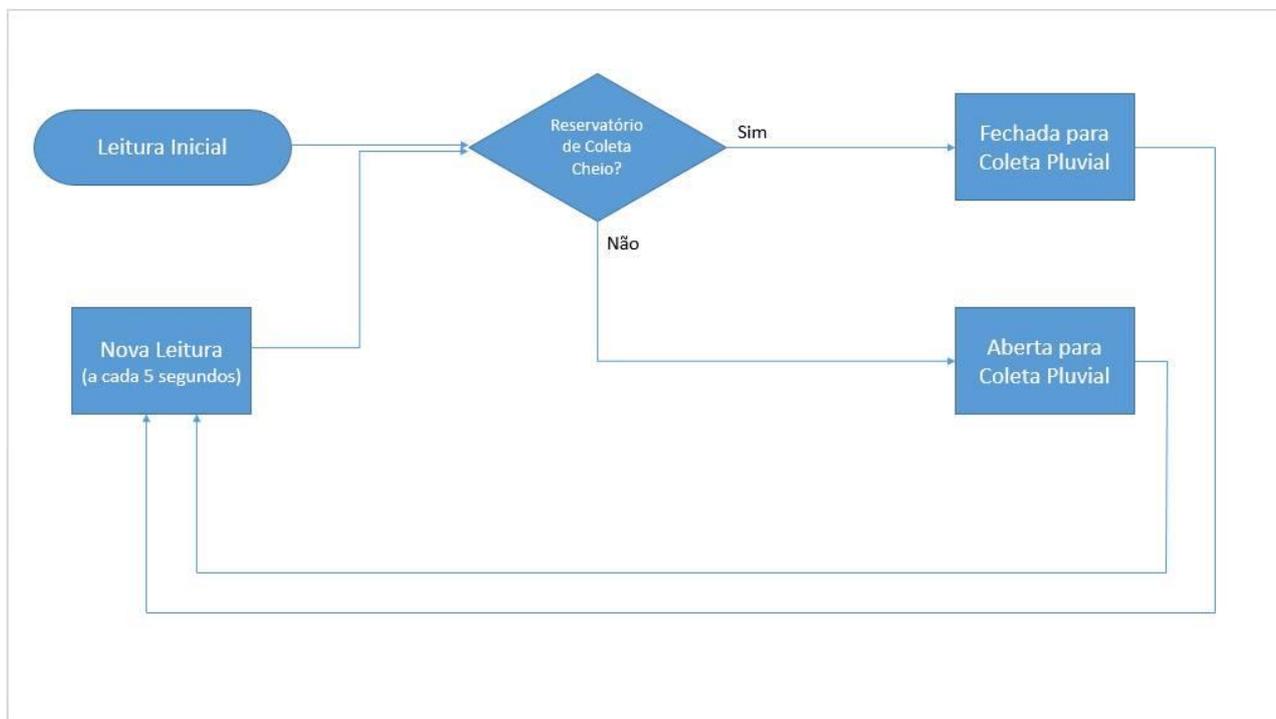


Figura 3.28. Fluxograma – Válvula Pluvial

4 CONSIDERAÇÕES FINAIS

Assim como o posicionamento de um ouvinte após atenta ponderação sobre tema explorado em palestra explanatória, o viés ou ponto de observação de um leitor deste texto deve ser complementado com um encerramento digno de todo o conteúdo até então disposto. Há de se evitar, no entanto, confundir esta revisitação das metas propostas inicialmente com o término de possibilidades de difusão de aplicações referentes à implementação de sistemas supervisórios nesse contexto, item que permanece em franca expansão devido não apenas à conjectura contemporânea que viabiliza e prioriza quesitos de automação, mas à natureza modular do trabalho proposto, que pode ser extraído e acoplado a novos sistemas passivos de otimização.

Quanto às vantagens de uma abordagem que gere alternativas viáveis para reuso de água cinza e pluvial, com parâmetros controláveis, é relevante notar que seus retornos são de caráter de médio a longo prazo, o que desagradaria visões mais imediatistas. O próprio conceito de reutilização e remanejamento dos recursos hídricos consiste em obter pequenas, mas constantes, facilidades que, associadas à rede de alimentação da construção, atendam à demanda do grupo usuário. Além disso, a integração em nível de grande escala endereçaria uma questão de caráter humanitário e não individual, que é a minimização dos dispêndios exagerados de água por parte da população. O cenário futuro, quando se analisam projeções de demanda hídrica, é alarmante. De acordo com a OCDE (Organização para Cooperação e Desenvolvimento Econômico), o incremento populacional que levará a população mundial a alcançar estrondosos 9 bilhões de indivíduos, também resultará em um acréscimo no consumo de água da magnitude de 55%. Um gerenciamento que pouco se interesse por questões futuras decorrerá em aproximadamente 40% de toda a população mundial vivendo em regiões de severo estresse hídrico. É a esses interesses que este projeto busca atender.

Para melhor compreensão da relevância dessa abordagem, é vital que se considerem os procedimentos para controle e leitura de variáveis da distribuição de líquidos no edifício em questão. O intuito é atacar os excedentes de demanda hídrica com as melhores ferramentas disponíveis. Enquanto lavar carros e calçadas com uso de baldes em vez de mangueiras e reduzir o tempo de banho são atitudes pontuais eficazes no combate ao desperdício, agir sobre a fonte prova-se ser mais que eficaz, atingindo também eficiência. Como um sistema automatizado busca respostas padronizadas para situações de vazamento, rotinas de racionamento também podem ser adotadas, interrompendo, por exemplo, o uso desmedido de água por algum usuário descuidado. Além disso, o gerenciamento de recursos externos, como os efluentes pluviais, adiciona um saldo positivo ao sistema, que passa a lançar mão

de uma porção outrora desprezada de líquido, em vez de apenas depender da quantidade limitada fornecida pelas redes usuais.

