



CENTRO UNIVERSITÁRIO CARIOCA – UNICARIOCA

LEANDRO NÉLIO PEIXOTO MAIA

**SISTEMA DE AUTOMAÇÃO PREDIAL:
UMA PESQUISA APLICADA**

Rio de Janeiro
2020

LEANDRO NÉLIO PEIXOTO MAIA

SISTEMA DE AUTOMAÇÃO PREDIAL: UMA PESQUISA APLICADA

Trabalho de conclusão de curso apresentado ao centro Universitário Unicarioca, como requisito parcial à obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Professor: Neury Nunes Cardoso

Rio de Janeiro
2020

M217

Maia, Leandro Nélio Peixoto

Sistema de automação predial: uma pesquisa aplicada/ Leandro Nélio Peixoto Maia. – 2020.

118 f ; il. ; 30 cm

Trabalho de Conclusão de Curso (Graduação em
Ciência da Computação) – Centro Universitário
Carioca, Rio de Janeiro, 2020.

1. Automação predial. I. Título.

LEANDRO NÉLIO PEIXOTO MAIA

SISTEMA DE AUTOMAÇÃO PREDIAL: UMA PESQUISA APLICADA

Trabalho de Conclusão de Curso
apresentado ao Centro Universitário
Carioca, como requisito parcial à
obtenção do grau de Bacharel em
Ciência da Computação.

Aprovado em 2020

Banca Examinadora

Prof. Neury Nunes cardoso – Orientador
Centro Universitário Carioca

Prof. Marcelo Perantoni
Centro Universitário Carioca

Prof. André Sobral - coordenação
Centro Universitário Carioca

Agradecimentos

Agradeço a Deus por me conceder tudo aquilo que eu desejava e além do que eu poderia merecer.

Agradeço de pé e com muitas palmas, as duas Marias da minha vida e que jamais serão substituídas. Agradeço a você “Maria Clara – Minha Esposa” O grande amor da minha vida, por nunca desistir de mim, me dando força quando não tínhamos para seguir em frente com os obstáculos da vida e por todos os conselhos. Sei que está orgulhosa de mim. Mãe, “Maria de Fátima”, que com sua imensa gratidão a tudo aquilo que foi concebido por Deus, me ensinou que ser humilde e honesto é o caminho mais correto para o crescimento pessoal e profissional. Amo vocês.

Aos meus irmãos, tios e sobrinhos, gratidão por tudo que fizeram por mim. Amo vocês.

A família que eu escolhi e também me escolheu. “Sol e Ade - meus pais do coração” e “Cristiane, minha irmã linda” sou grato por todos os ensinamentos e puxões de orelha. Amo vocês.

Em memória, “José – meu pai”, mesmo de longe, acredito que esteja orgulhoso de mim. Amo você.

Ao meu orientador “Neury Cardoso”, pois, mesmo com as demandas do dia a dia, não deixou faltar apoio.

A Universidade Unicarioca e seu corpo docente, obrigado por todo o ensinamento e por contribuir para minha formação profissional.

RESUMO

Esta pesquisa aplicada tem por objetivo principal observar a maneira a qual a Automação Predial pode trazer benefícios perante os processos tradicionais, demonstrando como esta inovação pode acarretar eficiência no consumo energético, redução de mão de obra e elaboração de melhores planos de manutenção preventiva e preditiva.

Para um melhor entendimento, será apresentado um estudo de caso acerca de um edifício inteligente que possui automação predial em diferentes processos, sendo esses administrados através de um Sistema de Gestão Predial, conhecido como BMS (*Building Management System*).

A importância deste estudo é demonstrar o quanto é vantajoso o uso da automação na administração das instalações prediais, em termos de redução de custos, e também os benefícios oferecidos pelos usuários, que têm à disposição um edifício inteligente, inovador e mais sustentável, capaz de antecipar e adaptar cada ambiente às necessidades dos utilizadores. Tal característica permite uma conectividade, mobilidade e controle ainda maior dos edifícios, já que praticamente tudo pode se conectar a rede.

Palavras-Chaves: Automação Predial, Edifícios Inteligentes, BMS (*Building Management System*).

ABSTRACTS

This applied research has as main objective to observe a way in which Building Automation can bring benefits in the face of traditional processes, demonstrating how this innovation can bring efficiency in energy consumption, reduction of manpower and elaboration of the best preventive and predictive maintenance plans.

For a better understanding, a case study will be presented about a building that has building automation in different processes, these being administered through a Building Management System, known as BMS (Building Management System).

The importance of the study is demonstration of how advantageous the use of automation in the administration of building installations is, in terms of cost reduction, and also the benefits offered by users, who have at their disposal an intelligent, innovative and more sustainable building, capable of anticipating and adapt each environment to the needs of users. This feature allows for even greater connectivity, mobility and control of buildings, since virtually everything can connect to the network.

Keywords: Building Automation, Intelligent Buildings, BMS (Building Management System).

LISTA DE SIGLAS

ABNT - Associação Brasileira de Normas Técnicas
ASCII - American Standard Code for Information Interchange
BMS - Building Management System
CFTV - Circuito Fechado de Televisão
DALI - Digital Addressable Lighting Interface
IoT - Internet of Things
IP - Internet Protocol
JSON - JavaScript Object Notation
NA - Normalmente Aberto
NBR - Norma Brasileira
NF - Normalmente Fechado
OPC A&E - Open Platform Communications - alarms and events
OPC DA - Open Platform Communications - Data Access
OPC UA - Open Platform Communications - Unified Architecture
OPC - Open Platform Communications
OSI - Open Systems Interconnection
PC - Personal computer
PLC - Programmable Logic Controller
RFID - Radio-Frequency IDentification
RTU - Remote Terminal Unit
SMS - Short Message Service
TCP - Transmission Control Protocol
UPS - Uninterruptible Power Supply
UTA - unidade de tratamento de ar

LISTA DE FIGURAS

Figura 1 Camadas de rede.....	16
Figura 2 Pirâmide de automação	20
Figura 3 Sistema de gerenciamento de energia do “Edifício Inteligente” – Análise de consumo de energia.....	21
Figura 4 Sistema de gerenciamento de energia do “Edifício Inteligente” – Medições	26
Figura 5 Sistema de gerenciamento de energia do “Edifício Inteligente” – Notificações.....	27
Figura 6 Sistema de gerenciamento de energia do “Edifício Inteligente” – Previsão e consumo.....	27
Figura 7 Central de água gelada	29
Figura 8 Condensação	29
Figura 9 Fancoil com controle de insuflamento e temperatura.....	30
Figura 10 Fancoil com controle de insuflamento, temperatura e umidade	30
Figura 11 Ventilador/Exaustor controle de vazão dos ambientes.....	30
Figura 12 UTA unidade de tratamento de ar	31
Figura 13 Programação fixa no tempo do edifício inteligente.....	32
Figura 14 Programação variável no tempo do edifício inteligente	33
Figura 15 Condições variáveis de luz do edifício inteligente	33
Figura 16 Motobomba de água pluvial e extravaso.....	35
Figura 17 Motobomba de água potável.....	35
Figura 18 Motobomba de drenagem, Motobombo de esgoto.....	36
Figura 19 Motobomba de incêndio	36

SUMÁRIO

1	INTRODUÇÃO	10
2	AUTOMAÇÃO E SUA HISTÓRIA	12
2.1	Revolução Industrial 1.0	13
2.2	Revolução Industrial 2.0	14
2.3	Revolução Industrial 3.0	14
2.4	Revolução Industrial 4.0	15
3	REDES DE AUTOMAÇÃO	16
3.1	Protocolos	17
3.1.1	MODBUS	17
3.1.2	OPC DA	17
3.1.3	OPC UA	17
3.1.4	TCP/IP	18
3.1.5	DALI	18
3.1.6	WIEGAND	18
4	BMS – <i>BUILDING MANAGEMENT SYSTEM</i> (SISTEMA DE GESTÃO PREDIAL)	19
4.1	Economia de recursos	20
4.2	Sustentabilidade e redução do impacto ambiental	21
4.3	Redução de custos operacionais e de manutenção	21
4.4	Segurança reforçada	22
4.5	Controle centralizado e acesso remoto	23
4.6	Sistema adaptável a necessidades específicas	23
4.7	Diferenciais e aplicabilidade	23
5	ESTUDO DE CASO – “EDIFÍCIO INTELIGENTE”	25
5.1	Subsistema de gerenciamento de energia	25
5.2	Subsistema de climatização	27

5.3	Subsistema de controle de iluminação	31
5.4	Subsistema de motobombas.....	34
5.5	Subsistema de detecção e alarme de incêndio	36
5.6	Subsistema de controle de acesso.....	37
5.7	Monitoramento das instalações de utilidades	38
6	CONCLUSÃO DA PESQUISA APLICADA	40
	REFERÊNCIAS.....	41
	ANEXO I.....	43
	ANEXO II.....	115

1 INTRODUÇÃO

Esta pesquisa aplicada tem por objetivo principal observar a maneira a qual a Automação Predial pode trazer benefícios perante os processos tradicionais, demonstrando como esta inovação pode acarretar eficiência no consumo energético, redução de mão de obra e elaboração de melhores planos de manutenção preventiva e preditiva. Foram feitas análises bibliográficas em trabalhos acadêmicos, artigos científicos e livro acerca do tema.

Na primeira parte será feita a definição e a revisão da literatura a respeito do assunto. Na segunda parte, será apresentado um estudo de caso acerca de um edifício que possui automação predial em diferentes processos, sendo esses administrados através de um Sistema de Gestão Predial, conhecido como BMS (*Building Management System*).

BMS é um sistema computadorizado implantado em edifícios, que serve para monitorar e controlar a parte mecânica e elétrica do prédio. Os serviços podem incluir iluminação, climatização, controle de acesso, detecção e alarmes contra incêndio, detecção e alarmes de intrusão, circuito fechado de TV (CFTV) e elevadores, entre outros.

A importância deste estudo é demonstrar o quanto é vantajoso o uso da automação na administração das instalações prediais, em termos de redução de custos, e também os benefícios oferecidos pelos usuários, que têm à disposição um edifício inteligente, inovador e mais sustentável, capaz de antecipar e adaptar cada ambiente às necessidades dos utilizadores. Tal característica permite uma conectividade, mobilidade e controle ainda maior dos edifícios, já que praticamente tudo pode se conectar a rede, fornecer informações e ser controlado remotamente.

A integração das áreas mecânica, eletrônica e ciência da informação dá origem a Mecatrônica, que trata de dispositivos que unem a eletrônica e a mecânica, a fim de realizar tarefas de modo inteligente e automático, tal ocorre na Automação Predial.

Ao longo dos anos, a tecnologia tem evoluído drasticamente, proporcionando novas formas de trabalho, mais eficiência, e fazendo cada vez mais parte de nossas vidas. A automação surge pela necessidade constante de adoção de novas inteligências, visando diminuição de custos e maior eficácia. Assim, visando melhorias, diversos

caminhos de desenvolvimento são oferecidos pelo mercado, sendo a automação predial uma alternativa eficaz para novas e antigas construções.

O controle automatizado de uma instalação é o objetivo de qualquer sistema de automação moderna. Automatizar tende a trazer maiores taxas de produtividade e facilidades, gerando previsões para possíveis problemas com antecedência. Segue breve descrição do objetivo da automação predial:

A automação predial abrange a integração e coordenação de todas as instalações e utilidades do edifício. Seu objetivo é a composição do “sistema de gerenciamento predial”. Com esse sistema o prédio otimiza seu funcionamento e reduz seu consumo energético. Trata-se do ponto de apoio ou ancoragem principal, embora se entenda que, sem os outros dois (redes de TI e telecomunicações), não teria razão de ser. É considerado o mais importante por possuir um ciclo de vida superior aos outros sistemas e envolver um investimento de maior vulto, uma vez que inclui o controle das instalações de climatização do edifício. Enquanto uma rede de informática possui um ciclo de vida em torno de dez anos, um sistema de automação poderá exceder o período de vinte. Em função do rápido desenvolvimento dessas tecnologias, os ciclos tendem a encurtar sua vigência. (DE LA CRUZ, 2019, p. 6)

De acordo com DE LA CRUZ (2019, p.15) “A automação predial é o conjunto de recursos tecnológicos, por excelência, requerido para compor o sistema de gerenciamento predial em determinado edifício”.

2 AUTOMAÇÃO E SUA HISTÓRIA

Segundo Victor Mendonça (2013, p. 20) “As primeiras iniciativas do homem para mecanizar atividades manuais ocorreram na pré-história com invenções como a roda. O moinho movido por vento ou força animal e as rodas d'água demonstram a criatividade do homem para poupar esforço”. Porém, a automação só ganhou destaque na sociedade quando o sistema de produção agrário e artesanal se transformou em industrial, a partir da segunda metade do século XVIII, inicialmente na Inglaterra.

Os sistemas inteiramente automáticos surgiram no início do século XX, entretanto, bem antes disso foram inventados dispositivos simples e semi-automáticos. Devido a necessidade de aumentar a produção e a produtividade, surgiram uma série de inovações tecnológicas:

- Máquinas modernas, capazes de produzir com maior precisão e rapidez em relação ao trabalho feito à mão;
- Utilização de fontes alternativas de energia, como o vapor, inicialmente aplicada a máquinas em substituição às energias hidráulica e muscular.

O desenvolvimento da automação iniciou por volta de 1788, quando James Watt desenvolveu um mecanismo de regulação do fluxo do vapor em locomotivas. A partir de 1870, a energia elétrica passou a ser utilizada e a estimular indústrias como a do aço, a química e a de máquinas-ferramenta e o setor de transportes progrediu bastante graças à expansão das estradas de ferro e à indústria naval. No século XX, a tecnologia da automação passou a contar com computadores, servomecanismos e controladores programáveis. Os computadores são o alicerce da tecnologia da automação contemporânea.

No Brasil, a automação predial é utilizada desde a década de 1980:

Na segunda metade da década de 80 aparecem as primeiras amostras de sistemas avançados de supervisão e automação predial no Brasil. Inicialmente, a automação estava baseada em uma Unidade Central, isto é, um microcomputador de padrão proprietário de determinado fornecedor. Essa unidade gerava as instruções para as Unidades Remotas do prédio e a ela chegava toda alteração e alarme. As unidades interligadas com a central possuíam escassos recursos de processamento ou limitadas funções locais. Ainda assim, tal tecnologia era considerada avançada. (DE LA CRUZ, 2019, p. 30)

Inicialmente, os controles eram desenvolvidos de forma individual, sendo um equipamento para cada unidade — assim, havia um acesso direto para o sistema de refrigeração e outro para os elevadores, por exemplo. Não havia, ainda, a possibilidade geral de integração oferecida pelos avanços da Internet das Coisas na automação predial.

A partir dos anos 90, com a tecnologia já um pouco mais evoluída, a junção entre as diferentes estruturas começou a ser trabalhada, unificando os controles em estações centrais.

Na trilha dos anos 90, especialmente no Brasil — após a abertura do mercado da informática —, a complexidade dos projetos de maior porte tornou-se evidente. Recebemos, em curto espaço de tempo, os sistemas tecnologicamente mais avançados; os mesmos oferecidos no âmbito mundial pelas indústrias que lideravam esse mercado. (...) O edifício corporativo se diferenciava por centralizar a operação e o controle das instalações de ar-condicionado, aumentar a capacidade de gerenciamento do prédio e automatizar todas as utilidades prediais. A partir desse momento a área de automação predial iniciou sua época de maior destaque até o presente momento. Além dos edifícios corporativos de usuário único, aparecem os edifícios institucionais e os de grandes lajes corporativas em regime de condomínio, modalidade comum a partir dessa época. (DE LA CRUZ, 2019, p. 31)

2.1 Revolução Industrial 1.0

Algumas divergências aparecem ao tentarem definir uma data exata a um marco importante historicamente como a primeira Revolução Industrial. Um dos itens a serem levado em conta para definir esse marco foi a presença da indústria e ainda dos processos automáticos autocontroláveis. Foi registrado como início da Automação Industrial o século XVIII, início da época das máquinas a Vapor Inglesas, obtendo então o resultado especulado de aumento de produtividade de produtos manufaturados, décadas da Revolução Industrial (LIMA, 2017).

Essa revolução foi determinante para a dinamização do transporte de matérias-primas, pessoas e distribuição de mercadorias, dando um novo panorama aos meios de se locomover e produzir.

A partir da década de 1930 que o Brasil começou a mudar seu modelo, incentivando o desenvolvimento do setor industrial no país. Os efeitos econômicos e sociais foram bem vantajosos ao Brasil, embora tardios. Os resultados foram a diminuição da

dependência da importação de produtos manufaturados; diminuição de custos de produção; geração de empregos na indústria; avanço na área de transporte.

Em contra partida o aumento da taxa de poluição foi enorme junto com a o uso de mão de obra infantil.

2.2 Revolução Industrial 2.0

A Segunda Revolução Industrial iniciou-se na segunda metade do século XIX (c. 1850 - 1870), e terminou durante a Segunda Guerra Mundial (1939 - 1945), envolvendo uma série de desenvolvimentos dentro da indústria química, elétrica, de petróleo e de aço. Outros progressos essenciais nesse período incluem a introdução de navios de aço movidos a vapor, o desenvolvimento do avião, a produção em massa de bens de consumo, o enlatamento de comidas, refrigeração mecânica e outras técnicas de preservação e a invenção do telefone eletromagnético.

Esse período marca também o advento da Alemanha e dos Estados Unidos como potências industriais, juntando-se à França e ao Reino Unido.

2.3 Revolução Industrial 3.0

Para alguns estudiosos, a terceira Revolução Industrial teve início nos Estados Unidos e em alguns países europeus, quando a ciência descobriu a possibilidade de utilizar a energia nuclear, do átomo. Para outros, seu início foi por volta de 1970, com o descobrimento da robótica, empregada na linha de montagem de automóveis. Para outro grupo, iniciou-se a partir dos anos 1990, com o uso do computador pessoal e a internet. A 3ª Revolução Industrial – também chamada de Revolução Técnico-Científica-Informacional – correspondeu ao processo de inovações no campo da informática e suas aplicações nos campos da produção e do consumo. As grandes realizações desse período são o desenvolvimento da chamada química fina, a biotecnologia, a escalada espacial, a robótica, a genética, entre outros importantes avanços.

2.4 Revolução Industrial 4.0

Atualmente, estamos passando pela fase de transição para a 4ª Revolução Industrial, que se baseia na incorporação de novas tecnologias e nas frequentes inovações que transformam a rotina de empresa e pessoas. A chamada Revolução 4.0 acontece após três processos históricos transformadores. A quarta mudança traz consigo uma tendência à automatização total das edificações. A partir do próximo tópico, daremos início ao processo de estudo de alguns pilares da Revolução Industrial 4.0.

3.1 Protocolos

Os protocolos são padrões que determinam como dois dispositivos vão se comunicar. Existem várias tarefas que um protocolo deve tratar, entre elas:

- Detecção e correção de erros;
- Roteamento de mensagens;
- Criptografia e segurança;
- Níveis de sinal entre dispositivos consistentes;
- Endereçamento de rede.

