

## Desenvolvimento de um sistema supervisório para planta didática de geração de potência

Development of a supervisory system for didactic power generation plant

Cristiano Garcia Rosa,<sup>\*</sup> Joubert Vieira Santos,<sup>†</sup> e Romeu Abrahão Pereira<sup>‡</sup>

*Centro Brasileiro de Pesquisas Físicas – CBPF, Rua Dr. Xavier Sigaud, 150 – Urca – Rio de Janeiro – RJ - Brasil*

Alexandre Mello<sup>§</sup>

*Centro Brasileiro de Pesquisas Físicas – CBPF, Rua Dr. Xavier Sigaud,  
150 – Urca – Rio de Janeiro – RJ - Brasil*

*Submetido: 30/07/2018 Aceito: 09/10/2019*

**Resumo:** Este trabalho apresenta o desenvolvimento de um sistema supervisório para operação e monitoramento de um protótipo de geração de potência usando o *software* Indusoft Web Studio. Trata-se de uma planta desenvolvida para fins didáticos baseada nos ciclos de geração Brayton-Rankine para o qual o controle de temperatura e pressão em determinados pontos do processo é primordial para que se tenha um desempenho mais próximo do projetado. Neste projeto, placas microcontroladas Arduino Uno são utilizadas no processamento dos sinais fornecidos por sensores instalados na planta. Em seguida, as informações são enviadas para a tela do supervisório, local onde o operador do equipamento realiza o monitoramento das variáveis e também executa comandos para o acionamento dos atuadores existentes no protótipo.

**Palavras-chave:** Geração de potência, supervisório, comunicação Modbus.

**Abstract:** This work presents the development of a supervisory system for the operation and monitoring of a power generation prototype using Indusoft Web Studio software. This prototype is a plant developed for didactic purposes based on the Brayton-Rankine generation cycles for which the control of temperature and pressure at certain points of the process is paramount so that its performance can be as close as possible to the designed one. In this project, Arduino Uno microcontrollable boards are used in the processing of the signals provided by the sensors installed in the plant and then this information is sent to the supervisor screen where the operator of the equipment performs the monitoring of the variables and also executes commands for the activation of the actuators present in the prototype.

**Keywords:** Power Generation, supervisory, Modbus communication.

### 1. INTRODUÇÃO

Torna-se mais constante a cada dia o avanço na pesquisa visando o desenvolvimento de novas técnicas para produção de energias renováveis, inclusive empregando tecnologias já disponíveis combinadas a técnicas clássicas de geração. Esta combinação resulta em processos mais eficientes e dispositivos mais seguros durante seu funcionamento.

A cogeração possibilita a produção conjunta de formas diferentes de energia útil, como a energia elétrica, mecânica e a térmica, a partir de uma mesma fonte de energia primária, tal como os derivados de petróleo, o gás natural, o carvão ou a biomassa (madeira, bagaço de cana de açúcar, etc.) [1].

A caldeira é um dos equipamentos mais comuns de uso e é extremamente útil. Porém, seu emprego requer várias medi-

das de segurança técnica e operacional [2]. A instrumentação e o controle automatizado desse sistema é essencial, pois permite uma operação segura, econômica e confiável [3].

O sistema supervisório descrito na presente Nota Técnica foi projetado para monitorar e operar um protótipo de geração de potência desenvolvido para fins didáticos, com o intuito de funcionar seguindo o princípio do ciclo térmico combinado Brayton-Rankine. No arranjo combinado Brayton-Rankine o calor liberado pelo ciclo de Brayton é usado parcial ou totalmente como entrada de calor para o ciclo de Rankine. No caso da geração de energia elétrica, os gases de exaustão da turbina do ciclo Brayton estão numa temperatura relativamente alta, condição que permite seu emprego na geração de vapor que alimenta o ciclo de Rankine [4]. As figuras 1 e 2 mostram o protótipo construído e seus detalhes internos. O combustível para sua operação é o GLP (gás liquefeito de petróleo). Por se misturar mais facilmente com o oxigênio, sua queima é rápida e eficiente e gera menor quantidade de resíduos (fuligem). Seus constituintes mais importantes são: propano (C<sub>3</sub>H<sub>8</sub>), propeno (C<sub>3</sub>H<sub>6</sub>), isobutano (C<sub>4</sub>H<sub>10</sub>), n-butano (C<sub>4</sub>H<sub>10</sub>) e buteno (C<sub>4</sub>H<sub>8</sub>) [5].

<sup>\*</sup>Electronic address: cristiano\_uberaba@yahoo.com

<sup>†</sup>Electronic address: joubertvs@gmail.com

<sup>‡</sup>Electronic address: romeuabraham@yahoo.com.br

<sup>§</sup>Electronic address: mello@cbpf.br

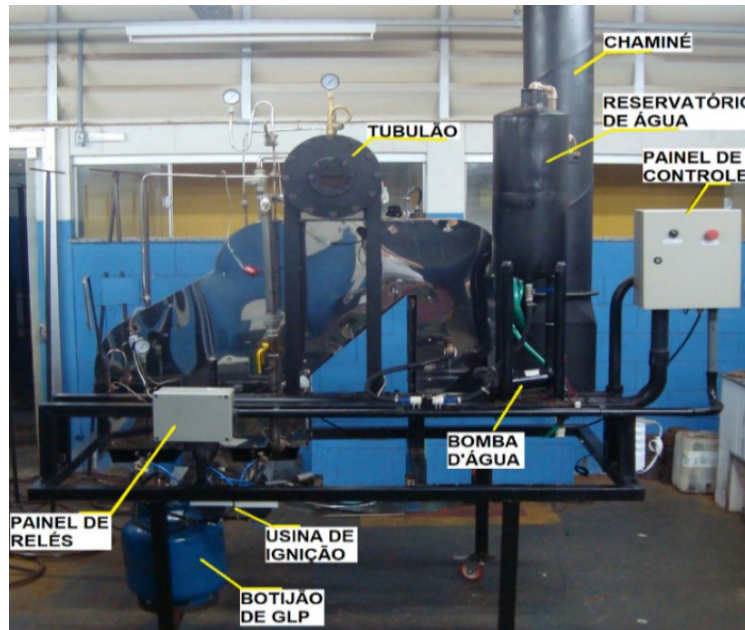


Figure 1: Protótipo de geração de potência.

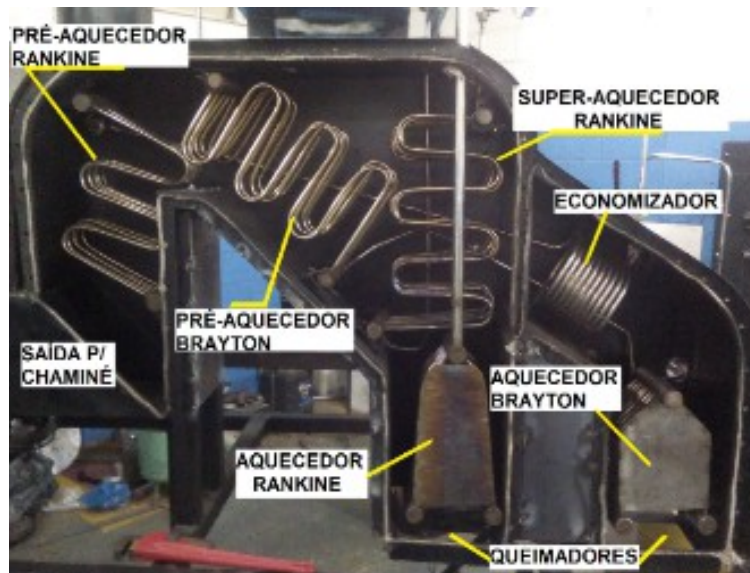


Figure 2: Parte interna do protótipo.

A figura 3 mostra o arranjo combinado entre os ciclos Brayton-Rankine. No ciclo Brayton, o ar atmosférico é continuamente sugado pelo compressor e comprimido até altas pressões. O ar comprimido entra na câmara de combustão, onde é misturado ao combustível criando condições para que ocorra a combustão, que tem como resultado uma violenta expansão dos gases devido às altas temperaturas atingidas durante o processo. Os gases provenientes da combustão se expandem através da turbina, acionando um gerador elétrico. Os gases liberados pela exaustão da turbina do ciclo Brayton ainda apresentam temperaturas relativamente altas, condição que permite sua utilização para troca de calor no ciclo Rankine. No ciclo Rankine, utiliza-se o calor proveniente da queima de combustíveis para geração de vapor, que pode

ser utilizado na geração de energia elétrica em um gerador acionado por uma turbina a vapor [6].

O desempenho eficiente da planta didática de geração de potência depende de um controle eficaz dos valores de temperatura e pressão em determinados estágios de cada ciclo térmico. Portanto, torna-se primordial a instalação de sensores nesses pontos e a realização de seu monitoramento. Considerando os instrumentos que trabalham conectados ao supervisor, para o presente projeto, estipulou-se a utilização de 12 sensores para medição de temperatura e 1 sensor para medição de pressão. A tabela 1 mostra os pontos selecionados onde os 12 sensores de temperatura e o sensor de pressão foram instalados na planta.

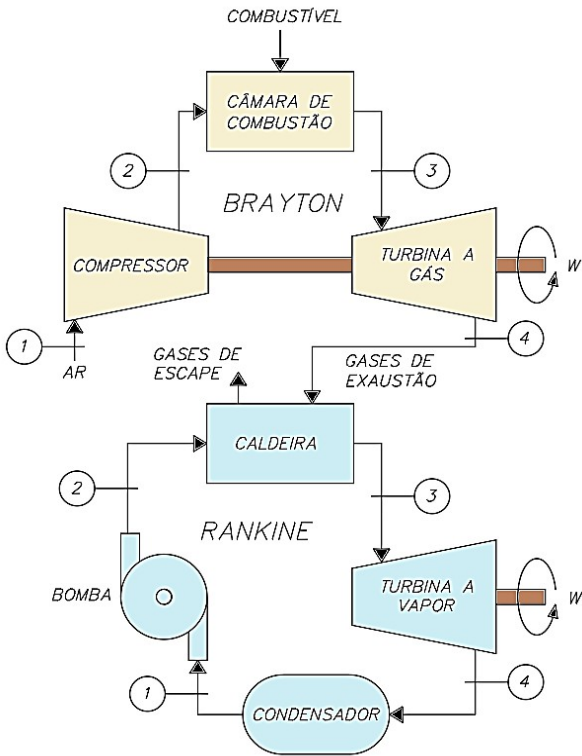


Figure 3: Ciclo combinado Brayton-Rankine.

TABELA 1 – Pontos de instalação de sensores de temperatura e pressão.

Sensores de Temperatura	
Ciclo Brayton	Ar ambiente
	Ar comprimido
	Saída do Brayton
Ciclo Rankine	Água ambiente
	Saída do economizador
	Entrada do aquecedor
	Saída do aquecedor
	Entrada do superaquecedor
	Saída do superaquecedor
Gases de Combustão	Área do Brayton
	Área do Rankine
	Área da Chaminé
Sensor de Pressão	
Ciclo Brayton	Ar comprimido

Justifica-se a escolha de se utilizar somente um sensor de pressão pelo fato de que apenas nesse ponto do processo (ar comprimido no ciclo Brayton) a temperatura é tal que permite a instalação deste modelo de sensor que, por sua vez, possui baixo custo de aquisição. Porém, deve-se ressaltar que manômetros analógicos foram utilizados nos

demais estágios. No Anexo A é mostrado o fluxograma de processo do protótipo detalhando a localização dos instrumentos sensores.

