

Briefing técnico — Desenvolvimento do CLP & IHM

****Projeto****: Mesa de medida/corte com batente motorizado (stop)

****Responsável pelo desenvolvimento****: Eduardo

****Entrega combinada****: valor R\$ ****250,00**** com pagamento de 50% na primeira reunião dia 18/10/2025 e ; resultados alcançados e tudo estiver pronto e aprovado ****até 30/10/2025** fica como bonificação por comprimento de prazo + R\$ 250,00 adicionais.

****Softwares**** - □ ****IHM****: Delta ****DOPSoft 4.0**** (DOP-107BV) - □ ****CLP****: Delta ****ISPSOft 3.22**** — ****Modelo do CLP****: ***DVP14SS2*** (ou equivalente). ****Confirmar a nomenclatura exata na etiqueta da CPU**** (ex.: DVP14SS2, DVP14SS211T).

1) Escopo funcional (visão limpa, sem código)

1.1 Modos de operação - □□ ****Manual (medida única)****: operador digita uma medida (mm) e confirma; o batente posiciona e aguarda corte. - □□ ****Automático (lista digitada)****: operador informa uma lista de medidas (mm) + quantidades; o sistema executa em sequência. - □□ ****Automático (receita via USB/CSV)****: operador carrega arquivo ****CSV**** (pendrive). Cada linha contém: `COD_PERFIL, MEDIDA_MM, QTD, TIPO_PECA`.

> ****Não usar agora****: impressão de etiquetas/relatórios.

1.2 Fluxo geral de ciclo 1. □ ****Selecionar modo**** (Manual / Automático Digitado / Automático CSV). 2. □ ****Posicionamento****: CLP calcula pulsos a partir da medida alvo e aciona o driver (*pulse/dir*). Perfil trapezoidal com limites de velocidade e aceleração configuráveis. 3. □ ****Chegada na posição****: considerar janela de tolerância configurável (ex.: ±0,3 mm). 4. ∞ ****Aguardar corte****: - □ ****Sensor de presença de peça**** (no início da esteira): - □ ****Antes do corte****: sensor ativado = peça presente. - □ ****Após retirada****: ao ****ficar livre****, contar atraso de segurança de ****2 s**** antes de iniciar próximo movimento (evitar colisão do batente com peça). - □ ****Se Automático****: ao detectar ****retirada**** da peça, ****decrementar QTD**** daquela linha. Se QTD chegar a 0, avançar para a próxima linha. - □ ****Se Manual****: permanece na medida até novo comando do operador. 5. □ ****Repetição / Próxima****: - □ Automático: segue a sequência até terminar. - □ Manual: aguarda nova medida.

1.3 Calibração e referência (ponto zero) - ⚙ Na tela ****Configurações**** haverá: - □ ****JOG ← / JOG →**** (passo contínuo) para conferir sentido do motor e aproximar o batente. - □ ****Zerar na posição atual****: grava a posição atual do batente como ****ponto zero lógico**** (em ****mm****), com possibilidade de informar ****offset de zero**** (ex.: zerar dizendo que esta posição é ****1000 mm**** ou ****1200 mm****). - □ ****Parâmetro PULSOS_POR_MM**** para conversão (ver 3.2).

1.4 Regras e detalhes importantes - □ ****Unidades**** sempre em ****mm**** na IHM. Internamente pode ser pulsos. - □ ****Atraso pós-corte****: ****2 s**** após o sensor indicar peça retirada, antes do próximo movimento (configurável). - □ ****Janela de posição****: tolerância configurável (mm) para considerar que “chegou”. - □ ****Fila de execução****: no automático, seguir a ordem das linhas. Permitir ****pausar**** e ****retomar****. - □ ****Contagem de peças****: cada retirada válida decrementa a ****QTD restante****. - □ ****Falhas**** (ver 2.5): tratar alarme do driver, falta de sensor, erro de seguimento/timeouts.

