

## Informações

- Docente: Dimitri Campos Viana;
- Data de entrega: 10/03/2016;
- Pontos distribuídos: 17 pontos (2 no cronograma + 15 no software);
- Grupos: 2 alunos por grupo.

## Orientações Gerais

- Desenvolvimento da aplicação:
  - Utilizar a ferramenta WinCC (disponível no laboratório);
  - A aplicação deverá funcionar conforme solicitado. Recursos adicionais não serão avaliados. Ausência de funcionalidades, mal funcionamento e erros de compilação dos *scripts* serão penalizados;
  - Muitas vezes o enunciado não utiliza o esquema de “passo-a-passo” adotado nos tutoriais realizados nas aulas práticas da disciplina. Presume-se que os procedimentos mostrados nos tutoriais foram assimilados pelos alunos.
- Entrega da pasta de projeto contendo a aplicação desenvolvida:
  - O subdiretório “**Archive Manager**” deverá ser excluído antes da compactação da pasta principal. Isso diminuirá significativamente o tamanho do arquivo sem prejudicar seu projeto, visto que apenas os dados armazenados em histórico serão eliminados;
  - Confira se o “Arquivo.pca” está dentro da pasta principal;
  - A pasta principal do projeto deverá ser compactada no formato “.rar” ou “.zip”;
  - O nome do arquivo compactado deverá conter o nome dos integrantes do grupo;
  - O arquivo compactado deverá ser postado na sala virtual da disciplina (AVA).
- Caso seja caracterizada cópia (comentários iguais, erros peculiares iguais, cores e posição iguais de vários objetos), os grupos dividirão os pontos distribuídos entre si;
- Em caso de dúvidas, entre em contato por meio do endereço [dimitri@deg.ufla.br](mailto:dimitri@deg.ufla.br).

# Enunciado

A partir de uma aplicação base do WinCC, distribuída pelo professor, desenvolva um Sistema de Supervisão e Controle para a planta de controle de nível cujo programa de CLP foi desenvolvido no primeiro trabalho prático dessa disciplina. Para tal, siga os passos de desenvolvimento indicados a seguir.

## 1 Configurações Iniciais

1. Adeque o *Computer Name* da aplicação recebida de acordo com o *Computer Name* da sua estação de trabalho.
2. Configure os seguintes módulos para serem ativados em *Runtime*:
  - Global Script Runtime;
  - Alarm Logging Runtime;
  - Tag Logging Runtime;
  - Report Runtime;
  - Graphics Runtime.
3. Para impedir que o operador saia do sistema sem ter permissão para tal, desabilite as seguintes combinações de teclas:
  - Ctrl + Alt + Del;
  - Alt + Tab (deixe por último, apenas quando seu desenvolvimento terminar);
  - Ctrl + Esc;
  - Alt + Esc.
4. Configure a tela “Principal.pdl” como tela inicial e configure o sistema para rodar em modo *Full Screen*.

## 2 Comunicação

1. Primeiramente, configure uma comunicação entre o *PC Access (OPC Server)* e o CLP, conforme indicado no “Anexo A” (última seção deste documento);
2. No módulo de gerenciamento de *tags*, adicione o *channel driver* “OPC.chn”;
3. Clique com o botão direito no objeto “OPC Groups (OPCHN Unit #1)” e selecione a opção “System Parameter”;
4. Na janela “OPC Item Manager”, siga o caminho: \\ < Local > → S7200.OPCServer e clique no botão “Browse Server”;
5. Na janela “Filter Criteria”, apenas clique no botão “Next”;

6. Utilize a janela “S7200.OPCServer” para navegar pelo servidor. Ao encontrar o grupo “Alarmes”, selecione todas as variáveis contidas no mesmo e clique no botão “Add Items”;
7. Responda “Sim” para a pergunta “Do you want to create a suitable connection?” e quando o nome dessa conexão for solicitado, utilize o nome sugerido por meio de um clique no botão “Ok”. Quando a janela “Add Tags” for apresentada, apenas clique no botão “Finish”;
8. Repita o penúltimo passo até que as variáveis contidas em todos os grupos sejam cadastradas no *WinCC*.

### 3 Mensagens de Alarme

1. A tabela a seguir mostra as mensagens de alarme que devem ser cadastradas:

Tag	Message Text	Point of Error
CLP01_Inativo	Comunicação inativa	Controlador S7-1200
Alr_GERAL_Emerge	Emergência atuada	Geral
Alr_VT101_HorFim	Limite de horas excedido	Ventilador 1
Alr_VT102_HorFim	Limite de horas excedido	Ventilador 2
Alr_LT101_ExcedHH	Limite “Muito Alto” excedido	Nível de água no Tanque
Alr_LT101_ExcedH	Limite “Alto” excedido	Nível de água no Tanque
Alr_LT101_ExcedL	Limite “Baixo” excedido	Nível de água no Tanque
Alr_LT101_ExcedLL	Limite “Muito Baixo” excedido	Nível de água no Tanque
Alr_BA101_DisjOff	Disjuntor desarmado	Bomba de Água
Alr_BA101_HorFim	Limite de horas excedido	Bomba de Água

### 4 Armazenamento de Valores Decimais

1. A tabela a seguir mostra as variáveis decimais que devem estar cadastradas para armazenamento de histórico após o cumprimento desta tarefa:

Arquivo	Tag	Aquisição
Níveis	Ana_LT101_Nivel	Acíclica (on change)
Diversos	Ana_LC102_SinCtrl	Acíclica (on change)
Diversos	Par_LC102_Setpoint	Cíclica (1s)

## 5 *Scripts* Globais

1. Crie uma *Global Action*, utilizando a linguagem *C*, para monitorar a comunicação entre o Sistema de Supervisão e Controle e o CLP;
2. Escreva o *script* mostrado a seguir e faça as configurações necessárias para que ele seja executado a cada 1 segundo;
3. Salve a *Global Action* com o nome “CLP01\_Watchdog”.

```
// declaração de variáveis (note as estáticas)
float SinalCLP;
static float SinalAux;
static int TimeOut;

// leitura do sinal do CLP
SinalCLP = GetTagFloat("Ana_GERAL_AtvcPU");

// verificação de congelamento do sinal do CLP
if(SinalCLP == SinalAux)
{
    TimeOut++;
}
else
{
    TimeOut = 0;
}

// atualização do sinal auxiliar
SinalAux = SinalCLP;

// disparo do alarme após 5s de congelamento
if(TimeOut >= 5)
{
    SetTagBit("CLP01_Inativo",TRUE);
}
else
{
    SetTagBit("CLP01_Inativo",FALSE);
}

// retorno obrigatório da action
return 0;
```