Em relação à correspondência do atingido a partir da metodologia aplicada, é notável que há relevância do trabalho desenvolvido no sentido de criar bases com parâmetros otimizados para evoluções seguintes no contexto de sistemas orientados a controle supervisorio aplicado a variáveis de vazão e nível, naturais de um ambiente de distribuição hídrica. O custo reduzido e a implementação facilitada por softwares *open source* tornam esse sistema adequado a uma diversidade de escopos, quando se consideram limitações financeiras. Além disso, este projeto pode ser facilmente ampliado para futuras atribuições que compreendam necessidades de construções específicas, residenciais ou não. Para isso, basta que sejam modificados os parâmetros de controle a partir do software de supervisão.

Sugere-se evitar a modificação do protocolo de comunicação utilizado para futuras realizações, visto que esse provou-se um ponto de gargalo quando considerado o tempo total de realização do projeto. Caso haja necessidade incontornável dessa modificação, protocolos proprietários cuja interface esteja atrelada ao dispositivo de processamento de dados são preferenciais (associação de um CLP ao supervisorio de sua empresa, por exemplo). Ainda para enquadramento futuro, espera-se que a relevância de um melhor gerenciamento de recursos hídricos permaneça como um dos principais fatores para definição de um sistema sustentável, tema de crescente preocupação para a nossa sociedade.

Dessa forma, uma visão puramente econômica talvez não corresponda ao bem maior na atual conjuntura ambiental. Embora existam bases para justificar a utilização do sistema considerado para redução de gastos a longo prazo, o verdadeiro benefício consiste em possibilitar edifícios sustentáveis e que não gerem um futuro comprometedor para todas as partes envolvidas no processo de distribuição desse bem irreduzível às necessidades humanas, que é a água.

REFERÊNCIAS BIBLIOGRÁFICAS

ALTUS EVOLUTION IN AUTOMATION. AL and PX Series. ALTUS. São Leopoldo, 2015. Disponível em <http://altus.com.br/site_en/index.php?option=com_content&view=article&id=22&Itemid=24>. Acesso em: 23 nov. 2015.

AMBIENTE BRASIL (Comp.). Legislação e Normatização do Reúso da Água. AMBIENTE BRASIL. Disponível em <http://ambientes.ambientebrasil.com.br/agua/uso_e_reuso_da_agua/legislacao_e_normatizacao_do_reuso_da_agua.html> Acesso em: 23 nov. 2015

AMERICAN NATIONAL STANDARDS INSTITUTE / HYDRAULIC INSTITUTE. ANSI/HI 9.1-9.5: Pumps – General Guidelines. Washington, DC: ANSI, 2000.

ARDUINO HOMEPAGE. Genuino Mega 2560 Rev3. ARDUINO. Disponível em <<http://www.solucoesindustriais.com.br/empresa/maquinas-e-equipamentos/lds-maquinas/produtos/bombas-e-motobombas/bomba-hidraulica>>. Acesso em: 23 nov. 2015.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 5626:1998: Instalação predial de água fria. Rio de Janeiro: ABNT, 1998.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 10844:1989: Instalações prediais de águas pluviais. Rio de Janeiro: ABNT, 1989.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 13969:1997: tanques sépticos: unidades de tratamento complementar e disposição final dos efluentes líquidos – projeto, construção e operação. Rio de Janeiro: ABNT, 1997.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR 15527:2007: Água de chuva – Aproveitamento de coberturas em áreas urbanas para fins não potáveis - Requisitos. São Paulo: ABNT, 2007.

BEM TE FIZ. Como tratar as águas cinzas. Zan Huke, Karin. São Paulo, 1 de mai. 2010. Disponível em <<http://www.bemtefiz.com.br/sustentabilidade/como-tratar-as-aguas-cinzas/>> Acesso em: 23 nov. 2015.

Bloch, D. Água: Direito à Vida. Disponível em: <<http://ircsa.org.br/semana/aguavida.htm>>. Acesso em: 23 nov. 2015.

CERTI-SCADABR. Iniciando com o ScadaBR. CERTI. Disponível em <https://sites.google.com/a/certi.org.br/certi_scadabr/home/minicursos/iniciando-scadabr>. Acesso em: 23 nov. 2015.

Da Silva, Ana Paula Gonçalves; Salvador, Marcelo. O que são sistemas supervisórios? São Paulo, 2005. Disponível em <http://www.wectrus.com.br/artigos/sist_superv.pdf>. Acesso em: 23 nov. 2015.

Dantas, Mário. Redes de Comunicação e Computadores. Visual Books, Florianópolis, 2009.

Decotignie, Jean D. Ethernet-based real-time and industrial communications. IEEE, 2005.

