

## PLANTSIM - UMA PLATAFORMA EM CÓDIGO LIVRE PARA DESENVOLVIMENTO DE PRÁTICAS DE ENSINO DE AUTOMAÇÃO INDUSTRIAL

**Geraldo Luis Bezerra Ramalho** - gramalho@ifce.edu.br

Instituto Federal do Ceará, Campus Fortaleza, Laboratório de Automação e Redes Industriais

**André Pimentel Moreira** - apmoreira@ifce.edu.br

Instituto Federal do Ceará, Campus Fortaleza, Laboratório de Sistemas Hidráulicos e Pneumáticos

**Resumo:** O PLANTsim foi desenvolvido para simular o comportamento de um processo industrial. Ele foi projetado para criar um gêmeo digital de uma planta de processo industrial didático usada para ensinar conceitos de automação e redes industriais. PLANTsim pode ser usado para demonstrar técnicas de automação e desenvolver programação de controlador industrial e habilidades de controle de supervisão. Este software foi desenvolvido em Python e será distribuído como uma plataforma open-source para ensinar conceitos e técnicas de automação industrial para estudantes de engenharia.

**Palavras-chave:** gêmeo digital, automação industrial, simulação de processo

### 1 INTRODUÇÃO

O ensino de tecnologias de automação industrial sempre é desafiador. As tecnologias estão em constante evolução, principalmente nos últimos 10 anos. Além disso, é difícil para as instituições de ensino, especialmente as públicas, manterem laboratórios didáticos que, pelo menos, acompanhem a velocidade com que as evoluções tecnológicas acontecem. Porém as aulas remotas apenas ampliaram o espectro das dificuldades no que se refere às aulas práticas, já carentes de recursos para a necessidade de desenvolver habilidades multidisciplinares (MARTINS *et al.*, 2019). Existem muitas soluções de software, comercialmente disponíveis, para atender a essa demanda. Também existem diversas alternativas de software gratuito (sejam livres ou versões de demonstração funcionais) para contornar essa dificuldade.

O PLANTsim foi criado para ser uma plataforma open-source para o desenvolvimento de atividades práticas de automação industrial, emuladas por computador. Por assim dizer, o PLANTsim pode ser usado para criar um gêmeo digital (SIEMENS, 2021) de uma planta de processo industrial com as características apresentadas por Medeiros *et al.* (2017) para um laboratório de curso de engenharia. O programa PLANTsim foi totalmente desenvolvido em linguagem Python e, portanto, é multiplataforma, rodando em computadores com sistema operacional Linux, Windows e macOS. No seu desenvolvimento foi priorizado o uso de bibliotecas públicas e somente aquelas essenciais para atender aos requisitos mínimos do programa. O seu código-fonte está em fase final de preparação para ser disponibilizado em plataformas de códigos open-source.

Como uma plataforma, o PLANTsim pode ser usado para estimular o desenvolvimento de trabalhos de conclusão de curso e agregar contribuições dos estudantes.



O restante deste trabalho é organizado como segue. A seção 2 descreve o conceito do PLANTsim e suas características principais. A seção 3 apresenta um exemplo de aplicação do PLANTsim em aulas práticas de uma disciplina de automação industrial em um curso de engenharia. A seção 4 apresenta as considerações finais e enumera as possibilidades de desenvolvimento.

## 2 PLANTsim

O simulador de planta de processos ou PLANTsim, é um programa desenvolvido em linguagem Python criado com o objetivo de emular os eventos de uma planta de processos industrial. O PLANTsim foi desenvolvido tendo em mente a necessidade de disponibilizar uma plataforma flexível e acessível para que alunos de cursos técnicos ou superiores possam desenvolver suas habilidades de programação de controladores programáveis, desenvolvimento de aplicações SCADA e tecnologias de redes industriais.