Existem inúmeros protocolos, dentre os quais iremos citar a seguir dois que foram utilizados no estudo de caso do “Edifício Inteligente”: MODBUS, OPC e DALI.

3.1.1 MODBUS

É um Protocolo de dados utilizado em sistemas de automação. Criado originalmente no final da década de 1970 pela fabricante Modicon, é um dos mais antigos protocolos de Controladores Lógicos Programáveis (PLC) para envio e recebimento de dados (0 ou 1) de instrumentos e atuadores. A rede modbus é dividida em três modos de transmissão: modbu/TCP, RTU e ASCII.

3.1.2 OPC DA

O OPC DA é uma sigla que representa o OPC *Data Access*. É um padrão de comunicação Cliente-Servidor que permite a aquisição de dados, em tempo real. Esses dados podem ser lidos ou escritos.

3.1.3 OPC UA

O OPC UA é uma sigla que significa plataforma de comunicação aberta de arquitetura unificada, do inglês *Open Platform Communications Unified Architecture*.

Pode ser entendido como um padrão de comunicação com foco em lidar com dados diversos em uma arquitetura baseada no modelo de cliente-servidor.

3.1.4 TCP/IP

O TCP/IP é um conjunto de protocolos de comunicação entre computadores em rede. Seu nome vem de dois protocolos: o TCP (*Transmission Control Protocol* - Protocolo de Controle de Transmissão) e o IP (*Internet Protocol*-Protocolo de Internet, ou ainda, protocolo de interconexão). O conjunto de protocolos pode ser visto como um modelo de camadas (Modelo OSI), onde cada camada é responsável por um grupo de tarefas, fornecendo um conjunto de serviços bem definidos para o protocolo da camada superior. As camadas mais altas estão logicamente mais perto do usuário (chamada camada de aplicação) e lidam com dados mais abstratos, confiando em protocolos de camadas mais baixas para tarefas de menor nível de abstração.

3.1.5 DALI

O DALI (*Digital Addressable Lighting Interface*) é um protocolo padrão, desenvolvido pensando em melhorar o controle da iluminação em aplicações de controle de ambientes, com a possibilidade de se conectar com sistemas de gerenciamento predial. De acordo com DE LA CRUZ (2019, p. 502), seguem as características do respectivo protocolo: “serial assíncrono, *half*-duplex, codificação Manchester em barramento diferencial de dois fios. Com protocolo de acesso ordenado por *polling*, comunicando unidades mestres e escravas”.

3.1.6 WIEGAND

O protocolo Wiegand possuem 24 bits: oito bits é código de acesso e 16 bits de código de identificação prefixados e pós- fixados com os bits de paridade.

4 BMS – *BUILDING MANAGEMENT SYSTEM* (SISTEMA DE GESTÃO PREDIAL)

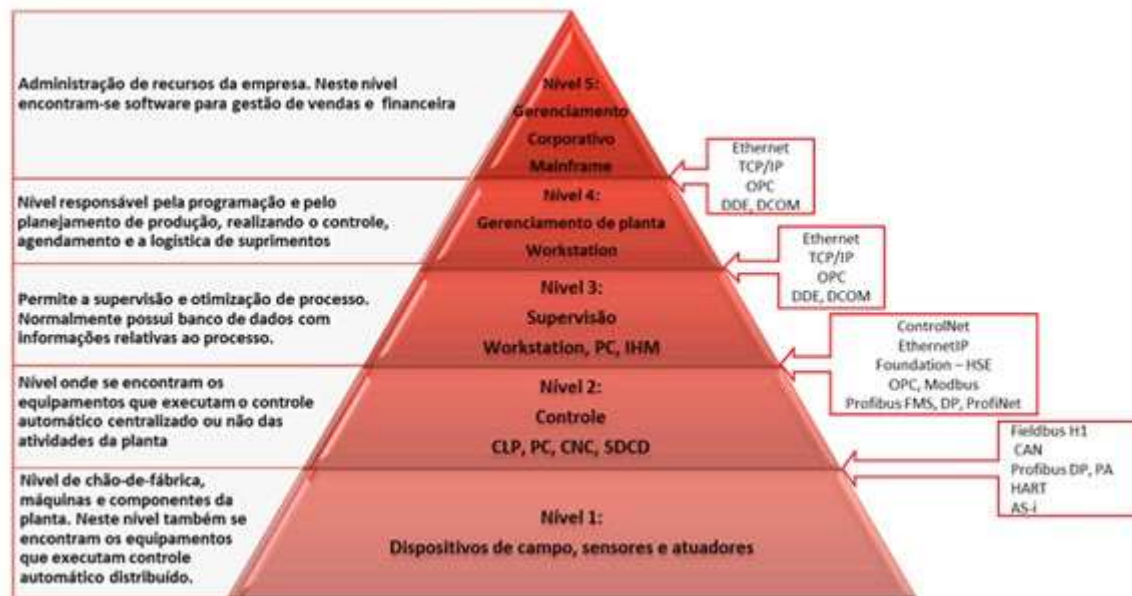
A sigla BMS vem do inglês *Building Management System*, que, em português, pode ser traduzido como Sistema de Gestão Predial. Em uma explicação bem simples, BMS é um sistema computadorizado implantado em edifícios, que serve para monitorar e controlar toda a parte mecânica e elétrica do prédio. Os serviços podem incluir iluminação, aquecimento, ventilação, ar-condicionado, controle de acesso físico, detecção e alarmes contra incêndio e elevadores, entre outros.

O BMS inclui um software, um banco de dados em um servidor e um sistema de sensores inteligentes conectados a uma rede de transmissão de dados. Os sensores espalhados pelo edifício recolhem informações que são enviadas ao BMS, onde são armazenadas em um banco de dados. Se um sensor colhe informações que não combinem com condições pré-definidas, o BMS dispara um alarme. Em um Data Center, por exemplo, um alarme será acionado quando a temperatura do rack de rede atingir um limite inaceitável para a segurança.

Dependendo do sistema, o BMS pode ser instalado como um aplicativo isolado ou então ser integrado a outros programas de monitoramento. Nos casos mais avançados, o BMS pode monitorar e gerenciar uma enorme variedade de serviços no prédio, por meio de múltiplas plataformas e protocolos, propiciando ao administrador do edifício uma visão eficiente de todas as operações em andamento. Pode-se dizer que o *Building Management System* é o futuro da administração predial: as vantagens oferecidas pelo sistema são de larga escala, o que o torna um investimento altamente benéfico e estratégico.

Um sistema BMS traz benefícios consideráveis, incluindo a limitação de impactos ambientais, economia de energia e aumento da segurança local, além de possibilitar um corte significativo de despesas. O processo de automação será demonstrado em seus diferentes níveis hierárquicos na. figura 2

Figura 2 Pirâmide de automação



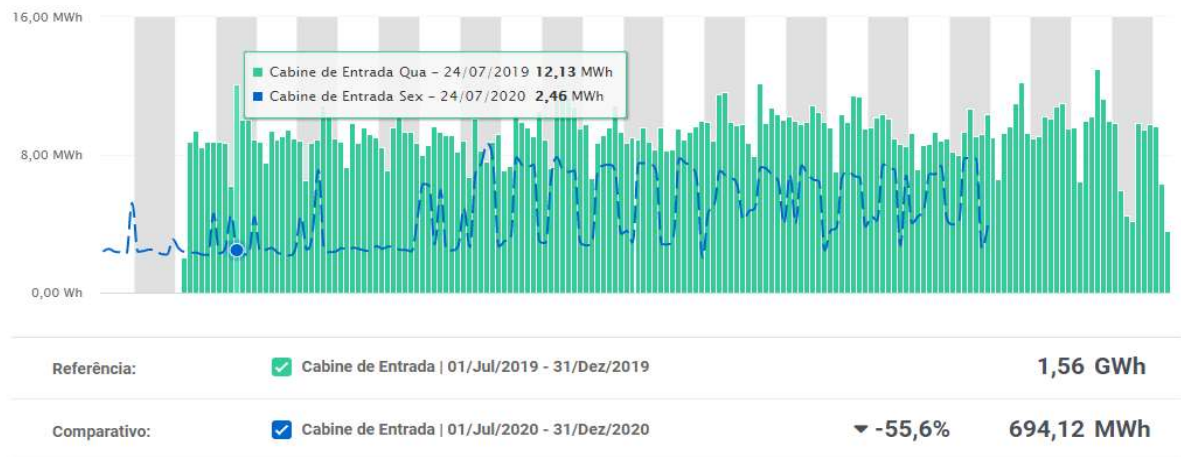
Fonte: <https://www.automacaoindustrial.info/a-piramide-da-automacao-industrial/>

4.1 Economia de recursos

Com a implantação do BMS no edifício inteligente observamos a redução de aproximadamente de 55% de consumo de energia conforme demonstrado na figura 3. A eficiência energética procedente desta implantação é uma das principais vantagens alcançadas, uma vez que o sistema permite:

- analisar e monitorar o consumo de eletricidade;
- controlar a iluminação, ligando e desligando luzes automaticamente, de acordo com a presença de pessoas no ambiente monitorado;
- ligar e desligar o ar-condicionado automaticamente;
- ajustar a temperatura do ar-condicionado à real necessidade do ambiente;
- criar medidas de economia para áreas desocupadas do edifício.

Figura 3 Sistema de gerenciamento de energia do “Edifício Inteligente” – Análise de consumo de energia



Fonte: "https://dashboard.greenant.com.br/#!/analysis"

Além da eficiência energética, a automação predial proporciona economia de recursos hídricos, reduz os custos operacionais e diminui a incidência de falhas humanas e de equipamentos.

4.2 Sustentabilidade e redução do impacto ambiental

Pela redução do consumo de energia em um edifício, o *Building Management System* (BMS) também reduz a emissão de gases nocivos à natureza, o que é fundamental para a preservação do meio ambiente. Um BMS pode operar de modo integrado ao sistema hidráulico do prédio, reduzindo o uso de água. Eliminando desperdícios, esse sistema favorece o uso eficiente de recursos e reduzem o impacto ambiental.

Ele também permite o levantamento de dados por terceiros, como agências governamentais, para a validação do consumo de energia no edifício, garantindo o cumprimento das leis ambientais.

4.3 Redução de custos operacionais e de manutenção

Os diagnósticos avançados disponibilizados por um BMS auxiliarão a criar medidas visando ao uso adequado de equipamentos e programar manutenções em tempo hábil. O sistema permite o monitoramento remoto de equipamentos, bem como a

integração de sistemas de câmeras e o controle de acesso ao local, por exemplo. O usuário pode definir alarmes para alertar sobre anormalidades, falhas em equipamentos e manutenções programadas, além de outros eventos.

O BMS recolhe informações sobre todos os setores do edifício e reporta os resultados ao sistema do computador. Analisando essas informações, o administrador pode identificar e diagnosticar problemas operacionais remotamente, sem a necessidade de contratar um técnico para localizar a falha. Identificando problemas operacionais com antecedência, é possível reduzir custos com manutenção e prevenir interrupções no funcionamento das operações.

4.4 Segurança reforçada

Edifícios apresentam diferentes necessidades no quesito segurança, e um sistema BMS oferece várias soluções possíveis. Veja alguns exemplos:

- Circuito fechado de televisão (CFTV) composto por câmeras de vídeo para o monitoramento do fluxo de pessoas e fiscalização de equipamentos, para garantir a segurança pessoal e a proteção do patrimônio. O sistema de câmeras pode ser integrado ao controle de acesso otimizando a segurança do edifício.
- Controle de acesso em diferentes áreas, como estacionamentos, entrada de prédios comerciais, Data Centers, salas de reuniões etc. O monitoramento do fluxo de pessoas no local pode ser feito por vários métodos, como biometria facial, ocular ou digital, ou então por cartão de identificação ou RFID (identificação por radiofrequência). Os projetos de segurança podem contar com catracas, fechaduras eletromagnéticas, sensores de portas, cancelas, portas automatizadas, leitoras por biometria ou por cartão e outros equipamentos.
- Controle de iluminação e de trancamento de portas de modo programado: se uma pessoa precisar permanecer em uma sala depois do horário determinado, um cartão permitirá o acesso e o acionamento das luzes somente naquele ambiente.

4.5 Controle centralizado e acesso remoto

É possível acessar informações pela internet, por meio de um navegador seguro e descomplicado, ou então ter o acesso limitado a uma rede interna ou a uma única estação de trabalho, de acordo com a conveniência.

No caso de acesso pela internet, o responsável pelo edifício não precisa estar no local para acessar as informações necessárias, o que facilita o trabalho do administrador. Em situações em que haja um conjunto de dois ou mais edifícios a serem monitorados, é possível integrar os sistemas de todos eles em um só computador, proporcionando maior comodidade e possibilitando a redução do quadro de empregados.

4.6 Sistema adaptável a necessidades específicas

Ainda entre as vantagens proporcionadas por um sistema de automação predial, citamos o fato de que um *Building Management System* (BMS) pode ser adaptado para atender necessidades específicas de cada edifício. É possível escolher os recursos do sistema que realmente atendam ao que cada gestor precisa para possibilitar uma administração eficiente e aumentar os níveis de conforto e de segurança para os ocupantes do prédio.

4.7 Diferenciais e aplicabilidade

O *Building Management System* atua como um grande diferencial na hora de aprimorar a administração de edificações sejam elas, residenciais ou comerciais. Para se ter uma ideia, esse conceito pode ser aplicado na gestão predial da indústria farmacêutica.

Ao monitorar e controlar condições ambientais, como temperatura e pureza do ar, por exemplo, há um ganho significativo na qualidade dos processos executados nesse ambiente, além da produtividade.

De modo geral, em qualquer tipo de aplicação — a depender do sistema — o BMS pode ser instalado como um aplicativo isolado ou então ser integrado a outros programas de monitoramento.

Nos casos mais avançados, o BMS pode monitorar e gerenciar uma enorme variedade de serviços na edificação, por meio de múltiplas plataformas e protocolos, propiciando ao administrador do ambiente uma visão ampliada e mais eficiente de todas as operações em andamento, facilitando a tomada de decisões.

5 ESTUDO DE CASO – “EDIFÍCIO INTELIGENTE”

Após trazermos alguns pontos teóricos relacionados ao BMS, apresentaremos o seu funcionamento na prática através de um estudo de caso, do “**Edifício Inteligente**”. Os Subsistemas de gerenciamento de energia, de automação do ar-condicionado, de gerenciamento de sistemas críticos, de controle de iluminação, de motobombas e reservatórios, de detecção e alarme de incêndio e de controle de acesso contemplam o BMS, que iremos detalhar a seguir.

Esse título de “edifício inteligente”, em grande parte, se deve a um conceito que vem se tornando cada vez mais comum: a *Internet of Things* (internet das coisas) - IoT. É ela que tem sido a responsável pela formulação de grande parte dos equipamentos tecnológicos empregados nas instalações e tem fortalecido a ideia de integração ampla entre todos os objetos e dispositivos presentes em um ambiente.

5.1 Subsistema de gerenciamento de energia

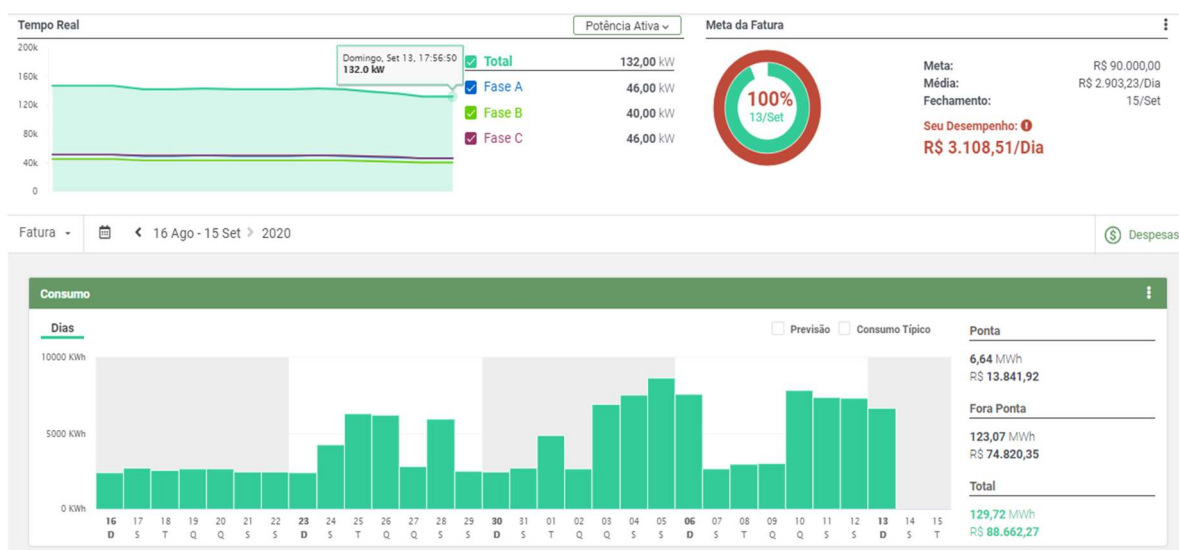
O objetivo deste subsistema é controlar com alta eficiência todas as demandas de consumo energético. A estratégia deste gerenciamento destina-se ao controle, administração e redução de custos com energia elétrica. A tarefa de controlar a demanda de energia elétrica é executada através de programas baseados em algoritmos de diferentes tipos. No sistema de gerenciamento de energia do “Edifício Inteligente” é utilizado o algoritmo preditivo adaptativo, que trabalha de forma sincronizada com o medidor da concessionária, dentro do tempo de intervalo de 15 minutos. É um método complexo e de ótimo resultado, tendo como características:

- Monitoramento de demanda em tempo real - através desse monitoramento, é possível acompanhar se a demanda utilizada está compatível com a demanda contratada;
- Monitoramento em tempo real de consumo por fase - através desse monitoramento, é possível verificar qual a fase que possui maior demanda, permitindo ajustes de balanceamento de carga;
- Monitoramento em tempo real do consumo monetário - através desse monitoramento, é possível definir metas de consumo diário e limitar as despesas.

- Relacionaremos a seguir os equipamentos que compõem o Subsistema de gerenciamento de energia do edifício inteligente e suas ações:
- Medidor de energia – equipamento eletromecânico capaz de mensurar o consumo de energia elétrica;
- Relé sepam – proteção digital para tensão ou corrente;
- Medidor de qualidade – analisador da qualidade da potência entregue pela concessionária.

O gerenciamento de energia do edifício inteligente conta com auxílio de um gerador para eventuais falhas de fornecimento. Esse gerador tem o papel de suprir o abastecimento de forma ininterrupta, para que eventuais falhas não afetem áreas críticas ou serviços essenciais, tais como sala dos servidores, elevadores, iluminação de emergência, controle de acesso e salas de rack dados.

Figura 4 Sistema de gerenciamento de energia do “Edifício Inteligente” – Medições



Fonte: "https://dashboard.greenant.com.br/#!/dashboard/meters/1475?startDate=2020-08-16&endDate=2020-09-15&period=day"

O sistema de gerenciamento de energia do “Edifício Inteligente” conta com notificações de alerta (enviados por *e-mail* ou mensagem SMS) quando são ultrapassados os limites pré-estabelecidos da demanda. Esses limites são baseados em um histórico de consumo e despesas do edifício por pontos de medição, conforme podemos observar na figura 5.

