INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TRIÂNGULO MINEIRO

CAMPUS UBERABA PARQUE TECNOLÓGICO

GABRIEL PAREDES FERREIRA

VITOR AUGUSTO GONÇALVES REIS

KELMSON LEANDRO RODRIGUES

AMBIENTE DE SIMULAÇÃO PSC EM PYTHON: ESTEIRA SELETORA DE CAIXAS

Uberaba - MG

Junho 2025

GABRIEL PAREDES FERREIRA
VITOR AUGUSTO GONÇALVES REIS
KELMSON LEANDRO RODRIGUES

AMBIENTE DE SIMULAÇÃO PSC EM PYTHON: ESTEIRA SELETORA DE CAIXAS

Trabalho relacionado ao curso de Engenharia de Computação, do Instituto Federal de Educação, Ciência e Tecnologia do Triângulo Mineiro, como parte dos requisitos da disciplina Controladores Programáveis.
Docente: Robson Borges Rodrigues

Uberaba - MG
Junho 2025

Índice

1. Introdução

Os controladores lógicos programáveis são dispositivos eletrônicos que atuam na monitoração e controle de máquinas e processos. No contexto da automação industrial, ele é projetado para ser capaz de atuar em ambientes diferentes e, por isso, são programáveis em uma determinada linguagem lógica. Dentre suas capacidades, destaca-se a de armazenar e executar instruções para automatizar tarefas, como lógica, sequenciamento, temporização, contagem e operações matemáticas. Em termos simples, o CLP (controlador lógico programável) atua como o cérebro de um sistema de automação industrial, recebendo informações de sensores, processando-as com base em um programa pré-definido e enviando sinais para atuadores que controlam as máquinas.

Em vista disso, o projeto final da disciplina é a criação de um ambiente de simulação de um CLP no computador. Estes tipos de ferramentas, como o LogixPro, são úteis para os projetistas testarem primeiro a criação de códigos de CLP antes de colocar a construção do protótipo em prática. Em muitos destes programas, é possível encontrar ambientes de simulação pré-programados com as entradas e saídas da aplicação, facilitando o aprendizado e teste desse tipo de dispositivo.

O projeto visa garantir a utilização de funções lógicas básicas (NOT, OR e AND), memórias booleanas locais, temporizadores com tempo baseado em 0,1 segundo e contadores. Além disso, o projeto também deve respeitar o ciclo de varredura comum de um CLP, sendo ele: inicializar o sistema, ler as entradas e armazenar na memória imagem, processar o programa do usuário e salvar as alterações da saída na memória imagem de saída e atualizar as saídas a partir da memória imagem de saída.

2. Objetivos

2.1 Objetivo Geral

Simular o comportamento de um CLP (Controlador Lógico Programável), um interpretador de linguagem de Lista de Instruções (IL), variáveis internas (entradas, saídas, memórias, temporizadores e contadores) e um ambiente gráfico simulando uma esteira seletora de caixas.

2.2 Objetivos Específicos

- Projetar e programar um simulador baseado no LogixPro (outra aplicação);
- Implementar código limpo para uma aplicação rápida e segura;
- Implementar um ambiente de simulação baseado na seleção de caixas por peso médio e pesado;
- Disponibilizar o projeto em um repositório público para a posterioridade.

3. Desenvolvimento

3.1 Estrutura do projeto

Arquivo / Módulo	Função Principal
CLPcore	Núcleo lógico do CLP: entradas, saídas, timers, contadores e modo de varredura.
ILInterpreter	Interpretador de linguagem IL (Lista de Instrução).
GUI	Interface gráfica com botões, LEDs, editor de código e simulação.
open_simulation_window	Simulação visual da esteira seletora.
main	Inicia a aplicação com interface feita em biblioteca Tkinter.