## 6 Administração de Usuários, Senhas e Permissões

1. A tabela a seguir mostra os usuários e senhas que devem existir no sistema após o cumprimento desta tarefa:

Usuário	Senha	Grupo	Observação
Dimitri	123456	Administradores	
Joana	123456	Mantenedores	
João	123456	Mantenedores	
José	123456	Operadores	
Maria	123456	Operadores	
{Nome do aluno 1}	123456	Administradores	A ser criado por você
{Nome do aluno 2}	123456	Administradores	A ser criado por você

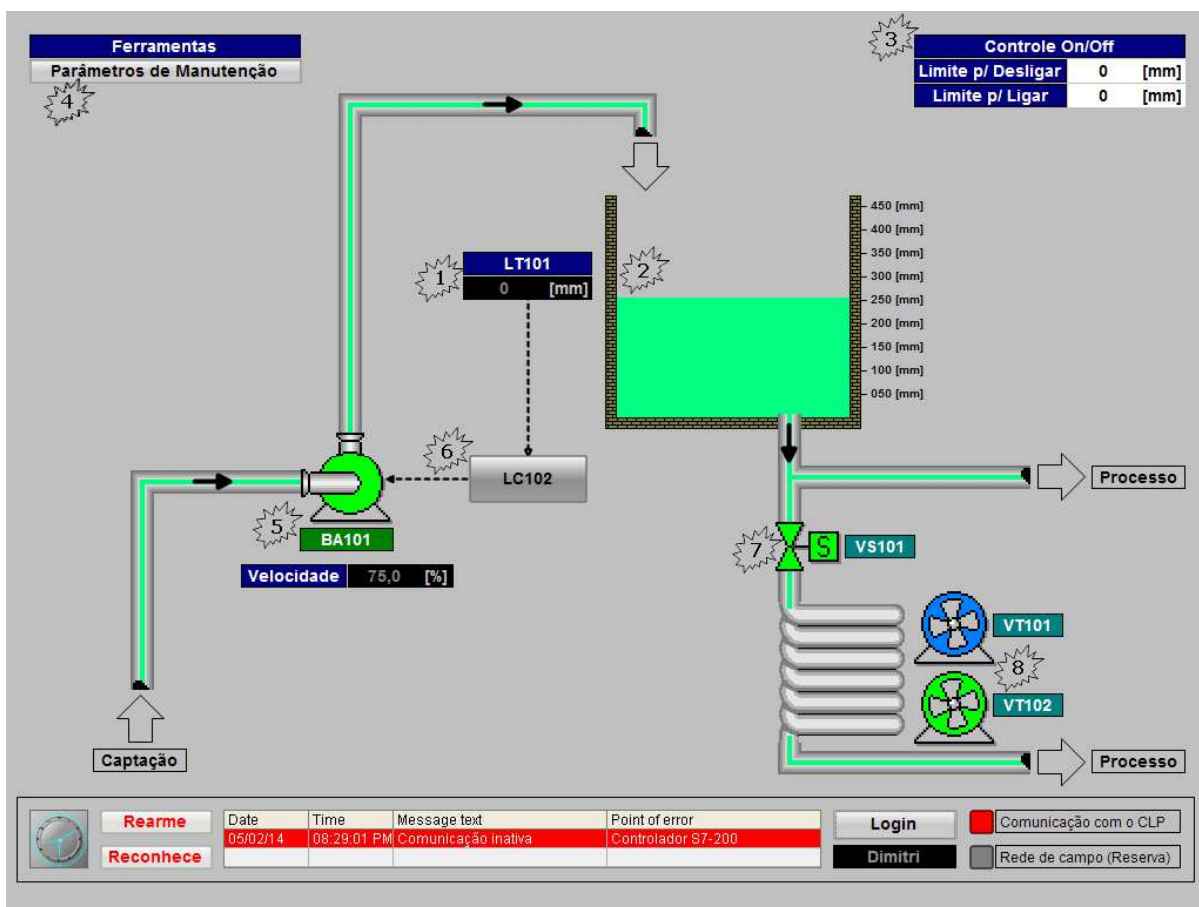
2. Associe os usuários recém-criados às seguintes permissões:

- “Operacao Basica”
- “Operacao Avancada”
- “Manutencao Basica”
- “Manutencao Avancada”
- “Administracao”
- “Engenharia”

## 7 Tela “Principal.Pdl”

Faça uma representação sinóptica do processo, similar a representação mostrada na figura abaixo. Para tal, siga a sequência de desenvolvimento indicada pelos “balões” inseridos na figura.

**Observação:** para alterar as cores de um objeto, primeiramente, é necessário alterar o valor da propriedade “Global Color Scheme” desse objeto para “No”.



1. Insira um campo para apresentar o valor do nível atual no tanque de água – Dois objetos do tipo *Static Text* devem indicar o *tag* de instrumentação do medidor e a unidade da medição, enquanto um objeto do tipo *I/O Field* deve apresentar efetivamente os valores medidos. Após inserir os objetos na tela, configure-os conforme indicado na próxima tabela.

Objeto	Atributo	Estático	Dinâmico	Evento
Static Text	Text	LT101		
	Tooltip Text	Nível no Tanque de água		
	Background Color	Azul escuro		
	Font Color	Branco		

Continua na próxima página ...

Objeto	Atributo	Estático	Dinâmico	Evento
I/O Field	Field Type	Output		
	Output Format	999		
	Output Value		Animação 1	
	Background Color	Preto		
	Font Color	Cinza Claro		
Static Text	Text	[mm]		
	Background Color	Preto		
	Font Color	Cinza Claro		

Adicionalmente, a animação indicada na tabela anterior deve ser implementada conforme mostrado a seguir.

Adicional	Tipo	Descrição	Execução
Animação 1	Tag	Ana_LT101_Nivel	1[s]

- Insira uma representação gráfica do nível atual no tanque de água – Um objeto do tipo *Bar* deve fornecer uma aproximação gráfica o valor medido. Os objetos *Graphic Object* e *Group* que, respectivamente, formam a moldura do tanque e a escala de nível, podem ser encontrados na tela “Biblioteca.Pdl”. Após inserir os objetos na tela, configure-os conforme indicado na próxima tabela.