Figura 5 Sistema de gerenciamento de energia do “Edifício Inteligente” – Notificações

Pontos de Medição	Potência Agora	FP Agora	Consumo	Despesa	Demanda Máxima no Período
Cabine de Entrada	117,0 kW	0,99 ind ①	129,90 MWh	R\$ 88.773,52	874,2 kW
ELE-QG8-1P-05	18,1 kW	0,98 Cap ①	12,63 MWh	R\$ 9.286,42	27,5 kW
TRA-SS01	27,6 kW	0,99 Cap ①	13,59 MWh	R\$ 9.900,26	85,5 kW
TRA-SS02	25,7 kW	0,91 Cap ①	23,80 MWh	R\$ 17.022,45	87,2 kW
TRA-SS03	12,6 kW	0,90 ind ①	14,68 MWh	R\$ 9.926,79	88,2 kW
TRA-SS04	1,3 kW	0,10 Cap ①	29,13 MWh	R\$ 17.814,47	500,3 kW
Total de itens: 6					

Fonte: <https://dashboard.greenant.com.br/#!/measuring-points>

Um dos benefícios apresentados pelo sistema de gerenciamento de energia do “Edifício Inteligente” são a previsão e o consumo típico para os demais meses, utilizando o algoritmo preditivo adaptativo, observado na figura 6.

Figura 6 Sistema de gerenciamento de energia do “Edifício Inteligente” – Previsão e consumo

Fonte: <https://dashboard.greenant.com.br/#!/dashboard/meters/1475?startDate=2020-09-16&endDate=2020-10-15&period=day>

5.2 Subsistema de climatização

Dentro do uso de energia em edifícios no Brasil, sabe-se que o principal consumo é para o sistema de climatização, representando 47% do uso de energia dos edifícios comerciais no país (ELETROBRAS, 2009). Desta forma, destaca-se a importância de racionalização deste consumo.

O objetivo do subsistema de climatização do edifício inteligente é controlar com eficiência a refrigeração dos ambientes.

Automatizar com eficiência as instalações de ar-condicionado é uma meta bem definida nos edifícios, pelo grande consumo de energia dessas instalações e correspondente custo operacional. (...) 50% da carga geral de energia no edifício (média estimada) corresponde a essa instalação. O montante é considerado elevado por se concentrar em um único item. O fato tem dado ênfase ao desenvolvimento contínuo de técnicas de controle economizadoras de energia com unidades CDD. (DE LA CRUZ, 2019, p. 384)

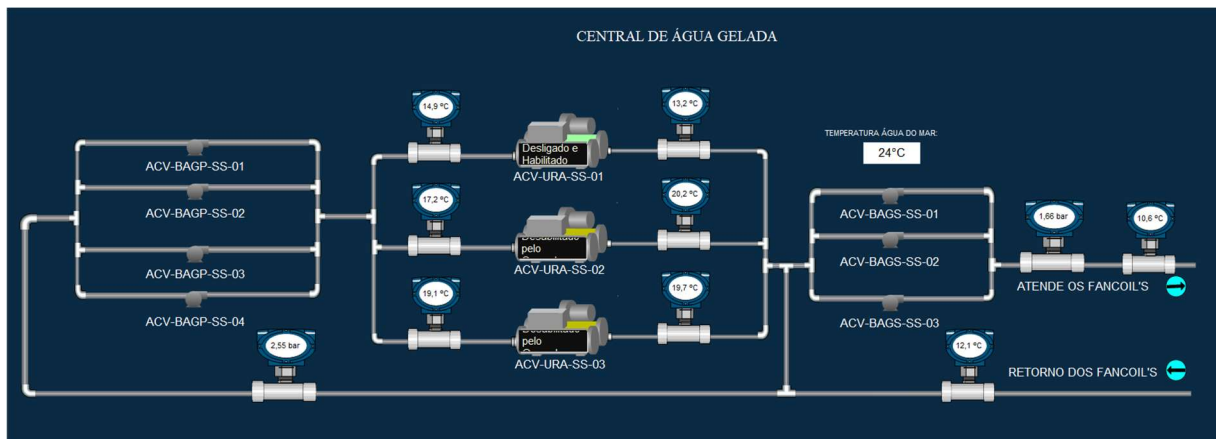
Relacionaremos a seguir os equipamentos que compõem o subsistema de climatização do edifício inteligente e suas ações:

- *Chiller* - responsável por resfriar água do edifício;
- Trocador de calor - responsável por realizar a troca de calor com água de condensação e com o circuito primário;
- Motobomba, responsável por abastecer água do circuito primário, secundário, condensação e trocador de calor;
- *Damper* - responsável por controlar a vazão de ar (instalados nos dutos que atende os ambientes);
- Ventilador/exaustor, responsável pela exaustão dos ambientes e pela renovação do ar;
- *Fan coil* - responsável pelo controle de umidade e temperatura dos ambientes;
- Transmissor de temperatura - responsável pelo controle de vazão dos circuitos primário, secundário, de condensação e controle de demanda;
- Transmissor de pressão - responsável pela operação das motobombas, garantindo o fluxo adequado no circuito;
- Transmissor de vazão - responsável pelo controle de demanda dos circuitos primeiro, secundário e de condensação;
- Válvulas motorizadas - responsáveis por controlar o fluxo de água;
- Reservatório de água - responsável pelo armazenamento da água que é utilizada com o trocador de calor;
- Transmissor de nível ultrassônico - responsável por fornecer o nível do reservatório de água;

- UTA - responsável por garantir a renovação do ar dos ambientes usando o ar externo, garantindo a qualidade do ar.

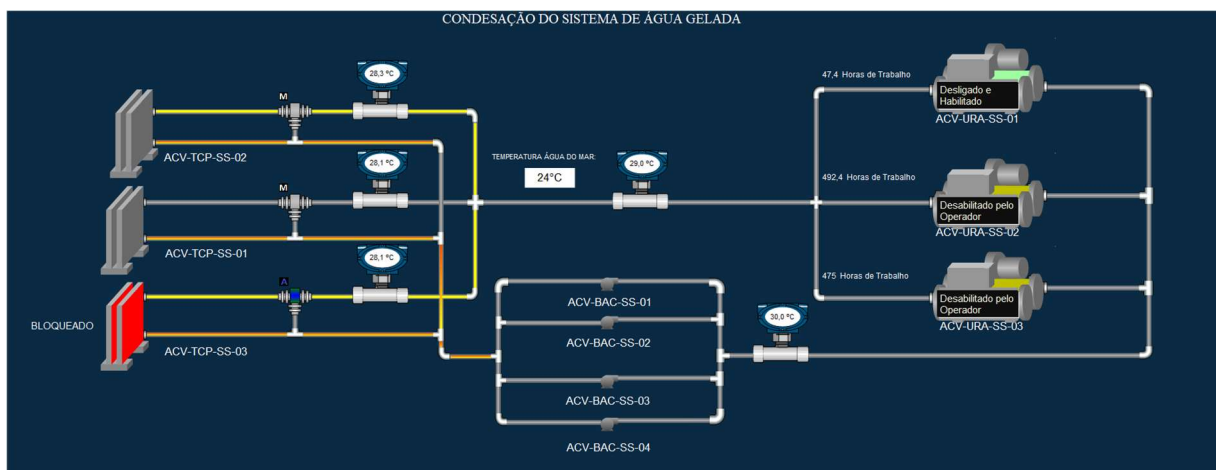
Podemos visualizar o processo de climatização do edifício inteligente nas figuras abaixo:

Figura 7 Central de água gelada



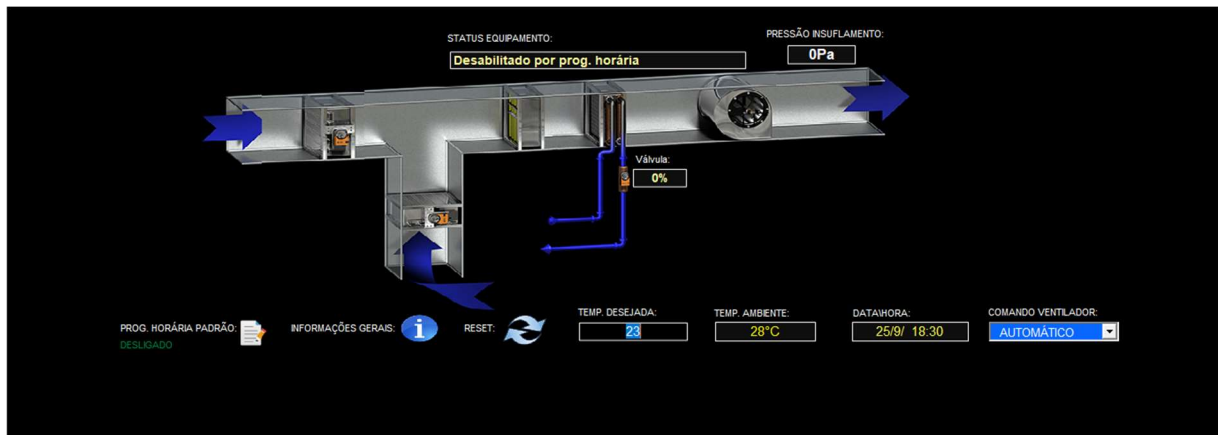
Fonte: BMS do edifício inteligente.

Figura 8 Condensação



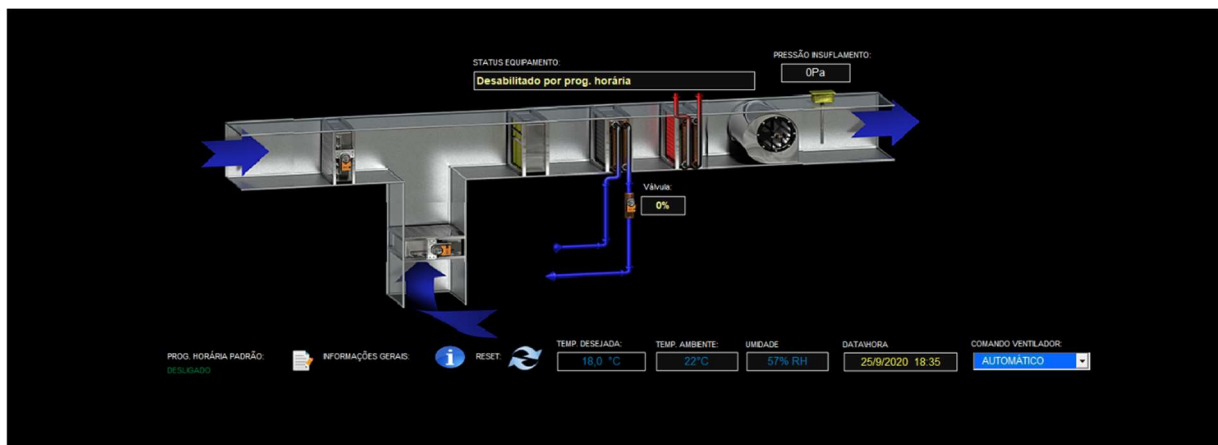
Fonte: BMS do edifício inteligente.

Figura 9 Fancoil com controle de insuflamento e temperatura



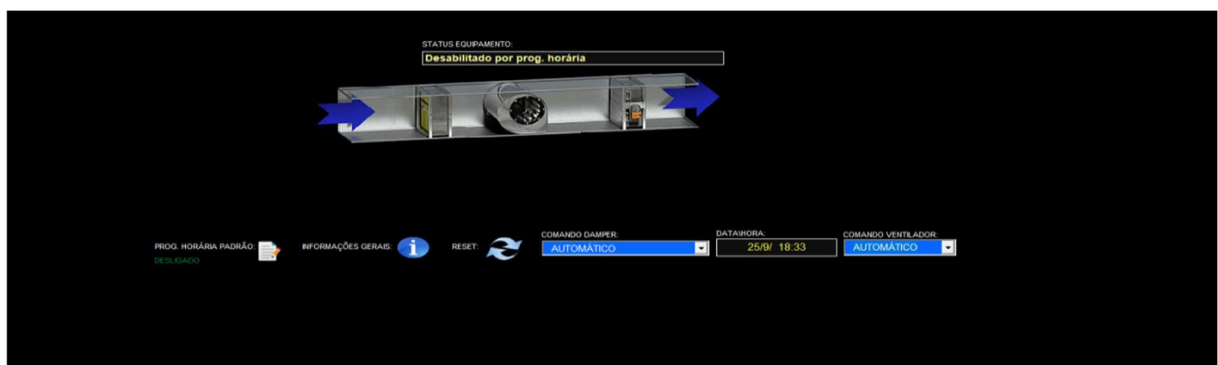
Fonte: BMS do edifício inteligente.

Figura 10 Fancoil com controle de insuflamento, temperatura e umidade



Fonte: BMS do edifício inteligente.

Figura 11 Ventilador/Exaustor controle de vazão dos ambientes



Fonte: BMS do edifício inteligente.

Figura 12 UTA unidade de tratamento de ar



Fonte: BMS do edifício inteligente.

5.3 Subsistema de controle de iluminação

O Subsistema de controle de iluminação do edifício inteligente permite a programação e controle das luzes de áreas internas e/ou externas, possibilitando iluminar os ambientes sem desperdício. Podemos citar como benefício relevante do referido sistema em relação a um sistema elétrico convencional a habilidade de controlar a luz de qualquer interface, além de possibilitar o controle de cenários de iluminação através das controladoras digitais, controladoras dimerizáveis, célula fotoelétrica e sensores de presença.

A operação de controle da iluminação do edifício inteligente utiliza a combinação de técnicas, que serão citadas a seguir:

- Programação fixa no tempo – a programação obedece a períodos fixos de operação no regime diário (figura 13);
- Programação variável no tempo – contempla diversos regimes pré-programados, dependentes de horário, calendário e eventos (figura 14);
- Detecção de presença – os acionamentos são operados por sensores de presença do tipo infravermelho ou ultrassônico;
- Incidência de eventos – neste caso, são combinados recursos de segurança com recursos de controle de iluminação;
- Condições variáveis de luz – combina recursos de controle digital com dispositivos dimerizáveis e sensores de luz (figura 15).

Relacionaremos a seguir os equipamentos que compõem o Subsistema de controle de iluminação do edifício inteligente e suas ações:

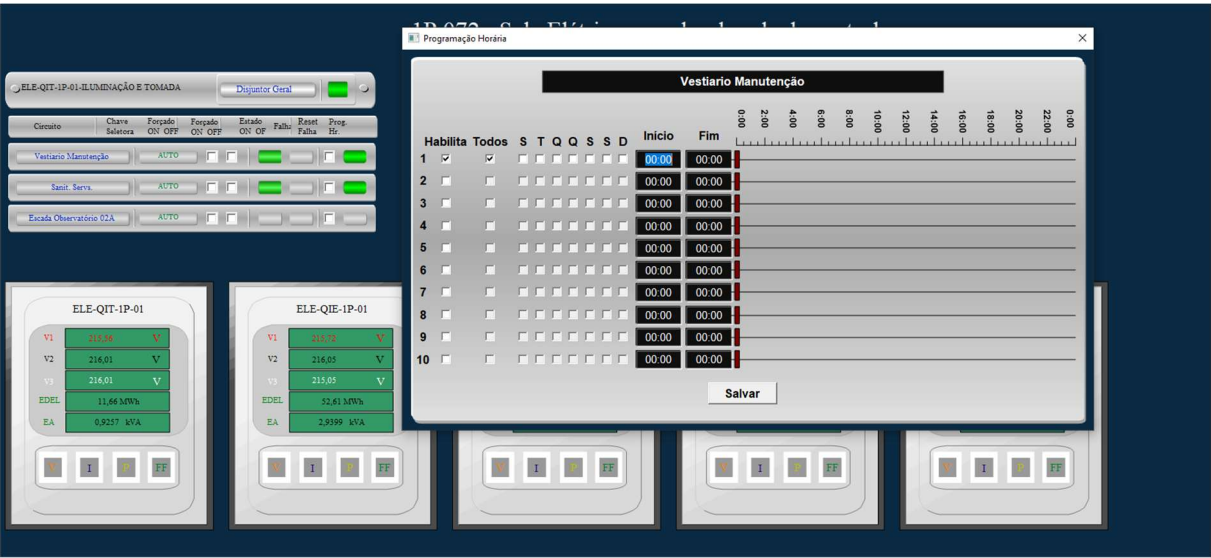
- Contator – é um dispositivo eletromecânico que realiza a comutação dos seus contatos de carga;
- Contator Auxiliar – é a interface que controla as bobinas magnéticas, usado como bloco auxiliar do contator;
- Relé – é um dispositivo que possui uma bobina que conecta os terminais nas posições Normalmente Aberto (NA) e Normalmente Fechado (NF);
- Disjuntor – é um dispositivo de proteção dos cabos e equipamento elétricos;
- Reator eletrônico – é o um equipamento auxiliar para o acendimento de lâmpada e para limitar a corrente e adequar a tensão da lâmpada;
- Reator dimerizável – controla nível de cena na iluminação definido a potência da lâmpada;
- Sensor de presença – dispositivo que tem função de detectar, sinalizar e responder de maneira rápida e eficiente quando detecta presença de qualquer objeto;
- Sensor ultrassônico – dispositivo que tem função de detectar, sinalizar e responder de maneira rápida e eficiente através da detecção de onda sonora.

Figura 13 Programação fixa no tempo do edifício inteligente



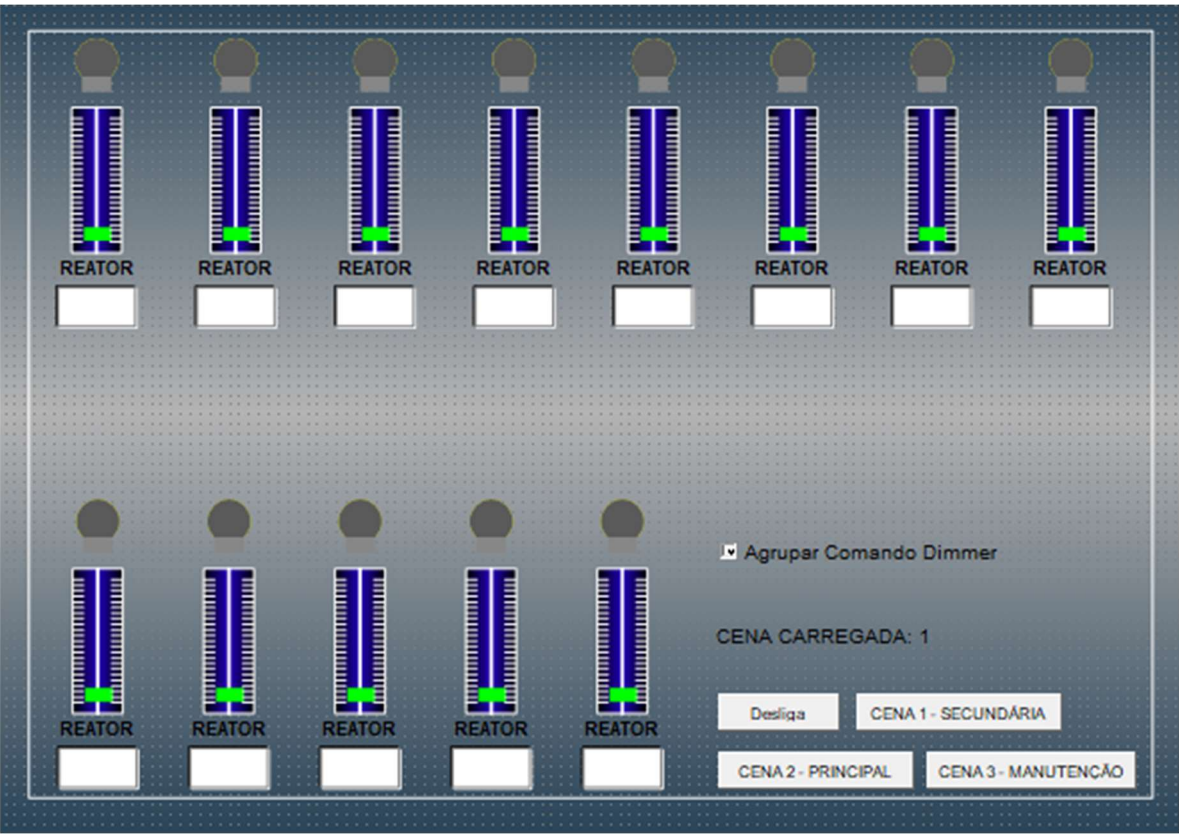
Fonte: BMS do edifício inteligente.