DMAES PONTE NOVA. Programa de Redução de Perdas do DMAES: economia de água chega a quase 500 milhões de litros. DMAES. Ponte Nova, 23 de mar. 2015. Disponível em <<http://www.dmaespontenova.com.br/index.php/noticias/413-programa-de-reducao-de-perdas-do-dmaes-economia-de-agua-chega-a-quase-500-milhoes-de-litros>> Acesso em: 23 nov. 2015

ECODESENVOLVIMENTO. Guia do consumo consciente de água. EcoD. Sem sede, 26 de abr. 2015. Disponível em <<http://www.ecodesenvolvimento.org/posts/2012/abril/guia-do-consumo-consciente-de-agua>> Acesso em: 23 nov. 2015

Eriksson, J.; Coester, M; Hennig, C.H. Redes Industriais – Panorama histórico e novas tendências – Revista Controle e Instrumentação, nº 119, ago. 2006, pg 86-89.

Felser, Max. Real-Time Ethernet Industry Prospective. IEEE, 2005.

Felser, Max; Sauter, Thilo. The fieldbus war: history or short break between battles? In: IEEE Int. Workshop Factory Communication Systems, 2002.

Forouzan, Behrouz A. Comunicação de Dados e Redes de Computadores / Behrouz A. Forouzan; tradução Glayson Eduardo de Figueiredo. Bookman, Porto Alegre, 2006.

G1 - GLOBO. Ministério do Meio Ambiente estuda criar norma nacional de reúso de água. Carvalho, Eduardo. São Paulo, 23 de abr. 2015. Disponível em <<http://g1.globo.com/natureza/noticia/2015/04/brasil-discute-criar-normas-nacionais-de-reuso-de-agua-diz-ministra.html>> Acesso em: 23 nov. 2015

Hermini, Helder Anibal. Sensores de Nível e Capacidade. Campinas, 2007. Disponível em <<http://www.fem.unicamp.br/~hermini/ES746/Aula%2007%20.ppt>>. Acesso em: 23 nov. 2015.

Hespanhol, I. Potencial de reúso de água no Brasil: agricultura, indústria, município e recarga de aquíferos. Bahia Análise & Dados, Salvador, 2003.

ICOS SENSORES PARA LÍQUIDOS. Sensor de Nível LA16M-40 – Água, Óleo, Combustíveis. ICOS SENSORES PARA LÍQUIDOS. Sorocaba, 2015. Disponível em <<https://www.icos.com.br/SensorDeNivel/LA16M40/>>. Acesso em: 23 nov. 2015.

INTERNATIONAL ELECTROTECHNICAL COMMISSION. IEC 61784-2, jun. 2009;

Kasberger, Johannes. Advantages of Industrial Ethernet – Comparison of Modbus over TCP/IP and PROFINET. Wien, 2011. Disponível em <<https://www.auto.tuwien.ac.at/bib/pdf/TR/TR0153.pdf>>. Acesso em: 23 nov. 2015.

KLEIN AUTOMAÇÃO. Válvula Solenoide 2/2 Vias – 2W. KLEIN AUTOMAÇÃO. Sorocaba, 2015. Disponível em <<http://kleinautomacao.com.br/br/produtos/pneumatica/valvulas-solenoides-acionamento-eletrico/-valvula-solenoide-2-2-vias-2w/>>. Acesso em: 23 nov. 2015.

- LIVRE PENSAR. As projeções sobre o futuro da água. LABORATÓRIO DE GARAGEM. Barueri, 2015. Disponível em <<http://www.labdegaragem.org/loja/sensor-de-fluxo-de-agua-g-3-4.html>>. Acesso em: 23 nov. 2015.
- LABORATÓRIO DE GARAGEM. Sensor de fluxo de água – G 3/4. Fred Seifert. 2014. Disponível em <<http://www.ideiasustentavel.com.br/2014/01/as-projecoes-sobre-o-futuro-da-agua/>>. Acesso em: 23 nov. 2015.
- May, Simone. Caracterização, tratamento e reuso de águas cinzas e aproveitamento de águas pluviais em edificações. São Paulo, 2009. Disponível em <http://www.teses.usp.br/teses/disponiveis/3/3147/tde-17082009-82126/publico/SIMONE_MAYOK.pdf>. Acesso em: 23 nov. 2015.
- Mendonça, Pedro de A. Ornelas. Reúso de água em edifícios públicos. O caso da escola politécnica. Salvador, 2004. 162 p. Dissertação (Mestrado em gerenciamento e tecnologias ambientais no processo produtivo) – Universidade Federal da Bahia, Bahia, 2004.
- Mountford, Helen. Water: The Environmental Outlook to 2050. Paris, 2011. Disponível em <<http://www.oecd.org/env/resources/49006778.pdf>>. Acesso em: 23 nov. 2015.
- PORTAL O SETOR ELÉTRICO. Da máquina a vapor aos softwares de automação. Goeking, Weruska. Santa Cecília, 2010. Disponível em <<http://www.osetoreletrico.com.br/web/a-revista/343-xxxx.html>>. Acesso em: 23 nov. 2015.
- REFRIPARTS. Válvula Solenoide York Hidronico 3 Vias 3/4" APCVLV3201V1. REFRIPARTS. Piracicaba, 2015. Disponível em <<http://kleinautomacao.com.br/br/produtos/pneumatica/valvulas-solenoides-acionamento-eletrico-valvula-solenoide-2-2-vias-2w/>>. Acesso em: 23 nov. 2015.
- SINDUSCON. Conservação e reuso de água em edificações. São Paulo: Prol Editora Gráfica, 2005.
- SOLUÇÕES INDUSTRIAIS. Página Inicial: Bomba Hidráulica Modelo. SOLUÇÕES INDUSTRIAIS. São Paulo, 2015. Disponível em <<http://www.solucoesindustriais.com.br/empresa/maquinas-e-equipamentos/lds-maquinas/produtos/bombas-e-motobombas/bomba-hidraulica>>. Acesso em: 23 nov. 2015.
- Thomasse, Jean P. Control Engineering Practice. IFAC (International Federation of Automatic Control), Laxenburg, 1999.
- Thomasse, Jean P. Time and industrial local area networks. In: 7th Annual European Computer Configuration, Computer Design, Manufacturing and Production (COMPEURO' 93), 1993.
- Torres, G. Redes de Computadores – Versão Revisada e Atualizada – 2ª Edição. NovaTerra, Vila Isabel, 2014.
- Wiechert, M.D. Modbus Serial – RTU Slave Arduino Sketch. Disponível em <<http://www.mediafire.com/download/wi1wmc6mavv6auy/arduino+library.rar>> Acesso em: 23 nov. 2015.