2. SISTEMA SUPERVISÓRIO

Atualmente, os sistemas de automação industrial fazem uso de tecnologias de computação e comunicação para facilitar o monitoramento e controle dos processos industriais. Estes sistemas realizam a coleta de dados em ambientes complexos, geralmente distantes, apresentando-os de modo amigável e objetivo ao operador, com elaborados recursos gráficos e conteúdos multimídias, facilitando, assim, a tomada de decisão. Um sistema supervisório, conhecido no ambiente industrial como SCADA (*Supervisory Control and Data Aquisition* – Controle Supervisório e Aquisição de Dados), pode ser definido como uma interface homem-máquina (IHM) amigável, permitindo simultaneamente: supervisionar e/ou controlar sistemas automatizados, alterar valores das variáveis, comandar os dispositivos físicos conectados ao sistema, reconhecer alarmes, acompanhar gráficos de tendência das variáveis e emitir relatórios [7]. A figura 4 mostra um fluxograma exemplificando o funcionamento de um sistema supervisório.

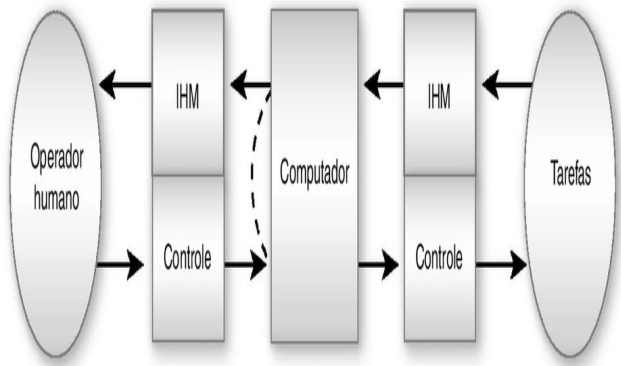


Figure 4: Fluxograma de sistema de supervisão e controle industrial. Rosário (2005).

Normalmente, tais aplicativos contam com dois modos básicos de utilização: o modo *Run Time*, que empacota os arquivos gerando um programa executável para ser utilizado num computador, e o modo de desenvolvimento, que vem a ser um ambiente que permite criar telas gráficas e utilizar as janelas, nas quais se elaboram os *layouts* do sistema monitorado e controlado, com a geração de aplicações com animação gráfica (denominadas aplicações dinâmicas).

2.1. Comunicação Modbus

O protocolo Modbus, inicialmente desenvolvido em 1979 pela Modicon, é uma estrutura de mensagem utilizada por vários fabricantes em diversos equipamentos. É um protocolo da camada de aplicação para comunicação entre dispo-

sitivos, principalmente utilizado em sistemas de automação industrial baseado em um modelo de comunicação mestre-escravo, no qual um único dispositivo, o mestre, pode iniciar transações denominadas *queries*. Os demais dispositivos da rede (escravos) respondem, suprimindo os dados solicitados pelo mestre ou executando uma ação por ele comandada. Geralmente, o mestre é um sistema supervisor e os escravos são controladores lógicos programáveis [8].

O protocolo Modbus pode ser utilizado com diversos padrões de meio físico, como RS-232, RS-485 e Ethernet TCP/IP. Para a comunicação entre os dispositivos utilizados nesse projeto foi adotado o padrão RS-485, que é um sistema multiponto, o que permite que vários dispositivos possam ser ligados ao mesmo barramento de comunicação. Sua implementação pode ser feita de dois modos: em ASCII (*American Standard Code for Information Interchange*) ou RTU (*Remote Terminal Unit*) [8]. Para este projeto, foi selecionado o modo RTU, no qual cada palavra é composta por um caractere no padrão hexadecimal, compostas por 11 bits, sendo 1 start bit, 8 data bits e 2 stop bits. A figura 5 apresenta um barramento Modbus RS-485 à dois fios.

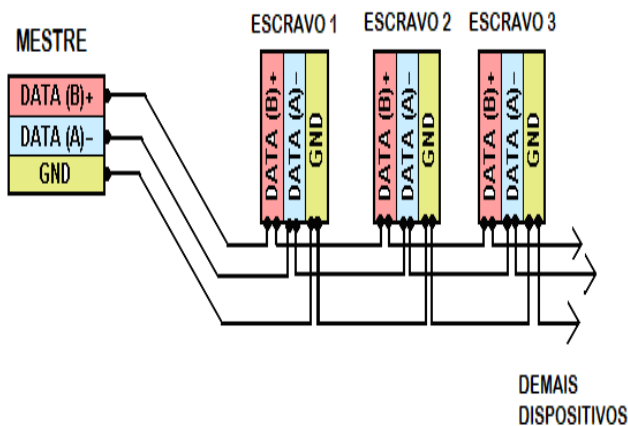


Figure 5: Barramento Modbus RS-485 à dois fios. FONTE: B&B Eletronics, 2003.

### 3. DESCRIÇÃO FÍSICA

O desenvolvimento do sistema supervisor para monitoramento e interação do operador com a planta de potência tornou-se possível a partir da presença de alguns elementos sensores e atuadores instalados no processo, conectados em algumas placas microcontroladas para processamento e transmissão dos sinais obtidos. A figura 6 ilustra o sentido que os sinais e os dados percorrem o sistema desde o processo até o supervisor.

#### 3.1. Elementos sensores

O sensor adotado para realizar a medição de temperatura nos 12 pontos escolhidos foi o termopar tipo K, sensor

este constituído pela junção de um fio de Chromel (Ni90%; Cr10%), como termo elemento positivo, e um fio de Alume (Ni95%; Mn2%; Si1%; Al2%), como termo elemento negativo presente dentro de uma sonda [3]. A figura 7 mostra a sonda do termopar utilizado. Os terminais negativo e positivo do termopar foram conectados a um módulo de leitura com um circuito integrado MAX6675, da empresa Maxim Integrated, conforme exibido na figura 8. O MAX6675 é um conversor AD de 12 bits de resolução que possui um sensor para compensação de junta fria, um controlador digital e a saída dos dados de temperatura já disponibilizados via protocolo SPI para conexão a qualquer tipo de microcontrolador podendo realizar medições de temperatura entre 0 e 1024°C [9]. A figura 9 mostra a pinagem do MAX6675, enquanto a tabela 2 detalha a função de cada pino.

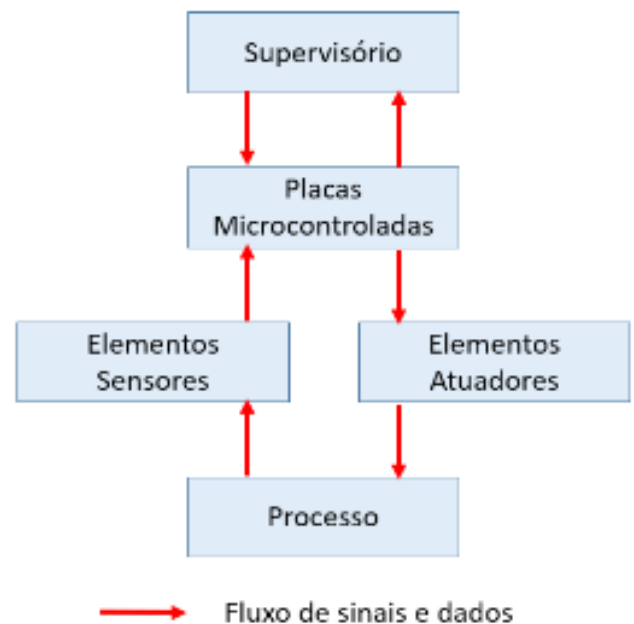


Figure 6: Fluxograma de sinais e dados.



Figure 7: Termo par K utilizado.



Figure 8: Módulo com o CI MAX6675.

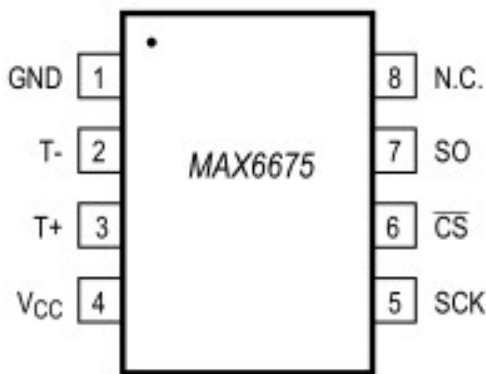


Figure 9: CI MAX6675. FONTE: Maxim Integrated, 2002.

TABELA 2 – Descrição dos pinos do MAX6675

PINO	FUNÇÃO
GND	Terra
T-	Conexão ao terminal negativo do termopar K (liga de Alumel)
T+	Conexão ao terminal positivo do termopar K (liga de Chromel)
VCC	Alimentação positiva (+5V)
SCK	Entrada para <i>clock</i> serial
CS	Seleção de chip. Configuração baixa habilita a interface serial
SO	Saída de dados serial
N.C.	Sem conexão

FONTE: Maxim Integrated, 2002.

Para se realizar a medição da pressão no ponto escolhido utilizou-se o transdutor piezoresistivo MPX5700AP, desenvolvido pela Freescale Semiconductor, conforme ilustrado

na figura 10. O MPX5700AP é um sensor de pressão de silício monolítico projetado para aplicações que empregam um microcontrolador ou microprocessador com entradas A/D. Este transdutor de elemento único combina técnicas de micromecanização, metalização de filme fino e processamento bipolar para fornecer um sinal de saída analógico preciso proporcional à pressão aplicada. O MPX5700AP é um sensor de pressão absoluta com faixa operacional entre 15 e 700 kPa empregado na medição de pressão de ar e gases com temperatura entre -40° e 125°C. Apesar de possuir 6 pinos, apenas 3 são utilizados nesta aplicação: saída de sinal (0,2 V a 4,7 V), terra e alimentação positiva (+5V) [10]. Com base na faixa de pressão operacional do sensor, e no intervalo de variação do sinal de saída, conclui-se que este sensor possui uma sensibilidade de 6,4 mV para cada 1 kPa.

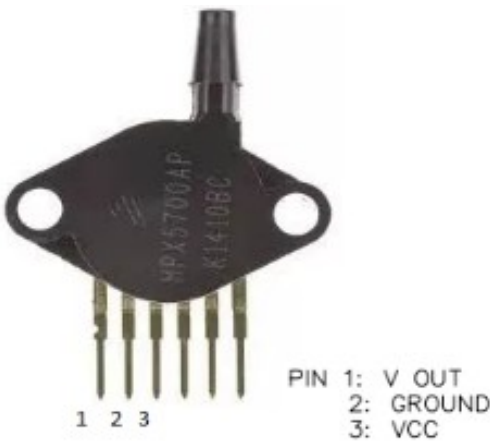


Figure 10: Sensor MPX5700AP.

3.2. Elementos atuadores

O funcionamento do ciclo combinado Brayton-Rankine depende dos seguintes atuadores controlados: uma válvula *on/off* para gás GLP, uma usina ignitora, um compressor de ar e uma bomba d’água.

A figura 11 mostra a válvula usada para permitir a passagem de gás combustível para os queimadores. Esta é uma válvula do tipo agulha, com seu corpo construído em latão forjado, dispositivo este geralmente usado em fornos de padaria. Sua abertura só ocorre quando se energiza com 12 V CC a solenoide nela existente.

O acendimento do queimador é feito através de um eletrodo centelhador que recebe um arco elétrico de 16 kV gerado por uma usina ignitora quando esta é energizada com 220 V CA. Estes equipamentos são os mesmos utilizados em fogões e fornos de cozinha. A figura 12 mostra a usina e o eletrodo de ignição.

