##### 1. Objetivo Geral do Projeto

Desenvolver um sistema automatizado que:

* Controle a **temperatura** de um forno industrial.
* Monitore **pressão** e **vazão de gás**.
* Ative alarmes e desligue o sistema em caso de **falhas** (por segurança).

**🔧 2. Variáveis do Sistema**

**➤ Variáveis Analógicas (valores contínuos):**

1. **Temperatura interna do forno (0–1000°C)**
   * Monitorada constantemente.
   * Se estiver fora da faixa, deve acionar alarmes.
2. **Pressão interna (0–10 bar)**
   * Deve manter-se em níve is seguros.
   * Pressão alta ativa alarme e desliga o sistema.
3. **Vazão de gás (0–100 L/min)**
   * Controla o consumo e segurança do gás.
4. **Volume do forno (2000 litros)**
   * 1 m (altura) × 1 m (largura) × 1 m (profundidade) = **1 m³**

**➤ Variáveis Digitais (ligado/desligado):**

1. **Válvula de entrada de gás (Input)**
   * Controla a entrada de gás; deve ser desligada em emergências.
2. **Sistema de aquecimento (Input)**
   * Aciona os elementos que aquecem o forno.
3. **Alarme de pressão alta (Output)**
   * Indicador visual ou sonoro.
4. **Alarme de falha no sensor de temperatura (Output)**
   * Detecta se o sensor está fora do ar ou danificado.
5. **Relé de segurança (Input)**
   * Se ativado, desliga todo o sistema por segurança.

**🧠 3. Lógica do Controle no SPDSW**

Você usará **diagrama LADDER** (LD) ou **Texto Estruturado** para programar.

**➤ Exemplo de Lógica para Alarme de Pressão Alta:**

ladder

CopiarEditar

IF Pressão >= 9.0 bar THEN

Ativar\_Alarme\_Pressão\_Alta := TRUE;

Desligar\_Válvula\_Gás := TRUE;

Desligar\_Aquecimento := TRUE;

ENDIF

**➤ Detecção de Falha no Sensor de Temperatura:**

* Se o valor for 0°C por muito tempo, considerar falha.
* Ou usar uma entrada digital de *watchdog* do sensor.

**➤ Ação do Relé de Segurança:**

ladder

CopiarEditar

IF Relé\_Segurança\_Ativado THEN

Desligar\_Válvula\_Gás := TRUE;

Desligar\_Aquecimento := TRUE;

Ativar\_Alarme\_Falha\_Grave := TRUE;

ENDIF

**🖥️ 4. Supervisório**

* Interface gráfica para monitorar variáveis em tempo real.
* Permite **simular falhas** (ex.: sensor fora do ar, válvula travada).
* Deve mostrar: temperatura, pressão, alarmes, status de válvulas e sensores.

**⚠️ 5. Pontos de Atenção (Alertas Importantes)**

1. **Segurança primeiro**: o sistema deve **desligar automaticamente** em qualquer situação de risco (pressão alta, falha de sensor, ativação do relé de segurança).
2. **Debounce digital**: evite falsos acionamentos de sensores digitais.
3. **Watchdogs**: monitore se os sensores estão funcionando corretamente.
4. **Simulações no supervisório**: devem ser claras e reversíveis (ex.: simular falha e restaurar).
5. **Escalamento correto**: para sensores analógicos, configure o escalamento para representar corretamente os valores reais (0–1000°C, etc.).

**💻 6. Programando no SPDSW (Altus) – Passo a Passo**

1. **Criar novo projeto**: selecione o modelo do CLP (ex.: ZAP91X).
2. **Definir variáveis**: insira todas as variáveis com tipo (AI, AO, DI, DO).
3. **Criar blocos de lógica**: use LADDER para implementar alarmes, desligamento e controle.
4. **Simular**: use a função de simulação para testar sem o CLP físico.
5. **Carregar no CLP**: após testes, carregue o programa no equipamento.
6. **Integrar com supervisório**: via Modbus ou protocolo Altus (se aplicável).

Se quiser, posso:

* **Montar os trechos de código LADDER** com base no projeto.
* **Criar um fluxograma lógico** para facilitar a visualização da lógica.
* **Ajudar na configuração de escalonamento de sensores analógicos**.

Você quer que eu gere um exemplo do programa no formato LADDER para SPDSW com base nesse projeto?