Objeto	Atributo	Estático	Dinâmico	Evento
Bar	Minimum Value	0		
	Maximum Value	500		
	Process Driver Connection		Animação 1	
	Scale	No		
	Zero Point	0		
	Bar Background Color	Cinza claro		
	Scale Color	Cinza claro		
	Border Weight	0		
Graphic Object				
Group				

Adicionalmente, a animação indicada na tabela anterior deve ser implementada conforme mostrado a seguir.

Adicional	Tipo	Descrição	Execução
Animação 1	Tag	Ana_LT101_Nivel	1[s]

- Insira os campos para entrada de dados que parametrizam o controlador On/Off – Um objeto do tipo *Static Text* deve dar título aos dois conjuntos de entrada de dados que aparecem abaixo dele, cada um composto por: dois objetos do tipo *Static Text* que devem descrever o parâmetro que pode ser alterado, bem como sua unidade, e um objeto do tipo *I/O Field* que deve transcrever valores digitados pelo usuário para o CLP. Após inserir os objetos na tela, configure-os conforme indicado na próxima tabela.

Objeto	Atributo	Estático	Dinâmico	Evento
Static Text	Text	Controle On/Off		
	Background Color	Azul escuro		
	Font Color	Branco		
Static Text	Text	Limite p/ Desligar		
	Background Color	Azul escuro		
	Font Color	Branco		
I/O Field	Authorization	Operação Avançada		
	Low Limit Value		Animação 1	
	High Limit Value	500,0		
	Field Type	I/O Field		
	Output Format	999		
	Output Value		Animação 2	
	Background Color	Branco		
	Border Color	Branco		
	Font Color	Preto		
Static Text	Text	[mm]		
	Background Color	Branco		
	Border Color	Branco		
	Font Color	Preto		
Static Text	Text	Limite p/ Ligar		
	Background Color	Azul escuro		
	Font Color	Branco		
I/O Field	Authorization	Operação Avançada		
	Low Limit Value	0,0		
	High Limit Value		Animação 3	
	Field Type	I/O Field		
	Output Format	999		
	Output Value		Animação 4	
	Background Color	Branco		
	Border Color	Branco		
	Font Color	Preto		
Static Text	Text	[mm]		
	Background Color	Branco		
	Border Color	Branco		
	Font Color	Preto		

Adicionalmente, as animações indicadas na tabela anterior devem ser implementadas conforme mostrado a seguir.

Adicional	Tipo	Descrição	Execução
Animação 1	Tag	Par.LC101.LimLig	1[s]
Animação 2	Tag	Par.LC101.LimDsg	1[s]
Animação 3	Tag	Par.LC101.LimDsg	1[s]
Animação 4	Tag	Par.LC101.LimLig	1[s]

- Insira uma chamada para a tela “Parametros.Pdl” – Um objeto do tipo *Static Text* deve dar título ao botão que aparece abaixo dele e à outros possíveis botões que poderão ser inseridos futuramente para adicionar funcionalidades ao sistema. Um objeto do tipo *Button* deve permitir que uma janela destinada à apresentação da tela em questão seja chamada pelo usuário. Após inserir os objetos na tela, configure-os conforme indicado na próxima tabela.



Objeto	Atributo	Estático	Dinâmico	Evento
Static Text	Text	Ferramentas		
	Background Color	Azul escuro		
	Font Color	Branco		
Button	Text	Parâmetros de Manutenção		Ação 1

Adicionalmente, a ação indicada na tabela anterior deve ser implementada conforme mostrado a seguir.

Adicional	Tipo	Descrição	Execução
Ação 1	VBS-Action	<pre> Dim PicWin  Set PicWin = ScreenItems("PicWin_Tools")  PicWin.Visible = False PicWin.PictureName = "Parametros.Pdl" PicWin.CaptionText = "Parâmetros" PicWin.Top = 30 PicWin.Left = 180 PicWin.Visible = True PicWin.Screen.Activate </pre>	Mouse Action

**Observação:** o *script* apresentado na tabela anterior, quando associado ao evento “Mouse Action” do objeto *Button*, altera as propriedades do objeto “PicWin\_Tools” (já inserido pelo professor na tela “Principal.Pdl”). A sequência de alterações é a seguinte: 1) torna o objeto invisível, ocultando alguma outra janela que por ventura estiver sendo exibida; 2) configura o nome da tela a ser apresentada na janela; 3) configura o *caption text* da janela; 4) define as coordenadas que determinam a posição inicial da janela; 5) torna o objeto visível, apresentado a janela ao usuário; 6) Leva o “foco” do sistema operacional para a janela recém-aberta.

- Insira uma representação gráfica da bomba BA101 – Os objetos *Rectangle* e *Graphic Object* que, quando agrupados, formam a representação da bomba, podem ser encontrados na tela “Biblioteca.Pdl”. O primeiro, deve indicar os estados desse equipamento, por meio do seguinte código de cores: vermelho claro para “defeito” (primeira prioridade), amarelo claro para “bloqueado” (segunda prioridade), verde claro para “ligado” (terceira prioridade) e azul claro em último caso, ou seja, “desligado”. O segundo, além de simbolizar o equipamento, deve permitir que uma janela destinada ao controle do mesmo seja chamada pelo usuário. Por sua vez, um objeto do tipo *Static Text*, além de indicar o *tag* de instrumentação do atuador, deve sinalizar seu modo de operação, da seguinte forma: amarelo escuro para “modo manual” e verde escuro caso contrário, ou seja, “modo automático”. Finalizando, dois objetos do tipo *Static Text* devem descrever e fornecer a unidade da estimativa de velocidade, enquanto um objeto do tipo *I/O Field* deve apresentar efetivamente os valores estimados. Após inserir os objetos na tela, configure-os conforme indicado na próxima tabela.