### 2.1 Características básicas

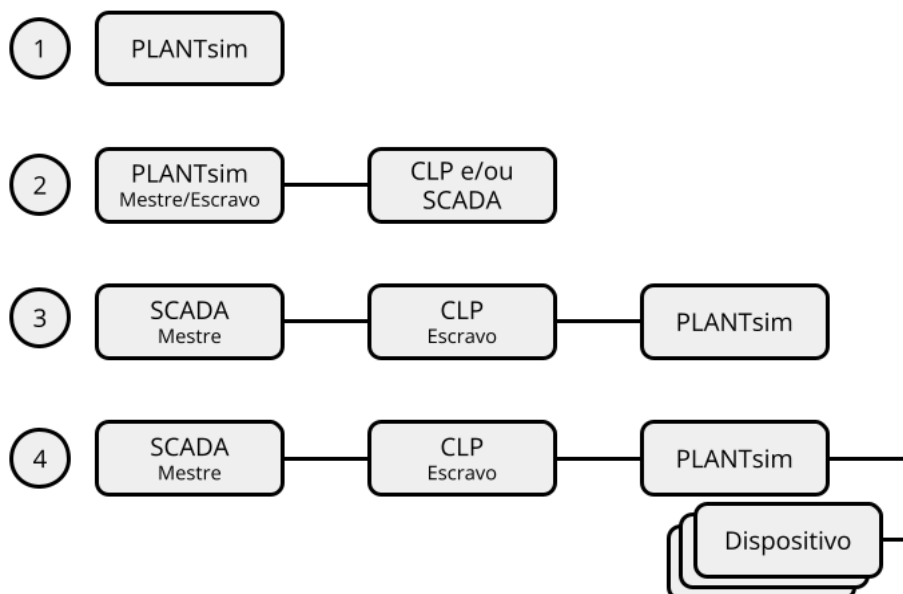
As principais características do PLANTsim visam incorporar a flexibilidade da definição de diferentes plantas de processos industriais que produzam sinais e que respondam a comandos. Dessa forma, um estudante usando um software de programação de CLP (controlador lógico programável) no padrão IEC 61131-3 que seja capaz de simular a comunicação em MODBUS TCP, um software SCADA (*supervisory control and data acquisition*) que adote as tecnologias recentes de comunicação e, preferencialmente, conceitos de desenvolvimento de IHM (interface homem-máquina) de alto desempenho (GOETZ, 2021), pode realizar as atividades de modelagem do processo, planejamento das etapas de projeto, documentação, programação e execução da programação, tudo isso usando seu próprio computador. Para isso, o programa PLANTsim disponibiliza:

- a modelagem de dispositivos de uma planta de processos emulando tanques, sensores e atuadores, encadeamento, ruído e vazamentos;
- comunicação com aplicação SCADA e software de simulação de programa do CLP usando protocolo MODBUS TCP;
- simulação de sequência pré-definida de eventos da planta;
- programação de ações dirigidas por eventos, usando uma versão adaptada da linguagem GRAFCET;
- possibilidade de encadear dispositivos (bombas, válvulas, medidor de vazão) e também construir processos com caminhos alternativos;
- os eventos ocorrem em tempo real, sendo que sua duração ou o intervalo entre os eventos depende da parametrização da planta.

O diagrama da Figura 1 ilustra as principais possibilidades de uso do PLANTsim em práticas de laboratório de automação. No modo 1, o PLANTsim pode ser usado sem nenhuma outra dependência, para modelagem da planta, simulação de eventos e programação de lógica de controle usando o GRAFCET adaptado para texto. Nos modos 2 e 3, o PLANTsim é utilizado como um dispositivo de controle ativo ou como gêmeo digital de uma planta de processos. No modo 4, ainda não implementado, o PLANTsim funcionará com interface entre os controladores e os dispositivos físicos, respondendo a eventos de sensores ou enviando sinal de controle para atuadores.



Figura 1 – Principais modos de uso do PLANTsim.