Para fornecer o ar comprimido necessário ao funcionamento do ciclo Brayton foi utilizado um compressor elétrico de pistão de 3,75 kW de potência com capacidade de fornecimento de 707,5 l/min à uma pressão de 175 PSI. As figuras 13 e 14 mostram o compressor e sua placa de identificação com seus dados de funcionamento.





Figure 11: Válvula de gás.

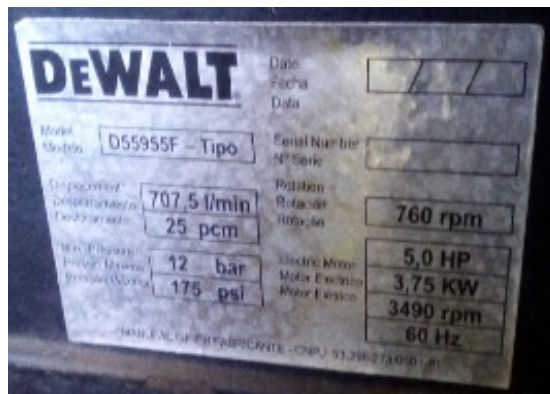


Figure 14: Compressor elétrico de pistão.



Figure 12: Usina (esquerda) e eletrodo de ignição (direita).

Para fornecer água para o funcionamento do ciclo Rankine, foi utilizada uma bomba elétrica tipo diafragma com capacidade de 10 l/min à uma pressão de 200 PSI. O uso desse equipamento é muito comum em embarcações para transporte de água para serviço e consumo. Seu acionamento ocorre quando o motor é energizado com 12 V CC. As figuras 15 e 16 mostram a bomba e sua placa de identificação com seus dados de funcionamento.



Figure 15: Bomba de diafragma.



Figure 13: Placa de identificação do compressor elétrico.



Figure 16: Placa de identificação da bomba d'água.

3.3. Placas microcontroladas

O microcontrolador é o componente responsável por receber um sinal eletrônico já condicionado por um circuito intermediário, e processá-lo por uma lógica de controle. Em função dessa lógica, e dos níveis dos sinais de entrada, o sistema interage com atuadores que interferem no sistema, seja para informar, seja para efetivamente interagir com o sistema [11].

Arduino é uma plataforma *open source*, com *software* livre baseado nas linhas de microcontroladores da ATMEL disponíveis em diferentes versões no mercado. O Arduino Uno é uma versão de placa baseada no microcontrolador ATmega328 de 8 bits da família AVR, com arquitetura RISC avançada, que pode ser encontrado em encapsulamento DIP28 ou SMD [12]. As principais características do Arduino Uno são mostradas na tabela 3. Sua foto é apresentada na figura 17.

TABELA 3 – Principais Especificações do Arduino Uno.

MICROCONTROLADOR	ATMEGA328
Tensão de funcionamento	5V
Tensão de entrada fonte externa (recomendada)	7-12V
Tensão de entrada (limites)	6-20V
Pinos I/O Digital	14 (dos quais 6 fornecem saída PWM)
Entradas Analógicas	6
Corrente DC para pinos I/O	40 mA
Corrente DC para pino 3,3V	50 mA
Memória Flash	32 kB (ATmega328) sendo 0,5 kB utilizado pelo bootloader
SRAM	2 kB (ATmega328)
EEPROM	1 kB (ATmega328)
Frequência de clock	16 MHz

FONTE: Stevan Jr. e Silva, 2015.



Figure 17: Placa Arduino Uno.

Para operar a planta didática de geração de potência foram utilizados cinco Arduinos Uno conectados, comunicando-se entre si e o supervisor. Quatro Arduinos foram programados para receber e processar os sinais gerados pelos sensores presentes na planta e transmiti-los ao supervisor; e um Arduino foi programado para tomar decisões e enviar sinais de acionamento para os atuadores. O acionamento de cada atuador é realizado através da energização de um relé correspondente presente em um módulo conectado diretamente às saídas digitais do Arduino. Ao receber um sinal alto da saída digital, o relé fecha contato permitindo a energização do atuador e seu consequente funcionamento. A figura 18 mostra um módulo com 4 relés utilizado no projeto.



Figure 18: Módulo com 4 relés.

Para se estabelecer a comunicação entre os Arduinos e o supervisor, empregou-se um módulo adaptador com *chip* MAX485 conectado em cada placa microcontrolada com a finalidade de converter o sinal TTL fornecido pelo Arduino em sinal padrão RS485 (EIA/TIA485). As figuras 19 e 20 mostram respectivamente o módulo adaptador e a pinagem do MAX485.



Figure 19: Módulo com *chip* MAX485.

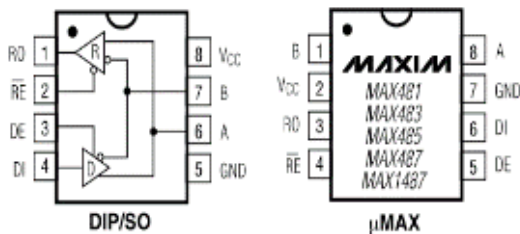


Figure 20: *Chip* MAX485. FONTE: Maxim Integrated, 2014.

O *chip* MAX485, desenvolvido pela Maxim Integrated, é um transceptor de baixa potência que trabalha alimentado

por uma única fonte de 5 V e apresenta corrente de consumo de 300  $\mu$ A, podendo atingir taxa de transmissão máxima de 2,5 Mbps. Para que se inicie a comunicação entre os dispositivos, os terminais DI e RO dos módulos MAX485 devem estar conectados respectivamente aos pinos TX e RX de cada Arduino e os terminais DE e RE do módulo devem ser conectados juntos em uma saída digital do Arduino. A variação de polaridade desta saída digital define a função do MAX485 durante a comunicação, trabalhando ora como transmissor (5 V), ora como receptor (0 V) [13].

Para que o computador no qual está instalado o programa supervisório possa se comunicar com barramento Modbus, é necessário utilizar um conversor RS485/USB. Esse conversor possui dois bornes para a ligação dos barramentos A e B de um lado, e do lado oposto um conector USB para se conectar ao computador. A figura 21 mostra o conversor RS485/USB utilizado.



Figure 21: Conversor RS 485 - USB.

Para maior praticidade nas ligações entre as partes - Arduinos, módulos MAX485 e conectores de sensores e atuadores - assim como para evitar mal contato que possa causar erros na transmissão de sinais dos medidores e na comunicação de dados, foi confeccionada uma placa de circuito impresso (PCI) para cada Arduino. Para fixação dos dispositivos eletrônicos na PCI, foram soldados barras de pino macho, barras de pino fêmea e conectores Molex para encaixe do Arduino, do módulo MAX485 e conectores dos sensores e atuadores respectivamente. As figuras 22 a 27 mostram os detalhes das placas de circuito impresso confeccionadas e os dispositivos montados.

Como apresentado na figura 28, as seis PCIs com os Arduinos acoplados (5 em funcionamento e 1 de reserva) foram fixadas em uma placa de montagem e acomodados dentro de uma caixa hermética colocada junto à planta, local para onde foram destinados os cabos dos sensores e atuadores através de tubulação própria. Na porta do painel montado foi colocada uma chave comutadora liga/desliga e uma botoeira de emergência para operação do sistema como mostra a figura 29.

#### 4. SUPERVISÓRIO DESENVOLVIDO

No desenvolvimento do supervisório da planta de geração de potência foi utilizada a versão educacional do *software* InduSoft Web Studio 8.0 (IWS), fornecido pela empresa InduSoft. Esse programa permite construir aplicações completas SCADA ou IHM para a indústria de automação. A figura 30 mostra a tela principal deste *software*.

O IWS viabiliza a execução de projetos com microcomputadores conectados em tempo real com processadores ou máquinas por meio de controladores programáveis, equipamentos de I/O remotos, e outros dispositivos de aquisição de dados. Estes projetos podem conter: telas com interfaces animadas de operação, *drivers* configuráveis de CLP (controlador lógico programável) e outros dispositivos controláveis de I/O, um banco de dados de *tags* do projeto, módulos opcionais como monitores de alarme, receitas, gráficos de tendências, lógica, organizadores e um sistema de segurança.

O desenvolvimento de um supervisório para uma planta de geração de potência consiste na elaboração de uma aplicação para monitoramento e operação do protótipo construído de tal forma que professores e alunos possam operá-lo de maneira prática e segura, visualizando as variáveis do ciclo combinado Brayton-Rankine em funcionamento de maneira intuitiva a partir de telas com diagramas de processo mostrados.

##### 4.1. Tags

*Tags* são variáveis usadas pelo IWS para receber e armazenar os dados obtidos a partir da comunicação com os dispositivos na planta, podendo ser utilizadas para mostrar informações nas telas, manipular objetos e controlar tarefas em execução. Ao se inserir uma *tag* no projeto, deve-se nomeá-la e defini-la quanto a um dos tipos de dado mostrados abaixo:

- Booleana - 0 (falso) e 1 (verdadeiro), usado para ligar e desligar objetos
- Inteira - Número Inteiro (positivo, negativo ou zero)
- Real – utilizado para medições ou para valores decimais ou fracionários
- *String* - dado alfanumérico com até 1024 caracteres

A versão educacional utilizada do IWS disponibiliza um total de 4000 *tags* para uso. Para esse projeto de supervisório, foram criadas um total de 17 *tags*; 12 para receber os dados referentes aos valores de temperatura e pressão e 5 *tags* correspondendo aos atuadores. A tabela 4 mostra as *tags* criadas, a estação (placa microcontrolada) na qual se encontra e o tipo de ação e de dado relativa a cada *tag*.



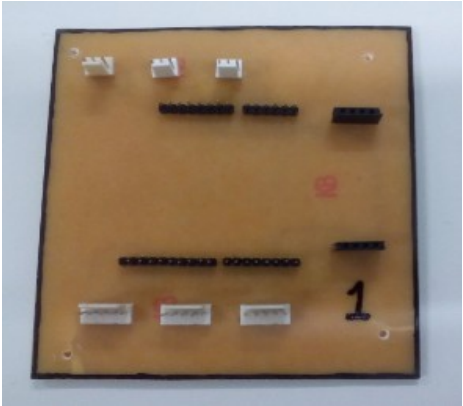


Figure 22: Modelo de PCI para sensores (face com os conectores para encaixe).

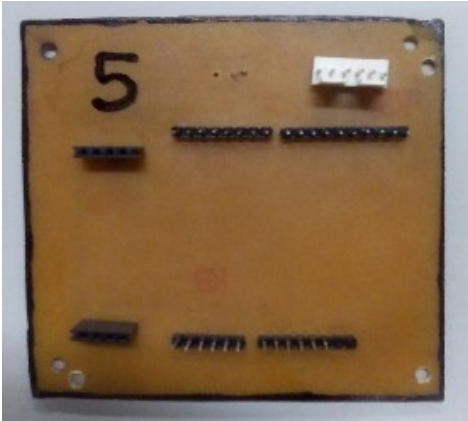


Figure 25: Modelo de PCI para atuadores (face com os conectores para encaixe).