Objeto	Atributo	Estático	Dinâmico	Evento
Rectangle	Background Color		Animação 1	
Graphic Object	Picture Transparent Color On	Yes		Ação 1
Static Text	Text	BA101		
	Background Color		Animação 2	
	Font Color	Branco		
Static Text	Text	Velocidade		
	Background Color	Azul escuro		
	Font Color	Branco		
I/O Field	Field Type	Output		
	Output Format	999,9		
	Output Value		Animação 3	
	Background Color	Preto		
	Font Color	Cinza Claro		
Static Text	Text	[%]		
	Background Color	Preto		
	Font Color	Cinza Claro		

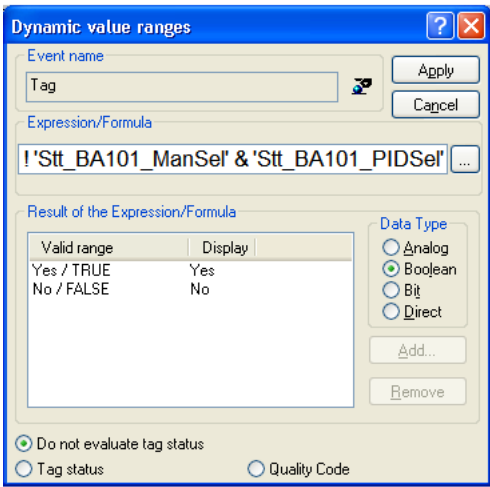
Adicionalmente, as animações e a ação indicadas na tabela anterior devem ser implementadas conforme mostrado a seguir.

Adicional	Tipo	Descrição	Execução
Animação 1	VBS-Action	<pre> Dim Tag01 Dim Tag02 Dim Tag03 Dim Defeito Dim Bloqueado Dim Ligado  Tag01 = "Stt_BA101_Defeito" Tag02 = "Stt_BA101_Bloq" Tag03 = "Stt_BA101_Ligada" Defeito = HMIRuntime.Tags(Tag01).Read Bloqueado = HMIRuntime.Tags(Tag02).Read Ligado = HMIRuntime.Tags(Tag03).Read  If Defeito Then     BackColor_Trigger = vbRed     Exit Function End If  If Bloqueado Then     BackColor_Trigger = vbYellow     Exit Function End If  If Ligado Then     BackColor_Trigger = vbGreen     Exit Function End If  BackColor_Trigger = RGB(0,128,255) </pre>	1[s]
Animação 2	VBS-Action	<pre> Dim Tag01 Dim Manual  Tag01 = "Stt_BA101_ManSel" Manual = HMIRuntime.Tags(Tag01).Read  If Manual Then     BackColor_Trigger = RGB(128,128,0) Else     BackColor_Trigger = RGB(0,128,0) End If </pre>	1[s]
Animação 3	Tag	Ana_BA101_VelBmb	1[s]
Ação 1	VBS-Action	<p>Após um clique no objeto <i>Graphic Object</i>, uma janela de controle deverá ser aberta. Para fazer isso, as propriedades do objeto <i>PicWin_Tools</i> (já inserido pelo professor na tela “Principal.Pdl”) devem ser modificadas em <i>Runtime</i>. Adapte o <i>script</i> desenvolvido anteriormente para abertura da janela de parâmetros para realizar essa tarefa.</p>	Mouse Action

6. Insira uma chamada para a janela de controle “Cmd\_LC102.Pdl” – Um objeto do tipo *Button*, deve simbolizar o controlador e permitir que uma janela destinada à operação do mesmo seja chamada pelo usuário. Dois objetos do tipo *Connector* devem simbolizar o fluxo dos sinais envolvidos nessa parte do sistema, ou seja, o sinal transmitido do sensor ao controlador e o sinal produzido por este último e enviado ao atuador. Após inserir os objetos na tela, configure-os conforme indicado na próxima tabela.

Objeto	Atributo	Estático	Dinâmico	Evento
Button	Text	LC102		Ação 1
Connector	Object name of the source	Static Text2		
	Connection point index of the source	2		
	Object name of the target	Button2		
	Connection point index of the target	0		
Connector	Object name of the source	Button2		
	Connection point index of the source	3		
	Object name of the target	GraphicObject3		
	Connection point index of the target	1		
	Display		Animação 1	

Em complemento, a animação e a ação indicadas na tabela anterior devem ser implementadas conforme mostrado a seguir.

Adicional	Tipo	Descrição	Execução
Animação 1	Dynamic Dialog		1[s]
Ação 1	VBS-Action	Após um clique no objeto <i>Button</i> , uma janela de controle deverá ser aberta. Para realizar essa tarefa, adapte o <i>script</i> desenvolvido anteriormente para abertura da janela de controle da bomba BA101.	Mouse Action

7. Insira uma representação gráfica da válvula VS101 – Os objetos *Rectangle* e *Graphic Object* que, quando agrupados, formam a representação da válvula, podem ser encontrados na tela “Biblioteca.Pdl”. O primeiro, deve indicar os estados desse equipamento, por meio do seguinte código de cores: verde claro para “aberta” e azul claro caso contrário, ou seja, “fechada”. O segundo, além de simbolizar o equipamento, deve permitir que uma janela destinada ao controle do mesmo seja chamada pelo usuário. Por sua vez, um objeto do tipo *Static Text*, além de indicar o *tag* de instrumentação do atuador, deve sinalizar seu modo de operação, da seguinte forma: magenta escuro para “modo local” e ciano escuro caso contrário, ou seja, “modo remoto”. Após inserir os objetos na tela, configure-os conforme indicado na próxima tabela.

Objeto	Atributo	Estático	Dinâmico	Evento
Rectangle	Background Color		Animação 1	
Graphic Object	Picture Transparent Color On	Yes		Ação 1
Static Text	Text	VS101		
	Background Color		Animação 2	
	Font Color	Branco		

Adicionalmente, as animações e a ação indicadas na tabela anterior devem ser implementadas conforme mostrado a seguir.

Adicional	Tipo	Descrição	Execução
Animação 1	C-Action	<pre>char* Tag01 = SysMalloc(49); BOOL Aberta;  Tag01 = "Stt_VS101_Aberta"; Aberta = GetTagBit(Tag01);  if(Aberta == TRUE) {     return CO_GREEN; } return CO_BLUE;</pre>	1[s]
Animação 2	C-Action	Para indicar que o modo local está selecionado, a cor de fundo desse objeto deve ser “CO_DKMAGENTA”. Já para o modo remoto, a cor deve ser “CO_DKCYAN”. Para tal, use o <i>tag</i> “Stt_GERAL_LocSel” e adapte o <i>script</i> apresentado logo acima.	1[s]
Ação 1	C-Action	<pre>char* Pic = SysMalloc(49); char* Obj = SysMalloc(49);  Pic = lpszPictureName; Obj = "PicWin_Tools";  SetVisible(Pic,Obj,FALSE); SetPictureName(Pic,Obj,"Cmd_VS101"); SetPropChar(Pic,Obj,"CaptionText","VS101"); SetTop(Pic,Obj,415); SetLeft(Pic,Obj,360); SetVisible(Pic,Obj,TRUE); Set_Focus("Cmd_VS101","Cmd_VS101");</pre>	Mouse Action