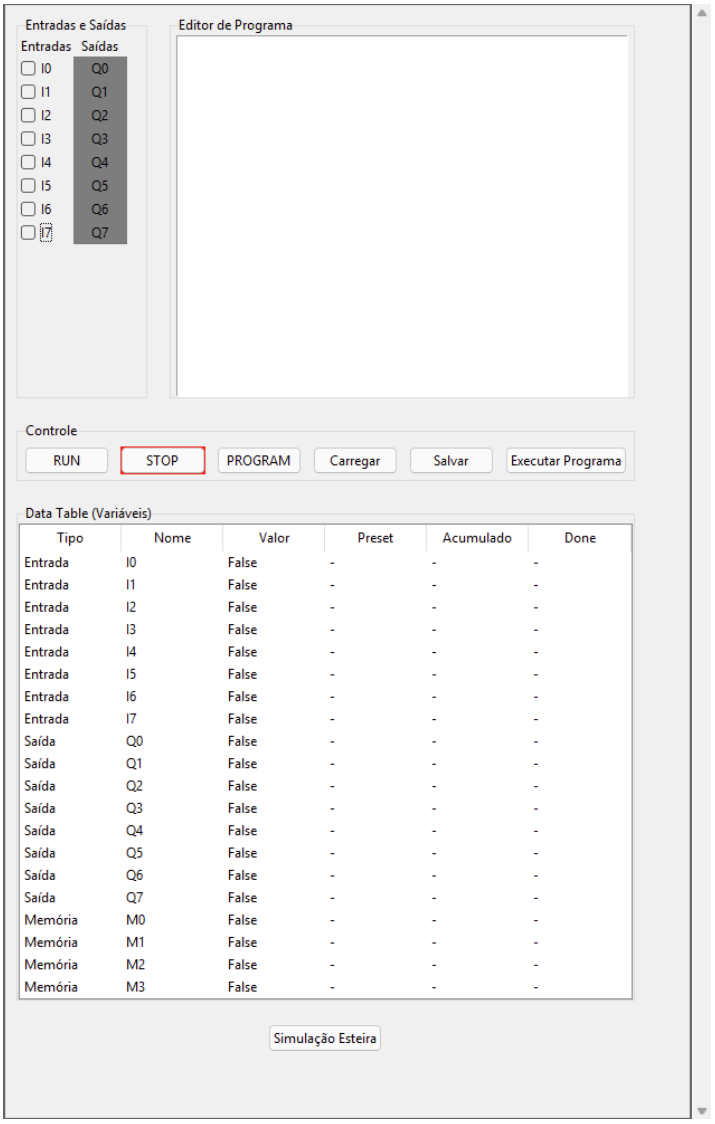
Para rodar a aplicação, basta baixar o projeto e rodar o arquivo gui.py. Utilizando a biblioteca pyinstaller, também fizemos a build da aplicação, o que significa que basta rodar o arquivo executável dentro da pasta “dist” para utilizar o programa. Exemplos de códigos prontos feitos em lista de instrução podem ser encontrados dentro da pasta “data” e carregados dentro do programa para testes.

3.2 Funções básicas do programa

- Simulação do CLP
    - Executa o ciclo de varredura típico de um CLP: lê entradas, executa o programa IL, atualiza temporizadores e contadores, atualiza saídas e repete o ciclo a cada 0,1s.
    - Permite simular entradas digitais (I0–I7) e saídas digitais (Q0–Q7) com interface gráfica.
- Interpretador de Linguagem IL
    - Suporta comandos IL como LD, AND, OR, NOT, OUT, além de temporizadores (TON, TOF) e contadores (CTU, CTD).
    - Permite carregar, salvar e executar programas IL personalizados pelo usuário.
- Interface Gráfica (CLPGUI)
    - Interface com botões para RUN/STOP/PROGRAM, checkbuttons para entradas, caixas que acendem em verde para saídas e editor de código IL.
    - Exibe Data Table com todas as variáveis do sistema em tempo real (entradas, saídas, memórias, temporizadores, contadores).
- Simulação Visual da Esteira
    - Simula uma esteira seletora de caixas por peso médio e pesado com sensores e atuadores.
    - Mostra o desvio de caixas médias e pesadas conforme a lógica do programa IL.

- Contadores e Memórias Internas
    - Mantém contadores de caixas totais, passadas, desviadas, médias e pesadas, mapeando para memórias internas (M10, M11, M20, etc).
- Botões Manuais
    - Entradas manuais (I5, I6, I7) para testes e controle direto de funções específicas, dentro do ambiente simulado da esteira.