Fonte: o autor

## 2.2 Interface com o usuário

A interface do PLANTsim é bastante simples. A entrada da configuração é realizada por meio de arquivos texto, editáveis com qualquer editor de texto disponível nativamente nos sistemas operacionais. As informações de saída são apresentadas no console em formato texto, como ilustrado na Figura 2. Cores são utilizadas para facilitar a identificação das informações. Os dados são apresentados segundo a referência de um controlador digital. Ou seja, os dados binários de leitura (entradas digitais, coluna Sensor), dados binários de escrita (saídas digitais, coluna Actuator), dados tipo word de leitura (como entradas analógicas e registradores) e de escrita (registradores tipo *setpoint* e variáveis manipuladas). Variáveis manipuladas e variáveis de processo são apresentados na forma numérica e em barra (ver coluna Word). Os identificadores dos objetos (memórias tipo sensor, actuator ou word) são intuitivos e uma descrição é detalhada na seção seguinte.

Vale ressaltar que a interface com o usuário é simples para não acrescentar complexidade desnecessária ao programa e também porque um dos objetivos do ensino por meio do PLANTsim é que o estudante desenvolva seu próprio IHM usando softwares SCADA.





Figura 2 – Exemplo de interface com o usuário do  
PLANTsim.

```

Sensor (r/o)      Actuator (r/w)      Word (0-9 r/o; 10-19 r/w)
0 tk0.lshh >95    0 tk0.fv0 ->level    0 10 .....100 tk0.level.pv
1 tk0.lsh >90     1 tk0.ty1 ->temp     1 24 .....100 tk0.temp.pv
2 tk0.lsl >20     2 tk0.ty2 ->temp     2 50 .....100 tk2.level.pv
3 tk0.tsh >70     3 tk2.fc02 0% ->level 3 24 .....100 tk2.temp.pv
4 tk2.lshh >95    4 tk2.mix ->temp     4 0 .....15 tk2.flowin2.pv
5 tk2.lsh >90     5 pip23.fv23 ->sis.aux123 5 0 .....100 tk3.level.pv
6 tk2.tsh >70     6 tk3.fc34 0% ->tk4.level 6 21 .....100 tk3.temp.pv
7 tk3.lsh >80     7 tk3.mix ->temp     7 10 .....100 tk4.level.pv
8 tk3.lsl >10     8 filler.fc4 0% ->tk4.level 8 25 .....100 tk4.temp.pv
9 tk3.tsh >60     9 .....              9 0 .....10 bkt.level.pv
10*tk3.tsl <50    10 .....             10 .....
11 tk4.lsh >50     11 tk3.fv43 0% ->level 11 .....
12 tk4.lsl >10     12 tk0.fc40 0% ->level,tk3.fc43 0% 12 0 ..... 1 sis.aux102.pv ?fv102
13 tk4.tsh >60     13 tk0.fv40 ->level 13 0 ..... 1 sis.aux123.pv ?fv123
14*tk4.tsl <50    14 .....             14 0 ..... 1 sis.aux143.pv ?fv143
15 bkt.lsh >9     15 filler.fv4 ->aux4 15 0 ..... 1 tk0.aux40.pv ?fv40,tk3.aux43.pv ?fv43
16 sis.lah12 >70  16*sis.fv102 ->tk2.level 0 .....
17 sis.lah13 >70  17 .....             1 .....
18 .....          18*sis.fv123 ->tk3.level 2 .....
19 .....          19*sis.fv143 ->tk3.level 3 .....
20 .....          20 .....             4 .....
21 .....          21 .....             5 .....
22 .....          22 .....             6 .....
23 .....          23 .....             7 .....
24 .....          24 .....             8 .....
25 .....          25 .....             9 .....
26 .....          26 .....            10 .....
27 .....          27 .....            11 .....
28 .....          28 .....            12 .....
29 .....          29 ctrl.bkt 13 .....
30 .....          30 ctrl.reset 14 .....
31 bkt.unity      31 ctrl.shtd 15 .....

| PLANTsim v0.1 T/O 567s | 350cps | SLAVE listening on port 1502 | PLANTsim_LARI_1.plant