8. Insira a representação gráfica do ventilador VT101 – O objeto *Status Display*, que deve indicar os estados do equipamento e permitir que uma janela destinada ao seu controle seja chamada pelo usuário, pode ser encontrado na tela “Biblioteca.Pdl”. Várias figuras podem ser associadas à esse objeto e a seleção de qual delas será apresentada ocorre por meio de um índice, nesse caso: 0 corresponde à figura com o ventilador azul claro (equipamento desligado), 1 corresponde à figura com o ventilador verde claro (equipamento ligado) e 2 corresponde à figura com o ventilador vermelho claro (equipamento com defeito). Um objeto do tipo *Static Text*, além de indicar o *tag* de instrumentação do atuador, deve sinalizar seu modo de operação, da seguinte forma: magenta escuro para “modo local” e ciano escuro caso contrário, ou seja, “modo remoto”.

**Observação:** uma única janela de controle, a ser parametrizada no momento em que o usuário solicita sua abertura, será utilizada para operação dos Ventiladores VT101 e VT102. Para permitir que isso seja feito, primeiramente, você deve criar os seguintes *tags* internos:

Nome	Tipo de Dado
Aux_Local	Text tag 8-bit character set
Aux_Ligado	Text tag 8-bit character set
Aux_Defeito	Text tag 8-bit character set
Aux_Liga	Text tag 8-bit character set
Aux_Desliga	Text tag 8-bit character set
Aux_HorRst	Text tag 8-bit character set
Aux_HorDec	Text tag 8-bit character set

Após criar os *tags* e inserir os objetos na tela, configure-os conforme indicado na próxima tabela.

Objeto	Atributo	Estático	Dinâmico	Evento
Status Display	Current Status		Animação 1	Ação 1
Static Text	Text	VT101		
	Background Color		Animação 2	
	Font Color	Branco		

Adicionalmente, as animações e a ação indicadas na tabela anterior devem ser implementadas conforme mostrado a seguir.

Adicional	Tipo	Descrição	Execução
Animação 1	VBS-Action	<pre> Dim Tag01 Dim Tag02 Dim Defeito Dim Ligado  Tag01 = "Stt_VT101_Defeito" Tag02 = "Stt_VT101_Ligado" Defeito = HMIRuntime.Tags(Tag01).Read Ligado = HMIRuntime.Tags(Tag02).Read  If Defeito Then     Index_Trigger = 2     Exit Function End If  If Ligado Then     Index_Trigger = 1     Exit Function End If  Index_Trigger = 0 </pre>	1[s]
Animação 2	VBS-Action	<p>Para indicar que o modo local está selecionado, a cor de fundo desse objeto deve ser "RGB(128,0,128)". Já para o modo remoto, a cor deve ser "RGB(0,128,128)". Para tal, adapte o <i>script</i> desenvolvido anteriormente para indicação do modo de operação da bomba BA101.</p>	1[s]

Adicional	Tipo	Descrição	Execução
Ação 1	VBS-Action	<pre> Dim Tag01 Dim Tag02 Dim Tag03 Dim Tag04 Dim Tag05 Dim Tag06 Dim Tag07 Dim PicWin  Tag01 = "Stt_GERAL_LocSel" Tag02 = "Cmd_VT101_Liga" Tag03 = "Cmd_VT101_Deslig" Tag04 = "Cmd_VT101_HorRst" Tag05 = "Stt_VT101_Ligado" Tag06 = "Stt_VT101_Defeito" Tag07 = "Ana_VT101_HorDec" HMIRuntime.Tags("Aux_Local").Write Tag01 HMIRuntime.Tags("Aux_Liga").Write Tag02 HMIRuntime.Tags("Aux_Desliga").Write Tag03 HMIRuntime.Tags("Aux_HorRst").Write Tag04 HMIRuntime.Tags("Aux_Ligado").Write Tag05 HMIRuntime.Tags("Aux_Defeito").Write Tag06 HMIRuntime.Tags("Aux_HorDec").Write Tag07  Set PicWin = ScreenItems("PicWin_Tools")  PicWin.Visible = False PicWin.PictureName = "Cmd_Tipo01" PicWin.CaptionText = "Ventilador 1" PicWin.Top = 480 PicWin.Left = 490 PicWin.Visible = True PicWin.Screen.Activate </pre>	Mouse Action

**Observação:** Analogamente, repita os passos necessários para inserir a representação gráfica do ventilador VT102 e permitir que este seja operado por meio da mesma janela de controle utilizada para o ventilador VT101.

9. Insira os demais objetos gráficos que ajudam a representar o processo, conforme mostrado na figura que aparece no início dessa seção. Tais objetos não possuem nenhum tipo de animação e podem ser encontrados na tela “Biblioteca.Pdl”.



## 8 Tela “Rodape.Pdl”

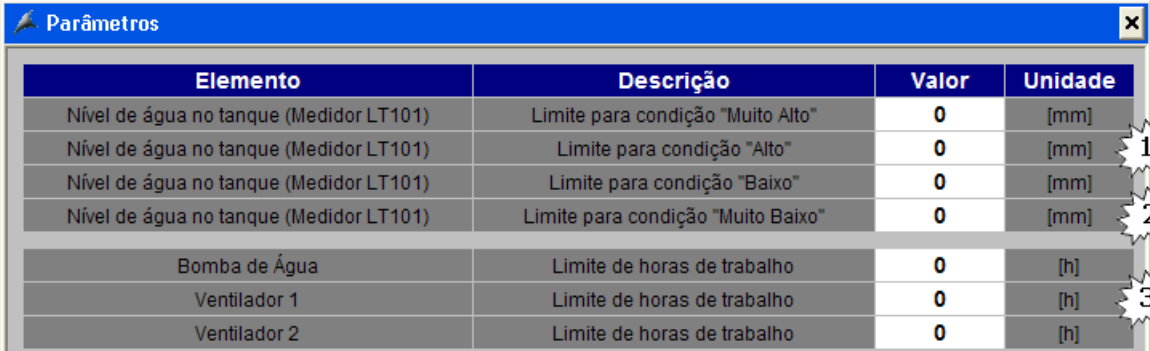
A tela “Rodape.Pdl”, que está incorporada na tela “Principal.Pdl” por meio de um objeto do tipo *Picture Window*, já foi desenvolvida pelo professor. Para se tornar completamente funcional, esta tela necessita apenas que os nomes de alguns *tags* sejam adequados nos seguintes locais:

Objeto	Atributo	Evento
Botão “Rearme”		Mouse Action

Após a adequação, analise os objetos utilizados nessa tela e entenda os objetivos e funcionalidades de cada um. Além disso, reflita: qual a vantagem de se utilizar uma *Picture Window* para incorporar a tela “Rodape.Pdl” na tela “Principal.Pdl” ao invés de configurar os objetos que você analisou diretamente na tela “Principal.Pdl”?