Figura 14 Programação variável no tempo do edifício inteligente



Fonte: BMS do edifício inteligente.

Figura 15 Condições variáveis de luz do edifício inteligente



Fonte: BMS do edifício inteligente.

5.4 Subsistema de motobombas

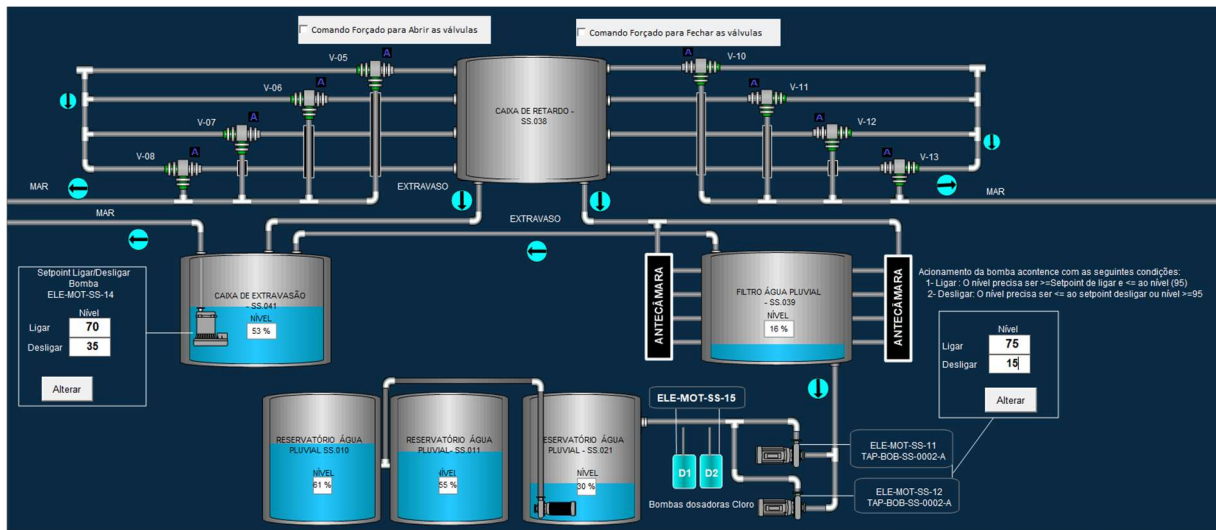
É notável a tendência atual de utilização de recursos hídricos visando a redução do desperdício, como o reaproveitamento de águas pluviais, a captação de água de chuva e os dispositivos economizadores de água em equipamentos hidráulicos encontrados no edifício inteligente. O subsistema de motobombas do referido edifício permite o controle das bombas de água pluvial, água potável, água de esgoto, drenagem, incêndio e irrigação dos jardins. Segundo DE LA CRUZ (2019), a operação do sistema de motobombas traz as seguintes vantagens, dentre outras:

- Permitir a qualquer momento a operação “convencional”, sem auxílio do sistema de automação;
- Evitar a operação das motobombas no horário de ponta, sem prejudicar o abastecimento da água;
- Substituir uma motobomba com defeito de forma automática;
- Providenciar a substituição de uma motobomba (com a partida da unidade de reserva) a partir do terminal de controle ou de forma automática.

Relacionaremos a seguir a composição do subsistema de motobombas do edifício inteligente e suas ações:

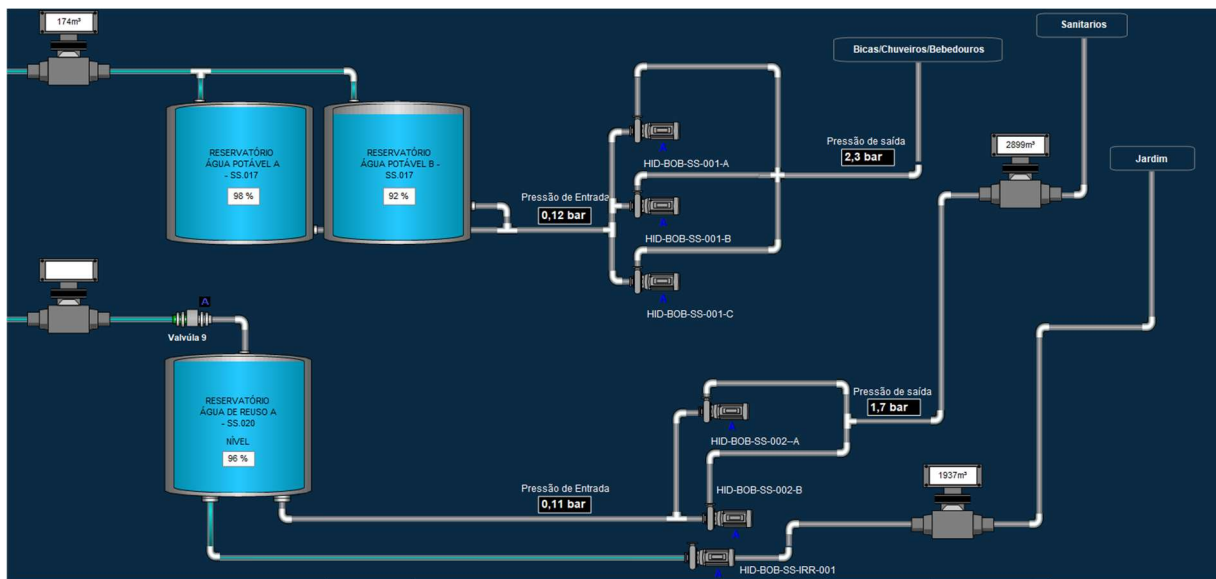
- Motobomba de água pluvial – responsável por transferir água do filtro de captação para os reservatórios de água;
- Motobomba de água potável – responsável por realizar o abastecimento das torneiras e chuveiros do difícil;
- Motobomba de drenagem – responsável por realizar a transferência de água dos posso de drenagem para os poços de esgoto;
- Motobomba de esgoto – responsável por realizar a transferência da água de esgoto do edifício para a linha de esgoto da concessionária;
- Motobomba de incêndio – responsável por realizar o abastecimento dos hidrantes;
- Motobomba de irrigação dos jardins – responsável por realizar a irrigação;
- Motobomba de caixa de extravaso – responsável por coletar água dos tanques quando ocorre transbordo.

Figura 16 Motobomba de água pluvial e extravaso



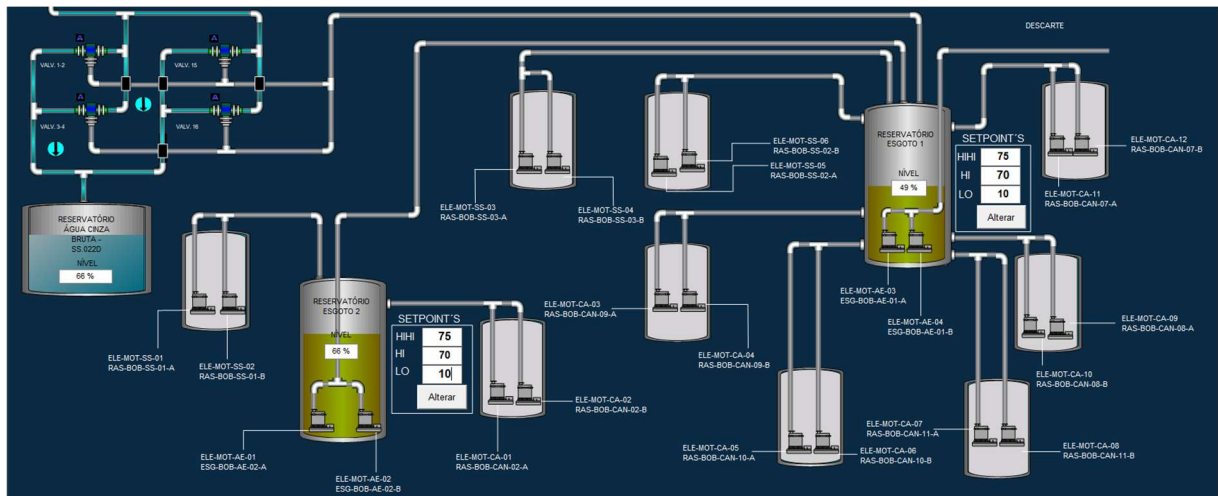
Fonte: BMS do edifício inteligente.

Figura 17 Motobomba de água potável



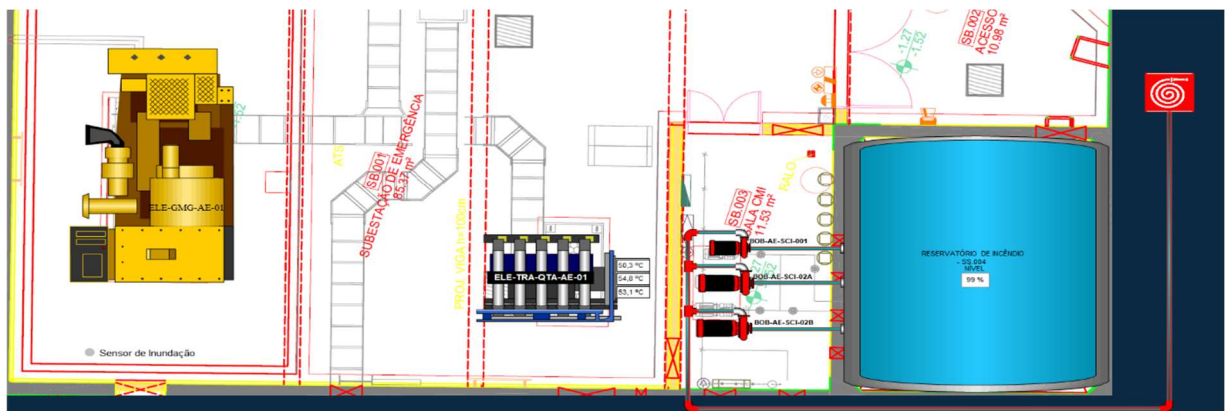
Fonte: BMS do edifício inteligente.

Figura 18 Motobomba de drenagem, Motobompo de esgoto



Fonte: BMS do edifício inteligente.

Figura 19 Motobomba de incêndio



Fonte: BMS do edifício inteligente.

5.5 Subsistema de detecção e alarme de incêndio

Segundo DE LA CRUZ (2019), o sistema de detecção de alarme de incêndio é uma especialidade baseada em normas e padrões técnicos. O referido subsistema tem o objetivo de proteger vidas e o patrimônio empresarial, sendo essas funções de extrema importância para o edifício inteligente. O subsistema de detecção e alarme de incêndio é diferenciado pelo modo de operação e instalação, que é classificado como endereçável. De acordo com a NBR 17240, da Associação Brasileira de Normas Técnicas (ABNT), um sistema completo é composto por alguns itens, que

funcionam em conjunto na detecção e sinalização de fumaça e fogo. Citaremos abaixo os equipamentos que compõem o subsistema:

- Central de alarme – é responsável por monitorar os dispositivos;
- Sinalizador audiovisual – é responsável por indicar área que ocorreu o sinistro;
- Acionador e detector de temperatura ou fumaça – é responsável por detectar a temperatura ou fumaças nos dutos e nas áreas;
- Acionadores manuais – é responsável por acionamento através de ação humana.

O subsistema de detecção de alarme de incêndio usa a técnica de endereçamento de dispositivo. Essa técnica oferece velocidade na tomada de ação e prevenção de incêndios, pois informa diretamente a central qual dispositivo foi acionado e sua localização exata. Isso reduz expressivamente os riscos e aumento do perigo devido sua precisão. Os sistemas endereçáveis são subdivididos em duas classes: A e B, que serão definidas a seguir. Cabe ressaltar que no edifício inteligente é usada a classe A.

- Classe A – Nesta classe, cada circuito do sistema possui fiação de retorno à central. Assim, quando ocorre algum problema em um dos dispositivos ou caso o cabo de ligação seja rompido, o funcionamento do sistema não é paralisado;
- Classe B – Na classe B, não há fiação de retorno à central, desta forma, caso ocorra interrupção em algum ponto, o sistema poderá ser paralisado parcial ou totalmente. O diferencial entre as duas classes é que na B são utilizados menos cabos, o que reduz o custo da instalação.

5.6 Subsistema de controle de acesso

O subsistema de controle de acesso é a combinação de software, hardware e ação humana, que se comunicam de forma ordenada tendo como função principal controlar o acesso das áreas no edifício inteligente.

De acordo com DE LA CRUZ (2019), a função do controle de acesso é imprescindível nos edifícios corporativos e institucionais onde critérios de segurança determinem o controle permanente de seus ocupantes. Ainda de acordo com o mesmo autor, a finalidade dessa instalação é controlar de forma dinâmica a identidade, o local e o tempo de acesso da pessoa que pretende adentrar em um acesso controlado. A seguir citaremos os componentes do subsistema.

- Hardware – é a parte física formada por componentes eletrônicos por exemplo controladora de acesso. Citaremos abaixo as suas vantagens.
- Possui a interface de comunicação TCP/IP;
- Permite armazenamento interno das listas de cartões e tabelas dos sistemas em caso de possíveis falhas de comunicação com o servidor;
- Notificação de alarme em caso de sinistros;
- Possui interface integradora suportando diferentes tipos de dispositivos de bloqueio através dos relés de saída - contato seco;
- Comunicação com o protocolo Wiegand para atender todos os tipos de leitores RFID e biométricos.
- Permite definir as áreas que devem ser acessadas assim como restrições de acordo com calendários e horários.
- Software – sistema de computação que realiza a interface com hardware e a ação humana. Citaremos abaixo sua arquitetura.
- Sistema é instalado e configurado em um servidor para realizar a interface com os hardwares de controle de acesso;
- Terminal de acesso, que é uma estação de trabalho usada para gerenciar e configurar todo o hardware, ação humana.

5.7 Monitoramento das instalações de utilidades

O BMS do edifício inteligente conta com o sistema integrador - MonitorAll. Ele é responsável por integrar todos os subsistemas. Citaremos abaixo suas atribuições.

- Monitoramento de Falha – quando o equipamento é impedido de operar exemplo (curto circuito);
- Monitoramento de Alerta – quando o equipamento apresenta problemas e não é impedido de operar;
- Monitoramento de status – indica se o equipamento está ligado, desligado, manual, automático, solicitado para operação e estado de funcionamento (atribuição somente para o subsistema de climatização);
- Monitoramento de nível – indica o nível do reservatório;
- Monitoramento de alarme de incêndio – indica quando algum alarme do subsistema de detecção de incêndio foi acionado;

- Monitoramento de alarme – indica quando algum subsistema entrou em estado crítico;
- Monitoramento de temperatura – indica a temperatura do ambiente e do sistema de climatização;
- Monitoramento de pressão – indica a pressão das tubulações e das motobombas;
- Monitoramento do controle de acesso – indica o status das portas aberta, fechada ou entrada forçada;
- Monitoramento de câmera – permitindo acesso as imagens em tempo real;
- Relatório – emissões de relatório de alarme, falha, consomem de água e energia;
- Monitoramento de hora de operação – indica a quantidade de horas que o equipamento está em operação;
- Monitoramento de energia – indica as variações de tensão;
- Monitoramento de energia secundaria – indica a operação do gerador como consumo, horas de trabalho, nível do tanque, tense, corrente;
- Monitoramento de elevador – responsável por monitorar os status do elevador;
- Controle de iluminação – responsável por programar e controlar os circuitos de iluminação;
- Controle de climatização – responsável por programar e controlar o subsistema de climatização;
- Controle de motobomba – responsável por controlar o seu acionamento;
- Controle de porta – responsável por acionar porta;

6 CONCLUSÃO DA PESQUISA APLICADA

O projeto teve como objetivo apresentar o funcionamento de um sistema de automação predial em uma edificação denominada edifício inteligente. A automação possibilita o controle eficaz dos subsistemas, garantindo melhoria na eficiência energética e redução dos custos de manutenção dos equipamentos.

No desenvolvimento deste trabalho, foram apresentados os conceitos gerais de um Sistema de Gestão Predial (BMS). No BMS é feita a integração de subsistemas que realizam o gerenciamento predial, sendo apresentada nesta pesquisa aplicada os seus equipamentos, protocolos de rede, camadas e subsistemas - de gerenciamento de energia, de climatização, de controle de iluminação, de motobombas, de detecção e alarme de incêndio e de controle de acesso.

Conclui-se que, com a implantação do BMS no edifício inteligente, foi possível identificar melhoria na sua operação geral e em sua eficiência energética, justificando, assim, sua implantação. Os principais benefícios observados foram a redução do custo de energia do edifício e das operações de manutenção preventiva, preditiva e corretiva de seus equipamentos.

REFERÊNCIAS

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6023**: informação e documentação: referências: elaboração. Rio de Janeiro, 2018.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 6027**: sumário. Rio de Janeiro, 2012.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 10520**: informação e documentação citações em documentos: apresentação. Rio de Janeiro, 2011.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR14724**: informação e documentação: trabalhos acadêmicos: apresentação. Rio de Janeiro, 2011.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 17240**: sistemas de detecção e alarme de incêndio – Projeto, instalação, comissionamento e manutenção de sistemas de detecção e alarme de incêndio – Requisitos. Rio de Janeiro, 2010.

DE LA CRUZ, Jaime Díaz; DE LA CRUZ, Eduardo Díaz. **Automação predial 4.0**: a automação predial na quarta revolução. Rio de Janeiro: Brasport, 2019.