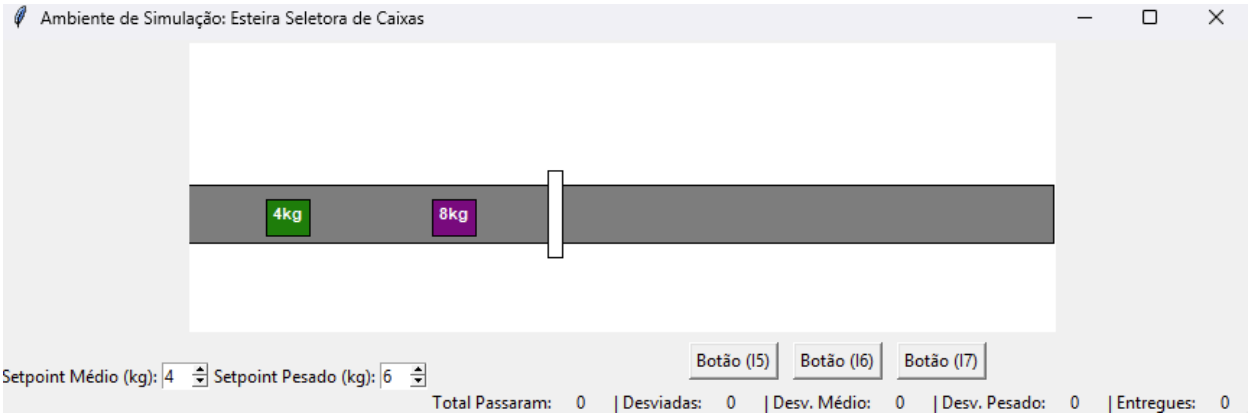
3.3 Interfaces e funcionamento do ambiente de simulação



Interface principal ao iniciar a aplicação. A primeira coisa a se fazer é desmarcar todas as entradas (I0 ao I7). Nota-se que ele inicia-se em modo STOP, no qual não se pode carregar nenhum exemplo de código da pasta “data” e nem digitar um código novo. Primeiro, é necessário iniciar o modo de programação pelo botão “PROGRAM”. O botão “Carregar” é responsável por carregar um código de exemplo já disponível ou carregar um código criado pelo usuário. Para criar e salvar um código, basta digitá-lo na caixa de Editor de Programa e clicar no botão “Salvar”, que inicia o explorador de arquivos do sistema operacional para que o usuário escolha o diretório em que seu código será salvo. Após escolher seu código, é necessário clicar no botão “Executar Programa” para que as instruções de sua lista sejam registradas nas

memórias, que serão mostradas preenchidas em Data Table. Caso o código interaja com alguma saída (representada por Q0 até Q7), o fundo cinza da saída se tornará verde, sinalizando o seu acionamento. Por fim, basta clicar em “RUN” para iniciar a simulação, e seu código está pronto para interagir com as entradas e saídas — e, caso projetado para isso, com o ambiente de simulação também.

A simulação será aberta ao clicar em “Simulação Esteira”.



O programa abrirá uma nova tela que mostra uma esteira e um retângulo no meio, simbolizando o ponto de fechamento da esteira e onde estão os sensores. No lado esquerdo da esteira são criadas caixas de pesos variados: azuis (2kg), verdes (4kg), laranjas (5kg) e roxas (8kg). A esteira se dirige da esquerda para a direita.
Os seletores de setpoint médio e pesado servem para modificar a maneira como os sensores médio e pesado interagem com as caixas. Caso o setpoint médio esteja em 4kg e o pesado em 8kg, todas as caixas de peso entre 4kg e menores que 8kg serão desviadas para baixo (ativa-se o pistão médio graças a detecção do sensor médio). Todas as caixas de peso 8kg ou maior serão desviadas para cima (ativa-se o pistão pesado graças a detecção do sensor pesado).
Contadores abaixo da tela evidenciam a totalidade de caixas, as caixas desviadas, as caixas desviadas pelo pistão médio, as caixas desviadas pelo pistão pesado e as caixas entregues (leves, ou seja, as que não sofreram desvio). Esses contadores são especiais ao ambiente de simulação e salvo nas memórias a seguir:

- M10: caixas peso médio desviadas;
- M11: caixas peso pesado desviadas;
- M20: total de caixas que passaram;
- M21: caixas que foram desviadas (desviadas médio e desviadas pesado);
- M22: total de caixas entregues.