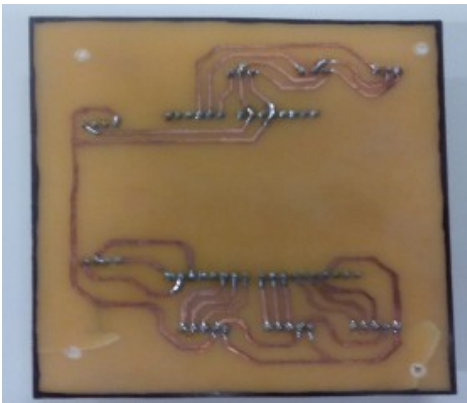


Figure 23: Modelo de PCI para sensores (face com as trilhas de ligação).

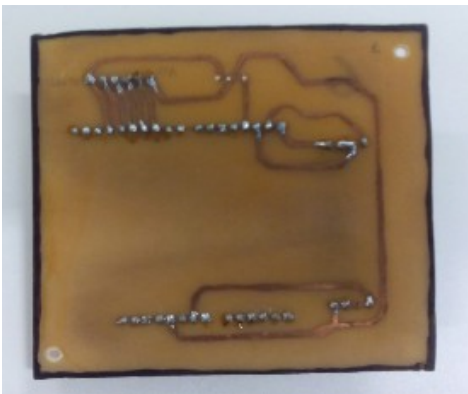


Figure 26: Modelo de PCI para atuadores (face com as trilhas de ligação).



Figure 24: Modelo de PCI para sensores com os dispositivos conectados.



Figure 27: Modelo de PCI para atuadores com os dispositivos conectados.



Figure 28: Acomodação dos Arduinos Uno dentro do painel.



Figure 29: Painel de armazenamento dos Arduinos Uno.

TABELA 4 – Lista de *tags* definidas no projeto.

TAG	ESTAÇÃO	AÇÃO	DADO
temp ar ambiente	1	Ler	Inteira
temp ar comprimido	1	Ler	Inteira
temp saida brayton	1	Ler	Inteira
pressao ar comprimido	1	Ler	Inteira
temp agua ambiente	2	Ler	Inteira
temp saida economizador	2	Ler	Inteira
temp entrada aquecedor	2	Ler	Inteira
temp saida aquecedor	3	Ler	Inteira
temp_entrada_superaquece	3	Ler	Inteira
temp saida superaquecedor	3	Ler	Inteira
temp camara brayton	4	Ler	Inteira
temp camara rankine	4	Ler	Inteira
temp chamine	4	Ler	Inteira
bomba	5	Escrever	Booleana
centelha	5	Escrever	Booleana
compressor	5	Escrever	Booleana
válvula_glp	5	Escrever	Booleana

4.2. Telas criadas

Para a criação da interface homem-máquina da planta de geração de potência, cinco telas de interação foram elaboradas, utilizando elementos gráficos disponíveis na biblioteca do IWS (motores, tanques e tubulação) e arquivos de imagens importadas e dispostas como pano de fundo, sobre as quais foram fixados botões de acionamento dos atuadores ou texto com *link* de dados onde são informados os valores de temperatura e pressão medidos.

As telas apresentadas nas figuras 30 a 35 compõem o painel de controle da planta de geração de potência. O painel de controle é formado por uma tela principal usada apenas para monitoramento, incluindo o diagrama de todo o processo Brayton-Rankine e os valores de temperatura e pressão medidos, indicados nos pontos onde os sensores estão instalados. Também foram elaboradas duas telas auxiliares detalhando o ciclo Brayton e o ciclo Rankine, cada uma com seu respectivo diagrama, mostrando os valores dos sensores e disponibilizando os botões para acionamento dos atuadores envolvidos. A quarta e a quinta tela mostram gráficos para acompanhamento individual, ao longo do tempo, da variação dos valores de temperatura e pressão de cada sensor.

4.3. Driver de comunicação

A comunicação com equipamentos de aquisição de dados, controladores, CLPs, UTRs, ou qualquer outro tipo de equipamento, é feita através de *drivers* de comunicação próprios ou servidores OPC, de acordo com o tipo do equipamento ou tipo de comunicação necessária. A aquisição e o envio de dados para os dispositivos conectados ao sistema supervisório é fundamental para manter o operador ciente do estado atual do sistema. Assim, o operador pode agir no sistema de forma coesa, controlando os equipamentos que estejam conectados ao sistema.

Um *driver* de comunicação é uma DLL que contém informações específicas sobre o equipamento remoto e implementa o protocolo de comunicação permitindo definir a interface de comunicação entre o supervisório e os equipamentos remotos, tais como um CLP, um *single-loop*, transmissores e medidores.

O IWS disponibiliza uma biblioteca com *drivers* de diversas empresas desenvolvedoras de tecnologia relacionada à automação, como Siemens e Allen Bradley. Para se estabelecer a comunicação RS485 entre o supervisório desenvolvido e as placas microcontroladas foi selecionado a DLL MODBU (MODBUS Protocol RTU/ASCII (CE) [v10.10]). As figuras 36 e 37 mostram as janelas destinadas à configuração do *driver* MODBU e os parâmetros selecionados.

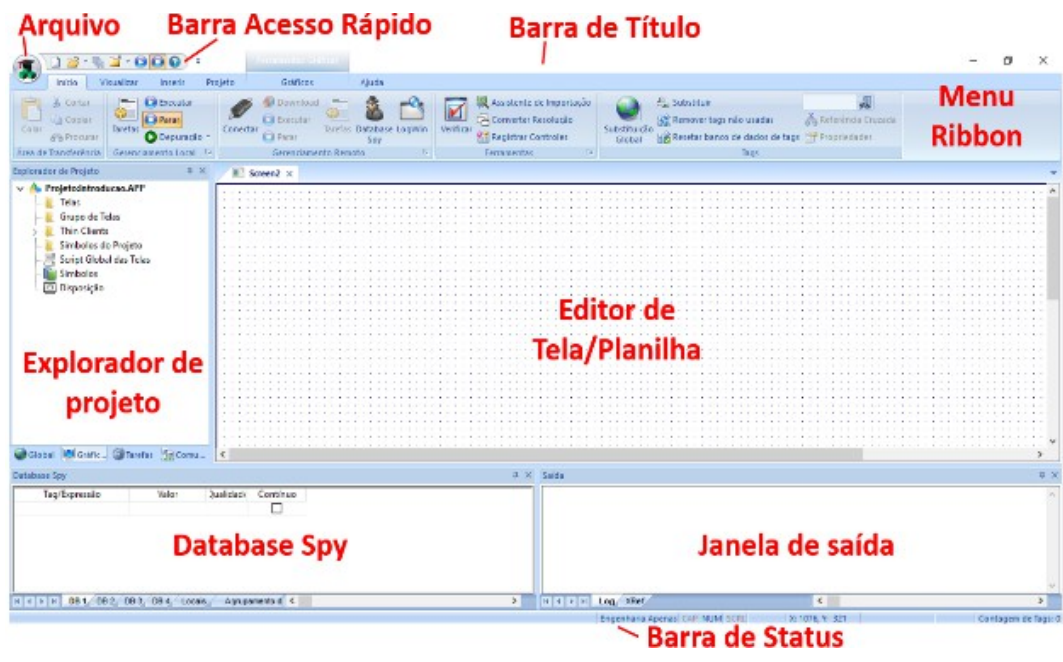


Figure 30: Tela principal do software Indusoft Web Studio.

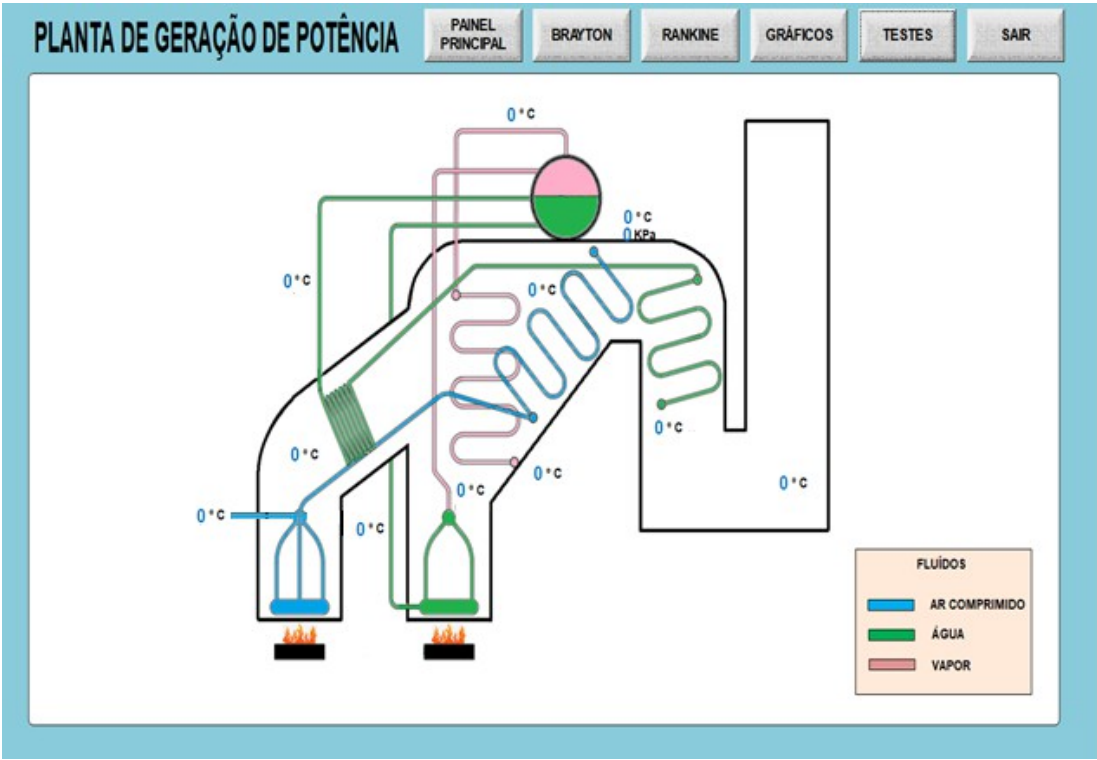


Figure 31: Tela principal do supervisor desenvolvido.

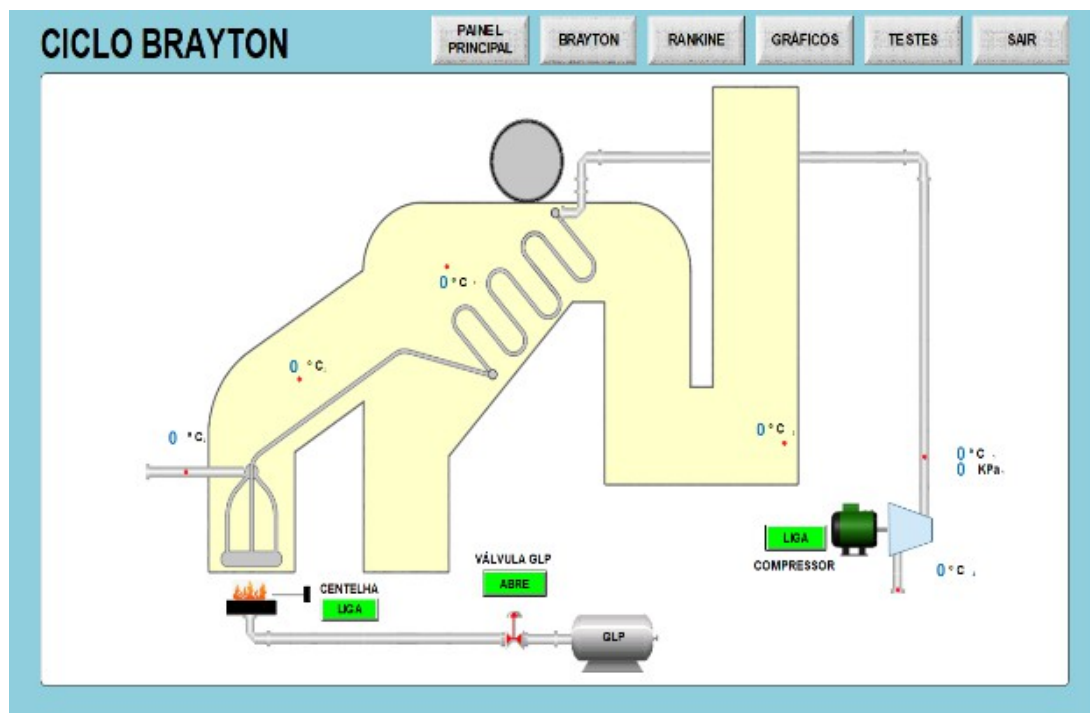


Figure 32: Tela específica para o ciclo Brayton no supervisório desenvolvido.