LIMA, Elaine Carvalho de; OLIVEIRA NETO, Calisto Rocha de. **Revolução industrial**: considerações sobre o pioneirismo industrial inglês. Revista Espaço Acadêmico, v. 17, n.194 2017. Disponível em: <http://www.periodicos.uem.br/ojs/index.php/EspacoAcademico/article/view/32912>. Acesso em 23 out. 2020.

ELETROBRAS. **Pesquisa de posse de equipamentos e hábitos de uso, ano base 2005**: classe residencial relatório Brasil - Sumário Executivo. Rio de Janeiro: ELETROBRAS; PROCEL, 2009. 187 p. (Avaliação do Mercado de Eficiência Energética no Brasil). Disponível em: <http://www.procelinfo.com.br/main.asp?View=%7B5A08CAF0-06D1-4FFE-B335-95D83F8DFB98%7D&Team=¶ms=itemID=%7B99EBBA5C-2EA1-4AEC-8AF2-5A751586DAF9%7D;&UIPartUID=%7B05734935-6950-4E3F-A182-629352E9EB18%7D>. Acesso em: 15 de set. 2020.

CÉSAR, Aldo. **Redes industriais**: o que são e para quê servem na indústria 4.0. Disponível em: <https://transformacaodigital.com/mercado/redes-industriais-o-que-sao-e-para-que-servem-na-industria-4-0/#:~:text=As%20redes%20industriais%20s%C3%A3o%20uma%20forma%20de%20automa%C3%A7%C3%A3o%20de%20ind%C3%BAstria,%2C%20atuadores%2C%20entre%20outros%20equipamentos>. Acesso em 15 de set. 2020.

MENDONÇA, Victor. **Sistemas de automação e controle**. 2013. Disponível em: <http://www.ebah.com.br/content/ABAAAchMAC/tudo-sobre-automacao>. Acesso em 21 out. 2020.

SPT Specialisty Product Technologies. **Diferenças entre mecatrônica e automação industrial.** Disponível em: <https://www.sptech.ind.br/post/diferen%C3%A7as-entre-mecatr%C3%B4nica-e-automa%C3%A7%C3%A3o-industrial>. Acesso em 11 de set. 2020.

DIGICOMP. **O que é building management system (BMS e suas principais vantagens.** Disponível em: <https://digicomp.com.br/o-que-e-building-management-system-bms-e-suas-principais-vantagens/>. Acesso em 19 de set. 2020.

INTELBRAS BLOG. **Veja as diferenças entre cada sistema de detecção e alarme de incêndio.** Disponível em: <http://blog.intelbras.com.br/veja-as-diferencas-entre-cada-sistema-de-deteccao-e-alarme-de-incendio/>. Acesso em: 02 de out. 2020.

ANEXO I

SCRIPTS

Script são instruções de computador, essa técnica executa ações no interior de um programa que serve para controlar e configurar o sistema. Segue abaixo o *script* usado no edifício inteligente.

- Subsistema de gerenciamento de energia – script de transmissão de dados e leitura no BMS. Foi usada a estrutura JSON para transferir os dados onde é feito o monitoramento em tempo real.

- Script de transmissão de dados do medidor de qualidade para gestão de consumo de energia;

```
Sub JSON_OnPreset()
```

```
Dim objXmlHttpMain , URL
```

```
Dim fCA,fCB,fCC,fCN 'Corrente
```

```
Dim fTA,ftB,ftC 'Tensao LN
```

```
Dim fPA,fPB,fPC 'Potencia Ativa
```

```
Dim QA,fQB,fQC 'Potencia Reativa
```

```
Dim iFEP,iREP 'Energia Ativa
```

```
Dim iFEQ,iREQ 'Energia Reativa
```

```
Dim id
```

```
Dim formattedTs
```

```
Dim j
```

```
Dim paddedId
```

```
Dim sHostName
```

```
j=0
```

```
sHostName = Application.GetObject("Dados.HostNameServer").Value
```

```
if sHostName = "EDRJ01SCP01" Then
```

```
for i = 4 to 9
```

```
j=j+1
```

```

formattedTs = FormataDataHora()
paddedId = Right("0000"&Cstr(i),4)
id= """"&"ED-ITELIGENTE "&paddedId&""""
if i = 8 then
'corrente
fCA=   Replace(Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_0_").Value,"",".")
fCB=   Replace(Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_1_").Value,"",".")
fCC=   Replace(Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_2_").Value,"",".")
fCN  =  Replace(Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_3_").Value,"",".")
'Tensao LN
fTA  =  Replace(Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_14_").Value,"",".")
fTB  =  Replace(Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_15_").Value,"",".")
fTC  =  Replace(Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_16_").Value,"",".")
'Potencia Ativa
fPA  =  Replace((Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_27_").Value*1000),"",".")
fPB  =  Replace((Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_28_").Value*1000),"",".")
fPC  =  Replace((Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_29_").Value*1000),"",".")
fQA  =  Replace((Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_31_").Value*1000),"",".")
fQB  =  Replace((Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_32_").Value*1000),"",".")
fQC  =  Replace((Application.GetObject("CT_UT_1P_01_Medidores.[ELE_QGB_1P-
05].rData4_33_").Value*1000),"",".")

```

```

iFEP                                                                    =
Replace(Application.GetObject("CT_UT_1P_01_Medidores.CONSUMO_ENERGIA_
QGB_1P_05.aConsumo_QGB_1_").Value,"",".")
iREP                                                                    =
Replace(Application.GetObject("CT_UT_1P_01_Medidores.CONSUMO_ENERGIA_
QGB_1P_05.aConsumo_QGB_2_").Value,"",".")
iFEQ                                                                    =
Replace(Application.GetObject("CT_UT_1P_01_Medidores.CONSUMO_ENERGIA_
QGB_1P_05.aConsumo_QGB_5_").Value,"",".")
iREQ                                                                    =
Replace(Application.GetObject("CT_UT_1P_01_Medidores.CONSUMO_ENERGIA_
QGB_1P_05.aConsumo_QGB_6_").Value,"",".")
elseif i = 9 then
fCA = Replace(Application.GetObject("ION7650.Corrente_A").Value,"",".")
fCB = Replace(Application.GetObject("ION7650.Corrente_B").Value,"",".")
fCC = Replace(Application.GetObject("ION7650.Corrente_C").Value,"",".")
fCN = "null"
fTA = Replace(Application.GetObject("ION7650.Voltage_In_A").Value,"",".")
fTB = Replace(Application.GetObject("ION7650.Voltage_In_B").Value,"",".")
fTC = Replace(Application.GetObject("ION7650.Voltage_In_C").Value,"",".")
fPA = Replace((Application.GetObject("ION7650.kW_a").Value*1000),"",".")
fPB = Replace((Application.GetObject("ION7650.kW_b").Value*1000),"",".")
fPC = Replace((Application.GetObject("ION7650.kW_c").Value*1000),"",".")
fQA = Replace((Application.GetObject("ION7650.kVar_a").Value*1000),"",".")
fQB = Replace((Application.GetObject("ION7650.kVar_b").Value*1000),"",".")
fQC = Replace((Application.GetObject("ION7650.kVar_c").Value*1000),"",".")
iFEP = Replace((Application.GetObject("ION7650.kWh_del").Value*1000),"",".")
iREP = Replace((Application.GetObject("ION7650.kWh_rec").Value*1000),"",".")
iFEQ = Replace((Application.GetObject("ION7650.kVARh_del").Value*1000),"",".")
iREQ = Replace((Application.GetObject("ION7650.kVARh_rec").Value*1000),"",".")
else

```

```

fCA                                                    =
Replace(Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_0_").Value,"",".")
fCB                                                    =
Replace(Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_1_").Value,"",".")
fCC                                                    =
Replace(Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_2_").Value,"",".")
fCN                                                    =
Replace(Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_3_").Value,"",".")
fTA                                                    =
Replace(Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_14_").Value,"",".")
fTB                                                    =
Replace(Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_15_").Value,"",".")
fTC                                                    =
Replace(Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_16_").Value,"",".")
fPA                                                    =
Replace((Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_27_").Value*1000),"",".")
fPB                                                    =
Replace((Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_28_").Value*1000),"",".")
fPC                                                    =
Replace((Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_29_").Value*1000),"",".")
fQA                                                    =
Replace((Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_31_").Value*1000),"",".")

```



```

fQB =
Replace((Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_32_").Value*1000),",",",".")
fQC =
Replace((Application.GetObject("CT_UT_SS_01.MEDIDORES.CommModbus.rData"
&i&"_33_").Value*1000),",",",".")
iFEP =
Replace(Application.GetObject("CT_UT_SS_01.CONSUMO_ENERGIA.aConsumo_
TRA_SS0"&j&"_1_").Value,",",",".")
iREP =
Replace(Application.GetObject("CT_UT_SS_01.CONSUMO_ENERGIA.aConsumo_
TRA_SS0"&j&"_2_").Value,",",",".")
iFEQ=Replace(Application.GetObject("CT_UT_SS_01.CONSUMO_ENERGIA.aCons
umo_TRA_SS0"&j&"_5_").Value,",",",".")
iREQ =
Replace(Application.GetObject("CT_UT_SS_01.CONSUMO_ENERGIA.aConsumo_
TRA_SS0"&j&"_6_").Value,",",",".")
end if
strJSONToSend = "{""uid"":""&id&",""hwvr"":""Elipse
0.1""",""swvr"":""v0.1""",""packets"":[{""samples"":[{""timestamp"":""&formattedTs&""",""
va"":""&fTA&",""vb"":""&fTB&",""vc"":""&fTC&",""ia"":""&fCA&",""ib"":""&fCB&",""ic"":""&fCC
&",""in"":""&fCN&",""pa"":""&fPA&",""pb"":""&fPB&",""pc"":""&fPC&",""qa"":""&fQA&",""qb"":
""&fQB&",""qc"":""&fQC&",""fep"":""&iFEP&",""rep"":""&iREP&",""feq"":""&iFEQ&",""req"":""
&iREQ&""}]}}}"
URL="https://cloud9.greenant.com.br/meter/" 'Endereço do webservice
Set objXmlHttpMain = CreateObject("Msxml2.ServerXMLHTTP")

On Error Resume Next
objXmlHttpMain.open "POST",URL, False
If Err Then 'handle errors
'Msgbox Err.Description & " [0x" & Hex(Err.Number) & "]"
End If
On Error Goto 0

```

```

objXmlHttpMain.setRequestHeader "Content-Type", "application/json"
objXmlHttpMain.send strJSONToSend
strResponse = objXmlHttpMain.responseText
'msgbox strJSONToSend
Next
end if
End Sub
function FormataDataHora()
DataHora = now()
set Shell = CreateObject("WScript.Shell")
atb = "HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\TimeZoneInformation\
ActiveTimeBias"
offsetMin = Shell.RegRead(atb)
UTC = DateAdd("n", offsetMin, DataHora) 'Soma o Offset na DataHora atual
iAno = Mid(UTC,7,4)
iMes = Mid(UTC,4,2)
iDia = Mid(UTC,1,2)
iHora = Mid(UTC,12,19)
FormataDataHora=iAno&"-"&iMes&"-"&iDia&" "&iHora&" +00"
end function
Sub Temp()
End Sub

```

- **Script de leitura dos medidores de energia;**

```

PROGRAM CommModbus
FUNCTION FUN_SWAP : REAL
VAR_INPUT
dwInput:DWORD;
END_VAR
VAR
rAux AT%MD500:REAL;

```

END_VAR

%MD600:=dwInput;

%MB500:=%MB602;

%MB501:=%MB603;

%MB502:=%MB600;

%MB503:=%MB601;

FUN_SWAP:=rAux;

FUNCTION FUN_Swplnt64 : LREAL

VAR_INPUT

dw1:DWORD;

dw2:DWORD;

END_VAR

VAR

dw3:DWORD;

dw4:DWORD;

S1:STRING;

END_VAR

dw3:=ROR(dw1,16);

dw4:=ROR(dw2,16);

S1:=CONCAT(DWORD_TO_STRING(dw3),DWORD_TO_STRING(dw4));

FUN_Swplnt64:=STRING_TO_LREAL(S1);

VAR

wStateMachine: WORD;

busy: BOOL;

bError: BOOL;

bBusy: BOOL;

bDone: BOOL;

T1: TON;

udError:modbus_errors;

k,i:WORD;

tTimeStateMachine:TIME:=T#100ms;

ModbusChannel1: ModbusRtuMaster_KL6x22B;

ModbusChannel2: ModbusRtuMaster_KL6x22B; (*eu fiz assim *)

bResetComModBus: BOOL;

bHabilitaMedicao: ARRAY [1..cNUM_MEDIDORES] OF BOOL; (*
TRUE=DESABILITADO,FALSE=HABILITADO *)

stErroModbusChannel: ARRAY[1..cNUM_MEDIDORES] OF ST_MODBUS_ERROR;
(* estrutura declarada em DATA_Types *)

Data1: ARRAY [0..100] OF DWORD;

Data2: ARRAY[0..100] OF DWORD;

aConsumo_QIT_GT_02:ARRAY[1..50] OF LREAL;(*~ (OPC :1 : available for OPC-
clients) *)

aConsumo_QAC_GT_02:ARRAY[1..50] OF LREAL;(*~ (OPC :1 : available for OPC-
clients) *)

rVoltage_1_AB: REAL;(*~ (OPC :1 : available for OPC-clients) *)

rVoltage_1_BC: REAL;(*~ (OPC :1 : available for OPC-clients) *)

rVoltage_1_CA: REAL;(*~ (OPC :1 : available for OPC-clients) *)

rVoltage_2_AB: REAL;(*~ (OPC :1 : available for OPC-clients) *)

rVoltage_2_BC: REAL;(*~ (OPC :1 : available for OPC-clients) *)

rVoltage_2_CA: REAL;(*~ (OPC :1 : available for OPC-clients) *)

rData1:(*~(OPC :1 : available for OPC-clients) *) ARRAY [0..101] OF REAL;
(*~(OPC :1 : available for OPC-clients) *)

rData2:(*~(OPC :1 : available for OPC-clients) *) ARRAY [0..101] OF REAL;
(*~(OPC :1 : available for OPC-clients) *)

fbReadRegisters : FB_MBReadRegs;

bReadRegisters : BOOL;

bReadRegistersBusy : BOOL;

```

bReadRegistersError      : BOOL;
nReadRegistersErrorId    : UDINT;
nReadRegistersCount      : UDINT;
nQuantity                : WORD := 8;
nMBAAddr                 : WORD := 0; (*offset*)
arrData2                 : ARRAY [1..8] OF UDINT; (* no Elipse configurar
P1/N1=1,2,3..., P2/N2=8=2 porem com swap words, P3/N3=0, P4/N4=OFFSET=0
*)
y : UDINT;
y2 : UDINT;
n: INT;
startAddr:WORD:=3203;
i1: INT;
END_VAR

```

```

VAR CONSTANT

```

```

cNUM_MEDIDORES: WORD :=2;

```

```

END_VAR

```

```

IF NOT bResetComModBus THEN (* bResetComModBus=1, reset não solicitado *)
CASE wStateMachine OF

```

```

0:      (* Inicio... prepara a comunicação! *)

```

```

bBusy := TRUE;

```

```

bDone:=FALSE;

```

```

ModbusChannel1.ReadInputRegs(EXECUTE:= FALSE);

```

```

wStateMachine:= 10;

```

```

10:

```

```

(*Multimedidor PM5100, ELE-QIT-GT-02*)

```

```

IF NOT bHabilitaMedicao[1] THEN

```

```

ModbusChannel1.ReadRegs(

```

```

UnitID:= 1,

```

```

Quantity:= 100,

```

```

MBAAddr:= 2999,

```

```

cbLength:= SIZEOF(Data1)/2,
pMemoryAddr:= ADR(Data1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
ModbusChannel1.ReadRegs(Execute:= FALSE);
IF ModbusChannel1.Error THEN
stErroModbusChannel[1].bError:=TRUE;
stErroModbusChannel[1].sTag:='PM5110-1ELE-QIT-GT-02';
stErroModbusChannel[1].sID:='01';
stErroModbusChannel[1].sIDMaq:='10';
stErroModbusChannel[1].sDescricao:='Erro Multimedidor ELE-QIT-GT-02';
stErroModbusChannel[1].eCodModbus:=ModbusChannel1.ErrorId;
wStateMachine:= 20;
ELSE
wStateMachine:= 20;
END_IF
END_IF
ELSE
wStateMachine:= 20;
END_IF
20:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=30;
END_IF
30:
(*Multimedidor PM5100, ELE-QAC-GT-02*)
IF NOT bHabilitaMedicao[2] THEN
ModbusChannel1.ReadRegs(
UnitID:= 2,

```

```

Quantity:= 100,
MBAAddr:= 2999,
cbLength:= SIZEOF(Data2)/2,
pMemoryAddr:= ADR(Data2),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
ModbusChannel1.ReadRegs(Execute:= FALSE);
IF ModbusChannel1.Error THEN
stErroModbusChannel[2].bError:=TRUE;
stErroModbusChannel[2].sTag:='PM5110-2 ELE-QAC-GT-02';
stErroModbusChannel[2].sID:='02';
stErroModbusChannel[2].sIDMaq:='20';
stErroModbusChannel[2].sDescricao:='Erro Multimedidor ELE-QAC-GT-02';
stErroModbusChannel[2].eCodModbus:=ModbusChannel1.ErrorId;
wStateMachine:=40;
ELSE
wStateMachine:= 40;
END_IF
END_IF
ELSE
wStateMachine:= 40;
END_IF
40:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=50;
END_IF
50:
(*Multimedidor PM5100, ELE-QIT-GT-02*)
IF NOT bHabilitaMedicao[1] THEN

```

```

ModbusChannel1.ReadRegs(
UnitID:= 1,
Quantity:= 100,
MBAAddr:= startAddr,
cbLength:= SIZEOF(Data1)/2,
pMemoryAddr:= ADR(Data1[50]),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
ModbusChannel1.ReadRegs(Execute:= FALSE);
IF ModbusChannel1.Error THEN
stErroModbusChannel[1].bError:=TRUE;
stErroModbusChannel[1].sTag:='PM5110-1ELE-QIT-GT-02';
stErroModbusChannel[1].sID:='01';
stErroModbusChannel[1].sIDMaq:='10';
stErroModbusChannel[1].sDescricao:='Erro Multimedidor ELE-QIT-GT-02';
stErroModbusChannel[1].eCodModbus:=ModbusChannel1.ErrorId;
wStateMachine:= 60;
ELSE
wStateMachine:= 60;
END_IF
END_IF
ELSE
wStateMachine:= 60;
END_IF
60:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=70;
END_IF
70:

```



```

(*Multimedidor PM5100, ELE-QAC-GT-02*)
IF NOT bHabilitaMedicao[2] THEN
ModbusChannel1.ReadRegs(
UnitID:= 2,
Quantity:= 100,
MBAAddr:= startAddr,
cbLength:= SIZEOF(Data2)/2,
pMemoryAddr:= ADR(Data2[50]),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
ModbusChannel1.ReadRegs(Execute:= FALSE);
IF ModbusChannel1.Error THEN
stErroModbusChannel[2].bError:=TRUE;
stErroModbusChannel[2].sTag:='PM5110-2 ELE-QAC-GT-02';
stErroModbusChannel[2].sID:='02';
stErroModbusChannel[2].sIDMaq:='20';
stErroModbusChannel[2].sDescricao:='Erro Multimedidor ELE-QAC-GT-02';
stErroModbusChannel[2].eCodModbus:=ModbusChannel1.ErrorId;
wStateMachine:=999;
ELSE
wStateMachine:= 999;
END_IF
END_IF
ELSE
wStateMachine:= 999;
END_IF
999:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=0;

```

```

END_IF
END_CASE
ELSE
bDone := FALSE;
bBusy := FALSE;
bError := FALSE;
udError:= 0;
wStateMachine:=0;
FOR i:=1 TO cNUM_MEDIDORES DO
stErroModbusChannel[i].bError:=FALSE;
stErroModbusChannel[i].sTag:='NO ERROR';
stErroModbusChannel[i].sID:='NO ERROR';
stErroModbusChannel[i].sIDMaq:='NO ERROR';
stErroModbusChannel[i].sDescricao:='NO ERROR';
stErroModbusChannel[i].eCodModbus:=0;
END_FOR
END_IF
(*Atribui para variável correspondente**)
rVoltage_1_AB:=FUN_SWAP(Data1[0]);
rVoltage_1_BC:=FUN_SWAP(Data1[1]);
rVoltage_1_CA:=FUN_SWAP(Data1[2]);
rVoltage_2_AB:=FUN_SWAP(Data2[0]);
rVoltage_2_BC:=FUN_SWAP(Data2[1]);
rVoltage_2_CA:=FUN_SWAP(Data2[2]);
IF n<=100 THEN
n:=n+1;
ELSE
n:=0;
END_IF
i1:=1;
FOR i:= 50 TO 98 BY 2 DO
aConsumo_QIT_GT_02[i1]:=FUN_Swplnt64(Data1[i],Data1[i+1]);
aConsumo_QAC_GT_02[i1]:=FUN_Swplnt64(Data2[i],Data2[i+1]);

```

```

i1:=i1+1;
END_FOR
rData1[n]:= FUN_SWAP(Data1[n]);
rData2[n]:= FUN_SWAP(Data2[n]);

```

- Subsistema de climatização – o script é responsável por controlar as ações de comando forçado a ligar, forçado a desligar, reset, programação horária.