## 9 Tela “Parâmetros.Pdl”

Complete a janela usada para parametrizar a planta, de forma que a mesma fique similar à apresentada na figura abaixo. Para tal, siga a sequência de desenvolvimento indicada pelos “balões” inseridos na figura.



Elemento	Descrição	Valor	Unidade
Nível de água no tanque (Medidor LT101)	Limite para condição "Muito Alto"	0	[mm]
Nível de água no tanque (Medidor LT101)	Limite para condição "Alto"	0	[mm]
Nível de água no tanque (Medidor LT101)	Limite para condição "Baixo"	0	[mm]
Nível de água no tanque (Medidor LT101)	Limite para condição "Muito Baixo"	0	[mm]
Bomba de Água	Limite de horas de trabalho	0	[h]
Ventilador 1	Limite de horas de trabalho	0	[h]
Ventilador 2	Limite de horas de trabalho	0	[h]

1. Insira os objetos relacionados ao parâmetro “Limite para condição Alto” do “Nível de água no tanque (Medidor LT101)”:
  - Os objetos do tipo *Static Text* que indicam o elemento, a descrição e a unidade do parâmetro devem ter cor de fundo cinza escuro e cor do texto preto;
  - O objeto do tipo *I/O Field* destinado ao parâmetro em questão deve aceitar valores entre o valor que estiver no *tag* “Par\_LT101\_LimL” e o valor que estiver no *tag* “Par\_LT101\_LimHH”, devendo transcrever os valores digitados pelo usuário para o *tag* “Par\_LT101\_LimH”, a cada 1 segundo. A entrada de dados deve ser bloqueada para usuários que não tenham a permissão “Manutencao Avancada”.
2. Insira os objetos relacionados ao parâmetro “Limite para condição Muito Baixo” do “Nível de água no tanque (Medidor LT101)”:
  - Os objetos do tipo *Static Text* que indicam o elemento, a descrição e a unidade do parâmetro devem ter cor de fundo cinza escuro e cor do texto preto;

- O objeto do tipo *I/O Field* destinado ao parâmetro em questão deve aceitar valores entre zero e o valor que estiver no *tag* “Par\_LT101\_LimL”, devendo transcrever os valores digitados pelo usuário para o *tag* “Par\_LT101\_LimLL”, a cada 1 segundo. A entrada de dados deve ser bloqueada para usuários que não tenham a permissão “Manutencao Avancada”.
3. Insira os objetos relacionados ao parâmetro “Limite de horas de trabalho” do “Ventilador 1”:
- Os objetos do tipo *Static Text* que indicam o elemento, a descrição e a unidade do parâmetro devem ter cor de fundo cinza escuro e cor do texto preto;
  - O objeto do tipo *I/O Field* destinado ao parâmetro em questão deve aceitar valores entre 0 e 999.999, devendo transcrever os valores digitados pelo usuário para o *tag* “Par\_VT101\_HorLim”, a cada 1 segundo. A entrada de dados deve ser bloqueada para usuários que não tenham a permissão “Manutencao Avancada”.

## 10 Tela “Cmd\_BA101.Pdl”

Faça uma janela para controlar a bomba BA101, similar a apresentada na figura abaixo. Para tal, siga a sequência de desenvolvimento indicada pelos “balões” inseridos na figura.



1. Insira uma interface para o modo de operação do equipamento:
- O objeto do tipo *Button*, que seleciona o modo “Manual”, deve enviar este comando ao CLP escrevendo o valor “1” no *tag* “Cmd\_BA101\_SelMan” (use VBS-Action);
  - O objeto do tipo *Button*, que seleciona o modo “Automático”, deve ser análogo ao anterior;
  - O objeto do tipo *Static Text*, que indica o modo de operação do equipamento, deve ter a cor de fundo preto e, tanto a propriedade que determina o texto a

ser apresentado como a que determina a cor do mesmo, devem ser animadas em *Runtime* (use VBS-Action). Ambas as propriedades devem verificar o *tag* “Stt\_BA101\_ManSel” de acordo com o *trigger* “Picture Cycle”. Se o valor desse *tag* for verdadeiro, o texto “Manual selecionado” deve ser apresentado na cor amarelo claro. Caso contrário, o texto “Automático selecionado” deve ser apresentado na cor verde claro.

2. Insira uma interface para o funcionamento automático do equipamento:

- O objeto do tipo *Button*, que seleciona o controle “On/Off”, deve enviar este comando ao CLP escrevendo o valor “1” no *tag* “Cmd\_BA101\_SelOnOff” (use VBS-Action);
- O objeto do tipo *Button*, que seleciona o controle “PID”, deve ser análogo ao anterior;
- O objeto do tipo *Static Text*, que indica o modo de controle do equipamento, deve ter a cor de fundo preto e cor do texto cinza claro. A propriedade que determina o texto a ser apresentado deve verificar o *tag* “Stt\_BA101\_PIDSel” de acordo com o *trigger* “Picture Cycle”. Se o valor desse *tag* for verdadeiro, o texto “Modo PID selecionado” deve ser apresentado. Caso contrário, o texto “Modo On/Off selecionado” deve ser apresentado.