```

Fonte: o autor

### 2.3 Modelo de planta de processos

No PLANTsim, um modelo de planta de processo, fica registrada em um arquivo-texto com extensão .PLANT. Neste arquivo são descritas as unidades da planta, compostas basicamente de variáveis de processo, sensores e atuadores com as seguintes características:

- atuadores com controle PID podem ser autocontrolados (basta definir o *setpoint* e ligar o dispositivo) ou podem ter a variável manipulada controlada pelo CLP ou pelo SCADA.
- Controladores PID possuem ganhos ajustáveis. Controle de atuadores via PID, *on-off* ou diretamente pela variável manipulada.
- Controlador PID (biblioteca *simple-pid*<sup>1</sup>, adaptada)
- Tipos de variável de processo definido por função de callback: ex. **generic()**, **flow\_rate()**, **reactor()**, **hydraulic()**
- Identificadores no formato <unidade>.<dispositivo>
- Atuadores tipo: **controller**, **relay**
- Sensores tipo: **switch**
- Fluxo de produto definido pelas cláusulas **from\_pv** e **to\_pv**
- Possibilidade de encadear dispositivos e também construir caminhos alternativos

<sup>1</sup> <https://github.com/m-lundberg/simple-pid>



A Figura 3 ilustra a definição da unidade **tk0** de uma planta, representando um tanque com chaves de nível e termostato (switch), bomba ou válvula posicionadora (controller) e válvulas solenóide, atuadores de aquecimento e resfriamento (relay). Em destaque os identificadores (negrito), as seções de uma unidade (itálico e sublinhado), palavras reservadas (itálico) e tipos de propriedades, sensores ou atuadores (sublinhado).

Figura 3 – Exemplo de definição de uma unidade de uma planta.

```
tk0
  property
    unity enabled true sequence 0 label tank_0
    level type process_var pv 10 wordr_pv 0 limits 0,100 sys generic()
    noise 0.002 loss 0
    temp type process_var pv 25 wordr_pv 1 limits 0,100 inertia 5
    sys generic() noise 0.001 loss 0.02
    aux40 type process_var wordr_pv 15 if fv40
  actuator
    fv0 type relay active false bit 0 flow 0.2 from_pv source.level
    to_pv level inlet true label onoff_valve
    ty1 type relay active false bit 1 flow 0.2 from_pv temp inlet true
    label onoff_heater
    ty2 type relay active false bit 2 flow 0.3 from_pv temp inlet
    false label onoff_cooler
    fc40 type controller active false bit 12 flow 0.3 ctrl PID()
    ctrl_par 0.5,0.00,0 ctrl_pv level from_pv tk4.level to_pv aux40
    inlet true label ctrl_valve
    fv40 type relay active false bit 13 flow 0.1 from_pv aux40 to_pv
    level inlet true label ctrl_valve
  sensor
    lshh type switch active false bit 0 from_pv level > 0.95
    lsh type switch active false bit 1 from_pv level > 0.9
    lsl type switch active false bit 2 from_pv level > 0.2
    tsh type switch active false bit 3 from_pv temp > 0.7
```

Fonte: o autor

## 2.4 Comunicação em rede MODBUS

São disponibilizados os modos de comunicação mestre ou escravo usando o protocolo MODBUS TCP (biblioteca pyModbusTCP<sup>2</sup>). O simulador responde como um dispositivo físico em rede, com dados organizados em memória no padrão MODBUS, conforme ilustrado na Figura 2. O MODBUS TCP foi escolhido por ser um protocolo aberto e para facilitar seu uso em redes de computadores no padrão Ethernet com TCP/IP.

## 2.5 Simulação de sequência eventos

O simulador integra um gerador de sequência de eventos (arquivo .SIM) baseado em condições pré-definidas, usando o tempo real decorrido e o estado das memórias (sensores e atuadores). Esse simulador é útil para verificar e resolver inconsistência no modelo da planta.