- Script genérico de comando forçado a ligar, desligar;

```
Sub XDropBox_Constructor()
```

```
'Adiciona os itens à lista do ComboBox.
```

```
Item("ComboBox1").Clear()
```

```
If QtdItens => 1 Then
```

```
Item("ComboBox1").AddItem Item1
```

```
End If
```

```
If QtdItens => 2 Then
```

```
Item("ComboBox1").AddItem Item2
```

```
End If
```

```
If QtdItens => 3 Then
```

```
Item("ComboBox1").AddItem Item3
```

```
End If
```

```
If QtdItens => 4 Then
```

```
Item("ComboBox1").AddItem Item4
```

```
End If
```

```
If QtdItens => 5 Then
```

```
Item("ComboBox1").AddItem Item5
```

```
End If
```

```
If QtdItens => 6 Then
```

```
Item("ComboBox1").AddItem Item6
```

```
End If
```

```
If QtdItens => 7 Then
```

```
Item("ComboBox1").AddItem Item7
```

```
End If
```

```
If QtdItens => 8 Then
```

```

Item("ComboBox1").AddItem Item8
End If
End Sub

```

- Script de comando de reset;

```

Sub btnRest_Click()
    resp = msgbox ("Voce realmente deseja Reiniciar o controlador
    ?",vbYesNo+vbExclamation+ vbDefaultButton2, "Reboot do Controlador")
    if resp = vbYes then
        resp = msgbox ("O controlador será reiniciado !!" +Chr(13)+Chr(10)+ "Voce
        realmente deseja Reiniciar o controlador ?",vbYesNo+vbCritical+
        vbDefaultButton2,"Reboot do Controlador")
        if resp = vbYes then
            FancoilTemperatura.Fonte.ResetFalha = true
        end if
    end if
End Sub

```

- Script de programação horaria;
- Script de seleção do dia da semana;

```

Sub XDroPBox_Constructor()
    'Adiciona os itens à lista do ComboBox.
    Item("ComboBox1").Clear()
    If QtdItens => 1 Then
        Item("ComboBox1").AddItem Item1
    End If
    If QtdItens => 2 Then
        Item("ComboBox1").AddItem Item2
    End If
    If QtdItens => 3 Then
        Item("ComboBox1").AddItem Item3
    End If
    If QtdItens => 4 Then
        Item("ComboBox1").AddItem Item4
    End If
End Sub

```

```

End If
If QtdItens => 5 Then
Item("ComboBox1").AddItem Item5
End If
If QtdItens => 6 Then
Item("ComboBox1").AddItem Item6
End If
If QtdItens => 7 Then
Item("ComboBox1").AddItem Item7
End If
If QtdItens => 8 Then
Item("ComboBox1").AddItem Item8
End If
End Sub

```

- Script para criar a programação horaria;

```

Sub btnSalvar_Click()
if XPH.Item("chkbHabilitaProgracaoHoraria").Value = -1 then
XPH.Item("Escala2").SetFocus()
aux = abs(Left(XPH.Item("Texto1").Value, 2)) & Right(XPH.Item("Texto1").Value, 2)
XPH.Fonte.PH1HoraInicial = int(aux)
aux = abs(Left(XPH.Item("Texto2").Value, 2)) & Right(XPH.Item("Texto2").Value, 2)
XPH.Fonte.PH1HoraFinal = int(aux)
aux = abs(Left(XPH.Item("Texto3").Value, 2)) & Right(XPH.Item("Texto3").Value, 2)
XPH.Fonte.PH2HoraInicial = int(aux)
aux = abs(Left(XPH.Item("Texto4").Value, 2)) & Right(XPH.Item("Texto4").Value, 2)
XPH.Fonte.PH2HoraFinal = int(aux)
aux = abs(Left(XPH.Item("Texto5").Value, 2)) & Right(XPH.Item("Texto5").Value, 2)
XPH.Fonte.PH3HoraInicial = int(aux)
aux = abs(Left(XPH.Item("Texto6").Value, 2)) & Right(XPH.Item("Texto6").Value, 2)
XPH.Fonte.PH3HoraFinal = int(aux)
aux = abs(Left(XPH.Item("Texto7").Value, 2)) & Right(XPH.Item("Texto7").Value, 2)

```

```

XPH.Fonte.PH4HoraInicial = int(aux)
aux = abs(Left(XPH.Item("Texto8").Value, 2)) & Right(XPH.Item("Texto8").Value, 2)
XPH.Fonte.PH4HoraFinal = int(aux)
aux = abs(Left(XPH.Item("Texto9").Value, 2)) & Right(XPH.Item("Texto9").Value, 2)
XPH.Fonte.PH5HoraInicial = int(aux)
aux = abs(Left(XPH.Item("Texto10").Value, 2)) & Right(XPH.Item("Texto10").Value,
2)
XPH.Fonte.PH5HoraFinal = int(aux)

aux = abs(Left(XPH.Item("Texto11").Value, 2)) & Right(XPH.Item("Texto11").Value,
2)
XPH.Fonte.PH6HoraInicial = int(aux)
aux = abs(Left(XPH.Item("Texto12").Value, 2)) & Right(XPH.Item("Texto12").Value,
2)
XPH.Fonte.PH6HoraFinal = int(aux)
aux = abs(Left(XPH.Item("Texto13").Value, 2)) & Right(XPH.Item("Texto13").Value,
2)
XPH.Fonte.PH7HoraInicial = int(aux)
aux = abs(Left(XPH.Item("Texto14").Value, 2)) & Right(XPH.Item("Texto14").Value,
2)
XPH.Fonte.PH7HoraFinal = int(aux)
MsgBox "Programação feita com sucesso !!!", vbExclamation, "Programação
Horaria"
else
MsgBox "É necesseraio Habilitar a programação horaria para alterar suas
programações. !!!", vbExclamation, "Programação Horaria"
end if
End Sub

```

- Subsistema de controle de iluminação – o script é responsável por controlar as ações de comando forçado a ligar, forçado a desligar, reset, programação horária, falha e status de funcionamento;

- Script para controlar as ações de comando forçado a ligar, forçado a desligar, reset, falha e status de funcionamento;

FUNCTION_BLOCK CONTATORATIPO1

VAR_INPUT

bDisjgeral : BOOL;

bSelAutMan: BOOL;

bProgHor: BOOL;

bLigaForçado: BOOL;

bDesLigaForçado: BOOL;

bEstadoFuncionamento: BOOL;

bReset: BOOL;

END_VAR

VAR_IN_OUT

bitsContatora1: WORD;

END_VAR

VAR_OUTPUT

bOUT1: BOOL;

bOUT2: BOOL;

bCMDOLIGADESLIGA: BOOL;

bAlarmContatora: BOOL;

bReleLiga: BOOL;

bReleDesliga: BOOL;

END_VAR

VAR

TON1:TON;

TON1PT: TIME :=T#3s;

TON1IN: BOOL ;

```

TON1ET: TIME;
TON1Q: BOOL;
TON2:TON;
TON2PT: TIME :=T#5s;
TON2IN: BOOL ;
TON2ET: TIME;
TON2Q: BOOL;
TON3:TON;
TON3PT: TIME :=T#6s;
TON3IN: BOOL ;
TON3ET: TIME;
TON3Q: BOOL;
TON4:TON;
TON4PT: TIME :=T#6s;
TON4IN: BOOL ;
TON4ET: TIME;
TON4Q: BOOL;
TPULSE1: TP;
bSetAlarme: BOOL;
bResetAlarme: BOOL;
END_VAR

bitsContatora1.0 := bDisjgeral ;
bitsContatora1.1 :=bSelAutMan ;
bitsContatora1.2 :=bEstadoFuncionamento;
bitsContatora1.3 :=bReleLiga;
bitsContatora1.4 :=bReleDesliga;
bitsContatora1.5 :=bProgHor;
bitsContatora1.6 :=bAlarmContatora ;
bLigaForcado:= bitsContatora1.7 ;
bDesLigaForcado:= bitsContatora1.8 ;
bResetAlarme:= bitsContatora1.9 ;

TPULSE1.PT:= T#200ms;

```



```

IF ( bDisjgeral AND bSelAutMan AND (bProgHor OR bLigaForcado) AND NOT
bAlarmContatora AND NOT bDesLigaForcado ) THEN bOUT1:=TRUE;
ELSE bOUT1 := FALSE;
END_IF
TON1( IN:=bOUT1, PT:=TON1PT, Q=>TON1Q, ET=>TON1ET);
bCMDOLIGADESLIGA:=TON1Q;
IF ( (bSelAutMan) AND ( TON1Q AND NOT bEstadoFuncionamento ) ) THEN
bOUT2 :=TRUE;
ELSE bOUT2 := FALSE;
END_IF
TON2(IN:=bOUT2, PT:=TON2PT, Q=>TON2Q, ET=>TON2ET);
IF (bSetAlarme AND NOT bSelAutMan) OR (bSetAlarme AND bResetAlarme)
THEN TPULSE1.IN :=TRUE;
bSetAlarme:=FALSE;
ELSE TPULSE1.IN:=FALSE;
END_IF
bResetAlarme:=TPULSE1.Q ;
IF bResetAlarme= TRUE THEN bSetAlarme:=FALSE;
END_IF
bResetAlarme:= FALSE;
bAlarmContatora:=bSetAlarme;
IF bCMDOLIGADESLIGA = 1 THEN
bReleLiga := 1;
bReleDesliga:= 0;
END_IF
IF bCMDOLIGADESLIGA = 0 THEN
bReleLiga := 0;
bReleDesliga:= 1;
END_IF
TON3(IN:=bReleLiga, PT:=TON3PT, Q=>TON3Q, ET=>TON3ET);
IF TON3Q THEN
bReleLiga := 0;

```

```

END_IF
TON4(IN:=bReleDesliga, PT:=TON4PT, Q=>TON4Q, ET=>TON4ET);
IF TON4.Q THEN
bReleDesliga := 0;
END_IF

```

- Script programação horaria;

```

FUNCTION_BLOCK ProgHorH
VAR_INPUT
eDataSecurityType0 : E_HVACDataSecurityType;
i0                  : INT;
dtSystemTime0      : DT;
uiOffsetBefore0    : UINT;
bALLDays0: ARRAY [1..10] OF BOOL;
uiOn_hh0: ARRAY [1..10] OF UINT;
uiOn_mm0: ARRAY [1..10] OF UINT;
uiOn_ss0: ARRAY [1..10] OF UINT;
uiOff_hh0: ARRAY [1..10] OF UINT;
uiOff_mm0: ARRAY [1..10] OF UINT;
uiOff_ss0: ARRAY [1..10] OF UINT;
bEnable0: ARRAY [1..10] OF BOOL;
bMonday0: ARRAY [1..10] OF BOOL;
bTuesday0: ARRAY [1..10] OF BOOL;
bWednesday0: ARRAY [1..10] OF BOOL;
bThursday0: ARRAY [1..10] OF BOOL;
bFriday0: ARRAY [1..10] OF BOOL;
bSaturday0: ARRAY [1..10] OF BOOL;
bSunday0: ARRAY [1..10] OF BOOL;

```

bResetAfterOn0: ARRAY [1..10] OF BOOL;

END_VAR

VAR_OUTPUT

bOutput0 : BOOL;

Q0 : ARRAY [1..10] OF BOOL;

tNextOff0 : TIME;

END_VAR

VAR

fbProfHorH0: FB_HVACScheduler28ch;

bInvalidParameter: BOOL;

END_VAR

VAR PERSISTENT

arrTimeChannel0 : ARRAY[1..28] OF ST_HVACTimeChannel;

END_VAR

Sub wCTRL_OnStartRunning()

int wControle

'Cada conjunto de bits corresponde ao dia da semana.

'Habilita programação = 1 bit

'todos os dias = 2 bits

'segunda = 4 bits

'terça = 8 bits

'quarta = 16 bits

'quinta = 32 bits

'sexta = 64 bits

'sabado = 128 bits

'domingo = 256 bits

Startup = True

if Controle <> "" then

wControle = Controle

if wControle >= 512 then

wControle = 511

Controle = 511

```
end if
if wControle >= 256 then
Item("chkDomingo").Value = true
wControle = wControle - 256
else
Item("chkDomingo").Value = false
end if
if wControle >= 128 then
Item("chkSabado").Value = true
wControle = wControle - 128
else
Item("chkSabado").Value = false
end if
if wControle >= 64 then
Item("chkSexta").Value = true
wControle = wControle - 64
else
Item("chkSexta").Value = false
end if
if wControle >= 32 then
Item("chkQuinta").Value = true
wControle = wControle - 32
else
Item("chkQuinta").Value = false
end if
if wControle >= 16 then
Item("chkQuarta").Value = true
wControle = wControle - 16
else
Item("chkQuarta").Value = false
end if
if wControle >= 8 then
Item("chkTerca").Value = true
```

```

wControle = wControle - 8
else
Item("chkTerca").Value = false
end if
if wControle >= 4 then
Item("chkSegunda").Value = true
wControle = wControle - 4
else
Item("chkSegunda").Value = false
end if
if wControle >= 2 then
Item("chkHabilita").Value = true
wControle = wControle - 2
else
Item("chkHabilita").Value = false
end if
if wControle >= 1 then
Item("chkSemanalInteira").Value = true
wControle = wControle - 1
else
Item("chkSemanalInteira").Value = false
end if
end if
Startup = False
End Sub

```

```

Sub btnSalvar_Click()
Dim iCount
Dim iControle
Dim iCountLinha1
Dim iCountLinha2
int iControle
int iCount

```

```
int iCountLinha1
int iCountLinha2
iCountLinha1 = 1
iCountLinha2 = 2
for iCount = 1 to 10
  iControle = 0
  if XPHIllum.Item("wCTRL"&iCount).Habilita = true then
    iControle = iControle + 2
  if XPHIllum.Item("wCTRL"&iCount).TodoDia = true then
    iControle = iControle + 1
  end if
  if XPHIllum.Item("wCTRL"&iCount).Segunda = true then
    iControle = iControle + 4
  end if
  if XPHIllum.Item("wCTRL"&iCount).Terca = true then
    iControle = iControle + 8
  end if
  if XPHIllum.Item("wCTRL"&iCount).Quarta = true then
    iControle = iControle + 16
  end if
  if XPHIllum.Item("wCTRL"&iCount).Quinta = true then
    iControle = iControle + 32
  end if
  if XPHIllum.Item("wCTRL"&iCount).Sexta = true then
    iControle = iControle + 64
  end if
  if XPHIllum.Item("wCTRL"&iCount).Sabado = true then
    iControle = iControle + 128
  end if
  if XPHIllum.Item("wCTRL"&iCount).Domingo = true then
    iControle = iControle + 256
  end if
```

```

Execute          "XPHIllum.Fonte.PH"&iCount&"HoraInicial"          =
abs((Left(XPHIllum.Item("Texto"&iCountLinha1&"").Value,          2)*60)          +
Right(XPHIllum.Item("Texto"&iCountLinha1&"").Value, 2))"
Execute          "XPHIllum.Fonte.PH"&iCount&"HoraFinal"          =
abs((Left(XPHIllum.Item("Texto"&iCountLinha2&"").Value,          2)*60)          +
Right(XPHIllum.Item("Texto"&iCountLinha2&"").Value, 2))"
end if
XPHIllum.Item("wCTRL"&iCount).Controle = iControle
iCountLinha1 = iCountLinha1 + 2
iCountLinha2 = iCountLinha2 + 2
next
End Sub

```

- Subsistema de motobombas – o script é responsável por controlar as ações de ligar, desligar e status de funcionamento.
- Script de comando e monitoramento da motobomba de água pluvial;

PROGRAM CTRL_BOMBAS_A_PLUVIAIS

VAR

dbomba1_dispAPLU: BOOL;

dbomba2_dispAPLU: BOOL;

(* NIVEIS RESERVATORIOS*)

ligarumabombaAPLU: BOOL;

ligarduasbombasAPLU: BOOL;

bombapriAPLU: INT :=1;

desligarbombasAPLU: BOOL;

falhasensoresAPLU: BOOL;

sensoranAPLU: REAL;

```

iAuxDesliga:INT;
tTimer01APLU: TON;
tTimer02APLU: TON;
dbomba1_preCMDAPLU AT %Q*: BOOL;
dbomba2_preCMDAPLU AT %Q*: BOOL;
END_VAR

```

- Script de comando e monitoramento da motobomba de esgoto;

```

PROGRAM CTRL_BOMBAS_ESGOTO
VAR
dbomba1_dispEEE1:BOOL;
dbomba2_dispEEE1:BOOL;
dbomba1_preCMDEEE1: BOOL;
dbomba2_preCMDEEE1: BOOL;
nivelaltoEEE1: BOOL;
nivelmuitoaltoEEE1: BOOL;
nivelbaixoEEE1: BOOL;
ligarumabombaEEE1: BOOL;
ligarduasbombasEEE1: BOOL;
habligarbombasEEE1: BOOL;
bombapriEEE1: INT :=1;
desligarbombasEEE1: BOOL;
varauxEEE1: INT;
falhasensoresEEE1: BOOL;
sensoranEEE1: REAL;
spointbEEE1: REAL;
spointaEEE1: REAL;
spointaaEEE1: REAL;
tTimer01EEE1: TON;
tTimer02EEE1: TON;
END_VAR