3. Insira uma interface para o funcionamento manual do equipamento:

- O objeto do tipo *Button*, destinado a ligar o equipamento, deve enviar este comando ao CLP escrevendo o valor “1” no *tag* “Cmd\_BA101\_Liga” (use VBS-Action);
- O objeto do tipo *Button*, destinado a desligar o equipamento, deve ser análogo ao anterior;
- O objeto do tipo *Static Text*, que indica os estados do equipamento, deve ter a cor de fundo preto e, tanto a propriedade que determina o texto a ser apresentado como a que determina a cor do mesmo, devem ser animadas em *Runtime* (use VBS-Action). Ambas as propriedades devem verificar os *tags* “Stt\_BA101\_Defeito”, “Stt\_BA101\_Bloq” e “Stt\_BA101\_Ligada” de acordo com o *trigger* “Picture Cycle”. A animação deve ser feita da seguinte forma: vermelho claro para o texto “Equipamento com defeito” (primeira prioridade), amarelo claro para o texto “Equipamento bloqueado” (segunda prioridade), verde claro para o texto “Equipamento ligado” (terceira prioridade) e azul claro para o texto “Equipamento desligado” (em último caso).

4. Insira um campo de entrada de dados para o parâmetro “Velocidade manual”:

- O objeto do tipo *Static Text* que dá título ao parâmetro em questão deve ter cor de fundo azul escuro e cor do texto branco;
- O objeto do tipo *I/O Field* deve aceitar valores entre 0,0 e 100,0 [%] (com 1 casa decimal), devendo transcrever os valores digitados pelo usuário para o *tag* “Par\_BA101\_RefVel” a cada 1 segundo.

5. Insira uma interface para o horímetro do equipamento:

- O objeto do tipo *Static Text* que dá título aos objetos em questão deve ter cor de fundo azul escuro e cor do texto branco;
- O objeto do tipo *I/O Field* deve ter a cor de fundo preto e cor do texto cinza claro e apresentar o valor contido no *tag* “Ana\_BA101\_HorDec”, a cada 1 segundo. Observação: este objeto não deve aceitar valores digitados pelo usuário.
- O objeto do tipo *Button*, utilizado para restabelecer as horas restantes do horímetro deve conter o texto “R” na cor preto. Sendo assim, ao se clicar com o botão esquerdo do *mouse* nesse botão, o valor “1” deve ser escrito no *tag* “Cmd\_BA101\_HorRst” (use VBS-Action). Esse comando deve ser bloqueado para usuários que não tenham a permissão “Manutencao Avancada”.

6. Insira indicações para os intertravamentos que podem bloquear o funcionamento do equipamento:

- O objeto do tipo *Static Text*, que dá título as essas indicações, deve ter cor de fundo azul escuro e cor do texto branco;
- O objeto do tipo *Rectangle* e o objeto do tipo *Static Text*, associados à condição “Emergência ativada”, respectivamente, devem ter as cores de fundo cinza escuro e cinza claro em ambiente de desenvolvimento. Já em *Runtime*, a cor de fundo do objeto do tipo *Rectangle* deve ser animada (use VBS-Action) para monitorar o *tag* “Alr\_GERAL\_Emerge” por meio do *trigger* “Picture Cycle”, da seguinte maneira: se o nível lógico for 1, a cor amarelo claro deve ser apresentada, caso contrário, a cor deve ser verde claro;
- O objeto do tipo *Rectangle* e o objeto do tipo *Static Text*, associados à condição “Nível muito alto no tanque”, respectivamente, devem ter as cores de fundo cinza escuro e cinza claro em ambiente de desenvolvimento. Já em *Runtime*, a cor de fundo do objeto do tipo *Rectangle* deve ser animada (use VBS-Action) para monitorar o *tag* “Alr\_LT101\_ExcddHH” por meio do *trigger* “Picture Cycle”, da seguinte maneira: se o nível lógico for 1, a cor amarelo claro deve ser apresentada, caso contrário, a cor deve ser verde claro;

## 11 Tela “Cmd\_LC102.Pdl”

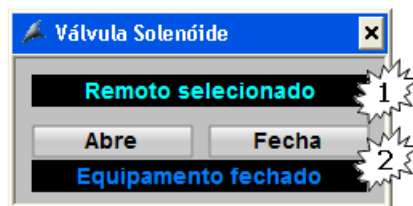
A tela “Cmd\_LI102.Pdl” já foi desenvolvida pelo professor. Para se tornar completamente funcional, esta tela necessita apenas que os nomes de alguns *tags* sejam adequados nos seguintes locais:

Objeto	Atributo	Evento
IO Field “Setpoint”	Output Value	
IO Field “Proporcional”	Output Value	
IO Field “Integral”	Output Value	
Tela “Cmd_LC102”		Open Picture

Após a adequação, analise os objetos utilizados nessa tela e entenda os objetivos e funcionalidades de cada um.

## 12 Tela “Cmd\_VS101.Pdl”

Faça uma janela para controlar a válvula VS101, similar a apresentada na figura abaixo. Para tal, siga a sequência de desenvolvimento indicada pelos “balões” inseridos na figura.



1. Insira uma interface para o modo de operação do equipamento:
  - O objeto do tipo *Static Text*, que indica o modo de operação do equipamento, deve ter a cor de fundo preto e, tanto a propriedade que determina o texto a ser apresentado como a que determina a cor do mesmo, devem ser animadas em *Runtime* (use C-Action). Ambas as propriedades devem verificar o *tag* “Stt\_GERAL\_LocSel” de acordo com o *trigger* “Picture Cycle”. Se o valor desse *tag* for verdadeiro, o texto “Local selecionado” deve ser apresentado na cor magenta claro. Caso contrário, o texto “Remoto selecionado” deve ser apresentado na cor ciano claro.
2. Insira uma interface para o funcionamento remoto do equipamento:
  - O objeto do tipo *Button*, destinado a abrir o equipamento, deve enviar este comando ao CLP escrevendo o valor “1” no *tag* “Cmd\_VS101\_Abre” (use C-Action);
  - O objeto do tipo *Button*, destinado a fechar o equipamento, deve ser análogo ao anterior;
  - O objeto do tipo *Static Text*, que indica os estados do equipamento, deve ter a cor de fundo preto e, tanto a propriedade que determina o texto a ser apresentado como a que determina a cor do mesmo, devem ser animadas em *Runtime* (use C-Action). Ambas as propriedades devem verificar o *tag* “Stt\_VS101\_Aberta” de acordo com o *trigger* “Picture Cycle”. A animação deve ser feita da seguinte forma: verde claro para o texto “Equipamento aberto” e azul claro para o texto “Equipamento fechado”, caso contrário.

### 13 Tela “Cmd\_Tipo01.Pdl”

Faça uma janela, parametrizável, para controlar os ventiladores VT101 e VT102, similar à apresentada na figura abaixo. Para tal, siga a sequência de desenvolvimento indicada pelos “balões” inseridos na figura.