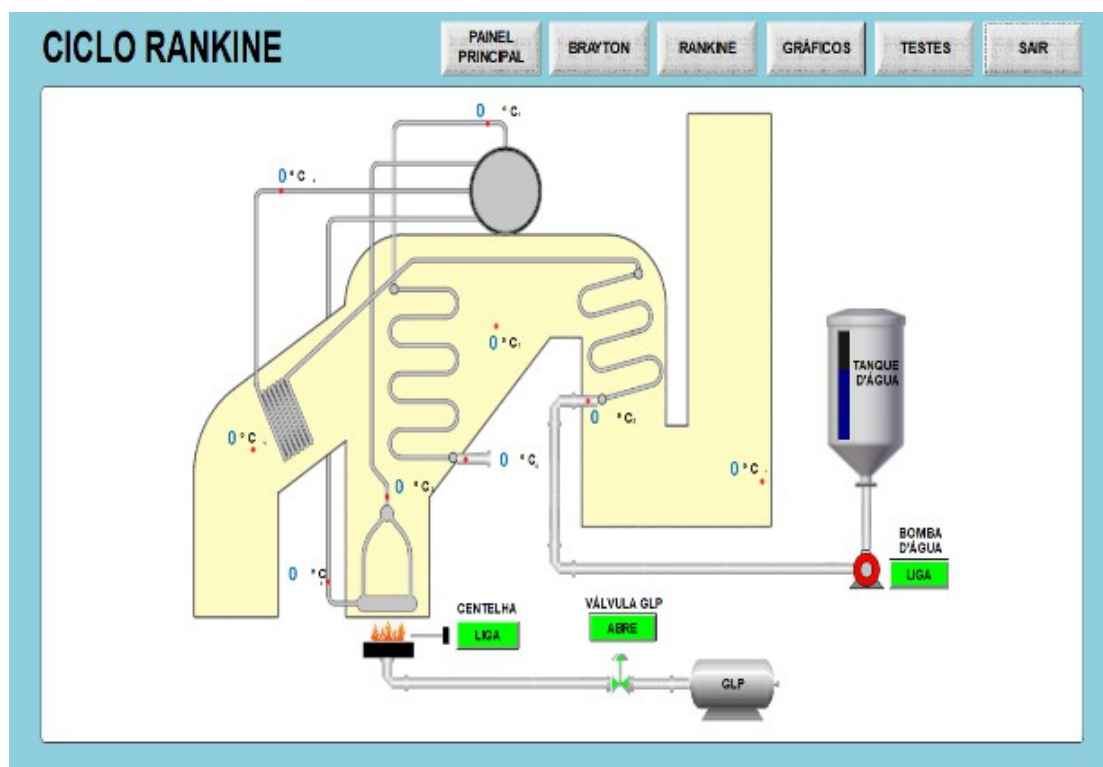


Figure 33: Tela específica para o ciclo Rankine no supervisório desenvolvido.

## 5. OPERAÇÃO E MONITORAMENTO DA PLANTA

Com o sistema supervisório desenvolvido pronto para realizar os primeiros experimentos, o protótipo de geração de potência foi colocado em operação, sendo efetuado seu mo-

nitoramento durante o período inicial de uma hora de teste com o funcionamento dos ciclos térmicos Brayton e Rankine.



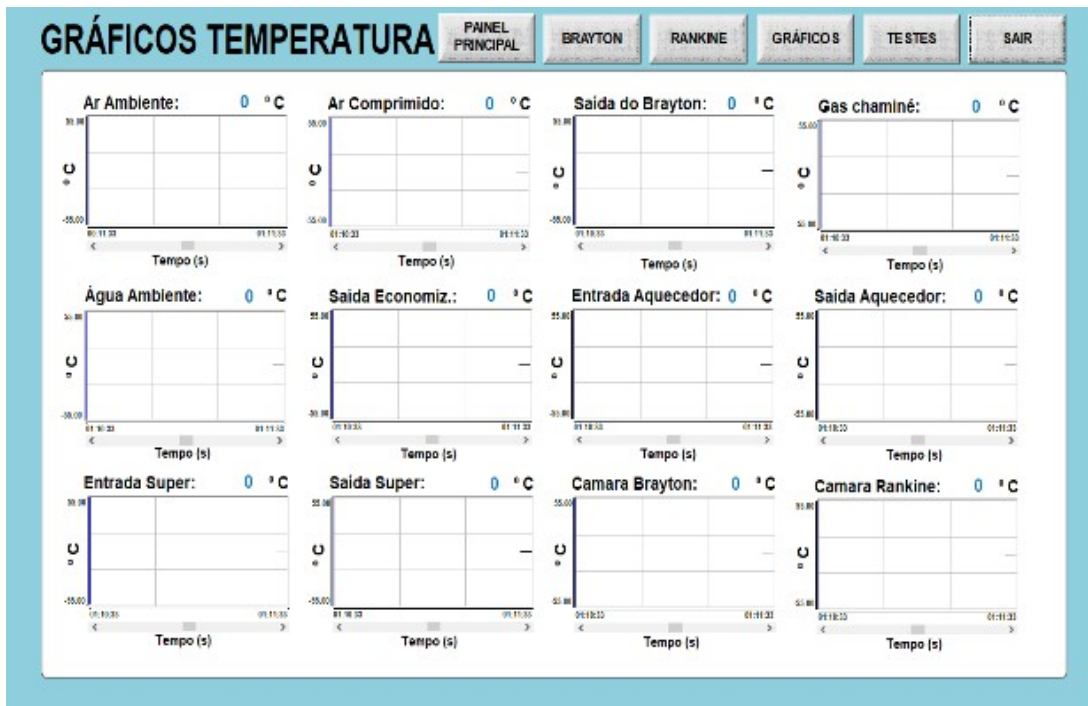


Figure 34: Tela reservada para os gráficos com as medições de temperatura.



Figure 35: Tela reservada ao gráfico com as medições de pressão.

### 5.1. Execução de comandos de partida

Com a planta preparada para funcionamento, conferido o nível de água no reservatório, o painel energizado, e um computador com o supervisor instalado preparado, o sistema foi colocado em modo *Run Time* através do botão “EXECUTAR” presente na barra de acesso rápido do IWS, como

mostra a figura 38.

O primeiro comando executado a partir do supervisor foi o acionamento da bomba d’água, o que foi feito através do botão “LIGA” presente na tela reservada ao ciclo Rankine, junto ao ícone do dispositivo mencionado, conforme mostrado na figura 39. Tal ação visava transferir a quantidade de água necessária do tanque reservatório ao tubulão

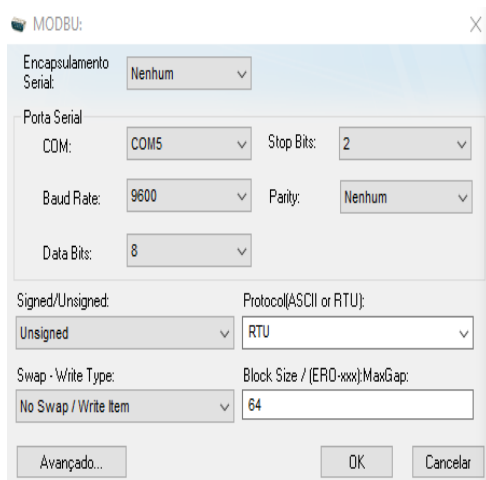


Figure 36: Janela de configuração de parâmetros do *driver* de comunicação (1/2).

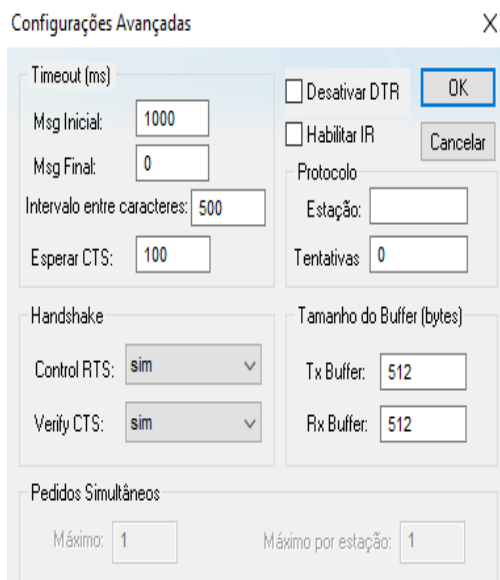


Figure 37: Janela de configuração de parâmetros avançados do *driver* de comunicação (2/2).

para o processo de geração de vapor. Quando foi constatado visualmente que o nível de água necessário no tubulão já havia sido atingido, o botão “DESLIGA” foi pressionado.

O segundo comando executado foi o acionamento do compressor a partir do botão “LIGA” junto ao ícone do respectivo equipamento presente na tela reservada ao ciclo Brayton, como mostra a figura 40. Com esse acionamento, iniciou-se o fornecimento de ar comprimido ao ciclo Brayton para realização da troca de calor durante o processo.

O terceiro e o quarto comandos foram, respectivamente, a abertura da válvula de gás e o acionamento da centelha de ignição, realizados nesta sequência através dos botões “ABRE” e “LIGA”, posicionados juntos aos ícones de cada dispositivo, conforme mostrado na figura 41. Com a execução desses comandos foi realizado o acendimento das chamas dos dois queimadores presentes na planta, dando início ao processo de funcionamento.

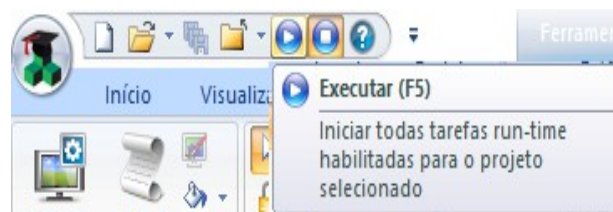


Figure 38: Botão executar no IWS.



Figure 39: Botão para acionamento da bomba d'água no supervisão.

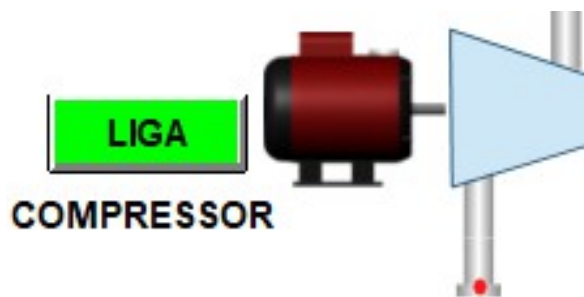


Figure 40: Botão para acionamento do compressor no supervisão.