```


- Script de comando e monitoramento da motobomba de extravaso;

```
PROGRAM CTRL_BOMBAS_EXT
```

```
VAR
```

```
dbomba1_dispEAPL:BOOL;
```

```
dbomba1_preCMD: BOOL;
```

```
nivelalto: BOOL;
```

```
nivelbaixo: BOOL;
```

```
falhasensores: BOOL;
```

```
sensoran: REAL;
```

```
spointb: REAL;
```

```
spointa: REAL;
```

```
tTimer01: TON;
```

```
END_VAR
```

```
EAPL_nivelalto := nivelalto;
```

```
EAPL_nivelbaixo := nivelbaixo;
```

```
EAPL_falhasensores := falhasensores;
```

```
sensoran := EAPL_sensoran;
```

```
spointb := EAPL_spointb;
```

```
spointa := EAPL_spointa;
```

```
IF (sensoran>spointb) THEN
```

```
  nivelbaixo:=1;
```

```
ELSE
```

```
  nivelbaixo:=0;
```

```
END_IF
```

```
IF (sensoran>spointa) THEN
```

```
  nivelalto:=1;
```

```
ELSE
```

```
  nivelalto:=0;
```

```
END_IF
```

```
IF (NOT dbomba1_FAUEAPL) AND (NOT dbomba1_WAREAPL) THEN
```

```
  dbomba1_dispEAPL:=TRUE ;
```

```
ELSE
```

```

dbomba1_dispEAPL:=FALSE;
END_IF
IF ((nivelalto=TRUE AND nivelbaixo=TRUE) OR (nivelbaixo=1 AND
dbomba1_preCMD=1)) AND dbomba1_dispEAPL=1 THEN
dbomba1_preCMD:=1;
ELSE
dbomba1_preCMD:=0;
END_IF
IF nivelalto=1 AND nivelbaixo=0 THEN
falhasensores:=1;
ELSE
falhasensores:=0;
END_IF
tTimer01 (IN:=dbomba1_preCMD, PT:=T#10s, Q=>dbomba1_CMDEAPL, ET=>);

```

- Script de comando e monitoramento motobomba de drenagem;

```

PROGRAM CommModbus_1
VAR
wStateMachine: WORD;
busy: BOOL;
bError: BOOL;
bBusy: BOOL;
bDone: BOOL;
T1: TON;
udError:modbus_errors;
k,i:WORD;
tTimeStateMachine:TIME:=T#100ms;
Modbus2Channel1: ModbusRtuMaster_KL6x22B;
bResetComModBus: BOOL;
bHabilitaMedicao: ARRAY [1..cNUM_MEDIDORES] OF BOOL;
stErroModbusChannel: ARRAY[1..cNUM_MEDIDORES] OF ST_MODBUS_ERROR;

```

Data1CommModbus_1: ARRAY [0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data2CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data3CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data4CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data5CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data6CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data7CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data8CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data9CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data10CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data11CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data12CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data13CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data14CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data15CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data16CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

Data17CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data18CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data19CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data20CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data21CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data22CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data23CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data24CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data25CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data26CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data27CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data28CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data29CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data30CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)
 Data31CommModbus_1: ARRAY[0..100] OF WORD; (*~(OPC :1 : available for OPC-clients) *)

 dbomba1_RTEEE2: BOOL; (*~(OPC :1 : available for OPC-clients) *)
 dbomba2_RTEEE2: BOOL; (*~(OPC :1 : available for OPC-clients) *)

```

dbomba1_RTEEE1: BOOL; (*~(OPC :1 : available for OPC-clients) *)
dbomba2_RTEEE1: BOOL; (*~(OPC :1 : available for OPC-clients) *)
dbomba1_RTAPLU: BOOL; (*~(OPC :1 : available for OPC-clients) *)
dbomba2_RTAPLU: BOOL; (*~(OPC :1 : available for OPC-clients) *)
dbomba1_RTEAPL: BOOL; (*~(OPC :1 : available for OPC-clients) *)

```

```

fbReadRegisters    : FB_MBReadRegs;
bReadRegisters     : BOOL;
bReadRegistersBusy : BOOL;
bReadRegistersError : BOOL;
nReadRegistersErrorId : UDINT;
nReadRegistersCount : UDINT;
nQuantity          : WORD := 8;
nMBAAddr           : WORD := 0;
arrData2           : ARRAY [1..8] OF UDINT;
y : UDINT;
y2 : UDINT;
n: INT;
conta_erro: INT;
Data1sCommModbus_1: WORD;
TON1B2EEE1: TON;
bQB2EEE1: BOOL;

```

```
END_VAR
```

```
VAR CONSTANT
```

```
cNUM_MEDIDORES: WORD := 29;
```

```
END_VAR
```

```
IF NOT bResetComModBus THEN
```

```
CASE wStateMachine OF
```

```
0:
```

```
bBusy := TRUE;
```

```
bDone:=FALSE;
```

```
Modbus2Channel1.ReadInputRegs(EXECUTE:= FALSE);
```

```

wStateMachine:= 10;
10:
IF NOT bHabilitaMedicao[1] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 1,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 200,
pMemoryAddr:= ADR(Data1CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[1].bError:=TRUE;
stErroModbusChannel[1].sTag:='TESYS-1';
stErroModbusChannel[1].sID:='01';
stErroModbusChannel[1].sIDMaq:='10';
stErroModbusChannel[1].sDescricao:='Erro Multimedidor 1';
stErroModbusChannel[1].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:= 20;
ELSE
wStateMachine:= 20;
END_IF
END_IF
ELSE
wStateMachine:= 20;
END_IF
20:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);

```

```

wStateMachine:=30;
END_IF
30:
IF NOT bHabilitaMedicao[2] THEN
Modbus2Channel1.ReadRegs(
UnitID:=2,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 200,
pMemoryAddr:= ADR(Data2CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[2].bError:=TRUE;
stErroModbusChannel[2].sTag:='TESYS-2';
stErroModbusChannel[2].sID:='02';
stErroModbusChannel[2].sIDMaq:='20';
stErroModbusChannel[2].sDescricao:='Erro Multimedidor 2';
stErroModbusChannel[2].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=40;
ELSE
wStateMachine:= 40;
END_IF
END_IF
ELSE
wStateMachine:= 40;
END_IF
40:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN

```

```

T1(IN:= FALSE);
wStateMachine:=50;
END_IF
50:
IF NOT bHabilitaMedicao[3] THEN
Modbus2Channel1.ReadRegs(
UnitID:=3,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 200,
pMemoryAddr:= ADR(Data3CommModbus_1),
Execute:= TRUE,
Timeout:= t#1s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[3].bError:=TRUE;
stErroModbusChannel[3].sTag:='TESYS-3';
stErroModbusChannel[3].sID:='03';
stErroModbusChannel[3].sIDMaq:='30';
stErroModbusChannel[3].sDescricao:='Erro Multimedidor 3';
stErroModbusChannel[3].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=60;
ELSE
wStateMachine:= 60;
END_IF
END_IF
ELSE
wStateMachine:= 60;
END_IF
60:
T1(IN:= TRUE, PT:= tTimeStateMachine);

```



```

IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=70;
END_IF
70:
IF NOT bHabilitaMedicao[4] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 4,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *) (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data4CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[4].bError:=TRUE;
stErroModbusChannel[4].sTag:='TESYS-4';
stErroModbusChannel[4].sID:='04';
stErroModbusChannel[4].sIDMaq:='40';
stErroModbusChannel[4].sDescricao:='Erro Multimedidor 4';
stErroModbusChannel[4].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=80;
ELSE
wStateMachine:= 80;
END_IF
END_IF
ELSE
wStateMachine:= 80;
END_IF
80:

```

```

T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=90;
END_IF
90:
IF NOT bHabilitaMedicao[5] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 5,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *) (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data5CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
(*Identificação detalhada do Erro, Igual para todos*)
stErroModbusChannel[5].bError:=TRUE;
stErroModbusChannel[5].sTag:='TESYS-5';
stErroModbusChannel[5].sID:='05';
stErroModbusChannel[5].sIDMaq:='50';
stErroModbusChannel[5].sDescricao:='Erro Multimedidor 5';
stErroModbusChannel[5].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=100;
ELSE
wStateMachine:= 100;
END_IF
END_IF
ELSE
wStateMachine:= 100;

```

```

END_IF
100:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=110;
END_IF
110:
IF NOT bHabilitaMedicao[6] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 6,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *) (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data6CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[6].bError:=TRUE;
stErroModbusChannel[6].sTag:='TESYS-6';
stErroModbusChannel[6].sID:='06';
stErroModbusChannel[6].sIDMaq:='60';
stErroModbusChannel[6].sDescricao:='Erro Multimedidor 6';
stErroModbusChannel[6].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=120;
ELSE
wStateMachine:= 120;
END_IF
END_IF
ELSE

```

```

wStateMachine:= 120;
END_IF
120:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=130;
END_IF
130:
IF NOT bHabilitaMedicao[7] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 7,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 250,
pMemoryAddr:= ADR(Data7CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
(*Identificação detalhada do Erro, Igual para todos*)
stErroModbusChannel[7].bError:=TRUE;
stErroModbusChannel[7].sTag:='TESYS-7';
stErroModbusChannel[7].sID:='07';
stErroModbusChannel[7].sIDMaq:='70';
stErroModbusChannel[7].sDescricao:='Erro Multimedidor 7';
stErroModbusChannel[7].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=140;
ELSE
wStateMachine:= 140;
END_IF

```

```

END_IF
ELSE
wStateMachine:=140;
END_IF
140:
IF (NOT bHabilitaMedicao[7] AND Data7CommModbus_1[5].14=0) THEN
Modbus2Channel1.WriteSingleRegister(
UnitID:= 7,
Quantity:= 1,
MBAAddr:= 704,
cbLength:= SIZEOF(wbomba1_CMDAPLU),
pMemoryAddr:= ADR(wbomba1_CMDAPLU),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.WriteSingleRegister(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[7].bError:=TRUE;
stErroModbusChannel[7].sTag:='TESYS-7';
stErroModbusChannel[7].sID:='7';
stErroModbusChannel[7].sIDMaq:='7';
stErroModbusChannel[7].sDescricao:='Erro Multimedidor 7';
stErroModbusChannel[7].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:= 145;
ELSE
wStateMachine:= 145;
END_IF
END_IF
ELSE
wStateMachine:= 145;
END_IF
145:

```

```

T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=170;
END_IF
170:
IF NOT bHabilitaMedicao[9] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 9,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 200,
pMemoryAddr:= ADR(Data9CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[9].bError:=TRUE;
stErroModbusChannel[9].sTag:='TESYS-9';
stErroModbusChannel[9].sID:='09';
stErroModbusChannel[9].sIDMaq:='90';
stErroModbusChannel[9].sDescricao:='Erro Multimedidor 9';
stErroModbusChannel[9].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=180;
ELSE
wStateMachine:= 180;
END_IF
END_IF
ELSE
wStateMachine:= 180;
END_IF

```

```

180:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=190;
END_IF
190:
IF NOT bHabilitaMedicao[10] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 10,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data10CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[10].bError:=TRUE;
stErroModbusChannel[10].sTag:='TESYS-10';
stErroModbusChannel[10].sID:='10';
stErroModbusChannel[10].sIDMaq:='100';
stErroModbusChannel[10].sDescricao:='Erro Multimedidor 10';
stErroModbusChannel[10].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=200;
ELSE
wStateMachine:= 200;
END_IF
END_IF
ELSE
wStateMachine:= 200;

```

```

END_IF
200:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=210;
END_IF
210:
IF NOT bHabilitaMedicao[11] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 11,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data11CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[11].bError:=TRUE;
stErroModbusChannel[11].sTag:='TESYS-11';
stErroModbusChannel[11].sID:='11';
stErroModbusChannel[11].sIDMaq:='110';
stErroModbusChannel[11].sDescricao:='Erro Multimedidor 11';
stErroModbusChannel[11].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=220;
ELSE
wStateMachine:= 220;
END_IF
END_IF
ELSE

```



```

wStateMachine:= 220;
END_IF
220:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=230;
END_IF
230:
IF NOT bHabilitaMedicao[12] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 12,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data12CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[12].bError:=TRUE;
stErroModbusChannel[12].sTag:='TESYS-12';
stErroModbusChannel[12].sID:='12';
stErroModbusChannel[12].sIDMaq:='120';
stErroModbusChannel[12].sDescricao:='Erro Multimedidor 12';
stErroModbusChannel[12].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=240;
ELSE
wStateMachine:= 240;
END_IF
END_IF

```

```

ELSE
wStateMachine:= 240;
END_IF
240:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=250;
END_IF
250:
IF NOT bHabilitaMedicao[13] THEN
Modbus2Channel1.ReadRegs(
UnitID:=13,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 200,
pMemoryAddr:= ADR(Data13CommModbus_1),
Execute:= TRUE,
Timeout:= t#1s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[13].bError:=TRUE;
stErroModbusChannel[13].sTag:='TESYS-13';
stErroModbusChannel[13].sID:='13';
stErroModbusChannel[13].sIDMaq:='130';
stErroModbusChannel[13].sDescricao:='Erro Multimedidor 13';
stErroModbusChannel[13].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=260;
ELSE
wStateMachine:= 260;
END_IF

```

```

END_IF
ELSE
wStateMachine:= 260;
END_IF
260:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=270;
END_IF
270:
IF NOT bHabilitaMedicao[14] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 14,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 200,
pMemoryAddr:= ADR(Data14CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[14].bError:=TRUE;
stErroModbusChannel[14].sTag:='TESYS-14';
stErroModbusChannel[14].sID:='14';
stErroModbusChannel[14].sIDMaq:='140';
stErroModbusChannel[14].sDescricao:='Erro Multimedidor 14';
stErroModbusChannel[14].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=280;
ELSE
wStateMachine:= 280;

```

```

END_IF
END_IF
ELSE
wStateMachine:= 280;
END_IF
280:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=290;
END_IF
290:
IF NOT bHabilitaMedicao[15] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 15,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data15CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[15].bError:=TRUE;
stErroModbusChannel[15].sTag:='TESYS-15';
stErroModbusChannel[15].sID:='15';
stErroModbusChannel[15].sIDMq:='150';
stErroModbusChannel[15].sDescricao:='Erro Multimedidor 15';
stErroModbusChannel[15].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=300;
ELSE

```

```

wStateMachine:= 300;
END_IF
END_IF
ELSE
wStateMachine:= 300;
END_IF
300:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=310;
END_IF
310:
IF NOT bHabilitaMedicao[16] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 16,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data16CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[16].bError:=TRUE;
stErroModbusChannel[16].sTag:='TESYS-16';
stErroModbusChannel[16].sID:='16';
stErroModbusChannel[16].sIDMaq:='160';
stErroModbusChannel[16].sDescricao:='Erro Multimedidor 16';
stErroModbusChannel[16].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=320;

```

```

ELSE
wStateMachine:= 320;
END_IF
END_IF
ELSE
wStateMachine:= 320;
END_IF
320:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=330;
END_IF
330:
IF NOT bHabilitaMedicao[17] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 17,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data17CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[17].bError:=TRUE;
stErroModbusChannel[17].sTag:='TESYS-17';
stErroModbusChannel[17].sID:='17';
stErroModbusChannel[17].sIDMaq:='170';
stErroModbusChannel[17].sDescricao:='Erro Multimedidor 17';
stErroModbusChannel[17].eCodModbus:=Modbus2Channel1.ErrorId;

```

```

wStateMachine:=340;
ELSE
wStateMachine:= 340;
END_IF
END_IF
ELSE
wStateMachine:=340;
END_IF
340:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=350;
END_IF
350:
IF NOT bHabilitaMedicao[18] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 18,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data18CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[18].bError:=TRUE;
stErroModbusChannel[18].sTag:='TESYS-18';
stErroModbusChannel[18].sID:='18';
stErroModbusChannel[18].sIDMaq:='180';
stErroModbusChannel[18].sDescricao:='Erro Multimedidor 18';

```

```

stErroModbusChannel[18].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=360;
ELSE
wStateMachine:= 360;
END_IF
END_IF
ELSE
wStateMachine:= 360;
END_IF
360:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=370;
END_IF
370:
IF NOT bHabilitaMedicao[19] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 19,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 200,
pMemoryAddr:= ADR(Data19CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[19].bError:=TRUE;
stErroModbusChannel[19].sTag:='TESYS-19';
stErroModbusChannel[19].sID:='19';
stErroModbusChannel[19].sIDMaq:='190';

```



```

stErroModbusChannel[19].sDescricao:='Erro Multimedidor 19';
stErroModbusChannel[19].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=380;
ELSE
wStateMachine:=380;
END_IF
END_IF
ELSE
wStateMachine:= 380;
END_IF
380:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=390;
END_IF
390:
IF NOT bHabilitaMedicao[20] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 20,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data20CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[20].bError:=TRUE;
stErroModbusChannel[20].sTag:='TESYS-20';
stErroModbusChannel[20].sID:='20';

```

```

stErroModbusChannel[20].sIDMaq:='200';
stErroModbusChannel[20].sDescricao:='Erro Multimedidor 20';
stErroModbusChannel[20].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=400;
ELSE
wStateMachine:= 400;
END_IF
END_IF
ELSE
wStateMachine:= 400;
END_IF
400:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=410;
END_IF
410:
IF NOT bHabilitaMedicao[22] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 22,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= SIZEOF(Data22CommModbus_1),
pMemoryAddr:= ADR(Data22CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
(*Identificação detalhada do Erro, Igual para todos*)
stErroModbusChannel[22].bError:=TRUE;

```

```

stErroModbusChannel[22].sTag:='TESYS-22';
stErroModbusChannel[22].sID:='22';
stErroModbusChannel[22].sIDMaq:='220';
stErroModbusChannel[22].sDescricao:='Erro Multimedidor 22';
stErroModbusChannel[22].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=415;
ELSE
wStateMachine:= 415;
END_IF
END_IF
ELSE
wStateMachine:= 415;
END_IF
415:
IF (NOT bHabilitaMedicao[22] AND Data22CommModbus_1[5].14=0) THEN
Modbus2Channel1.WriteSingleRegister(
UnitID:= 22,
Quantity:= 1,
MBAAddr:= 704,
cbLength:= SIZEOF(wbomba1_CMDEEEE2),
pMemoryAddr:= ADR(wbomba1_CMDEEEE2),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.WriteSingleRegister(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[22].bError:=TRUE;
stErroModbusChannel[22].sTag:='TESYS-22';
stErroModbusChannel[22].sID:='22';
stErroModbusChannel[22].sIDMaq:='22';
stErroModbusChannel[22].sDescricao:='Erro Multimedidor 1';
stErroModbusChannel[22].eCodModbus:=Modbus2Channel1.ErrorId;