APÊNDICES

Apêndice 1 – Código de Simulação de Circuito Virtual no Proteus 8.

O programa contido no Anexo 1 utiliza parte da Biblioteca Open Source desenvolvida por Juan Pablo Zometa para interação do protocolo Modbus. O código está formatado para uso na IDE do Arduino, assim como os comentários que são realizados nas diversas linhas de código representados por "//", que faz com que todos os escritos da linha a frente deste comando não sejam prejudiciais à programação, ou "/* */", que faz com que todo os escritos que estejam entre os dois asteriscos não sejam prejudiciais à lógica de programação.

/*

Modbus serial - RTU Slave Arduino Sketch

Marcos Daniel Wiechert wiechertdaniel@yahoo.com.br

Baseado na biblioteca de Juan Pablo Zometa : jpmzometa@gmail.com

<http://sites.google.com/site/jpmzometa/>

and Samuel Marco: sammarcoarmengol@gmail.com

and Andras Tucsni.

As funções do protocolo MODBUS implementadas neste código:

3 - Read holding registers;

6 - Preset single register;

16 - Preset multiple registers.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

The functions included here have been derived from the Modicon Modbus Protocol Reference Guide which can be obtained from Schneider at www.schneiderautomation.com.

This code has its origins with

paul@pmcrae.freemove.co.uk (<http://www.pmcrae.freemove.co.uk>)

who wrote a small program to read 100 registers from a modbus slave.

*/

/*

* configure_mb_slave(baud, parity, tx_en_pin)

* configuração dos parametros da porta serial.

* baud: taxa de transmissão em bps (valores típicos entre 9600, 19200... 115200)

* parity: seta o modo de paridade:

* 'n' sem paridade (8N1); 'e' paridade ímpar (8E1), 'o' paridade par (8O1).

* tx_en_pin: pino do arduino que controla a transmissão/recepção em uma linha RS485.

* 0 or 1 desliga esta função (para rede RS232)

```

* >2 para uma rede multiponto.
*/
void configure_mb_slave(long baud, char parity, char txenpin);
/*
* update_mb_slave(slave_id, holding_regs_array, number_of_regs)
* verifica se há qualquer pedido válido do mestre modbus. Se houver,
* executa a ação solicitada
* slave: endereço do escravo (arduino) (1 to 127)
* regs: uma matriz com os holding registers. Eles começam no endereço 1 (mestre ponto de
vista)
* Regs_size: número total de holding registers.
* Retorna: 0 se não houver pedido do mestre,
* NO_REPLY (-1) se nenhuma resposta é enviada para o mestre
* Caso um código de exceção (1 a 4) em algumas exceções de modbus
* O número de bytes enviados como resposta (> 4) se OK.
*/
int update_mb_slave(unsigned char slave, int *regs,
unsigned int regs_size);
/* Aqui começa o código do exemplo */
/* Parâmetros Modbus RTU de comunicação, o Mestre e os escravos devem usar os
mesmos parâmetros */
enum {
    COMM_BPS = 9600, /* baud rate */
    MB_SLAVE = 1, /* endereço do escravo modbus */
    PARITY = 'n' /* paridade */
};
/* registros do escravo (holding registers)
* Aqui ficam ordenados todos os registros de leitura e escrita
* da comunicação entre o mestre e o escravo (SCADA e arduino)
*/
enum {
    MB_PINO_3, /* Controle do Led no pino 3 (desliga=0 liga=1) */
    MB_PINO_4, /* Controle do Led no pino 4 (desliga=0 liga=1) */
    MB_PINO_5, /* Controle do Led no pino 5 (desliga=0 liga=1) */
    MB_PINO_6, /* Controle do Led no pino 6 (desliga=0 liga=1) */
    MB_PINO_7, /* Controle do Led no pino 7 (desliga=0 liga=1) */
    MB_PINO_8, /* Controle do Led no pino 8 (desliga=0 liga=1) */
    MB_PINO_9, /* Controle do Led no pino 9 (desliga=0 liga=1) */