Esquema das outras entradas e saídas da simulação:

- Caixas de diferentes pesos circulam pela esteira.;
- Sensor de presença (I1) detecta caixas;
- I2 / I3 indicam se a caixa é média ou pesada;

- Q1 aciona o fechamento do portão da esteira;
- Q2 aciona o pistão para desviar caixas médias;
- Q3 aciona o pistão para desviar caixas pesadas;
- A lógica de desvio depende das saídas do CLP.

3.4 Comandos IL suportados

- Lógicos
  - LD, LDN, AND, ANDN, OR, ORN, NOT, OUT
- Temporizadores
  - TON Tn X → Temporizador On Delay
  - TOF Tn X → Temporizador Off Delay
- Contadores
  - CTU Cn X → Contador ascendente
  - CTD Cn X → Contador descendente

3.5 Exemplos de códigos IL

Codificações e suas descrições dos códigos disponíveis na pasta “data”.

```
# // Código 1: Esteira sempre ligada, desvia apenas caixas pesadas.
  LD TRUE
  OUT Q1
  LD I1
  AND I3
  OUT Q3
  NOT
  OUT Q1

# // Código 2: Desvia médias e pesadas.
  LD I5
  OUT Q1
  LD I1
  AND I2
  OUT Q2
  NOT
  OUT Q1
  LD I1
  AND I3
  OUT Q3
  NOT
  OUT Q1
```

// Código 3: Manual com Botões - I5 desbloqueia a esteira, I6 desvia médias e I7 pesadas.

```
LD I5
OUT Q1
LD I6
OUT Q2
LD I7
OUT Q3
```

// Código 4: Depois de 30 segundos os pesados passam e as médios não.

```
LD TRUE
OUT Q1
LD TRUE
TON T0 300
LD T0
OUT Q2
LD I1
AND I3
ANDN T0
OUT Q3
```

// Código 5: Deixa passar tudo, até contar 5 caixas, depois aciona os 2 pistões (médias e pesadas.

```
LD TRUE
OUT Q1
LD M20
ANDN M10
CTU C0 5
LD M20
OUT M10
LD C0
OUT Q2
LD C0
OUT Q3
```

4. Conclusão

O aplicativo atende integralmente aos requisitos funcionais do trabalho final, simulando um CLP com interface gráfica, interpretador IL, modos de operação, visualização de variáveis e simulação visual da esteira seletora. Possíveis melhorias plausíveis são a criação de perfis de caixa com peso ajustável, a exportação dos dados da simulação para CSV e a criação de ais cenários de simulação. Observa-se que, graças a boa integração do programa e sua robustez, é plenamente possível adicionar mais cenários de simulação evitando retrabalho, visto que o ambiente de simulação da esteira utiliza as próprias memórias e recursos do interpretador, entradas e saídas.

Conclui-se que, ainda, por optar na criação de um novo simulador em uma linguagem que o grupo está familiarizado, flui-se melhor produção e, por tratar-se de uma aplicação criada pelos mesmos alunos até o presente momento do envio, seus recursos são mais ágeis e robustos, visto o conhecimento da totalidade do projeto e não somente de uma parte.

Na bibliografia, e também no envio final, estão disponíveis um vídeo de demonstração de uso do simulador e o repositório final do projeto.

5. Referências bibliográficas

THE LEARNING PIT. LogixPro PLC Simulator: Setting the Standard in PLC Hands-On Training. LogixPro Simulator. Disponível em: https://canadu.com/lp/logixpro.html. Acesso em: 27 jun. 2025.

FERREIRA, Gabriel Paredes. *Tutorial como usar CLP* [vídeo]. YouTube, 28 jun. 2025. Disponível em: https://youtu.be/MqdqrlUBxjM. Acesso em: 28 jun. 2025.

FERREIRA, Gabriel Paredes; REIS, Vitor Augusto Gonçalves; FERREIRA, Kelmson Leandro de Carvalho. *TrabalhoFinalCLP-2025-01* [repositório]. GitHub, 2025. Disponível em:https://github.com/Pareedes/TrabalhoFinalCLP-2025-01. Acesso em: 28 jun. 2025.