```

```

wStateMachine:= 425;
ELSE
wStateMachine:= 425;
END_IF
END_IF
ELSE
wStateMachine:= 425;
END_IF
425: T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=430;
END_IF
430:
IF NOT bHabilitaMedicao[21] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 21,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data21CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[21].bError:=TRUE;
stErroModbusChannel[21].sTag:='TESYS-21';
stErroModbusChannel[21].sID:='21';
stErroModbusChannel[21].sIDMaq:='210';
stErroModbusChannel[21].sDescricao:='Erro Multimedidor 21';
stErroModbusChannel[21].eCodModbus:=Modbus2Channel1.ErrorId;

```

```

wStateMachine:=435;
ELSE
wStateMachine:= 435;
END_IF
END_IF
ELSE
wStateMachine:= 435;
END_IF
435:
IF (NOT bHabilitaMedicao[21] AND Data21CommModbus_1[5].14=0) THEN
Modbus2Channel1.WriteSingleRegister(
UnitID:= 21,
Quantity:= 1,
MBAAddr:= 704,
cbLength:= SIZEOF(wbomba2_CMDEEEE2),
pMemoryAddr:= ADR(wbomba2_CMDEEEE2),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.WriteSingleRegister(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[21].bError:=TRUE;
stErroModbusChannel[21].sTag:='TESYS-21';
stErroModbusChannel[21].sID:='21';
stErroModbusChannel[21].sIDMaq:='210';
stErroModbusChannel[21].sDescricao:='Erro Multimedidor 1';
stErroModbusChannel[21].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:= 440;
ELSE
wStateMachine:= 440;
END_IF
END_IF

```

```

ELSE
wStateMachine:= 440;
END_IF
440:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=450;
END_IF
450:
IF NOT bHabilitaMedicao[23] THEN
Modbus2Channel1.ReadRegs(
UnitID:=23,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data23CommModbus_1),
Execute:= TRUE,
Timeout:= t#1s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[23].bError:=TRUE;
stErroModbusChannel[23].sTag:='TESYS-23';
stErroModbusChannel[23].sID:='23';
stErroModbusChannel[23].sIDMaq:='230';
stErroModbusChannel[23].sDescricao:='Erro Multimedidor 23';
stErroModbusChannel[23].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=460;
ELSE
wStateMachine:= 460;
END_IF

```

```

END_IF
ELSE
wStateMachine:= 460;
END_IF
460:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=470;
END_IF
470:
IF NOT bHabilitaMedicao[24] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 24,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data24CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[24].bError:=TRUE;
stErroModbusChannel[24].sTag:='TESYS-24';
stErroModbusChannel[24].sID:='24';
stErroModbusChannel[24].sIDMaq:='240';
stErroModbusChannel[24].sDescricao:='Erro Multimedidor 24';
stErroModbusChannel[24].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=480;
ELSE
wStateMachine:= 480;

```

```

END_IF
END_IF
ELSE
wStateMachine:= 480;
END_IF
480:
IF (NOT bHabilitaMedicao[24] AND Data24CommModbus_1[5].14=0) THEN
Modbus2Channel1.WriteSingleRegister(
UnitID:= 24,
Quantity:= 1,
MBAAddr:= 704,
cbLength:= SIZEOF(wbomba1_CMDEAPL),
pMemoryAddr:= ADR(wbomba1_CMDEAPL),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.WriteSingleRegister(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[24].bError:=TRUE;
stErroModbusChannel[24].sTag:='TESYS-24';
stErroModbusChannel[24].sID:='24';
stErroModbusChannel[24].sIDMaq:='24';
stErroModbusChannel[24].sDescricao:='Erro Multimedidor 24';
stErroModbusChannel[24].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:= 485;
ELSE
wStateMachine:= 485;
END_IF
END_IF
ELSE
wStateMachine:= 485;
END_IF

```



```

485:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=490;
END_IF
490:
IF NOT bHabilitaMedicao[25] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 25,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 200,
pMemoryAddr:= ADR(Data25CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[25].bError:=TRUE;
stErroModbusChannel[25].sTag:='TESYS-25';
stErroModbusChannel[25].sID:='25';
stErroModbusChannel[25].sIDMaq:='250';
stErroModbusChannel[25].sDescricao:='Erro Multimedidor 25';
stErroModbusChannel[25].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=495;
ELSE
wStateMachine:= 495;
END_IF
END_IF
ELSE
wStateMachine:= 495;

```

```

END_IF
495:
IF (NOT bHabilitaMedicao[25] AND Data25CommModbus_1[5].14=0) THEN
Modbus2Channel1.WriteSingleRegister(
UnitID:= 25,
Quantity:= 1,
MBAAddr:= 704,
cbLength:= SIZEOF(wbomba2_CMDAPLU),
pMemoryAddr:= ADR(wbomba2_CMDAPLU),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.WriteSingleRegister(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[25].bError:=TRUE;
stErroModbusChannel[25].sTag:='TESYS-25';
stErroModbusChannel[25].sID:='25';
stErroModbusChannel[25].sIDMaq:='25';
stErroModbusChannel[25].sDescricao:='Erro Multimedidor 25';
stErroModbusChannel[25].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:= 500;
ELSE
wStateMachine:= 500;
END_IF
END_IF
ELSE
wStateMachine:= 500;
END_IF
500:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);

```

```

wStateMachine:=510;
END_IF
510:
IF NOT bHabilitaMedicao[26] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 26,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 200,
pMemoryAddr:= ADR(Data26CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[26].bError:=TRUE;
stErroModbusChannel[26].sTag:='TESYS-26';
stErroModbusChannel[26].sID:='26';
stErroModbusChannel[26].sIDMaq:='260';
stErroModbusChannel[26].sDescricao:='Erro Multimedidor 26';
stErroModbusChannel[26].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=520;
ELSE
wStateMachine:= 520;
END_IF
END_IF
ELSE
wStateMachine:= 520;
END_IF
520:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN

```

```

T1(IN:= FALSE);
wStateMachine:=530;
END_IF
530:
IF NOT bHabilitaMedicao[27] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 27,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 200,
pMemoryAddr:= ADR(Data27CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[27].bError:=TRUE;
stErroModbusChannel[27].sTag:='TESYS-27';
stErroModbusChannel[27].sID:='27';
stErroModbusChannel[27].sIDMaq:='270';
stErroModbusChannel[27].sDescricao:='Erro Multimedidor 27';
stErroModbusChannel[27].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=540;
ELSE
wStateMachine:=540;
END_IF
END_IF
ELSE
wStateMachine:=540;
END_IF
540:
T1(IN:= TRUE, PT:= tTimeStateMachine);

```

```

IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=550;
END_IF
550:
IF NOT bHabilitaMedicao[28] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 28,
Quantity:= 100,
MBAAddr:= 450,
cbLength:= 200,
pMemoryAddr:= ADR(Data28CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[28].bError:=TRUE;
stErroModbusChannel[28].sTag:='TESYS-28';
stErroModbusChannel[28].sID:='28';
stErroModbusChannel[28].sIDMaq:='28';
stErroModbusChannel[28].sDescricao:='Erro Multimedidor 28';
stErroModbusChannel[28].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=560;
ELSE
wStateMachine:=560;
END_IF
END_IF
ELSE
wStateMachine:=560;
END_IF
560:

```

```

IF (NOT bHabilitaMedicao[28] AND Data28CommModbus_1[5].14=0) THEN
Modbus2Channel1.WriteSingleRegister(
UnitID:= 28,
Quantity:= 1,
MBAAddr:= 704,
cbLength:= SIZEOF(wbomba1_CMDEEE1),
pMemoryAddr:= ADR(wbomba1_CMDEEE1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.WriteSingleRegister(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[28].bError:=TRUE;
stErroModbusChannel[28].sTag:='TESYS-28';
stErroModbusChannel[28].sID:='28';
stErroModbusChannel[28].sIDMaq:='28';
stErroModbusChannel[28].sDescricao:='Erro Multimedidor 28';
stErroModbusChannel[28].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:= 565;
ELSE
wStateMachine:= 565;
END_IF
END_IF
ELSE
wStateMachine:= 565;
END_IF
565:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=570;
END_IF

```

```

IF NOT bHabilitaMedicao[29] THEN
Modbus2Channel1.ReadRegs(
UnitID:= 29,
Quantity:= 100,
MBAAddr:= 450, (*era MBAAddr:= 3019, *)
cbLength:= 200,
pMemoryAddr:= ADR(Data29CommModbus_1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.ReadRegs(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[29].bError:=TRUE;
stErroModbusChannel[29].sTag:='TESYS-29';
stErroModbusChannel[29].sID:='29';
stErroModbusChannel[29].sIDMaq:='29';
stErroModbusChannel[29].sDescricao:='Erro Multimodidor 29';
stErroModbusChannel[29].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:=575;
ELSE
wStateMachine:=575;
END_IF
END_IF
ELSE
wStateMachine:=575;
END_IF
575:
IF (NOT bHabilitaMedicao[29] AND Data29CommModbus_1[5].14=0) THEN
Modbus2Channel1.WriteSingleRegister(
UnitID:= 29,
Quantity:= 1,
MBAAddr:= 704,

```

```

cbLength:= SIZEOF(wbomba2_CMDEEE1),
pMemoryAddr:= ADR(wbomba2_CMDEEE1),
Execute:= TRUE,
Timeout:= t#10s,
Busy => busy);
IF NOT busy THEN
Modbus2Channel1.WriteSingleRegister(Execute:= FALSE);
IF Modbus2Channel1.Error THEN
stErroModbusChannel[29].bError:=TRUE;
stErroModbusChannel[29].sTag:='TESYS-29';
stErroModbusChannel[29].sID:='29';
stErroModbusChannel[29].sIDMaq:='29';
stErroModbusChannel[29].sDescricao:='Erro Multimedidor 29';
stErroModbusChannel[29].eCodModbus:=Modbus2Channel1.ErrorId;
wStateMachine:= 999;
ELSE
wStateMachine:= 999;
END_IF
END_IF
ELSE
wStateMachine:= 999;
END_IF
999:
T1(IN:= TRUE, PT:= tTimeStateMachine);
IF (T1.Q) THEN
T1(IN:= FALSE);
wStateMachine:=0;
conta_erro:=conta_erro+1;
END_IF
END_CASE
ELSE
bDone := FALSE;
bBusy := FALSE;

```



```

bError := FALSE;
udError:= 0;
wStateMachine:=0;

FOR i:=1 TO cNUM_MEDIDORES DO
stErroModbusChannel[i].bError:=FALSE;
stErroModbusChannel[i].sTag:='NO ERROR';
stErroModbusChannel[i].sID:='NO ERROR';
stErroModbusChannel[i].sIDMaq:='NO ERROR';
stErroModbusChannel[i].sDescricao:='NO ERROR';
stErroModbusChannel[i].eCodModbus:=0;
END_FOR
END_IF
IF n<99THEN
n:=n+1;
ELSE
n:=0;
END_IF
wbomba1_CMDAPLU.0 := dbomba1_CMDAPLU ;
IF      (dbomba1_FAUAPLU=1      OR      dbomba1_WARAPLU=1)      AND
dbomba1_RSTAUTAPLU=1 AND dbomba1_RTAPLU=1
THEN wbomba1_CMDAPLU.3 := TRUE ;
ELSE wbomba1_CMDAPLU.3 := FALSE ;
END_IF
dbomba1_FAUAPLU := Data7CommModbus_1[5].2 ; (* 5 = REG 455 *)
dbomba1_WARAPLU := Data7CommModbus_1[5].3 ;
dbomba1_READYAPLU := Data7CommModbus_1[5].0 ;
dbomba1_SYSONAPLU := Data7CommModbus_1[5].1 ;
dbomba1_RSTAUTAPLU := Data7CommModbus_1[5].5 ;
dbomba1_RUNAPLU := Data7CommModbus_1[5].7 ;
wbomba2_CMDAPLU.0 := dbomba2_CMDAPLU ;
IF      (dbomba2_FAUAPLU=1      OR      dbomba2_WARAPLU=1)      AND
dbomba2_RSTAUTAPLU=1 AND dbomba2_RTAPLU=1

```

```

THEN wbomba2_CMDAPLU.3 := TRUE ;
ELSE wbomba2_CMDAPLU.3 := FALSE ;
END_IF
dbomba2_FAUAPLU :=Data25CommModbus_1[5].2 ;
dbomba2_WARAPLU :=Data25CommModbus_1[5].3 ;
dbomba2_READYAPLU :=Data25CommModbus_1[5].0 ;
dbomba2_SYSONAPLU :=Data25CommModbus_1[5].1 ;
dbomba2_RSTAUTAPLU :=Data25CommModbus_1[5].5 ;
dbomba2_RUNAPLU :=Data25CommModbus_1[5].7 ;
wbomba1_CMDEEE1.0 := dbomba1_CMDEEE1 ;
IF      (dbomba1_FAUEEE1=1      OR      dbomba1_WAREEE1=1)      AND
dbomba1_RSTAUTEEE1=1 AND dbomba1_RTEEE1=1
THEN wbomba1_CMDEEE1.3 := TRUE ;
ELSE wbomba1_CMDEEE1.3 := FALSE ;
END_IF
dbomba1_FAUEEE1 := Data28CommModbus_1[5].2 ; (* 5 = REG 455 *)
dbomba1_WAREEE1 := Data28CommModbus_1[5].3 ;
dbomba1_READYEEE1 := Data28CommModbus_1[5].0 ;
dbomba1_SYSONEEE1 := Data28CommModbus_1[5].1 ;
dbomba1_RSTAUTEEE1 := Data28CommModbus_1[5].5 ;
dbomba1_RUNEEE1 := Data28CommModbus_1[5].7 ;
wbomba2_CMDEEE1.0 := dbomba2_CMDEEE1 ;
IF      (dbomba2_FAUEEE1=1      OR      dbomba2_WAREEE1=1)      AND
dbomba2_RSTAUTEEE1=1 AND dbomba2_RTEEE1=1
THEN wbomba2_CMDEEE1.3 := TRUE ;
ELSE wbomba2_CMDEEE1.3 := FALSE ;
END_IF
dbomba2_FAUEEE1 :=Data29CommModbus_1[5].2 ;
dbomba2_WAREEE1 :=Data29CommModbus_1[5].3 ;
dbomba2_READYEEE1 :=Data29CommModbus_1[5].0 ;
dbomba2_SYSONEEE1 :=Data29CommModbus_1[5].1 ;
dbomba2_RSTAUTEEE1 :=Data29CommModbus_1[5].5 ;
dbomba2_RUNEEE1 :=Data29CommModbus_1[5].7 ;

```

```

wbomba1_CMDEEEE2.0 := dbomba1_CMDEEEE2 ;
IF      (dbomba1_FAUEEEE2=1      OR      dbomba1_WAREEEE2=1)      AND
dbomba1_RSTAUTEEE2=1 AND dbomba1_RTEEEE2=1
THEN wbomba1_CMDEEEE2.3 := TRUE ;
ELSE wbomba1_CMDEEEE2.3 := FALSE ;
END_IF

dbomba1_FAUEEEE2 := Data22CommModbus_1[5].2 ; (* 5 = REG 455 *)
dbomba1_WAREEEE2 := Data22CommModbus_1[5].3 ;
dbomba1_READYEEE2 := Data22CommModbus_1[5].0 ;
dbomba1_SYSONEEE2 := Data22CommModbus_1[5].1 ;
dbomba1_RSTAUTEEE2 := Data22CommModbus_1[5].5 ;
dbomba1_RUNEEE2 := Data22CommModbus_1[5].7 ;
wbomba2_CMDEEEE2.0 := dbomba2_CMDEEEE2 ;
IF      (dbomba2_FAUEEEE2=1      OR      dbomba2_WAREEEE2=1)      AND
dbomba2_RSTAUTEEE2=1 AND dbomba2_RTEEEE2=1
THEN wbomba2_CMDEEEE2.3 := TRUE ;
ELSE wbomba2_CMDEEEE2.3 := FALSE ;
END_IF

dbomba2_FAUEEEE2 :=Data21CommModbus_1[5].2 ;
dbomba2_WAREEEE2 :=Data21CommModbus_1[5].3 ;
dbomba2_READYEEE2 :=Data21CommModbus_1[5].0 ;
dbomba2_SYSONEEE2 :=Data21CommModbus_1[5].1 ;
dbomba2_RSTAUTEEE2 :=Data21CommModbus_1[5].5 ;
dbomba2_RUNEEE2 :=Data21CommModbus_1[5].7 ;
wbomba1_CMDEAPL.0 := dbomba1_CMDEAPL ;
IF      (dbomba1_FAUEAPL=1      OR      dbomba1_WAREAPL=1)      AND
dbomba1_RSTAUTEAPL=1 AND dbomba1_RTEAPL=1
THEN wbomba1_CMDEAPL.3 := TRUE ;
ELSE wbomba1_CMDEAPL.3 := FALSE ;
END_IF

dbomba1_FAUEAPL := Data24CommModbus_1[5].2 ; (* 5 = REG 455 *)
dbomba1_WAREAPL := Data24CommModbus_1[5].3 ;
dbomba1_READYEAPL := Data24CommModbus_1[5].0 ;

```

```
dbomba1_SYSONEAPL := Data24CommModbus_1[5].1 ;  
dbomba1_RSTAUTEAPL := Data24CommModbus_1[5].5 ;  
dbomba1_RUNEAPL := Data24CommModbus_1[5].7 ;
```

ANEXO II

EQUIPAMENTOS UTILIZADOS PARA A PESQUISA APLICADA

Foram utilizados os seguintes equipamentos nessa pesquisa:

- 1- Elipse E3 – Studio;
- 2- Lutron – Sistema de dimmer;
- 3- Cogito – Averics – sistema de controle de acesso;
- 4- Mobitix – sistema de Câmera de segurança;
- 5- Mobitix-mx – câmera de segurança;
- 6- Twincat 2;
- 7- Banco de dados – SQL server 2012;
- 8- Controladora de acesso;
- 9- PC-CX5020
- 10- Cartão de saída analógico;
- 11- Cartão de entrada analógica;
- 12- Cartão de entrada digital;
- 13- Cartão de saída digital;
- 14- Transmissor de vazão;
- 15- Relé tesys;
- 16- Relé falta de fase;
- 17- Relé termomagnético;
- 18- Medidor de energia;
- 19- Medidor de qualidade;
- 20- Disjuntor;
- 21- Contator;
- 22- Relé;
- 23- Contator auxiliar;
- 24- Controlador dimmer;
- 25- Sensor de presença;
- 26- Sensor ultrassônico;
- 27- Transmissor de temperatura;

- 28- Transmissor de vazão;
- 29- Sensor de temperatura – pt1000;
- 30- Sensor de temperatura – pt100;
- 31- Transmissor de pressão;
- 32- Central de incêndio – edward;
- 33- Acionador manual – edward;
- 34- Detector de fumaça edward;
- 35- Detecto de temperatura – edward;
- 36- Motobomba centrífuga;
- 37- Motobomba;
- 38- Servidor Windows 2000;
- 39- Atuadores;
- 40- Leitor de cartão RFID;
- 41- UPS;
- 42- Switch gerenciável;
- 43- OPC server;
- 44- OPC Client;
- 45- Protocolo OPC A&E;
- 46- Protocolo OPC DA;
- 47- Protocolo Modbus;
- 48- Gateway modbus TCP/IP.