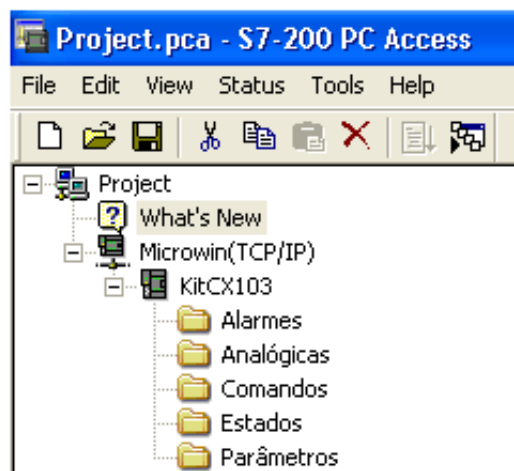
1. Insira uma interface para o modo de operação do equipamento:
  - O objeto do tipo *Static Text*, que indica o modo de operação do equipamento, deve ter a cor de fundo preto e, tanto a propriedade que determina o texto a ser apresentado como a que determina a cor do mesmo, devem ser animadas em *Runtime* (use VBS-Action). Ambas as propriedades devem verificar o *tag* “Stt\_GERAL\_LocSel”, indiretamente, por meio do *tag* interno “Aux\_Local”, de acordo com o *trigger* “Picture Cycle”. Se o valor apurado for verdadeiro, o texto “Local selecionado” deve ser apresentado na cor magenta claro. Caso contrário, o texto “Remoto selecionado” deve ser apresentado na cor ciano claro.
2. Insira uma interface para o funcionamento remoto do equipamento:
  - O objeto do tipo *Button*, destinado a ligar o equipamento, deve enviar este comando ao CLP escrevendo o valor “1” no *tag* “Cmd\_VT101\_Liga” ou no *tag* “Cmd\_VT102\_Liga”, indiretamente, por meio do *tag* interno “Aux\_Liga” (use Direct Connection);
  - O objeto do tipo *Button*, destinado a desligar o equipamento, deve ser análogo ao anterior;
  - O objeto do tipo *Static Text*, que indica os estados do equipamento, deve ter a cor de fundo preto e, tanto a propriedade que determina o texto a ser apresentado como a que determina a cor do mesmo, devem ser animadas em *Runtime* (use VBS-Action). Ambas as propriedades devem verificar o *tag* “Cmd\_VT101\_Ligado” ou o *tag* “Cmd\_VT102\_Ligado”, indiretamente, por meio do *tag* interno “Aux\_Ligado”, de acordo com o *trigger* “Picture Cycle”. A animação deve se dar da seguinte forma: verde claro para o texto “Equipamento ligado” e azul claro para o texto “Equipamento desligado”, caso contrário.
3. Insira uma interface para o horímetro do equipamento:
  - O objeto do tipo *Static Text* que dá título aos objetos em questão deve ter cor de fundo azul escuro e cor do texto branco;

- O objeto do tipo *I/O Field* deve ter a cor de fundo preto e cor do texto cinza claro e apresentar o valor contido no *tag* “Ana\_VT101\_HorDec” ou no *tag* “Ana\_VT102\_HorDec”, indiretamente, por meio do *tag* interno “Aux\_HorDec” (associe esse último *tag* à propriedade a ser animada e marque o *checkbox* “Indirect”).
- O objeto do tipo *Button*, utilizado para restabelecer as horas restantes do horímetro deve conter o texto “R” na cor preto. Sendo assim, ao se clicar com o botão esquerdo do *mouse* nesse botão, o valor “1” deve ser escrito no *tag* “Cmd\_VT101\_HorRst” ou no *tag* “Cmd\_VT102\_HorRst”, indiretamente, por meio do *tag* interno “Aux\_HorRst” (use Direct Connection). Esse comando deve ser bloqueado para usuários que não tenham a permissão “Manutencao Avancada”.

## Anexo A

Com o objetivo de estabelecer uma comunicação entre o *PC Access* (*OPC Server* para o S7-200 e S7-1200) e o CLP, proceda da seguinte maneira:

- Abra o programa *PC Access* a partir do caminho: Iniciar → Simatic → S7-200 PC Access V1.0.6.25 → S7-200 PC Access;
- Assim que o programa for aberto, clique com o botão direito no ícone “MicroWin” e selecione a opção “PG/PC Interface...”;
- Na janela “Set PG/PC Interface”, você deve selecionar sua placa de rede na lista “Interface Parameter Assignment Used” (provavelmente, o parâmetro a ser escolhido terá o nome “TCP/IP → Broadcom NetXtreme...”). Após a seleção, clique no botão “OK” e, quando solicitado, confirme sua escolha;
- Voltando ao PC Access, clique com o botão direito no ícone “MicroWin (TCP/IP)” e selecione a opção “New CLP...”;
- Assim que a janela “PLC Properties” for aberta, faça as seguintes configurações:
  - Name: KitXC103
  - IP Address: 192.168.0.11 ou 192.168.0.12 (depende da sua bancada)
  - TSAP Local: 10.00
  - TSAP Remote: 03.01
- Clique com o botão direito no recém-criado ícone “KitXC103” e selecione “New → Folder”, criando as seguintes pastas:



- Clique com o botão direito na pasta “Alarmes” e selecione “New → Item...”. Na janela “Item Properties”, cadastre as informações relacionadas ao primeiro sinal de alarme da aplicação que você desenvolveu no 1º Trabalho Prático da disciplina, por exemplo:



- Name: Alr\_Geral\_Emerge
  - Address: M301.0
  - Data Type: BOOL
- Repita o último passo até que as variáveis contidas em todos os grupos (Alarmes, Analógicas, Comandos, Estados e Parâmetros) sejam cadastradas no *OPC Server*.
  - Salve suas configurações e mantenha o “Arquivo.pca” com você. Esse arquivo será necessário para estabelecer futuras comunicações entre o CLP e o Sistema de Supervisão e Controle a ser desenvolvido. **Caso você tenha que trabalhar em outro microcomputador, será necessário abrir esse arquivo com o *PC Access* e salvá-lo no *HD* de sua nova estação de trabalho.**
    - Dica: um bom lugar para salvar este arquivo é dentro da pasta principal do projeto, criada pelo WinCC. Assim, se você copiar a pasta do projeto para outro microcomputador ou compactá-la para enviar pelo AVA, o “Arquivo.pca” não corre risco de ser esquecido.