2) IHM — Telas mínimas e elementos

2.1 Telas (enxutas, conforme pedido) 1. □ ****HOME**** - □ ****Botões****: ****Selecionar Operação****, ****Configurações****. - □ ****Status rápidos****: posição atual (mm), modo, estado do driver (pronto/alarme), sensor de peça (on/off). 2. □ ****Selecionar Operação**** - □ ****Opções****: ****Manual****, ****Automático (Digitado)****, ****Automático (CSV/USB)****. 3. □

****Manual (Medida única)**** - □ ****Campo****: ****Medida alvo (mm)****. - □ ****Botões****: ****Mover****, ****Parar**** (parada suave), ****Voltar****. - □ ****Indicadores****: posição atual, tolerância, sensor peça, driver pronto/alarme. - □ não permitir movimentar batente se peça detectada no sensor, 4. □ ****Automático (Digitado)**** - □ ****Tabela editável****: ****MEDIDA_MM**** | ****QTD**** (linhas múltiplas). - □ ****Botões****: ****Iniciar****, ****Pausar/Retomar****, ****Cancelar****, ****Voltar****. - □ ****Indicadores****: linha atual, QTD restante da linha, próxima medida, progresso. 5. □ ****Automático (CSV/USB)**** - □ ****Botões****: ****Carregar CSV (USB)****, ****Iniciar****, ****Pausar/Retomar****, ****Cancelar****, ****Voltar****. - □ ****Grid**** somente leitura: ****COD_PERFIL | MEDIDA_MM | QTD | TIPO_PECA****. - □ ****Indicadores****: linha atual, QTD restante, progresso. 6. ⚙ ****Configurações / Calibração**** - □ ****Jog ← / Jog →**** (contínuo, pressionar=move; soltar=para) - □ ****Definir PULSOS_POR_MM**** (inteiro/real) - □ ****Definir velocidade (mm/s)**** e ****aceleração (mm/s²)**** - □ ****Definir tolerância (mm)**** - □ ****Definir atraso pós-corte (s)**** - □ ****Zerar na posição atual**** (campo opcional para ****definir esta posição como X mm****) - □ ****Salvar parâmetros**** - □ (Opcional) ****Diagnóstico I/O****: X/Y/M bits e alarmes

> Observação: Evitar multiplicidade de telas. O ****Diagnóstico**** pode ser uma aba dentro de ****Configurações****.

2.2 CSV de Receita (USB) - □ ****Formato****:

`COD_PERFIL,MEDIDA_MM,QTD,TIPO_PECA` - □ ****Exemplo****: D-078,753,5,VERGA
SU-111,1200,2,PEITORIL TUB-4599,455,8,BARRA - □ ****Validações****: - □ `MEDIDA_MM > 0`,
`QTD >= 1` - □ Ignorar linhas vazias/comentários - □ Exibir relatório de erros de leitura na IHM (linhas inválidas)

2.3 Estados e comandos IHM ****Comandos****: START, PAUSE, CANCEL, MOVE_MANUAL, JOG_LEFT, JOG_RIGHT, SET_ZERO, LOAD_CSV.

****Indicadores****: POS_MM_ATUAL, ALVO_MM, TOL_MM, SENSOR_PECA, DRIVER_READY, DRIVER_ALARM, MODO, LINHA_ATUAL, QTD_RESTO.

2.4 Comportamentos específicos - □ ****Jog****: ativo enquanto o botão estiver pressionado (momentâneo). Definir velocidade de jog (pode usar a velocidade nominal ou parâmetro dedicado menor). - □ ****Parar****: parar suave (desaceleração). Não é stop ativo travado. - □ ****Pausa****: congela a sequência sem perder progresso; ****Retomar**** continua da mesma linha/quantidade.

2.5 Alarmes e tratativas - □ ****ALM_DRIVER****: entrada de alarme do driver easy-servo. Ação: bloquear movimento, exibir mensagem, exigir Reset. - □ ****TIMEOUT_POS****: não atingiu posição dentro de tempo máximo (parametrizável). Ação: parar e exibir erro. - □ ****SENSOR_FAIL****: inconsistência do sensor (ex.: nunca ativa). Ação: alertar e permitir modo manual com confirmação do operador (configurável). - □ ****USB_CSV_ERR****: erro de leitura/parse do CSV. Ação: listar linhas inválidas.

3) CLP — Lógica e mapeamento (alto nível, sem código)

3.1 Máquina de estados (sugestão) - □ IDLE → aguardando comando/mode - □ MOVE → gera pulsos até alvo (considerar DIR e ENABLE) - □ SETTLE → aguarda estabilização/tolerância - □ WAIT_CUT → espera retirada da peça (sensor = livre) e aplica atraso de 2 s - □ NEXT → decrementa QTD/avança linha - □ PAUSED → pausa segura (saída de passo inativa, posição mantida) - □ FAULT → bloqueia até reset