```

```

    MB_PINO_10, /* Controle do Led no pino 10 (desliga=0 liga=1) */
    MB_PINO_11, /* Controle do Led no pino 11 (desliga=0 liga=1) */
    MB_PINO_12, /* Controle do Led no pino 12 (desliga=0 liga=1) */
    MB_PINO_13, /* Controle do Led no pino 13 (desliga=0 liga=1) */
    MB_A0,      /* Leitura da entrada analógica 0 (0 a 1023) */
    MB_A1,      /* Leitura da entrada analógica 1 (0 a 1023) */
    MB_A2,      /* Leitura da entrada analógica 2 (0 a 1023) */
    MB_A3,      /* Leitura da entrada analógica 3 (0 a 1023) */
    MB_A4,      /* Leitura da entrada analógica 4 (0 a 1023) */
    MB_A5,      /* Leitura da entrada analógica 5 (0 a 1023) */
    MB_REGS     /* número total de registros do escravo */
};

int regs[MB_REGS];
int ledPin3 = 3;
int ledPin4 = 4;
int ledPin5 = 5;
int ledPin6 = 6;
int ledPin7 = 7;
int ledPin8 = 8;
int ledPin9 = 9;
int ledPin10 = 10;
int ledPin11 = 11;
int ledPin12 = 12;
int ledPin13 = 13;

int potvalor0,potvalor1,potvalor2,potvalor3,potvalor4,potvalor5;
unsigned long wdog = 0; /* watchdog */
unsigned long tprev = 0; /* tempo anterior do último comando*/
unsigned long tanalogprev = 0; /* tempo anterior da leitura dos pinos analogicos*/
void setup()
{
    /* configura comunicação modbus
    * 9600 bps, 8N1, RS485 network */
    configure_mb_slave(COMM_BPS, PARITY, 2);
    pinMode(ledPin3, OUTPUT);
    pinMode(ledPin4, OUTPUT);
    pinMode(ledPin5, OUTPUT);
    pinMode(ledPin6, INPUT);
    pinMode(ledPin7, INPUT);

```

```

pinMode(ledPin8, INPUT);
pinMode(ledPin9, OUTPUT);
pinMode(ledPin10, OUTPUT);
pinMode(ledPin11, OUTPUT);
pinMode(ledPin12, OUTPUT);
pinMode(ledPin13, OUTPUT);
}
void loop()
{
    /* verifica se há solicitações do mestre */
    if(update_mb_slave(MB_SLAVE, regs, MB_REGS))
        wdog = millis();

    if ((millis() - wdog) > 10000) { /* desliga as saidas se não recebe comando a mais de
10 segundos */
        regs[MB_PINO_3] = 0; /* desliga led 3 */
        regs[MB_PINO_4] = 0; /* desliga led 4 */
        regs[MB_PINO_5] = 0; /* desliga led 5 */
        regs[MB_PINO_6] = 0; /* desliga led 6 */
        regs[MB_PINO_7] = 0; /* desliga led 7 */
        regs[MB_PINO_8] = 0; /* desliga led 8 */
        regs[MB_PINO_9] = 0; /* desliga led 9 */
        regs[MB_PINO_10] = 0; /* desliga led 10 */
        regs[MB_PINO_11] = 0; /* desliga led 11 */
        regs[MB_PINO_12] = 0; /* desliga led 12 */
        regs[MB_PINO_13] = 0; /* desliga led 13 */
    }

    if ((millis() - tanalogprev) > 1000) { /* atualiza as entradas analogica a cada 1
segundo */
        regs[MB_A0] = analogRead(0); /* ler entrada analógica 0 */
        regs[MB_A1] = analogRead(1); /* ler entrada analógica 1 */
        regs[MB_A2] = analogRead(2); /* ler entrada analógica 2 */
        regs[MB_A3] = analogRead(3); /* ler entrada analógica 3 */
        regs[MB_A4] = analogRead(4); /* ler entrada analógica 4 */
        regs[MB_A5] = analogRead(5); /* ler entrada analógica 5 */
        tanalogprev = millis();
    }

    /* os valores dos registros são definidos pelo mestre modbus (SCADA) */

```

```

potvalor0 = analogRead(0); //Lê o potenciômetro e armazena na variável potvalor
potvalor1 = analogRead(1);
potvalor2 = analogRead(2);
potvalor3 = analogRead(3);
potvalor4 = analogRead(4);
potvalor5 = analogRead(5);
regs[MB_A0] = analogRead(0);    /* ler entrada analógica 0 */
regs[MB_A1] = analogRead(1);
regs[MB_A2] = analogRead(2);    /* ler entrada analógica 0 */
regs[MB_A3] = analogRead(3);
regs[MB_A4] = analogRead(4);    /* ler entrada analógica 0 */
regs[MB_A5] = analogRead(5);

regs[MB_PINO_7]= digitalRead(6);
regs[MB_PINO_8]= digitalRead(7);
regs[MB_PINO_9]= digitalRead(8);

digitalWrite(9,regs[MB_PINO_10]);
digitalWrite(10,regs[MB_PINO_11]);
digitalWrite(11,regs[MB_PINO_12]);

if(abs(potvalor0-potvalor1)>70||(digitalRead(8)== HIGH)){
    digitalWrite(2,HIGH);
    regs[MB_PINO_3]=HIGH;
}
else{
    digitalWrite(2,LOW);
    regs[MB_PINO_3]=LOW;
}
if(abs(potvalor4-potvalor5)>70){
    digitalWrite(5,HIGH);
    regs[MB_PINO_6]=HIGH;
}
else{
    digitalWrite(5,LOW);
    regs[MB_PINO_6]=LOW;
}
if(potvalor2>950||(digitalRead(6)== HIGH)){