<sup>2</sup> <https://pymodbustcp.readthedocs.io>



Porém, como ilustra a Figura 4, ele também pode ser utilizado para definir um comportamento específico para a planta a fim de testar condições previstas ou não no programa de controle do CLP ou no sistema de supervisão em uma aplicação SCADA.

Figura 4 – Exemplo de uma definição de sequência de eventos.

```
at 22 tk2.fc.active false
if tk3.lsh.active = True tk3.fv.active false
at 24 link_2_3.fc.active true
at 40 tk3.level.sp 2000
```

Fonte: o autor

## 2.6 Programação GRAFCET

O PLANTsim possui um interpretador GRAFCET interno, que pode ser utilizado para controlar a planta, simulando a lógica de controle do CLP. Uma linguagem GRAFCET adaptada para texto foi desenvolvida para escrever o programa GRAFCET em arquivos de texto (arquivo .PRG). Essa adaptação mantém as mesmas estruturas da sintaxe da linguagem GRAFCET: etapa-ação, transição-receptividade. Como ilustrado na Figura 5, estão disponíveis ações contínuas, temporizadas, com atraso, e condicionais. As transições suportam sincronismo e concorrência.

Figura 5 – Exemplo de um trecho de programa GRAFCET.

```
step 0 true
  action cont tk3.fv.active=true, tk3.fc.active=true
  transition 0 to 10,20 tk2.level.pv<=5
step 10
  action timed 10 tk3.fv.active=false
  transition 10 to 30 true
step 20
  action if tk3.lsh.active = false then tk3.fv.active
  transition 20 to 30 True
step 30
  action cont tk3.fv.active=false
  transition 30 to 40 true
step 40
  action cont tk2.fc.active=true , tk3.level.sp = 4
  transition 40 to 0 tk2.level.pv>=7
```

Fonte: o autor

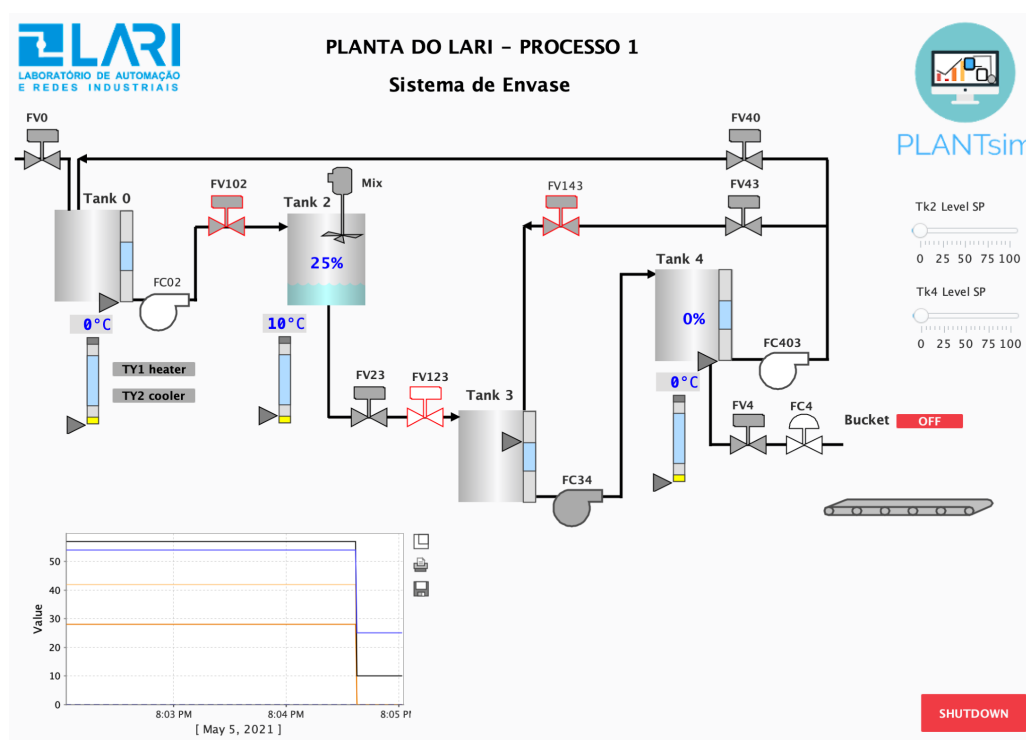
## 3 Exemplo de atividade prática

O emprego do PLANTsim em atividades de ensino obviamente depende de um bom planejamento das aulas. O programa deve estar facilmente acessível para os estudantes. Por isso,