3.2 Conversão mm ↔ pulsos - □ Parâmetro: PULSOS_POR_MM = (passo por volta × microstep) / (avanço_mm_por_volta) - ⚙ Se houver relação por correia/cremalheira/polia, incorporar relação na fórmula. - □ Conversão: PULSOS_ALVO = round(MEDIDA_MM × PULSOS_POR_MM) - □ Exemplo mecânico desta máquina (informado): HTD-8M, polia 16 dentes → avanço linear por volta ≈ 8 mm × 16 = 128 mm. Assim: - □ PULSOS_POR_MM =

$(\text{passos_por_volta} \times \text{microsteps}) / 128$. (Ex.: motor 200 passos/volta, microstep=1600 → $200 \times 1600 / 128 \approx 2500$ pulsos/mm)

3.3 Interface com driver (easy-servo) - □ Saídas: Y_STEP, Y_DIR, Y_ENA - □ Entradas: X_ALM (alarme driver), X_RDY (pronto, se disponível) - □ Sensor de peça: X_SPC (NPN/PNP conforme hardware)

3.4 Intertravamentos - □ Bloquear geração de pulso se ALM_DRIVER=1 ou ENA=0. - □ Manual: não movimentar se SENSOR_PECA=1 (peça presente). Exibir aviso ao operador. - □ Fins de curso: se FC_INICIO=1 e comando for para o início, inibir movimento; se FC_FIM=1 e comando for para o fim, inibir movimento. Jog somente no sentido de liberação. - □ Em PAUSE/CANCEL, garantir Y_STEP=0 e manter Y_ENA ligado (configurável) para segurar posição.

3.5 Temporizações - □ T_SETTLE (ms) após chegada dentro da tolerância - □ T_POST_CUT = 2000 ms (parametrizável) - □ T_TIMEOUT_POS para falha de posicionamento

3.6 I/O e memória — sugestão de mapa (Delta DVP) Entradas (X) - □ X0 = SENSOR_PECA (presença) - □ X1 = DRIVER_READY (opcional) - □ X2 = DRIVER_ALARM - □ X3 = FC_INICIO (fim de curso início) - □ X4 = FC_FIM (fim de curso fim)

Saídas (Y) - ☼ Y0 = STEP - □ Y1 = DIR - □ Y2 = ENABLE - □ Y3 = LÂMPADA/BUZZER (opcional)

Bits internos (M) - ► M100 = CMD_START - □ M101 = CMD_PAUSE - □ M102 = CMD_CANCEL - □ M103 = CMD_MOVE_MANUAL - ◀ M104 = CMD_JOG_LEFT - ► M105 = CMD_JOG_RIGHT - □ M110 = STATE_IDLE - □ M111 = STATE_MOVE - □ M112 = STATE_SETTLE - ⚡ M113 = STATE_WAIT_CUT - □ M114 = STATE_NEXT - □ M115 = STATE_PAUSED - △ M116 = STATE_FAULT

Palavras (D) - □ D1000 = POS_MM_ATUAL (float/int mm) - □ D1001 = ALVO_MM - □ D1002 = TOL_MM - □ D1003 = VEL_MM_S - □ D1004 = ACC_MM_S2 - □ D1005 = PULSOS_POR_MM - □ D1006 = VEL_JOG_MM_S (velocidade dedicada do JOG) - □ D1010 = PULSOS_ALVO - □ D1011 = PULSOS_ATUAL - □ D1020 = ATRASO_POS_CORTE_MS (2000) - □ D1030 = TIMEOUT_POS_MS - □ D1100..D1199 = Tabela de medidas/QTD (lista digitada) - □ D1200.. = Buffer da receita CSV parseada (somente leitura pela IHM)

Observação: No DOPSoft, mapear campos/teclados para estes D/M. Usar Enhanced Recipe apenas para exibir/editar lista digitada, se desejarem, ou manipular via grid padrão.

3.7 Tratamento do sensor de peça - □ Presença verdadeira: X0=1. - □ Para avançar no Automático: aguardar transição 1→0 (retirada), iniciar T_POST_CUT, ao expirar decrementar QTD e ir ao próximo alvo. - □ Em Manual: somente liberar novo movimento após transição 1→0 + atraso (ou botão do operador).

3.8 Pausa/Cancelar - □ Pausa: congela estado → PAUSED (desliga STEP; mantém ENABLE). - □ Cancelar: aborta sequência, limpa linha atual, vai para IDLE.