```

```

digitalWrite(3,HIGH);
regs[MB_PINO_4]=HIGH;
}
else{
digitalWrite(3,LOW);
regs[MB_PINO_4]=LOW;
}
if(potvalor2<100||((digitalRead(7)== HIGH)){
digitalWrite(4,HIGH);
regs[MB_PINO_5]=HIGH;
}
else{
digitalWrite(4,LOW);
regs[MB_PINO_5]=LOW;
}
}
/*****
* INICIO DAS FUNÇÕES ESCRAVO Modbus RTU
*****/

/* variaveis globais */
unsigned int Txenpin = 2;    /*Definir o pino usado para colocar o driver
                             RS485 em modo de transmissão, utilizado
                             somente em redes RS485 quando colocar em 0
                             ou 1 para redes RS232 */

/* Lista de códigos de função modbus suportados. Se você implementar um novo, colocar o
seu código de função aqui! */
enum {
    FC_READ_REGS = 0x03, //Read contiguous block of holding register (Ler um bloco
contíguo de registros)
    FC_WRITE_REG = 0x06, //Write single holding register (Escrever em um único
registro)
    FC_WRITE_REGS = 0x10 //Write block of contiguous registers (Escrever em um
bloco contíguo de registros)
};

/* Funções suportadas. Se você implementar um novo, colocar seu código em função nessa
matriz! */
const unsigned char fsupported[] = { FC_READ_REGS, FC_WRITE_REG,
FC_WRITE_REGS };

```

```

/* constantes */
enum {
    MAX_READ_REGS = 0x7D,
    MAX_WRITE_REGS = 0x7B,
    MAX_MESSAGE_LENGTH = 256
};
enum {
    RESPONSE_SIZE = 6,
    EXCEPTION_SIZE = 3,
    CHECKSUM_SIZE = 2
};
/* código de exceções */
enum {
    NO_REPLY = -1,
    EXC_FUNC_CODE = 1,
    EXC_ADDR_RANGE = 2,
    EXC_REGS_QUANT = 3,
    EXC_EXECUTE = 4
};
/* posições dentro da matriz de consulta / resposta */
enum {
    SLAVE = 0,
    FUNC,
    START_H,
    START_L,
    REGS_H,
    REGS_L,
    BYTE_CNT
};
/*

```

CRC

INPUTS:

buf -> Matriz contendo a mensagem a ser enviada para o controlador mestre.

start -> Início do loop no crc do contador, normalmente 0.

cnt -> Quantidade de bytes na mensagem a ser enviada para o controlador mestre

OUTPUTS:

temp -> Retorna byte crc para a mensagem.

COMMENTÁRIOS:

Esta rotina calcula o byte crc alto e baixo de uma mensagem.

Note que este CRC é usado somente para Modbus, não em Modbus PLUS ou TCP.

```
*****/
```

```
unsigned int crc(unsigned char *buf, unsigned char start,  
unsigned char cnt)
```

```
{  
    unsigned char i, j;  
    unsigned temp, temp2, flag;  
    temp = 0xFFFF;  
    for (i = start; i < cnt; i++) {  
        temp = temp ^ buf[i];  
        for (j = 1; j <= 8; j++) {  
            flag = temp & 0x0001;  
            temp = temp >> 1;  
            if (flag)  
                temp = temp ^ 0xA001;  
        }  
    }  
    /* Inverter a ordem dos bytes. */  
    temp2 = temp >> 8;  
    temp = (temp << 8) | temp2;  
    temp &= 0xFFFF;  
    return (temp);  
}
```

```
*****
```

```
*    As seguintes funções constroem o frame de  
*    um pacote de consulta modbus.
```

```
*****/
```

```
/*
```

```
* Início do pacote de uma resposta read_holding_register
```

```
*/
```

```
void build_read_packet(unsigned char slave, unsigned char function,  
unsigned char count, unsigned char *packet)
```

```
{  
    packet[SLAVE] = slave;  
    packet[FUNC] = function;  
    packet[2] = count * 2;  
}
```

```

/*
 * Início do pacote de uma resposta preset_multiple_register
 */
void build_write_packet(unsigned char slave, unsigned char function,
unsigned int start_addr,
unsigned char count,
unsigned char *packet)
{
    packet[SLAVE] = slave;
    packet[FUNC] = function;
    packet[START_H] = start_addr >> 8;
    packet[START_L] = start_addr & 0x00ff;
    packet[REGS_H] = 0x00;
    packet[REGS_L] = count;
}
/*
 * Início do pacote de uma resposta write_single_register
 */
void build_write_single_packet(unsigned char slave, unsigned char function,
    unsigned int write_addr, unsigned int reg_val, unsigned char* packet)
{
    packet[SLAVE] = slave;
    packet[FUNC] = function;
    packet[START_H] = write_addr >> 8;
    packet[START_L] = write_addr & 0x00ff;
    packet[REGS_H] = reg_val >> 8;
    packet[REGS_L] = reg_val & 0x00ff;
}
/*
 * Início do pacote de uma resposta exceção
 */
void build_error_packet(unsigned char slave, unsigned char function,
unsigned char exception, unsigned char *packet)
{
    packet[SLAVE] = slave;
    packet[FUNC] = function + 0x80;
    packet[2] = exception;
}