uma versão compilada fica disponível na página do laboratório, junto com o link para download dos programas complementares: software de programação do CLP e software de programação da aplicação SCADA.

A Figura 6 ilustra a tela de um IHM desenvolvido para uso em conjunto com o PLANTsim. Neste exemplo, o software Ignition SCADA<sup>3</sup> (mestre) comunica com o PLANTsim (escravo) enviando os comandos dos atuadores ao serem clicados pelo estudante. As seguintes práticas podem ser desenvolvidas: planejamento e elaboração de IHM, configuração de TAGs e da comunicação serial, verificação e teste da sequência de etapas do processo.

Figura 6 – Exemplo de sistema SCADA da simulação da planta física do laboratório.



Fonte: o autor

As Figuras 7 e 8 ilustram, respectivamente, um trecho de programa LADDER e o resultado da simulação do programa do CLP Modicom M221 no software fornecido pelo fabricante, neste caso o software EcoStruxure Machine Expert Basic<sup>4</sup>. A tela de um IHM desenvolvido para uso em conjunto com o PLANTsim. Neste exemplo, o software SCADA (mestre) comunica com o PLANTsim (escravo) enviando os comandos dos atuadores ao serem clicados pelo estudante. As seguintes práticas podem ser desenvolvidas: planejamento e elaboração de IHM, configuração de TAGs e da comunicação serial, verificação e teste da sequência de etapas do processo. As atividades práticas desenvolvidas neste exemplo são: programação do CLP em linguagem LADDER, Lista de Instruções ou SFC (*sequential function chart*); a simulação, teste e diagnóstico de erros de lógica do

<sup>3</sup> <https://inductiveautomation.com>

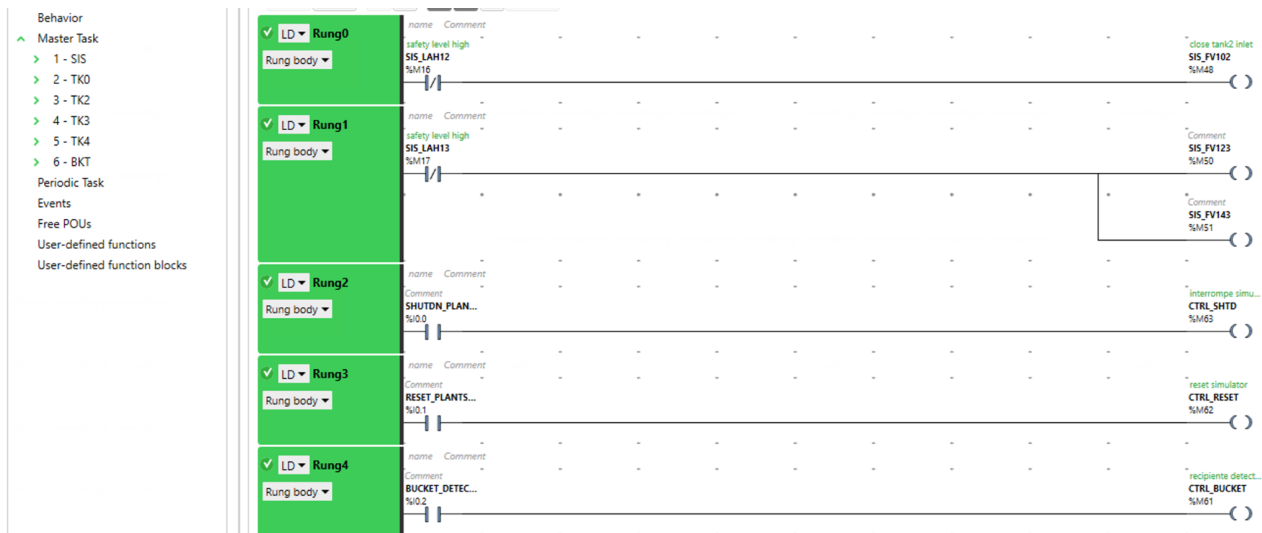
<sup>4</sup> <https://lab4sys.com/en/download-ecostruxure-machine-expert-basic-v1-0-sp2>