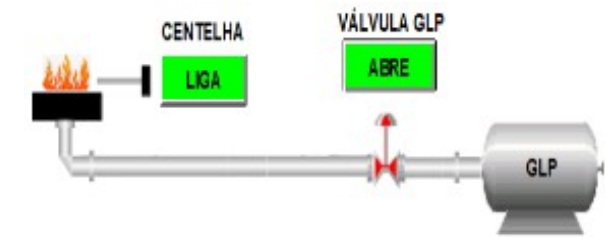


Figure 41: Botões para acionamento da válvula de gás e da centelha de ignição no supervisório.

5.2. Execução de comandos de parada

Com a planta em funcionamento, foi realizado seu monitoramento durante o período de uma hora conforme detalhado a seguir. Após esse período, foram enviados comandos para o desligamento do protótipo. Na sequência, foram enviados os seguintes comandos: fechamento da válvula de gás, apagando as chamas dos queimadores e, após dez minutos, desligamento do compressor, cortando o suprimento de ar comprimido à planta. O período em que foi mantida a passagem de ar comprimido tinha como objetivo desaquecer as serpentinas do ciclo Brayton para evitar possíveis danos causados pela dilatação térmica.

Com todos os atuadores desligados, foi então encerrado o modo *Run Time* através do botão “SAIR”, presente no canto superior direito de todas as telas do supervisório, como mostra a figura 42.



Figure 42: Botão para sair do modo *Run Time*.

5.3. Monitoramento e geração de histórico de medições

Durante o período em que a planta esteve em funcionamento, o operador do supervisório pôde acompanhar a variação instantânea dos valores de temperatura nos doze pontos em que foram instalados os termopares e também pôde acompanhar a variação instantânea do valor de pressão fornecido pelo sensor MPX5700AP. O acompanhamento destes valores pôde ser feito pelas telas reservadas aos gráficos e também pelas telas do ciclo Brayton e Rankine. Após o período de funcionamento do protótipo o IWS gerou um arquivo texto com o histórico de medição para cada ponto. A partir desses arquivos, foram gerados os gráficos mostrados entre as figuras 43 e 55.

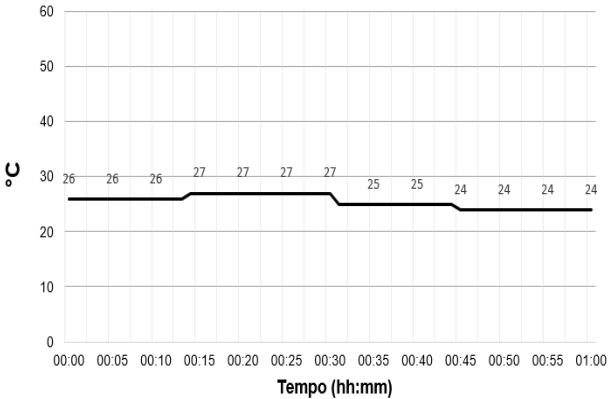


Figure 43: Temperatura do ar ambiente.

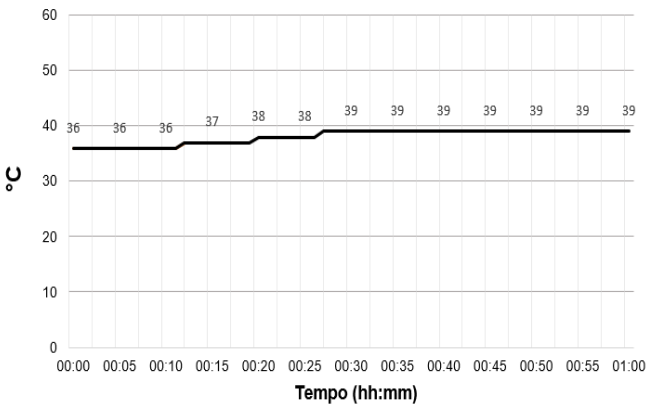


Figure 44: Temperatura do ar comprimido na entrada do ciclo Brayton.

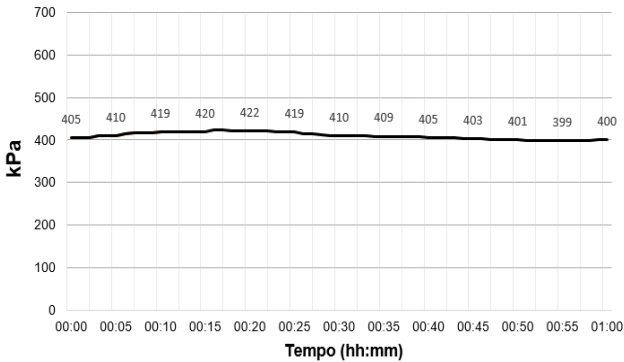


Figure 45: Pressão do ar comprimido na entrada do ciclo Brayton.

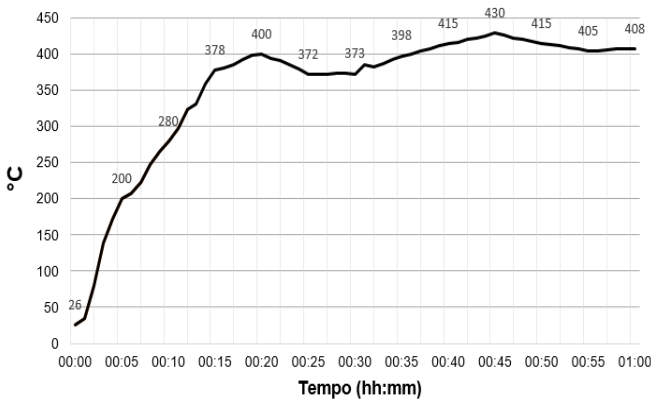


Figure 46: Temperatura do ar na saída do ciclo Brayton.

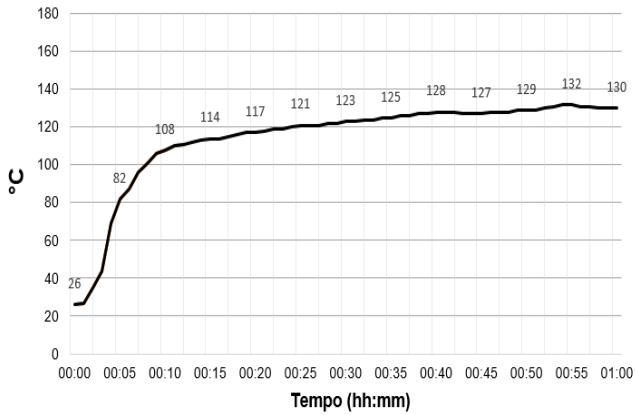


Figure 49: Temperatura na entrada do aquecedor no ciclo Rankine.

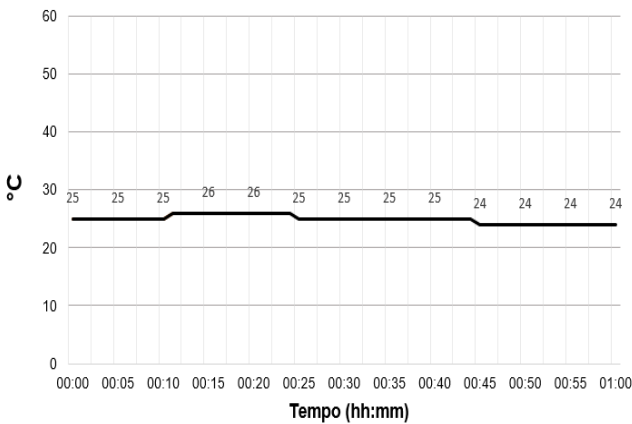


Figure 47: Temperatura da água na entrada do ciclo Rankine.

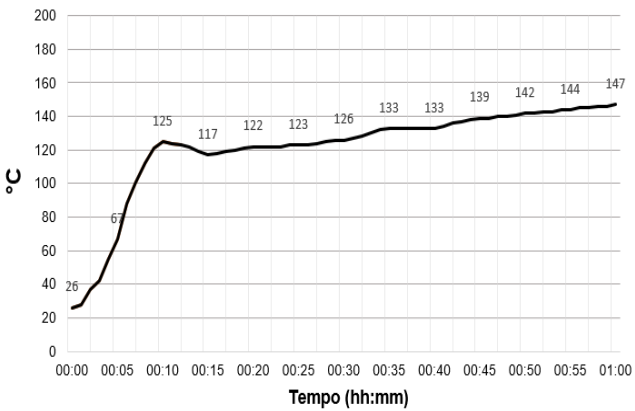


Figure 50: Temperatura na saída do aquecedor no ciclo Rankine.

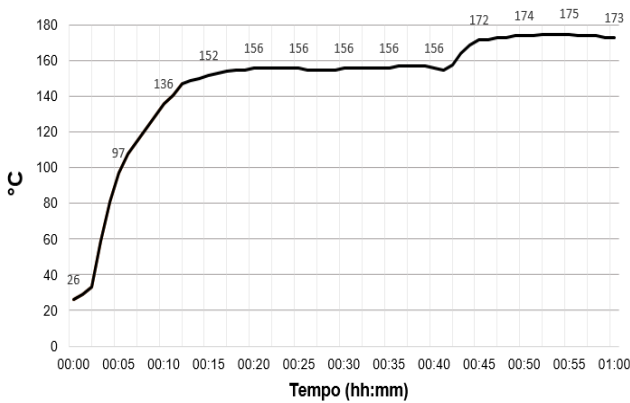


Figure 48: Temperatura após o economizador no ciclo Rankine.

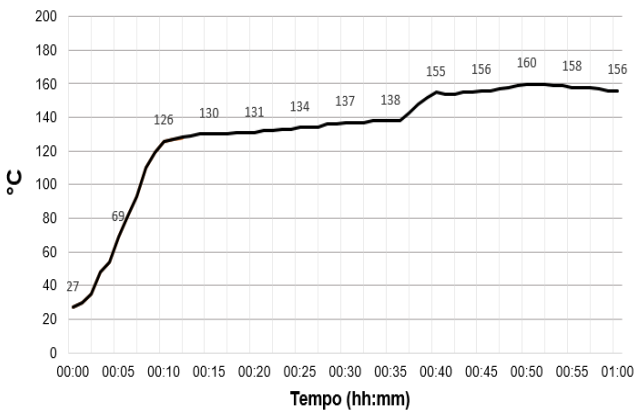


Figure 51: Temperatura na entrada do superaquecedor no ciclo Rankine.



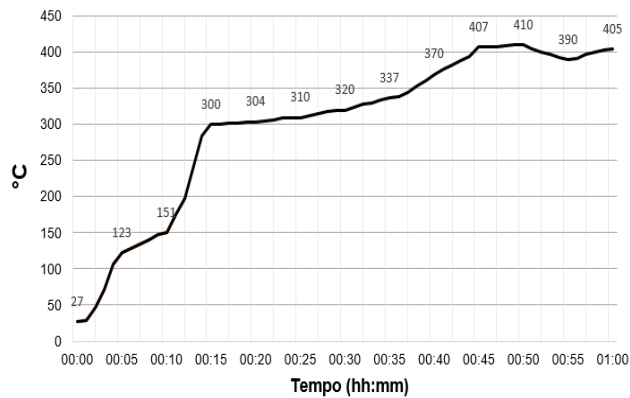


Figure 52: Temperatura na saída do superaquecedor no ciclo Rankine.

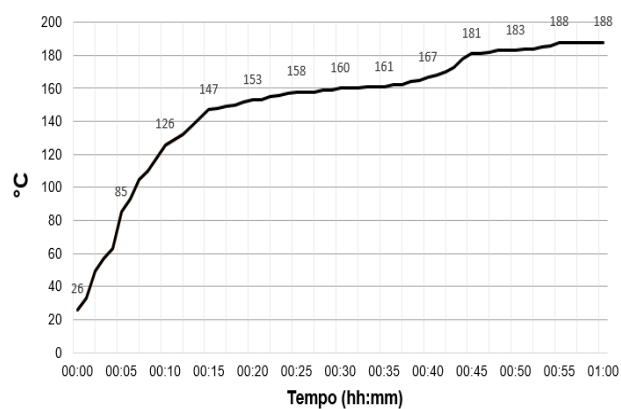


Figure 55: Temperatura dos gases de combustão na região da chaminé.

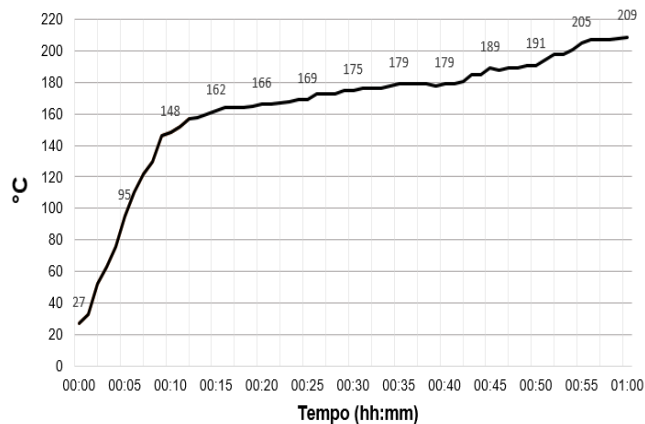


Figure 53: Temperatura dos gases de combustão na região do ciclo Brayton.

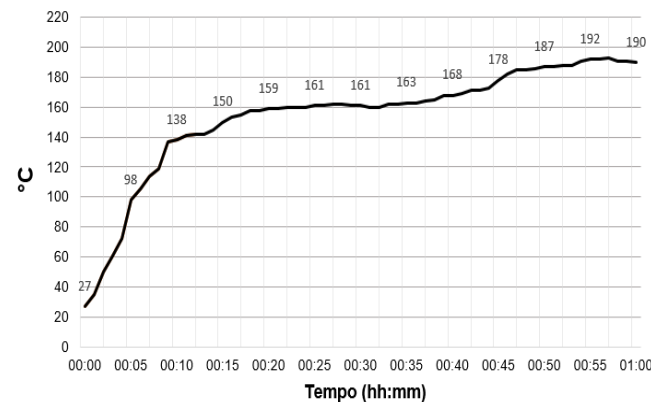


Figure 54: Temperatura dos gases de combustão na região do ciclo Rankine.

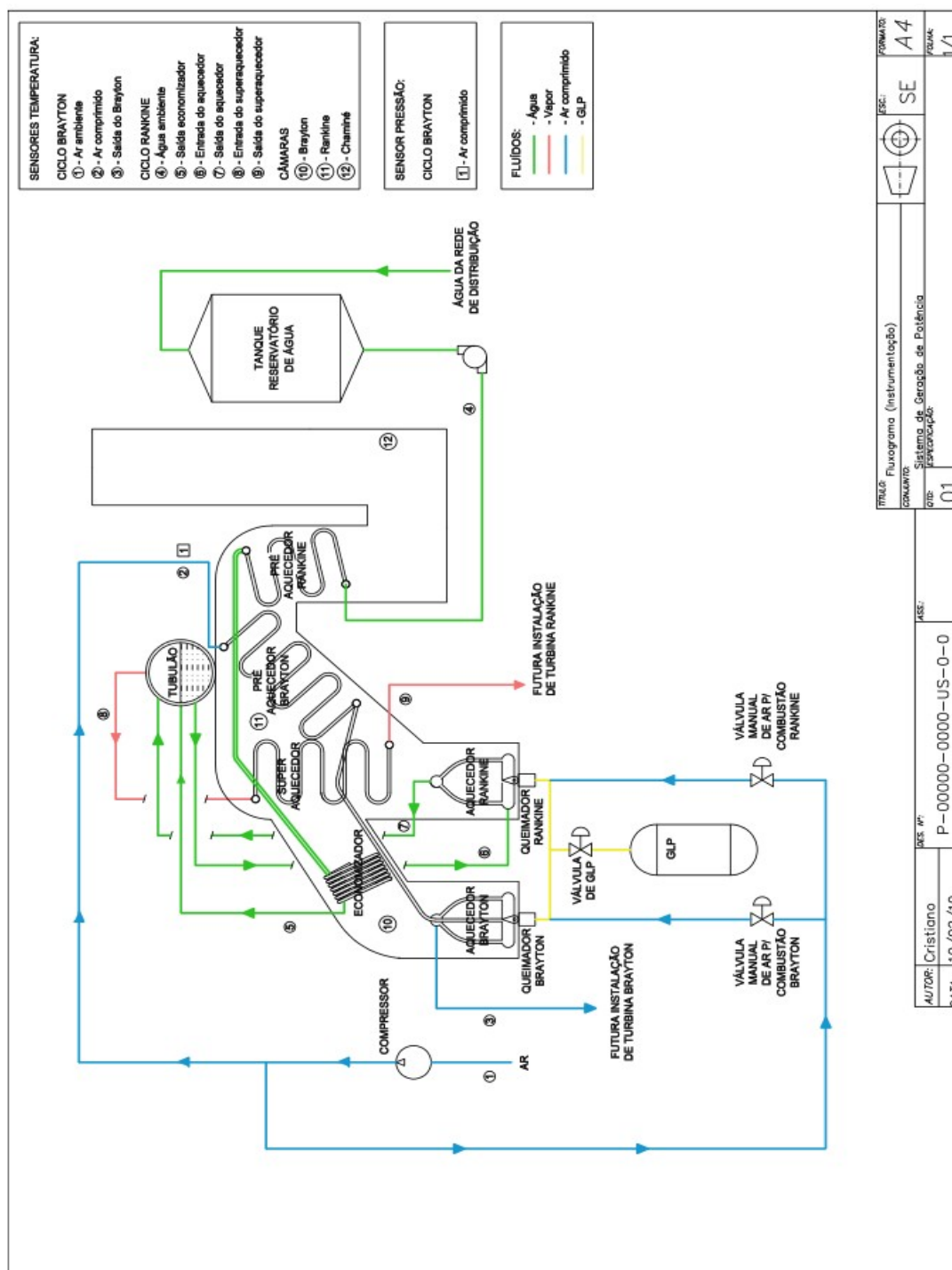
6. CONCLUSÃO

A etapa inicial de concepção de um supervisor para monitoração e controle de uma planta de geração de potência foi concluída. A planta está em operação para testes, atendendo o objetivo deste trabalho que foi de apresentar o processo de desenvolvimento de um supervisor para monitorar e operar um protótipo de geração de potência desenvolvido para fins didáticos. No atual estágio, um operador habilitado pode acessar a tela de um computador que tenha o supervisor desenvolvido instalado e a partir dele realizar o monitoramento das variáveis de processo e executar comandos em determinados atuadores, os quais foram detalhados ao longo deste documento. Considerando que o projeto como um todo ainda está em fase de testes, esta versão do supervisor não pode ser considerada uma versão definitiva. Conforme os testes vão sendo realizados, correções vão se mostrando necessárias e novas necessidades vão surgindo e exigindo a atualização dos *softwares* e dos códigos presentes nas placas microcontroladas para que possam ser inseridos novos itens e funcionalidades, visando maior segurança e praticidade na operação do equipamento. Tais atualizações eram esperadas e qualquer projeto, ao ser colocado em prática, apresenta essa demanda. Por fim, tal sistema desenvolvido está operando dentro da instituição de ensino superior Fac-tus, sendo utilizado por alunos que desenvolvem projetos de pesquisa dentro dessa área.

## Bibliography

- [1] PAULA, C. P. **Geração distribuída e cogeração no setor elétrico: Avaliação sistêmica de um plano de inserção incentivada**. Tese de doutorado de Pós-Graduação em Energia, IEE USP, 2004. Disponível em: <http://www.iee.usp.br>. Acesso em setembro/2018.
- [2] BOTELHO, M. H. C.; BIFANO, H. M. **Operação de caldeiras – gerenciamento, controle e manutenção**. São Paulo: Edgard Blücher, 2011.
- [3] BEGA, E. A. **Instrumentação aplicada ao controle de caldeiras**. 3 ed. Rio de Janeiro: Interciência, 2003.
- [4] VAN WYLEN, Gordon et al. **Fundamentos da Termodinâmica Clássica**. 4. ed. São Paulo: Edgard Blücher, 1994.
- [5] KIDNAY, A. J.; PARRISH, W. R.; MCARTNEY, D. G. **Fundamentals of Natural Gas Processing**. 2 ed. CRC Press, 2011.
- [6] LOGRADO, C. L. **2000 - Análise e otimização de turbinas a gás e ciclos combinados**. Dissertação de mestrado, Departamento de Engenharia Mecânica, Universidade de Brasília, 2000.
- [7] ROSÁRIO, J. M. **Princípios de Mecatrônica**. São Paulo: Pearson Education do Brasil, 2005.
- [8] MODICON. **Modbus Protocol Reference Guide**, AEG, PI-MBUS-300 Rev. E. 1993.
- [9] MAXIM INTEGRATED PRODUCTS. **Cold-Junction-Compensated K-Thermocouple-to-Digital Converter**. 2002. Disponível em: <https://datasheets.maximintegrated.com/en/ds/MAX6675.pdf>. Acesso em: 20 nov. 2018.
- [10] NXP. **Integrated Silicon Pressure Sensor On-Chip Signal Conditioned, Temperature Compensated and Calibrated**. 2012. Disponível em: <https://www.nxp.com/docs/en/data-sheet/MPX5700.pdf>. Acesso em: 13 out. 2018.
- [11] BOLTON, W. **Instrumentation and control systems**. 2 ed. Elsevier, 2015.
- [12] STEVAN JR. S. L.; SILVA, R. A. **Automação e instrumentação industrial com Arduino: teoria e projetos**. São Paulo: Érica, 2015.
- [13] MAXIM INTEGRATED PRODUCTS. **Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers**. 2014. Disponível em: <https://datasheets.maximintegrated.com/en/ds/MAX1487-MAX491.pdf>. Acesso em: 20 nov. 2018.