```

```

/*****
* modbus_query( packet, length)
* Função para adicionar uma soma de verificação para o fim de um pacote.
* Por favor, note que a matriz pacote deve ser de pelo menos 2 campos mais do que
* String_length.
*****/
void modbus_reply(unsigned char *packet, unsigned char string_length)
{
    int temp_crc;
    temp_crc = crc(packet, 0, string_length);
    packet[string_length] = temp_crc >> 8;
    string_length++;
    packet[string_length] = temp_crc & 0x00FF;
}
/*****
* send_reply( query_string, query_length )
* Função para enviar uma resposta a um mestre Modbus.
* Retorna: o número total de caracteres enviados
*****/
int send_reply(unsigned char *query, unsigned char string_length)
{
    unsigned char i;
    if (Txenpin > 1) { // coloca o MAX485 no modo de transmissão
        UCSR0A=UCSR0A |(1 << TXC0);
        digitalWrite( Txenpin, HIGH);
        delayMicroseconds(3640); // aguarda silencio de 3.5 caracteres em 9600bps
    }
    modbus_reply(query, string_length);
    string_length += 2;
    for (i = 0; i < string_length; i++) {
        Serial.write(byte(query[i]));
    }
    if (Txenpin > 1) { // coloca o MAX485 no modo de recepção
        while (!(UCSR0A & (1 << TXC0)));
        digitalWrite( Txenpin, LOW);
    }
    return i;          /* isso não significa que a gravação foi bem sucedida */
}

```

```

/*****
*   receive_request( array_for_data )
* Função para monitorar um pedido do mestre modbus.
* Retorna: Número total de caracteres recebidos se OK
* 0 se não houver nenhum pedido
* Um código de erro negativo em caso de falha
*****/
int receive_request(unsigned char *received_string)
{
    int bytes_received = 0;
    /* FIXME: não Serial.available esperar 1.5T ou 3.5T antes de sair do loop? */
    while (Serial.available()) {
        received_string[bytes_received] = Serial.read();
        bytes_received++;
        if (bytes_received >= MAX_MESSAGE_LENGTH)
            return NO_REPLY; /* erro de porta */
    }
    return (bytes_received);
}

/*****
*   modbus_request(slave_id, request_data_array)
* Função que é retornada quando o pedido está correto
* e a soma de verificação está correto.
* Retorna: string_length se OK
* 0 se não
* Menos de 0 para erros de exceção
* Nota: Todas as funções usadas para enviar ou receber dados via
* Modbus devolver esses valores de retorno.
*****/
int modbus_request(unsigned char slave, unsigned char *data)
{
    int response_length;
    unsigned int crc_calc = 0;
    unsigned int crc_received = 0;
    unsigned char rcv_crc_hi;
    unsigned char rcv_crc_lo;
    response_length = receive_request(data);
    if (response_length > 0) {

```

```

    crc_calc = crc(data, 0, response_length - 2);
    rcv_crc_hi = (unsigned) data[response_length - 2];
    rcv_crc_lo = (unsigned) data[response_length - 1];
    crc_received = data[response_length - 2];
    crc_received = (unsigned) crc_received << 8;
    crc_received =
        crc_received | (unsigned) data[response_length - 1];
    /****** verificar CRC da resposta *****/
    if (crc_calc != crc_received) {
        return NO_REPLY;
    }
    /* verificar a ID do escravo */
    if (slave != data[SLAVE]) {
        return NO_REPLY;
    }
}
return (response_length);
}
/*****
*   validate_request(request_data_array, request_length, available_regs)
* Função para verificar se o pedido pode ser processado pelo escravo.
* Retorna: 0 se OK
* Um código de exceção negativa em caso de erro
*****/
int validate_request(unsigned char *data, unsigned char length,
unsigned int regs_size)
{
    int i, fcnt = 0;
    unsigned int regs_num = 0;
    unsigned int start_addr = 0;
    unsigned char max_regs_num;
    /* verificar o código de função */
    for (i = 0; i < sizeof(fsupported); i++) {
        if (fsupported[i] == data[FUNC]) {
            fcnt = 1;
            break;
        }
    }
}

```

```

if (0 == fcnt)
    return EXC_FUNC_CODE;
if (FC_WRITE_REG == data[FUNC]) {
    /* Para a função de escrever um reg único, este é o registro alvo.*/
    regs_num = ((int) data[START_H] << 8) + (int) data[START_L];
    if (regs_num >= regs_size)
        return EXC_ADDR_RANGE;
    return 0;
}
/* Para as funções de leitura / escrita de registros, este é o intervalo. */
regs_num = ((int) data[REGS_H] << 8) + (int) data[REGS_L];
/* verifica a quantidade de registros */
if (FC_READ_REGS == data[FUNC])
    max_regs_num = MAX_READ_REGS;
else if (FC_WRITE_REGS == data[FUNC])
    max_regs_num = MAX_WRITE_REGS;
if ((regs_num < 1) || (regs_num > max_regs_num))
    return EXC_REGS_QUANT;
/* verificará a quantidade de registros, endereço inicial é 0 */
start_addr = ((int) data[START_H] << 8) + (int) data[START_L];
if ((start_addr + regs_num) > regs_size)
    return EXC_ADDR_RANGE;
return 0;      /* OK, sem exceção */
}
/*****
*   write_regs(first_register, data_array, registers_array)
*   escreve nos registradores do escravo os dados em consulta,
*   A partir de start_addr.
*   Retorna: o número de registros escritos
*****/
int write_regs(unsigned int start_addr, unsigned char *query, int *regs)
{
    int temp;
    unsigned int i;
    for (i = 0; i < query[REGS_L]; i++) {
        /* mudar reg hi_byte para temp */
        temp = (int) query[(BYTE_CNT + 1) + i * 2] << 8;
        /* OR com lo_byte      */

```

```

        temp = temp | (int) query[(BYTE_CNT + 2) + i * 2];
        regs[start_addr + i] = temp;
    }
    return i;
}
}
/*****
*   preset_multiple_registers(slave_id, first_register, number_of_registers,
*   data_array, registers_array)
*   Escreva os dados na matriz dos registos do escravo.
*****/
int preset_multiple_registers(unsigned char slave,
unsigned int start_addr,
unsigned char count,
unsigned char *query,
int *regs)
{
    unsigned char function = FC_WRITE_REGS; /* Escrever em múltiplos registros */
    int status = 0;
    unsigned char packet[RESPONSE_SIZE + CHECKSUM_SIZE];
    build_write_packet(slave, function, start_addr, count, packet);
    if (write_regs(start_addr, query, regs)) {
        status = send_reply(packet, RESPONSE_SIZE);
    }
    return (status);
}
/*****
*   write_single_register(slave_id, write_addr, data_array, registers_array)
*   Escrever um único valor inteiro em um único registo do escravo.
*****/
int write_single_register(unsigned char slave,
    unsigned int write_addr, unsigned char *query, int *regs)
{
    unsigned char function = FC_WRITE_REG; /* Função: Write Single Register */
    int status = 0;
    unsigned int reg_val;
    unsigned char packet[RESPONSE_SIZE + CHECKSUM_SIZE];
    reg_val = query[REGS_H] << 8 | query[REGS_L];
    build_write_single_packet(slave, function, write_addr, reg_val, packet);
}

```

```

regs[write_addr] = (int) reg_val;
/*
written.start_addr=write_addr;
written.num_regs=1;
*/
status = send_reply(packet, RESPONSE_SIZE);
return (status);
}
/*****
*   read_holding_registers(slave_id, first_register, number_of_registers,
* registers_array)
* lê os registros do escravo e envia para o mestre Modbus
*****/
int read_holding_registers(unsigned char slave, unsigned int start_addr,

unsigned char reg_count, int *regs)
{
    unsigned char function = 0x03;      /* Função 03: Read Holding Registers */
    int packet_size = 3;
    int status;
    unsigned int i;
    unsigned char packet[MAX_MESSAGE_LENGTH];
    build_read_packet(slave, function, reg_count, packet);
    for (i = start_addr; i < (start_addr + (unsigned int) reg_count);
        i++) {
        packet[packet_size] = regs[i] >> 8;
        packet_size++;
        packet[packet_size] = regs[i] & 0x00FF;
        packet_size++;
    }
    status = send_reply(packet, packet_size);
    return (status);
}

void configure_mb_slave(long baud, char parity, char txenpin)
{
    Serial.begin(baud);
    switch (parity) {
    case 'e': // 8E1

```

```

        UCSR0C |= ((1<<UPM01) | (1<<UCSZ01) | (1<<UCSZ00));
        // UCSR0C &= ~((1<<UPM00) | (1<<UCSZ02) | (1<<USBS0));
        break;
    case 'o': // 8O1
        UCSR0C |= ((1<<UPM01) | (1<<UPM00) | (1<<UCSZ01) | (1<<UCSZ00));
        // UCSR0C &= ~((1<<UCSZ02) | (1<<USBS0));
        break;
    case 'n': // 8N1
        UCSR0C |= ((1<<UCSZ01) | (1<<UCSZ00));
        // UCSR0C &= ~((1<<UPM01) | (1<<UPM00) | (1<<UCSZ02) | (1<<USBS0));
        break;
    default:
        break;
}

if (txenpin > 1) { // pino 0 & pino 1 são reservados para RX/TX
    Txenpin = txenpin; /* definir variável global */
    pinMode(Txenpin, OUTPUT);
    digitalWrite(Txenpin, LOW);
}

return;
}

/*
 * update_mb_slave(slave_id, holding_regs_array, number_of_regs)
 *
 * verifica se há qualquer pedido válido do mestre modbus. Se houver,
 * executa a ação solicitada
 */

unsigned long Nowdt = 0;
unsigned int lastBytesReceived;
const unsigned long T35 = 5;
int update_mb_slave(unsigned char slave, int *regs,
unsigned int regs_size)
{
    unsigned char query[MAX_MESSAGE_LENGTH];
    unsigned char errpacket[EXCEPTION_SIZE + CHECKSUM_SIZE];
    unsigned int start_addr;
    int exception;

```

```

int length = Serial.available();
unsigned long now = millis();
if (length == 0) {
    lastBytesReceived = 0;
    return 0;
}
if (lastBytesReceived != length) {
    lastBytesReceived = length;
    Nowdt = now + T35;
    return 0;
}
if (now < Nowdt)
    return 0;
lastBytesReceived = 0;
length = modbus_request(slave, query);
if (length < 1)
    return length;
exception = validate_request(query, length, regs_size);
if (exception) {
    build_error_packet(slave, query[FUNC], exception,
        errpacket);
    send_reply(errpacket, EXCEPTION_SIZE);
    return (exception);
}
start_addr = ((int) query[START_H] << 8) +
    (int) query[START_L];
switch (query[FUNC]) {
    case FC_READ_REGS:
        return read_holding_registers(slave,
            start_addr,
            query[REGS_L],
            regs);
        break;
    case FC_WRITE_REGS:
        return preset_multiple_registers(slave,
            start_addr,
            query[REGS_L],
            query,

```

```
        regs);
break;
case FC_WRITE_REG:
    write_single_register(slave,
        start_addr,
        query,
        regs);
break;
    }
}
```