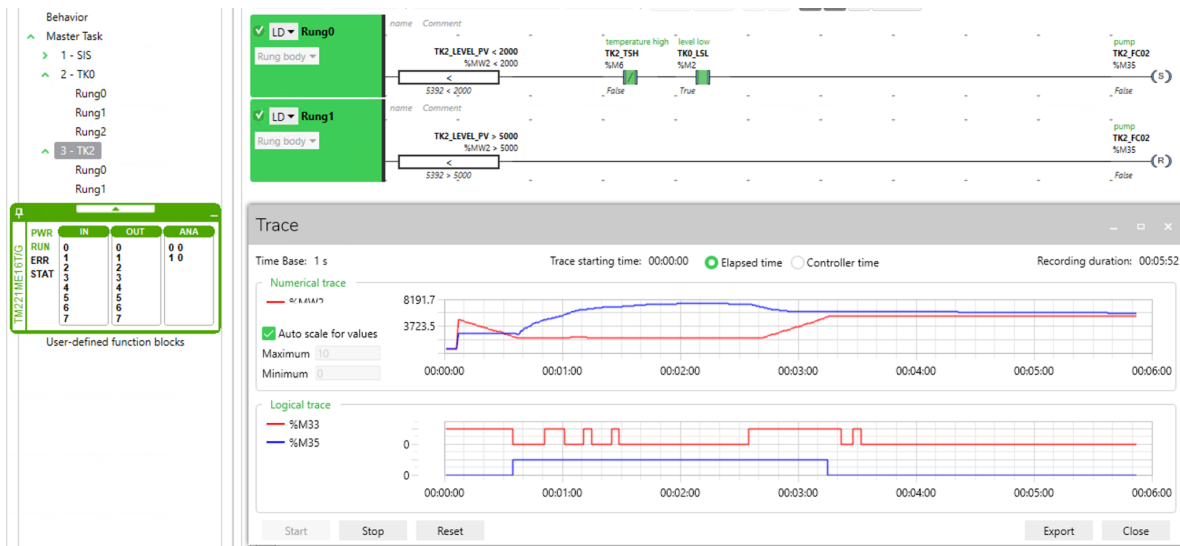
programa de controle; o acompanhamento simultâneo em um IHM e o teste das funções de uma aplicação SCADA para controle supervisório.

Figura 7 – Exemplo de programa LADDER.



Fonte: o autor

Figura 8 – Exemplo de programa LADDER - gráfico da simulação do programa controlando o PLANTsim.



Fonte: o autor

## 4 Considerações FINAIS

O PLANTsim foi desenvolvido, inicialmente, para atender à necessidade das aulas remotas de laboratório. No entanto, o seu conceito é anterior e mais amplo pois ele pode ser usado para criar um gêmeo digital de plantas didáticas, ou até mesmo processos industriais. Isto permite que o







estudante tenha um contato inicial com as principais características da planta simulada: comunicação em rede industrial, ajuste de *setpoint*, ajustes de posição de sensores, diagnóstico de falhas de sequência de operação, experiência aprimorada na utilização dos softwares dos CLPs e SCADA. Após o primeiro contato com o gêmeo digital, o estudante poderá desenvolver suas habilidades aplicando seus conhecimentos na planta física do laboratório, porém com mais confiança e diminuindo a chance de erros. Além disso, o estudante não fica restrito à planta do laboratório e pode agregar sua própria experiência às práticas, modelando sua proposta de planta ou uma planta conhecida.