3.9 Reset de falhas - ☼ Botão RESET limpa STATE_FAULT quando ALM_DRIVER=0.

4) Periféricos e especificações IHM: Delta DOP-107BV (RS-232/RS-485/USB)

CLP: Delta DVP14SS2 (confirmar variante exata: ex. DVP14SS211T) — programado no ISPSOft 3.22

Motor/Driver: Easy-servo (modo pulse/dir) com encoder integrado modelo hybrid servo motorrnema 34 modelo 86hse4n-bc38 com driver hss86 hybrid servo driver

Sensor de presença de peça: proximidade indutivo LJ12A3-4-Z/BX— NPN

Sensor de presença fim de curso (início e fim do curso : proximidade indutivo LJ8A3-2-Z/BX— NPN polia para correia htd8m de 20mm 16t

Demais: buzzer/luz de torre opcional; fonte 24 Vdc; bornes; cabos blindados para STEP/DIR

Sinais elétricos principais - □ STEP, DIR, ENABLE (saídas do CLP para driver) - □ ALARM, READY (entradas no CLP, se disponíveis no driver) - □ SENSOR_PECA (entrada no CLP)

Observação: Mapa de ligação: entregar diagrama indicando bornes X/Y do CLP, pinos do driver (PUL+/PUL-, DIR+/DIR-, ENA+/ENA-), alimentação, e sensor (NPN PNP e resistores/COM adequados).

5) Requisitos de entrega (o que esperamos do Eduardo) 1. □ Projeto IHM (DOPSoft 4.0): - □ Arquivo do projeto (.dopprj) com as 6 telas acima, funcionamento testado em simulação, teclado numérico e grids conforme descrito. 2. □ Programa CLP (ISPSOft 3.22): - □ Projeto (.dvprj) com máquina de estados descrita, parametrizações e mapeamento X/Y/M/D conforme sugestão (ou equivalente documentado). - ✱ Geração de pulsos STEP/DIR com rampas; conversão mm↔pulsos via PULSOS_POR_MM. 3. □ Mapa de Ligações: - □ Diagrama elétrico com todas as conexões: CLP ↔ driver ↔ motor; sensor de peça; alimentação; aterramento; blindagem. 4. □ Arquivos & Exemplos: - □ CSV de exemplo (conforme 2.2) - □ Lista de parâmetros (defaults de velocidade, aceleração, tolerância, atraso pós-corte) - □ Manual curto (README) explicando sequência e ajustes. 5. □ Teste: - □ Simulação/bench básico que comprove: Manual funciona; Automático Digitado executa lista; Automático CSV lê arquivo e executa; decremento de QTD por retirada de peça; atraso de 2 s antes do próximo movimento; pausa/retomar; tratamento de alarme do driver.

Critérios de aceite (checklist) - [] □ Manual posiciona dentro da tolerância configurada e bloqueia movimento com SENSOR_PECA=1 - [] □ Automático (Digitado) executa todas as linhas e quantidades em ordem - [] □ Automático (CSV) carrega arquivo, valida e exibe grid corretamente - [] □ Decremento de QTD ocorre na retirada (transição SENSOR_PECA 1→0) + atraso 2 s antes do próximo - [] □ Jog ←/→ funciona, com inibição no sentido do fim de curso ativo - [] □ Zerar na posição atual com offset grava e persiste após reinício - [] □ Pausa/Retomar não perde progresso - [] □ Tratamento de ALM_DRIVER exige Reset antes de novo movimento - [] □ Timeout de posição gera falha e cancela movimento com mensagem clara

6) Observações e combinações comerciais - □ Sem impressão nesta fase. - □ Prazo: pronto e entregue até 30/10/2025. - □ Pagamento: R\$ 250,00 (50% na primeira reunião em 18/10/2025). Bônus de R\$ 250,00 adicionais se resultados forem alcançados e tudo estiver pronto e aprovado até 30/10/2025 (cumprimento de prazo). - □ Origem da solicitação: Éder Caligari — (11) 91343-9396.

7) Anexos a produzir pelo desenvolvedor - □ PROJ_IHM_DOPSoft_4_0.dopprj - □ PROJ_CLP_ISPSOft_3_22.dvprj - □ MAPA_LIGACOES.pdf (ou .dwg) - □ RECEITA_EXEMPLO.csv - □ README_RAPIDO.pdf

Nota final para Eduardo Qualquer divergência de endereçamento/tags pode ser ajustada, mas documento claramente a tabela final de X/Y/M/D na entrega. Priorizamos simplicidade e robustez: poucas telas, operação direta, e parâmetros acessíveis na Configuração/Calibração.

Saudações,

Éder Caligari