## ANEXO A – Fluxograma resumido do protótipo de geração de potência



**ANEXO B – Código fonte do Arduino 1**

```
//ARDUINO 01 - Monitoramento:
//TEMPERATURA: CAMARA RANKINE / SAIDA
ECONOMIZADOR / ENTRADA AQUECEDOR
//PRESSAO: AR COMPRIMIDO
//-----

#include "max6675.h"
#include "SimpleModbusSlave.h"
unsigned long currentMillis = 0, previousMillis = 0;
unsigned int amostragem = 0;
float temp_termo1 = 0;
float temp_termo2 = 0;
float temp_termo3 = 0;
//Definições para calculo da pressão
int valorAD;
int offset = 105;
int scalaMaxima = 963;
float pressaoFinal;
enum {
temp_camara_rankine,
temp_saida_economizador,
temp_entrada_aquecedor,
press_ar_comprimido,
HOLDING_REGS_SIZE
};
int holdingRegs[HOLDING_REGS_SIZE];
//Configuração dos pinos para max6675
temp_camara_rankine
int therm1DO = 11;
int therm1CS = 12;
int therm1CLK = 13;
MAX6675 therm1(therm1CLK, therm1CS, therm1DO);
//Configuração dos pinos para max6675
temp_saida_economizador
int therm2DO = 8;
int therm2CS = 9;
int therm2CLK = 10;
MAX6675 therm2(therm2CLK, therm2CS, therm2DO);
//Configuração dos pinos para max6675
temp_entrada_aquecedor
int therm3DO = 5;
int therm3CS = 6;
int therm3CLK = 7;
MAX6675 therm3(therm3CLK, therm3CS, therm3DO);
void setup() {
modbus_configure(&Serial, 9600, SERIAL_8N2, 1, 2,
HOLDING_REGS_SIZE, holdingRegs);
modbus_update_comms(9600, SERIAL_8N2, 1);
holdingRegs[temp_camara_rankine] = 0;
holdingRegs[temp_saida_economizador] = 0;
holdingRegs[temp_entrada_aquecedor] = 0;
holdingRegs[press_ar_comprimido] = 0;
}
void loop() {
currentMillis = millis();
if (currentMillis - previousMillis >= 250) {
previousMillis = currentMillis;
holdingRegs[temp_camara_rankine] = normalizarTemperatura(therm1);
```

```
holdingRegs[temp_saida_economizador] = normalizarTemperatura(therm2);
holdingRegs[temp_entrada_aquecedor] = normalizarTemperatura(therm3);
holdingRegs[press_ar_comprimido] = calcularPressaoComleituraDoSensor();
}
modbus_update();
}
float calcularPressaoComleituraDoSensor() {
valorAD = 0;
valorAD = analogRead(A0);
pressaoFinal = (valorAD - offset) * 700.0 / (scalaMaxima - offset);
return pressaoFinal;
}
double normalizarTemperatura(MAX6675 sensor) {
double temperatura = 0;
for (int i = 0; i <= 20; i++) {
temperatura += sensor.readCelsius();
delay(??);
}
return temperatura / 20;
}
```



**ANEXO C – Código fonte do Arduino 2**

```
//ARDUINO 02 - Monitoramento:
//TEMPERATURA: SAIDA BRAYTON / SAIDA SUPER-
//AQUECEDOR / AR AMBIENTE
//_____

#include "max6675.h"
#include "SimpleModbusSlave.h"
unsigned long currentMillis = 0, previousMillis = 0;
unsigned int amostragem = 0;
float temp_termo4 = 0;
float temp_termo5 = 0;
float temp_termo6 = 0;
enum {
temp_saida_brayton,
temp_saida_superaquedor,
temp_ar_ambiente,
HOLDING_REGS_SIZE
};
int holdingRegs[HOLDING_REGS_SIZE];
//Configuração dos pinos para max6675 temp_saida_brayton
int therm4DO = 11;
int therm4CS = 12;
int therm4CLK = 13;
MAX6675 therm4(therm4CLK, therm4CS, therm4DO);
//Configuração dos pinos para max6675
temp_saida_superaquedor
int therm5DO = 8;
int therm5CS = 9;
int therm5CLK = 10;
MAX6675 therm5(therm5CLK, therm5CS, therm5DO);
//Configuração dos pinos para max6675 temp_ar_ambiente
int therm6DO = 5;
int therm6CS = 6;
int therm6CLK = 7;
MAX6675 therm6(therm6CLK, therm6CS, therm6DO);
void setup() {
modbus_configure(&Serial, 9600, SERIAL_8N2, 2, 2,
HOLDING_REGS_SIZE, holdingRegs);
modbus_update_comms(9600, SERIAL_8N2, 2);
holdingRegs[temp_saida_brayton] = 0;
holdingRegs[temp_saida_superaquedor] = 0;
holdingRegs[temp_ar_ambiente] = 0;
}
void loop() {
currentMillis = millis();
if (currentMillis - previousMillis >= 250) {
previousMillis = currentMillis;
holdingRegs[temp_saida_brayton] = normalizarTemperatura(therm4);
holdingRegs[temp_saida_superaquedor] = normalizarTemperatura(therm5);
holdingRegs[temp_ar_ambiente] = normalizarTemperatura(therm6);
}
modbus_update();
}
double normalizarTemperatura(MAX6675 sensor) {
double temperatura = 0;
```

```
for (int i = 0; i <= 20; i++) {
temperatura += sensor.readCelsius();
delay(??);
}
return temperatura / 20;
}
```

**ANEXO D – Código fonte do Arduino 3**

```
//ARDUINO 03 - Monitoramento:
//TEMPERATURA: CAMARA BRAYTON / GAS
//CHAMINE / ENTRADA SUPERAQUECEDOR
//
//-----

#include "max6675.h"
#include "SimpleModbusSlave.h"
unsigned long currentMillis = 0, previousMillis = 0;
unsigned int amostragem = 0;
float temp_termo7 = 0;
float temp_termo8 = 0;
float temp_termo9 = 0;
enum {
temp_camara_brayton,
temp_gas_chamine,
temp_entrada_superaquecedor,
HOLDING_REGS_SIZE
};
int holdingRegs[HOLDING_REGS_SIZE];
//Configuração dos pinos para max6675
temp_camara_brayton
int therm7DO = 11;
int therm7CS = 12;
int therm7CLK = 13;
MAX6675 therm7(therm7CLK, therm7CS, therm7DO);
//Configuração dos pinos para max6675 temp_gas_chamine
int therm8DO = 8;
int therm8CS = 9;
int therm8CLK = 10;
MAX6675 therm8(therm8CLK, therm8CS, therm8DO);
//Configuração dos pinos para max6675
temp_entrada_superaquecedor
int therm9DO = 5;
int therm9CS = 6;
int therm9CLK = 7;
MAX6675 therm9(therm9CLK, therm9CS, therm9DO);
void setup() {
modbus_configure(&Serial, 9600, SERIAL_8N2, 3, 2,
HOLDING_REGS_SIZE, holdingRegs);
modbus_update_comms(9600, SERIAL_8N2, 3);
holdingRegs[temp_camara_brayton] = 0;
holdingRegs[temp_gas_chamine] = 0;
holdingRegs[temp_entrada_superaquecedor] = 0;
}
void loop() {
currentMillis = millis();
if (currentMillis - previousMillis >= 250) {
previousMillis = currentMillis;
holdingRegs[temp_camara_brayton] = normalizarTemperatura(therm7);
holdingRegs[temp_gas_chamine] = normalizarTemperatura(therm8);
holdingRegs[temp_entrada_superaquecedor] = normalizarTemperatura(therm9);
}
modbus_update();
}
double normalizarTemperatura(MAX6675 sensor) {
```

```
double temperatura = 0;
for (int i = 0; i <= 20; i++) {
temperatura += sensor.readCelsius();
delay(??);
}
return temperatura / 20;
}
```

**ANEXO E – Código fonte do Arduino 4**

```
//ARDUINO 04 - Monitoramento:
//TEMPERATURA: SAIDA AQUECEDOR / AR COM-
//PRIMIDO / AGUA AMBIENTE
//
//_____

#include "max6675.h"
#include "SimpleModbusSlave.h"
unsigned long currentMillis = 0, previousMillis = 0;
unsigned int amostragem = 0;
float temp_termo7 = 0;
float temp_termo8 = 0;
float temp_termo9 = 0;
enum {
temp_saida_aquecedor,
temp_ar_comprimido,
temp_agua_ambiente,
HOLDING_REGS_SIZE
};
int holdingRegs[HOLDING_REGS_SIZE];
//Configuração dos pinos para max6675
temp_saida_aquecedor
int therm7DO = 11;
int therm7CS = 12;
int therm7CLK = 13;
MAX6675 therm7(therm7CLK, therm7CS, therm7DO);
//Configuração dos pinos para max6675
temp_ar_comprimido
int therm8DO = 8;
int therm8CS = 9;
int therm8CLK = 10;
MAX6675 therm8(therm8CLK, therm8CS, therm8DO);
//Configuração dos pinos para max6675
temp_agua_ambiente
int therm9DO = 5;
int therm9CS = 6;
int therm9CLK = 7;
MAX6675 therm9(therm9CLK, therm9CS, therm9DO);
void setup() {
modbus_configure(&Serial, 9600, SERIAL_8N2, 4, 2,
HOLDING_REGS_SIZE, holdingRegs);
modbus_update_comms (9600, SERIAL_8N2, 4);
holdingRegs[temp_saida_aquecedor] = 0;
holdingRegs[temp_ar_comprimido] = 0;
holdingRegs[temp_agua_ambiente] = 0;
}
void loop() {
currentMillis = millis();
if (currentMillis - previousMillis >= 250) {
previousMillis = currentMillis;
holdingRegs[temp_saida_aquecedor] = normalizarTemper-
atura(therm7);
holdingRegs[temp_ar_comprimido] = normalizarTemper-
atura(therm8);
holdingRegs[temp_agua_ambiente] = normalizarTemper-
atura(therm9);
}
modbus_update();
}
```

```
double normalizarTemperatura(MAX6675 sensor) {
double temperatura = 0;
for (int i = 0; i <= 20; i++) {
temperatura += sensor.readCelsius();
delay(??);
}
return temperatura / 20;
}
```

**ANEXO F – Código fonte do Arduino 5**

```
//ARDUINO 05 - Acionamentos:
//CENTELHA / VALVULAGLP / COMPRESSOR / BOMBADAGUA
//
//-----

#include "SimpleModbusSlave.h"
unsigned long currentMillis = 0, previousMillis = 0;
unsigned int amostragem = 0;
////////// registers of slave //////////
enum {
CENTELHA,
VALVULAGLP,
COMPRESSOR,
BOMBADAGUA,
HOLDING_REGS_SIZE
};
int holdingRegs[HOLDING_REGS_SIZE];
//Configuração dos pinos
int rele_centelha = 10;
int rele_valvula_glp = 11;
int rele_compressor = 12;
int rele_bomba = 13;
//////////
void setup() {
modbus.configure(&Serial, 9600, SERIAL_8N2, 5, 2,
HOLDING_REGS_SIZE, holdingRegs);
holdingRegs[CENTELHA] = 0;
holdingRegs[VALVULAGLP] = 0;
holdingRegs[COMPRESSOR] = 0;
holdingRegs[BOMBADAGUA] = 0;
pinMode(rele_centelha, OUTPUT);
pinMode(rele_valvula_glp, OUTPUT);
pinMode(rele_compressor, OUTPUT);
pinMode(rele_bomba, OUTPUT);
digitalWrite(rele_centelha,HIGH);
digitalWrite(rele_valvula_glp,HIGH);
digitalWrite(rele_compressor,HIGH);
digitalWrite(rele_bomba,HIGH);
}
void loop() {
modbus.update();
currentMillis = millis();
if (holdingRegs[CENTELHA] == 1) {
digitalWrite (rele_centelha, LOW);
}
else {
digitalWrite (rele_centelha,HIGH);
}
if (holdingRegs[VALVULAGLP] == 1) {
digitalWrite (rele_valvula_glp, LOW);
}
else {
digitalWrite (rele_valvula_glp,HIGH);
}
if (holdingRegs[COMPRESSOR] == 1){
digitalWrite (rele_compressor, LOW);
}
else {
```

```
digitalWrite (rele_compressor, HIGH);
}
if (holdingRegs[BOMBADAGUA] == 1){
digitalWrite (rele_bomba, LOW);
}
else {
digitalWrite (rele_bomba,HIGH);
}
}
}
```