O PLANTsim ainda está em desenvolvimento. Atualmente está na versão beta e, por isso, ainda em testes com a primeira turma da disciplina de Automação Industrial. No entanto, em breve estará disponível em uma plataforma de código-livre para que a comunidade de desenvolvedores, inclusive alunos e professores, possa melhorá-lo e ampliar suas funcionalidades. Dentre as principais melhorias projetadas para o PLANTsim estão uma visualização gráfica do modelo da planta de processos e do diagrama GRAFCET; a implementação de características físicas como a transferência de calor e outras não linearidades inerentes aos processos industriais; melhoria na apresentação do código fonte e padronização e ampliação do uso de funções *callback* permitindo que o estudante crie suas próprias funções de controle (Fuzzy, por exemplo).

O passo seguinte para tornar o PLANTsim um gêmeo digital robusto da planta didática do laboratório de automação será a implementação de reproduções de sinais reais coletados diretamente dos equipamentos da planta, assim como a reprodução desses sinais usando algoritmos de inteligência artificial.

Todas as funcionalidades, presentes e futuras, fazem do PLANTsim uma plataforma bastante útil para o desenvolvimento de atividades complementares e trabalhos de conclusão de curso.

### ***Agradecimentos***

Agradeço a todos os meus alunos de Automação Industrial e Laboratório de Automação Industrial que, por meio das suas dúvidas e sugestões, ajudaram a desenvolver as ideias conceituais por trás deste trabalho.

### **REFERÊNCIAS**

GOETZ, Heicker F. **Metodologia para Desenvolvimento de IHMs de Alta Performance Visual**. ELIPSE Software. Disponível em:

<https://kb.elipse.com.br/metodologia-para-desenvolvimento-de-ihms-de-alta-performance-visual/>.

Acesso em: 06 maio 2021.

MARTINS, Geovani. ALVES Jr, James G. V. Proposta Interdisciplinar na Educação do Engenheiro de Mecatrônica. In: XLVII Congresso Brasileiro de Educação em Engenharia, 2019, Fortaleza. **Anais**.



**COBENGE**  
2021

XLIX Congresso Brasileiro  
de Educação em Engenharia  
e IV Simpósio Internacional  
de Educação em Engenharia  
da ABENGE

28 a 30 de SETEMBRO

Evento Online

"Formação em Engenharia:  
Tecnologia, Inovação e Sustentabilidade"

MEDEIROS, Cláudio M. de S; SOUSA, José R. de B.; ALBUQUERQUE, Pedro U. B. de. Uma Experiência Pedagógica Vivenciada Em Curso de Automação Baseada em Ações Multidisciplinares Sobre Planta Industrial Didática. In: XLV Congresso Brasileiro de Educação em Engenharia, 2017, Joinville. **Anais**.

SIEMENS. **Gêmeos Digitais - O Futuro da Indústria**. Estadão. Disponível em:  
<http://patrocinados.estadao.com.br/siemens/digitalizacao/gemeos-digitais-o-futuro-da-industria/>.  
Acesso em: 06 maio 2021.

## PLANTSIM - AN OPEN-SOURCE FRAMEWORK TO DEVELOP PRACTICES FOR INDUSTRIAL AUTOMATION LEARNING

**Abstract:** *PLANTsim was developed to emulate the behaviour of an industrial process. It was designed to create a digital twin of a didactic industrial process plant used to teach automation and industrial networks concepts. PLANTsim can be used to demonstrate automation techniques and develop industrial controller programming and supervisory control skills. This software was developed in Python and will be distributed as an open-source platform to teach industrial automation concepts and techniques for engineering students.*

**Keywords:** *digital twin, industrial automation, process simulation*

Promoção:



Realização:

