



Minas Gerais

**SERVIÇO NACIONAL DE APRENDIZAGEM INDUSTRIAL**

**CENTRO TECNOLÓGICO  
DE ELETROELETRÔNICA "CÉSAR RODRIGUES"**

**CENATEC**

# ***SISTEMAS DIGITAIS DE CONTROLE***

---

Rua Santo Agostinho 1717 - Horto - Belo Horizonte - MG - CEP 31035-490

Tel.: (031) 482-5576 - FAX (031) 482-5577

*email: [cetel@fiemg.com.br](mailto:cetel@fiemg.com.br) - home page: [www.senai-mg.org.br/cetel](http://www.senai-mg.org.br/cetel)*

## ÍNDICE

1.	SISTEMAS DE CONTROLE	3
1.1.	Evolução	3
1.1.1.	Controles Locais	3
1.1.2.	Controles Centralizados	4
1.1.3.	Sistemas Digitais	8
1.2.	Arquiteturas De Sistemas Digitais	8
1.2.1.	Sistemas Centralizados	8
1.2.2.	Sistemas Digitais De Controle Distribuido-Sdcd	13
1.3.	Comparação Entre Sistemas Convencionais E Sdcd	16
2.	ESTRUTURA DE UM SDCD	16
2.1.	Sub-sistema de aquisição de dados e controle	20
2.1.1.	Componentes básicos de uma estação de controle	21
2.2.	Sub-sistema de monitoração e operação	24
2.2.1.	Janelas (Telas)	24
2.2.2.	Componentes básicos de uma estação de operação	28
2.3.	Sub-sistema de supervisão e monitoração	30
2.3.1.	Componentes básicos do subsistema de supervisão e otimização	32
3.	SOFTWARE SUPERVISÓRIO	33
3.1.	Arquitetura	34
3.1.1.	Base de Dados	36
3.1.2.	Configuração da interface de operação	36
3.1.3.	Configuração dos módulos aplicativos	41
3.1.4.	Configuração do módulo de comunicação	46
4.	Exercícios de Configuração do Software Supervisório UNISOFT	52
4.1.	Exercício 1	52
4.1.1.	Grupo UniSoft	52
4.1.2.	Selecionando uma Aplicação do UniSoft	53

4.1.3. Criando uma Nova Aplicação	54
4.1.4. Apresentação do Toolbar	55
4.1.5. Criando um bargraph	55
4.1.6. Capturando um Símbolo da Biblioteca	61
4.1.7. Agrupando Vários Objetos	63
4.2. Exercício 2	66
4.2.1. Construindo um Display para Saída de Dados	66
4.2.2. Criando um tag contador	70
4.2.3. Criando um Algoritmo Matemático	71
4.3. Exercício 3	75
4.3.1. Criando um Novo Tag na Planilha Matemática	75
4.3.2. Criando um Botão ON/OFF para a Bomba do Tanque	78
4.3.3. Dinâmica de Command	78
4.3.4. Botões Enchendo / Esvaziando	82
4.3.5. Criando uma nova tela	85
4.4. Exercício 4	91
4.4.1. Configurando um alarme	91
4.4.2. Relacionando um Tag a um alarme	92
4.5. Exercício 5	96
4.5.1. Criando a saída gráfica	97
5. ANEXO: Capítulo 6 do Manual UNISOFT	102

## SISTEMAS DE CONTROLE

O desenvolvimento do controle industrial tem sido fortemente influenciado pela tecnologia dos microprocessadores que tende a tornar o controle de processo totalmente distribuído. Mesmo antes da revolução da eletrônica, o controle automático de processos contínuos já era verdadeiramente distribuído.

A diferença fundamental é que estes não tinham um funcionamento integrado, muito pelo contrário, funcionavam de forma desorganizada através de um conjunto de malhas de controle independentes.

As primeiras aplicações do computador no controle de processos foram forçosamente centralizadas devido ao alto custo do mesmo.

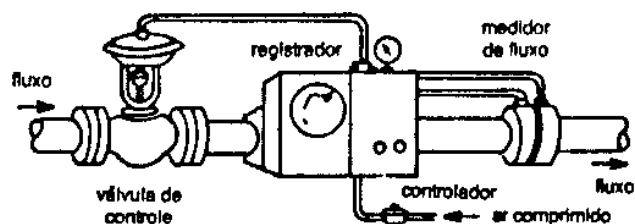
A tendência atual é integrar todo o sistema, permitindo um controle hierarquizado através da introdução de níveis de controle.

### Evolução

O controle automático de processos basicamente desde seus primórdios atravessou algumas modalidades de como o operador iria atuar no processo. Passamos a fazer um resumo histórico dessas fases.

## CONTROLES LOCAIS

Antes da segunda Guerra Mundial o controle automático era verdadeiramente distribuído. Um fluxo poderia ser controlado tal como mostra na ilustração.



**Controlador de fluxo**

O controlador era fixado próximo à unidade que ele controlava, ao alcance do medidor de fluxo e da válvula de controle de fluxo. Um registrador permitia ao operador conhecer o comportamento do fluxo durante sua ausência, bem como estimar a eficiência do processo, pela comparação com outros dados também registrados em outros pontos da planta.

Embora razoavelmente sofisticados, os controladores de então não solucionavam três grandes dificuldades:

O operador estando em um ponto de controle não tinha informações do que estava ocorrendo no resto da planta industrial;

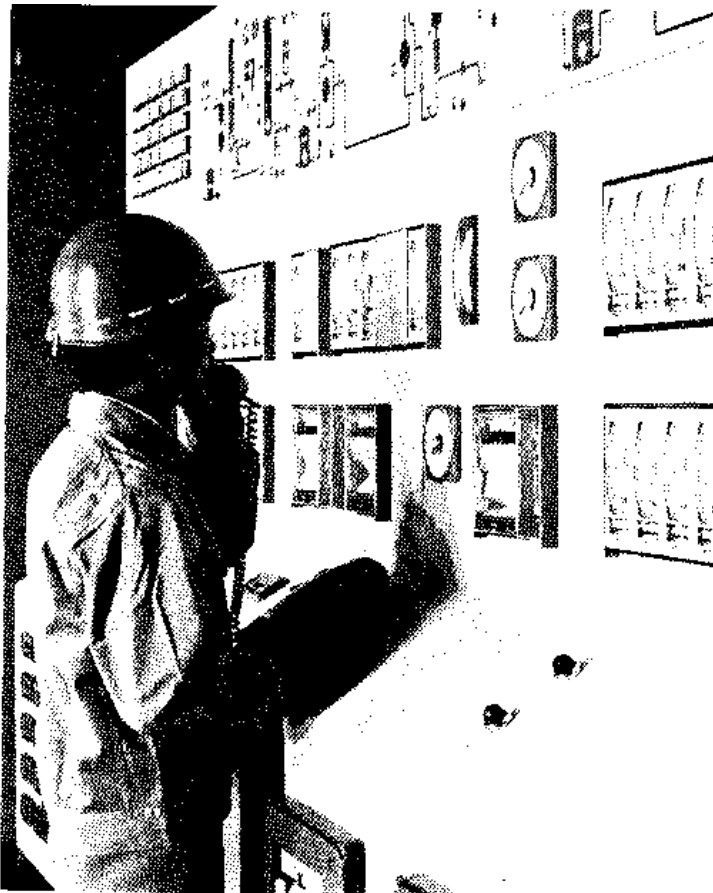
Alguns controles dependiam do inter-relacionamento de medidas diversas, em pontos diversos, para maior eficiência do processo;

o ajuste dos controladores exigia deslocamentos constantes dos operadores até a área;

Assim sendo esse modo de controle foi ultrapassado, porém cabe salientar que ainda hoje existem malhas de controle de pequena influência na produção de uma fábrica onde esse modo de controle é encontrado.

## **CONTROLES CENTRALIZADOS**

Com o desenvolvimento dos transmissores, foi possível levar os sinais da variável de processo até uma sala de controle central onde os controladores/indicadores realizariam as devidas correções e os sinais retornariam ao campo para atuar os elementos finais de controle.



Quando do uso da instrumentação pneumática isso representa um fator negativo para o controle de processos, devido ao tempo de resposta que o sistema agora teria. Isso porque aumentava-se consideravelmente a distância entre sensor-controlador e controlador-atuador.

Esse inconveniente limitava essas distâncias em aproximadamente 50 metros, sem prejuízo significativo para a malha de controle.

Com o advento da instrumentação eletrônica esse inconveniente foi ultrapassado visto que a velocidade de propagação dos sinais eletrônicos é muito rápida, sendo então a distância entre campo e sala de controle quase que desprezível no que se refere ao tempo de resposta.

Essa evolução que se faz presente até hoje, tem ainda um inconveniente:

Numa unidade fabril não existe somente uma sala de controle, mas várias. Cada uma concentra informações sobre determinada área da planta industrial. Outra sala de controle outra área e assim por diante. Podemos perceber que não existe uma integração de todas as informações das variáveis da planta industrial.

A não existência de uma única sala de controle se deve a fatores técnicos, custos mas principalmente os operacionais.

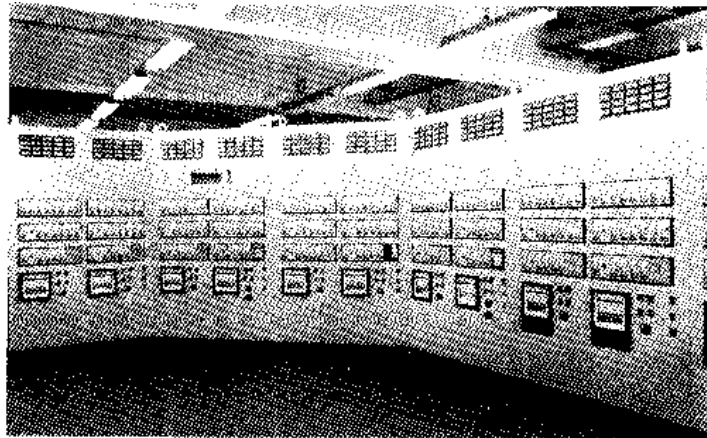
Os fatores técnicos se justificam basicamente que mesmo utilizando instrumentação eletrônica, o tempo de resposta praticamente é nulo, mas grandes distâncias irão fazer os sinais percorrer diversos pontos da fábrica até a chegada na sala de controle. Isso poderá gerar degradação dos sinais devidos a ruídos induzidos, resistência e capacitâncias elevadas nos cabos de sinais, etc.

Outro fator que pesa é o custo de implementação de uma sala de controle pois o material, equipamentos e mão de obra envolvidos numa instalação desse tipo é muito mais onerosa quanto maior for a distância envolvida.

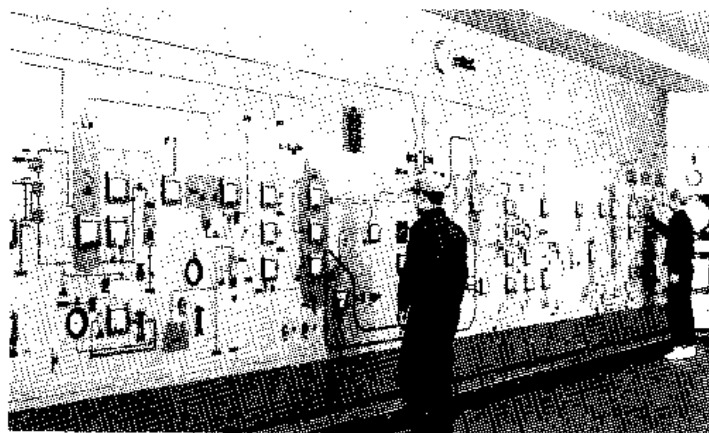
O fator operacional que pesa é que mesmo na sala de controle tendo somente os controladores, indicadores e registradores o tamanho desses painéis começa a ficar muito grande o que dificulta enormemente o trabalho do operador, podendo o mesmo incorrer em erros de operação pela atuação em um controlador errado por exemplo, devido ao fato dos instrumentos ficarem alinhados um ao lado do outro.

Uma forma para tentar se contornar esse inconveniente foi o uso de painéis gráficos, onde os instrumentos eram montados diretamente num painel sinótico da planta. Isso era excelente do ponto de vista operacional, pois o operador ao atuar num controlador pelo sinótico já percebia suas consequências no processo.

A grande desvantagem desse sistema é justamente o enorme espaço que esse painel ocuparia tornando-o inviável em processos com portes consideráveis.



Conventional panel

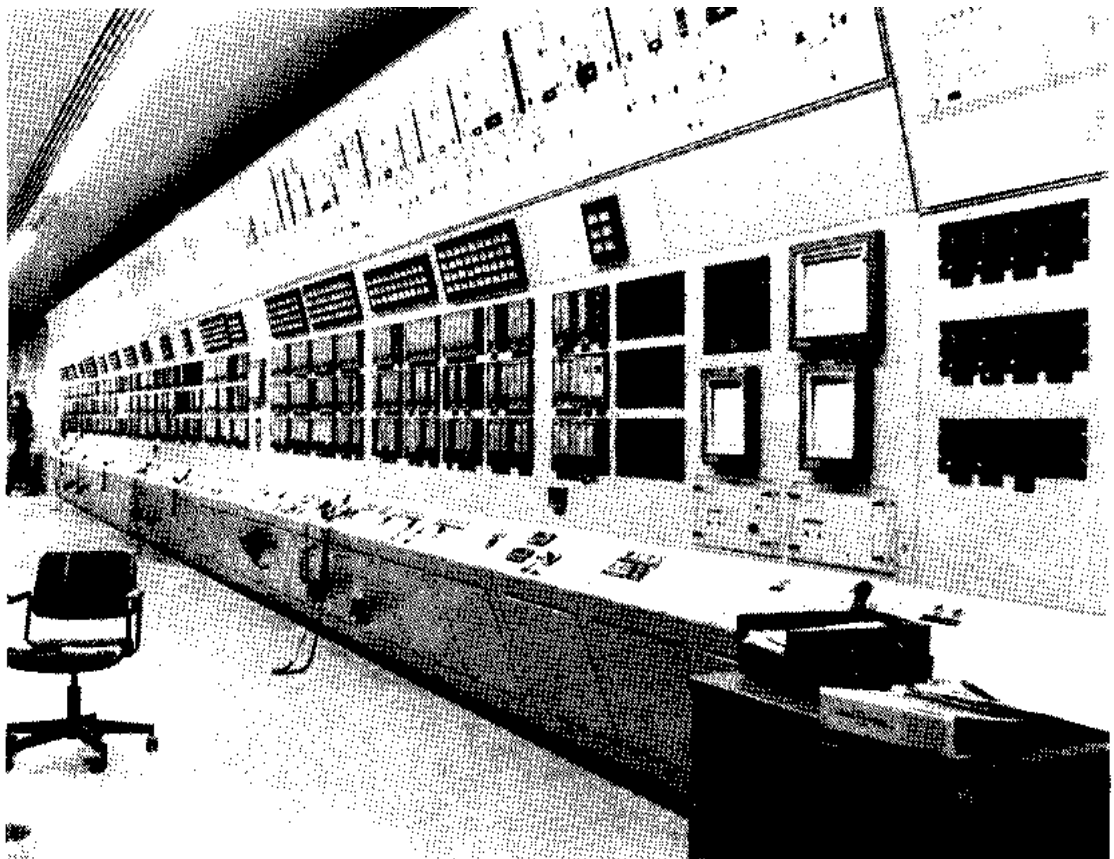


Full-graphic panel



Semi-graphic panel

Hoje em dia existe o que chamamos de painel semi-gráfico onde temos em sua parte inferior os instrumentos alinhados e em sua parte superior o painel sinóptico do mesmo. Convém ressaltar que o nível de animação do sinóptico é bem reduzida devido ao fator espaço.



Pode-se perceber que toda vez que desejamos concentrar informações estamos esbarrando no problema de espaço e confiabilidade de operação.

Com a chegada dos instrumentos microprocessados o problema de espaço foi atenuado, visto que com a instrumentação convencional cada instrumento tinha uma função definida. Pôr exemplo numa malha de controle e totalização de vazão com transmissor de  $\Delta p$ , teríamos no painel os instrumentos: extrator de raiz quadrada, controlador de vazão, estação auto-manual, totalizador de vazão.

Já a mesma malha de controle com instrumentos microprocessados ficaria reduzida a um único instrumento no painel: o controlador de vazão que incorpora os demais instrumentos.

E ainda mais, um único controlador com o poder de realizar o controle de diversas malhas (controladores multi-loop).

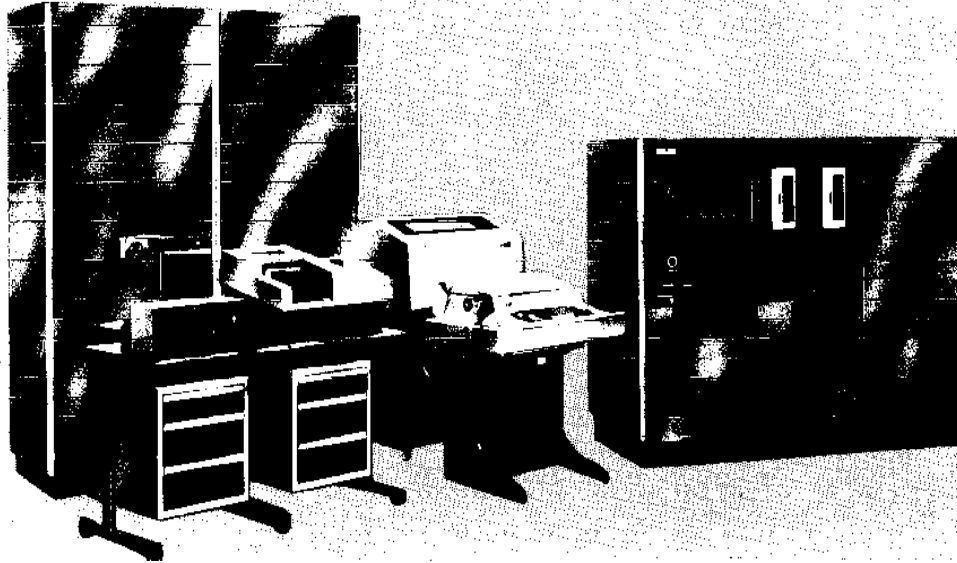
Porem agora temos informações concentradas demais, fazendo com que o operador tenha que ter atenção redobrada, pois um único instrumento controla varias malhas, ou seja antes o mesmo poderia se enganar no instrumento e agora pode se enganar no ajuste específico do instrumento o que é muito mais fácil de ocorrer.

Veremos mais adiante que o que parece ser uma desvantagem nesse sistema será uma vantagem nos sistemas digitais.



## **SISTEMAS DIGITAIS**

Com a evolução de sistemas eletrônicos de controle, naturalmente surgiram oportunidade de uso dos computadores eletrônicos digitais e trabalhando inicialmente com os conceitos de sistemas centralizados do tipo:DAS (Sistemas de Aquisição de dados "Data Acquisition System"), SPC ( Sistemas de Controle Supervisório "Set-Point Control) e DDC (Controle Digital Direto "Direct Digital Control") e os atuais sistemas distribuídos do tipo SDCD.



## **ARQUITETURAS DE SISTEMAS DIGITAIS**

Com a evolução de sistemas eletrônicos de controle, naturalmente surgiram oportunidade de uso dos computadores eletrônicos digitais e trabalhando inicialmente com os conceitos de sistemas centralizados do tipo: DAS, SPC e DDC e os atuais sistemas distribuídos do tipo SDCD. Veremos agora os conceitos dessas duas fases dos sistemas digitais:

### **SISTEMAS CENTRALIZADOS**

#### **SISTEMAS DE AQUISIÇÃO DE DADOS-DAS (DATA ACQUISITION SYSTEMS)**

Antes de executar uma tarefa de controle é necessário medir as variáveis do processo e, por isto, uma das primeiras aplicações de computadores digitais em processo foi a de aquisição ou coleta de dados.

Os valores das variáveis analógicas são multiplexados e convertidas para valores digitais. Os valores destas variáveis sofrem um tratamento no qual são comparados com valores limites para validação, transformados em unidades de engenharia, comparados com limites de alarme alto e baixo e armazenados para posterior processamento, que pode envolver calculo de performance, balanços materiais e de energia, acumulações, médias, processamento estatístico e outros.

Estes sistemas apresentam inumeras vantagens para o pessoal de operação e de engenharia de produção, mas pôr serem pouco mensu ráveis, na maioria dos casos torna-se difícil justificar investimentos em tais sistemas.

Suas importantes vantagens:

Coleta de dados do processo com precisão, periodicidade e confiabilidade apuradas.

Informação dos valores das variáveis de processo em unidades de engenharia.

Informação de valores de variáveis calculadas em função de outras.

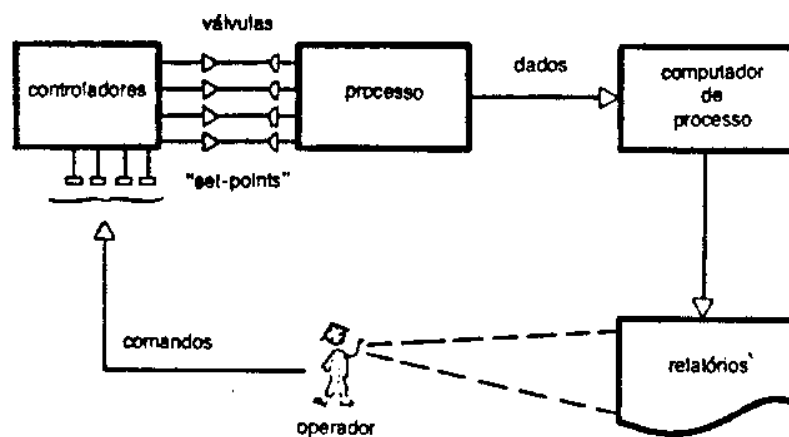
Informação de calculo de performance de equipamentos e do processo.

Relatórios de produção e calculo de rendimento de produtos.

Supervisão de variáveis gerando alarmes quando detectadas condições anormais.

Envio de mensagens de alerta para o operador.

Armazenamento de dados históricos para verificação de tendências e realização de estatísticas.



**Sistema de aquisição de dados**

Todas estas vantagens irão proporcionar aos operadores um acompanhamento mais apurado do processo tornando as ações de controle mais eficientes. Pôr outro lado, a engenharia de processo e de produção disporão de um maior volume de informações, permitindo conhecer melhor o processo e facilitando assim a sua modelagem e, como consequencia, seu melhor controle e otimização.

O operador obtém informações sobre o processo, dialogando com o computador através de terminais de vídeo, analisando relatórios de balanços, cálculos, registros, alarmes e eventos que são fornecidos pelos periféricos de impressão.

Baseando-se nestas informações, o operador ajusta os pontos de controle dos instrumentos convencionais de controle analógico aos quais cabe o efetivo controle das variáveis do processo.

## **SISTEMAS DE CONTROLE SUPERVISÓRIO - SPC(SET POINT CONTROL)**

O sucesso dos sistemas DAS geraram a confiança necessária para o uso dos minicomputadores em tarefas mais críticas como, por exemplo, o controle de algumas variáveis importantes do processo. Nos sistemas SPC o computador mantém as mesmas funções de um sistema DAS e além disso, utiliza o resultado de cálculo para atualizar os pontos de controle dos instrumentos analógicos convencionais que controlam efetivamente o processo.

Aplica-se o controle SPC em processos em batelada, no controle da sequência de operações de abertura e fechamento de válvulas e dos pontos de controle dos controladores convencionais, que variam de acordo com uma curva pré-determinada.

Este tipo de controle é também muito utilizado quando se quer otimizar a operação de um processo contínuo, como por exemplo, determinar os pontos ótimos de controle para minimizar o consumo de energia ou maximizar a produção ou o rendimento de determinados produtos.

Os controladores convencionais, neste caso, devem assumir o controle da unidade quando ocorrer alguma falha no computador e, para tanto, eles devem possuir as seguintes características:

Circuito de memória de set point para armazenamentos do último valor de set point recebido.

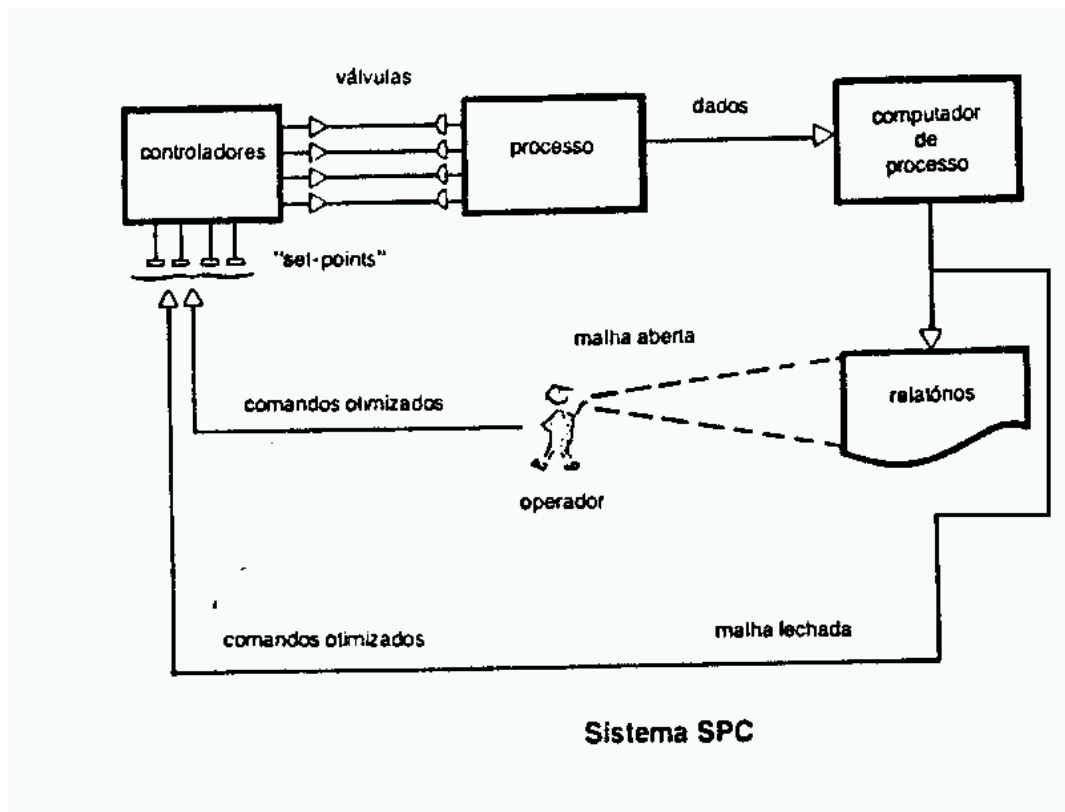
Contato para detecção de falha no computador.

Dispositivo de chaveamento de comando pelo computador para comando pelo controlador.

Opções de retomada do set point em caso de falha do computador considerando o seu retorno:

- ao último valor lido da variável
- ao último valor do set point do computador
- ao último valor lido da variável e em seguida para um valor pré-estabelecido
- a um valor pré-estabelecido

Os sistemas SPC podem ser de malha aberta ou fechada, sendo no primeiro caso chamado de sistema de "Instrução ao operador". Em ambos os casos o computador calcula os set points ótimos utilizando modelos matemáticos e teóricos de otimização, sendo que no primeiro, ele apenas informa aos operadores o valor ótimo a ser ajustado e no segundo, manipula diretamente os set points dos controladores.



Como vantagens teríamos:

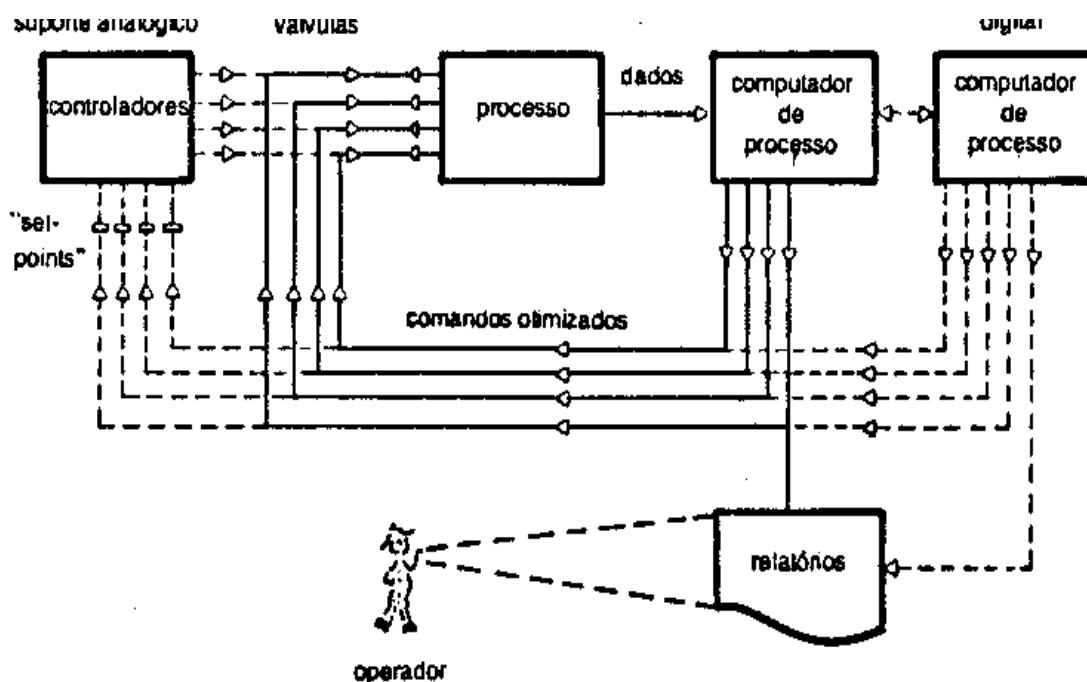
- Adequação tanto em processos contínuos como discretos
- Capacidade de executar algoritmos de teoria moderna de controle
- Capacidade de executar um gerenciamento da produção
- Grande precisão das ações de controle
- Aumento da produtividade
- Redução do consumo de energia
- Aumento do tempo útil de operação

Como limitações temos:

- Grande complexidade
- Alto custo do desenvolvimento e manutenção do software
- Alto custo de engenharia
- Baixa tolerância a falhas
- Alto custo de alta disponibilidade
- Alto custo da instrumentação analógica de back-up

## SISTEMAS DE CONTROLE DIGITAL DIRETO-DDC (DIRECT DIGITAL CONTROL)

Num sistema DDC também são operacionais as funções dos sistemas DAS e, além disso, o computador atua diretamente sobre os elementos finais de controle, contornando os controladores analógicos convencionais. Neste caso, os algoritmos de controle estão armazenados na memória do computador o que permite uma extensa gama de funções de transferência além dos tradicionais P, I E D.



Originalmente, esperava-se que os sistemas DDC pudessem ser justificados pela substituição direta da instrumentação convencional. No entanto, posteriormente foi provado que apesar da elevada disponibilidade do computador, sempre haveria a possibilidade de falhas e os operadores relutam em operar sem uma instrumentação analógica de back-up, já que um mau funcionamento do computador poderia causar a perda simultânea de todo o controle do processo.

Entretanto, ainda assim o DDC foi muito utilizado em combinação com o SPC, devido a sua grande flexibilidade e habilidade no uso de técnicas de controle impossíveis de se obter com instrumentos analógicos convencionais.

Como uma solução para os problemas de falha do computador em sistemas DDC, os fornecedores de instrumentos lançaram no mercado as estações de back-up computador-manual.

Num evento de falha do computador, ou caso ele seja posto fora de serviço por qualquer outro motivo, a estação memoriza o último valor de saída calculado, e o elemento final de controle permanece na sua última posição anterior a falha ou desconexão do computador. Caso o tempo em que o computador esteja fora seja prolongado, a estação pode ser chaveada para o modo de controle manual, onde o sinal de saída pode ser manipulado diretamente pelo operador.

As vantagens e limitações de um sistema DDC são as mesmas apresentadas anteriormente para os sistemas SPC.

As limitações acrescenta-se o alto custo das estações de back-up necessárias para prevenir a perda do controle de todo o processo no caso de falha do computador.

### **SISTEMAS DIGITAIS DE CONTROLE DISTRIBUIDO-SDCD**

Questionado o uso dos sistemas DDC, face à concentração da total responsabilidade de operação num único equipamento, evoluiu-se para uma filosofia de distribuição das funções de controle, tornada possível com o advento dos microprocessadores de baixo custo, elevado potencial e alta confiabilidade.

Nas arquiteturas SDCD o controle não é concentrado em um dispositivo central, mas distribuído entre as estações remotas. A estação central não é um elemento essencial à continuidade da operação, mas apenas um dispositivo para facilitar e oferecer maiores recursos para a interface do operador com o processo.

Nesta arquitetura as informações são centralizadas embora possa existir salas de controle locais e uma central, o controle é funcionalmente distribuído e os controladores são geograficamente centralizados ou não, possuindo as facilidades e recursos da eletrônica moderna dos microprocessadores, redes locais e fibras óticas. -

O fato da tecnologia baseada em microprocessadores ter tornado-se economicamente aplicável no projeto de instrumentação para controle de processos industriais, abriu a porta para muitas idéias inovadoras e permitiu filosofias de controle que podiam manipular funções de controle significativamente complexas, com a mesma facilidade e tão bem como se fossem malhas simples.

Novos avanços na tecnologia de microprocessadores vem possibilitando o desenvolvimento de um grande número de equipamentos de aquisição de dados e controle de processos que podem ser distribuídos ao longo de uma via de dados em uma planta industrial.

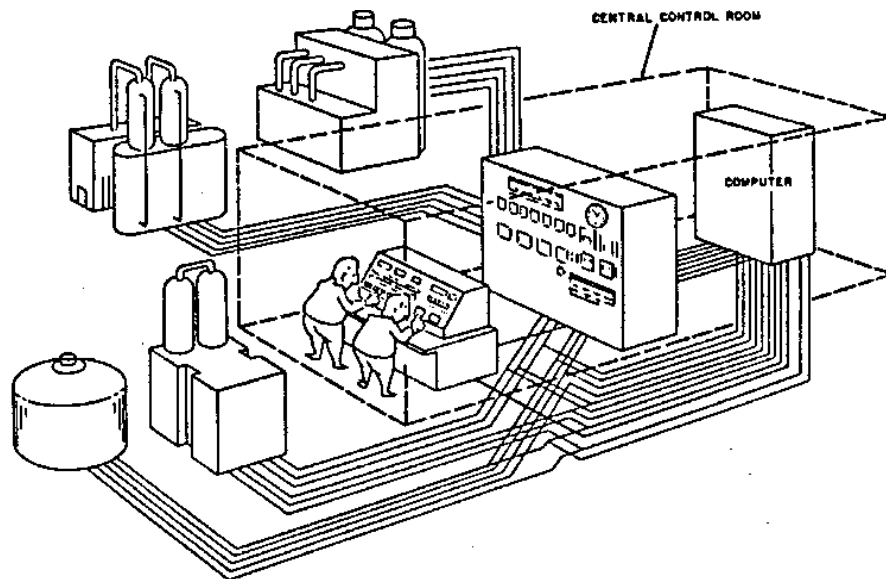
Cada um desses equipamentos é dotado de inteligência e executa funções específicas. Esta recente disponibilidade encontrada no mercado com grande quantidade de micro computadores com excepcional capacidade computacional e preços relativamente baixos tem tornado possível a implantação de sistemas digitais de controle distribuído possuindo capacidade e facilidades similares encontradas em um grande, poderoso e caro sistema de controle por computador centralizado e seus periféricos.

Os sistemas de processamento distribuído são adequados para uso tanto no controle de processo com também em condições de aplicações comerciais. Devido a seu baixo custo e simplicidade os microprocessadores podem ser distribuídos geograficamente ou funcionalmente para executarem funções dedicadas, gerando-se desta forma os sistemas de controle distribuído. Então os sistemas de controle distribuídos são uma série de microcomputadores(controladores programáveis) dedicados e altamente modularizados interligados por uma rede de comunicação digital.

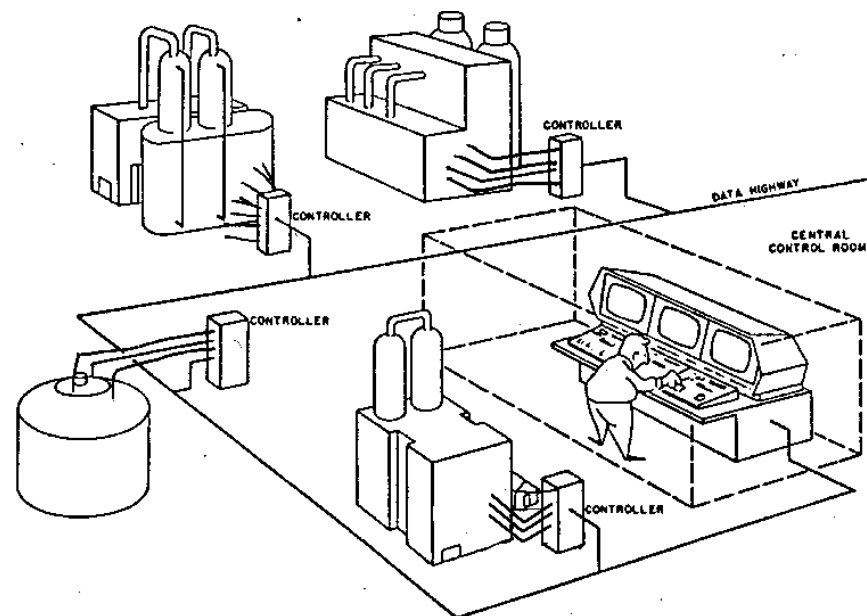
Um sistema digital de controle distribuído combina as vantagens do conceito de controle distribuído dos sistemas analógicos mais as vantagens do conceito de operação centralizada dos sistemas de computadores.

Assim sendo o computador fica livre para executar aplicações mais sofisticadas tais como modelagem matemática e otimização do processo.

A título de comparação temos nas ilustrações à seguir a representação de um sistema DDC e um SDCD sendo fácil verificar como um sistema SDCD é muito mais flexível do que um DDC.



Sistema Centralizado



Sistema de Controle Distribuído

Como vantagens temos no controle distribuído:

Elevada confiabilidade, garantida pôr:

- sub sistema de comunicação redundante

- disponibilidade de módulos de back-up (baixo custo)
- rotinas de auto-diagnóstico

Elevada flexibilidade de configuração e reconfiguração

- Baixo custo de configuração ou reconfiguração
- Facilidade de alteração da estratégia de controle
- Utilização de consoles de vídeo com linguagem interativa

Interface homem-máquina de alto nível

- Uso de consoles de vídeo coloridos, tecla funcionais, linguagem interativa
- Fácil aprendizado pelos operadores
- Telas padronizadas de fácil compreensão e manipulação
- relatórios impressos

funções - acesso a maior número de informações e execução de maior número de

Menores custos de instalação

- Custos de fiação drasticamente reduzidos
- Menores painéis e salas de controle
- Menores problemas com interferência por indução em sinais DC de baixo nível

Maior facilidade de interligação com computadores digitais

- Interface facilitada pelo uso de um "Data Highway"  
- Alivia carga de CPU do computador na medida em que as funções encontram-se distribuídas

Menores custos de desenvolvimento de Software

- Grande número de funções previamente programadas em firmware



## COMPARAÇÃO ENTRE SISTEMAS CONVENCIONAIS E SDCD

Relacionados abaixo alguns fatores em que um sistema SDCD permite maior conforto e segurança no controle de processos:

SITUAÇÃO	PAINEL CONVENCIONAL	SDCD
Cabeamento	Praticamente 1 par de cabos para cada informação, ligando o sensor no campo ao painel	Cabo do sensor ligado a uma ECL no campo e a paritr da ECL, barramento digital de dados até a sala de controle
Configuração	Dedicada e fixa definidas pela construção do painel	Fácil mudança pela configuração do sistema (software)
Visão da planta	Instrumentos montados em painéis de até 30m de comprimento identificados pelo TAG	Diferentes tipos de telas gráficas em diferentes níveis de detalhes
Registro cronológico dos eventos	Não possui grande capacidade de registro cronológico	Registra todos os eventos de forma cronológica
Falhas nos equipamentos	São detectados somente através de observação de anomalias da planta	Através de alarmes de sistema e auto diagnósticos, são detectados as falhas assim que ocorrem
Registro de variáveis	Utiliza registradores comuns do tipo suporte papel gráfico e penas sem nenhuma flexibilidade sem grande consumo de papel e tinta	Utiliza registradores em suporte magnético com níveis de infor mações elevadas podendo inclusive excluir, sobrepor variáveis a analisar

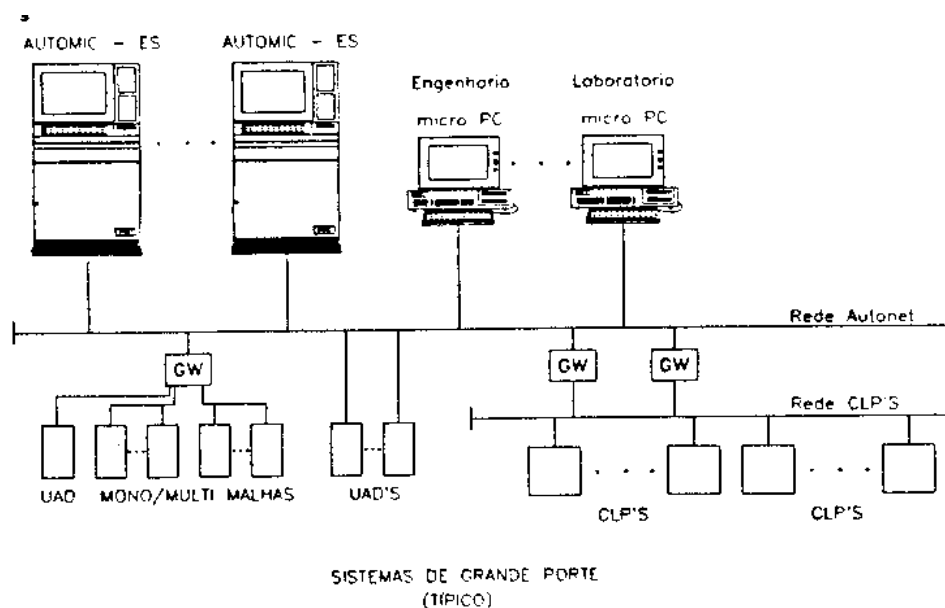
## ESTRUTURA DE UM SDCD

Sistemas de controle de processo e outros sistemas usados em condições industriais típicas envolvem aquisição de dados de sensores e subsequente controle em malha fechada via atuadores acoplados a controladores individualmente. As tarefas a serem executadas por estes controladores podem ser claramente definidas e uma configuração ótima pode ser especificada.

Uma forma simplificada de processamento distribuído horizontalmente permite que a carga seja compartilhada entre diferentes processadores sem envolver transferência de programas aplicativos e grande quantidade de dados. Este tipo de processamento distribuído é adequado para aquisição de dados e controle de processos industriais.

O processamento de dados geral e as funções de controle são analisadas, particionadas e alocadas a diferentes processadores. Assim cada processador tem que executar somente um conjunto específico e bem determinado de funções. O programa de aplicação requerido para executar tais funções estará residente na memória daquele processador ou sobre um dispositivo de armazenagem de massa acoplado ao mesmo. Similarmente, os dados a serem usados por estes programas de aplicação poderão ser tanto armazenados sobre um dispositivo de memória principal ou secundária acoplado ao mesmo ou adquirido diretamente do processo por meio de sensores adequados.

Temos então uma base de dados geral da planta distribuída localmente nos subsistemas formados por cada controlador e seus dispositivos associados.



Em condições de operação, cada controlador é responsável por aquisição de dados, calibração e pela execução de qualquer pré-processamento necessário. Estes dados são então usados em um ou mais algoritmos de controle que determinam a ação de controle requerida, a qual é executada via atuadores interfaceados ao mesmo. Cada um dos processadores será responsável pela execução de qualquer cálculo de otimização necessários para aquela seção do processo. Uma interface para o operador separada com facilidades de aquisição e controle pode ser prevista.

A coordenação do processo é obtida pela transferência, via linha de comunicação, de pequenas quantidades de dados necessárias aos outros controladores. É difícil executar uma

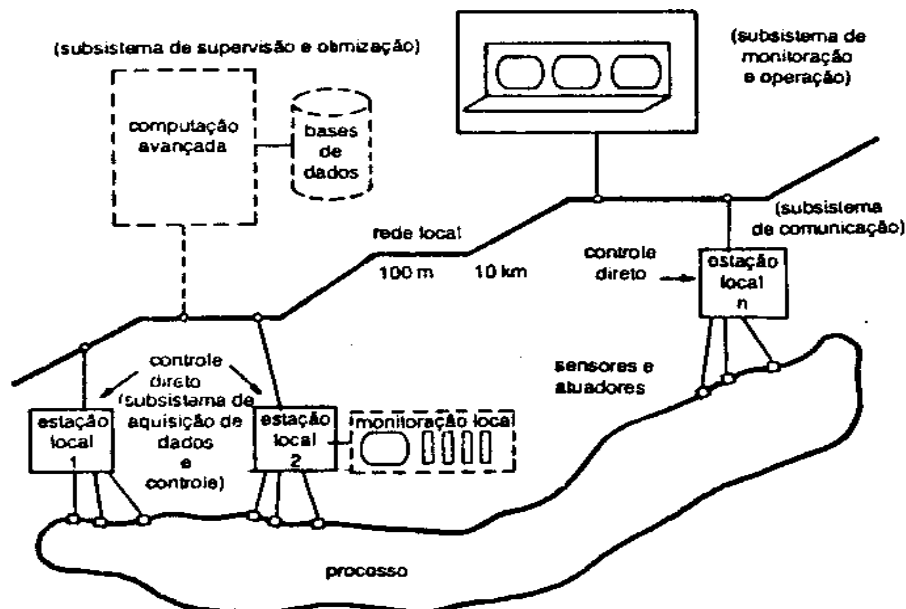
otimização de processo geral se nenhum dos computadores tem informação completa sobre o estado geral de todo o processo.

Desta forma, caso se deseje realizar uma otimização geral da planta é recomendável a existência de um computador central com acesso rápido a base de dados de toda a planta e geralmente com uma capacidade computacional maior do que os processadores distribuídos. Este computador é conhecido como hospedeiro e fica em um nível hierárquico superior aos processadores distribuídos horizontalmente.

Temos então, um sistema com arquitetura mista, isto é, processadores distribuídos horizontalmente e verticalmente. A maioria dos SDCD associados a outras atividades tais como supervisão, coordenação e controle de produção possuem arquitetura mista.

De uma forma geral, as funções exercidas por um SDCD podem ser estruturadas de maneira hierárquica, sendo definidos diversos níveis de atividades.

De modo a melhor caracterizar um SDCD, vamos agrupar os elementos que o compõem em quatro subsistemas de acordo com suas características funcionais, e mostrar como o atendimento aos níveis hierárquicos acima se coaduna com a caracterização proposta.



### SUBSISTEMA DE AQUISIÇÃO DE DADOS E CONTROLE

Está diretamente ligado ao processo. A sua principal finalidade é a realização das funções de controle, que são exercidas pelas Estações de Controle Local (ECL, níveis 0, 1 e 2).

### SUBSISTEMA DE MONITORAÇÃO E OPERAÇÃO

Nele se concentra a maior parte das funções de Interface Homem-Máquina (Nível 3).

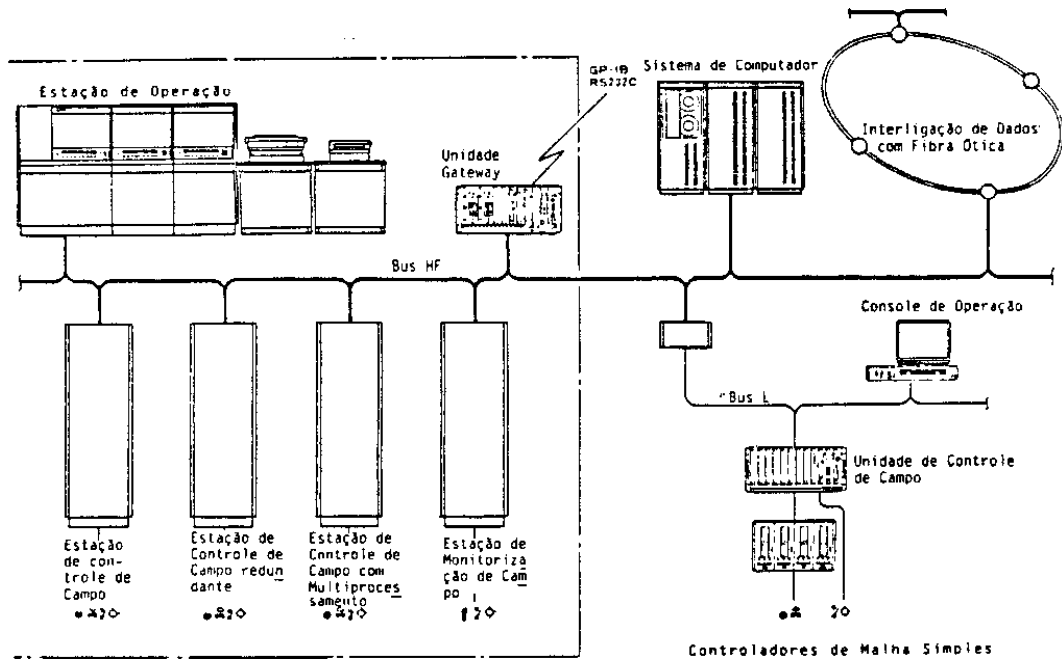
### SUBSISTEMA DE SUPERVISÃO E OTIMIZAÇÃO

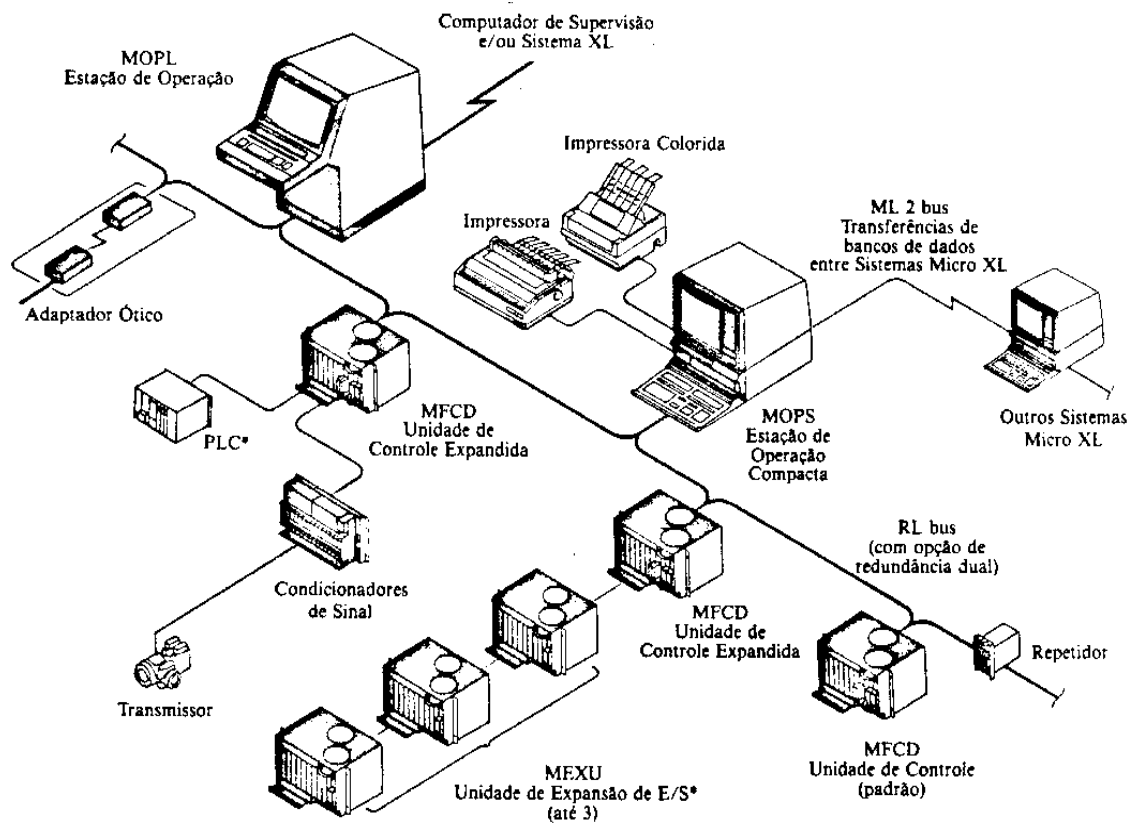
É onde são realizadas as funções de otimização e gerenciamento de informações (Níveis 3 e 4).

## SUBSISTEMA DE COMUNICAÇÃO

Para que seja possível a realização de um controle integrado, é necessário que exista uma infra-estrutura de comunicação entre os diversos subsistema. Então este subsistema será responsável pela integração dos diversos módulos autônomos do sistema.

Em vista do exposto até agora, mostraremos alguns modelos de configurações de sistemas digitais:





#### Sub-sistema de aquisição de dados e controle

O objetivo deste grupo de elementos é promover a interface direta com o processo e realizar as funções de controle local.

É importante ressaltar a característica de autonomia destes módulos, pois mesmo na ausência das funções de níveis superiores ele deve continuar operando as funções de controle, embora podendo estar degradado segundo algum aspecto específico.

Este subsistema apresenta, na maioria dos SDCD disponíveis no mercado, além dos algoritmos de controle do tipo PID, comuns na instrumentação analógica convencional, uma variada gama de funções que inclui, por exemplo:

Controle multivariável

Algoritmos de nível superior

Controle "feed-forward"

Controle de sequência

Controle lógico

Intertravamento

Soma, subtração, multiplicação e divisão

Raiz quadrada, Compensadores de pressão e temperatura

Polinômios e logarítimos

Alarmes de nível, desvio, velocidade

Linearizações

Etc.

Dele também fazem parte os cartões de interface de entrada e saída com o processo, tais como:

Entradas e saídas analógicas

Entradas e saídas digitais

Entradas de pulsos

Multiplexadores

Conversores AD e DA

Etc.

Este subsistema contém também as placas de memória que armazenam os microprogramas das funções executáveis, das rotinas de diagnóstico de falha, das rotinas de "back-up", etc..., as placas de módulos para redundância parcial ou total e os circuitos necessários à segurança intrínseca.

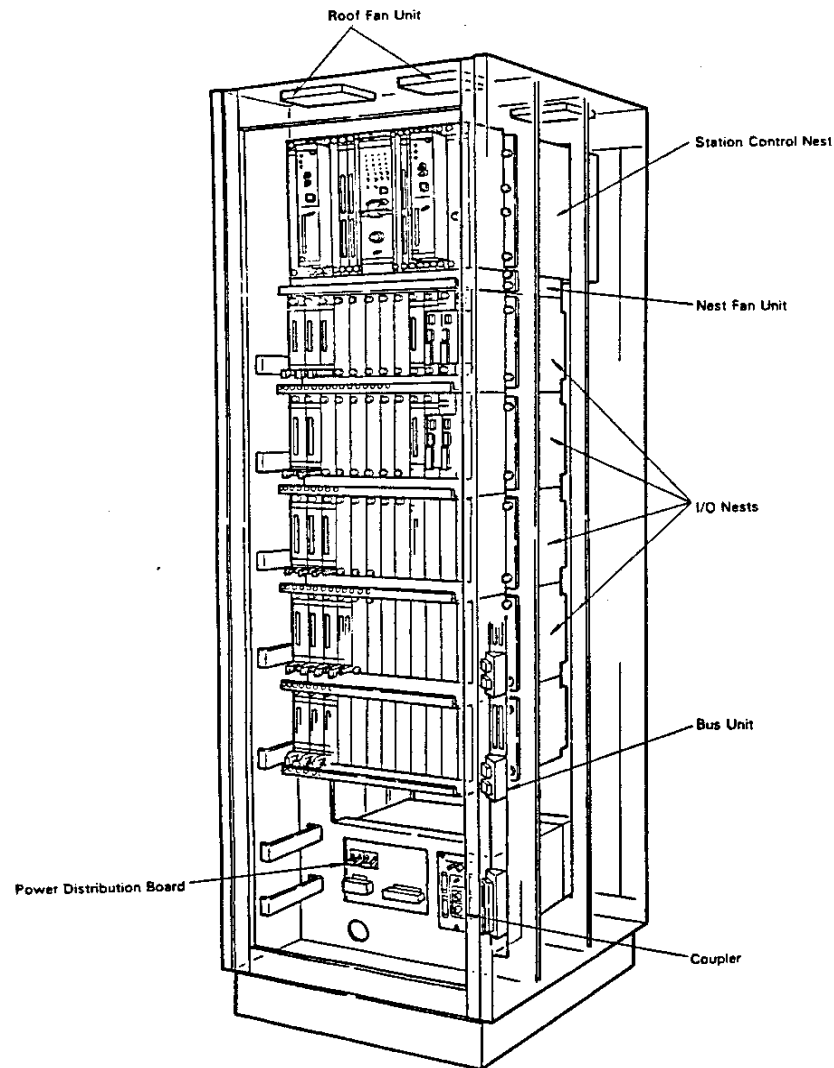
No nível deste subsistema poderá ou não haver um outro subsistema de monitoração e operação local simplificado, conforme mostrado na figura do modelo de referência. Este subsistema interfaceia-se com os subsistemas de comunicação e com um eventual subsistema simplificado de monitoração local.

Componentes básicos de uma estação de controle

É formado por um conjunto de controladores capazes de executar as tarefas descritas anteriormente no subsistema de Aquisição de Dados e Controle.

Cada um destes controladores é implantado por meio de um processador de propósito especiais locado remotamente, podendo receber informações de poucas ou várias entradas digitais e/ou analógicas.

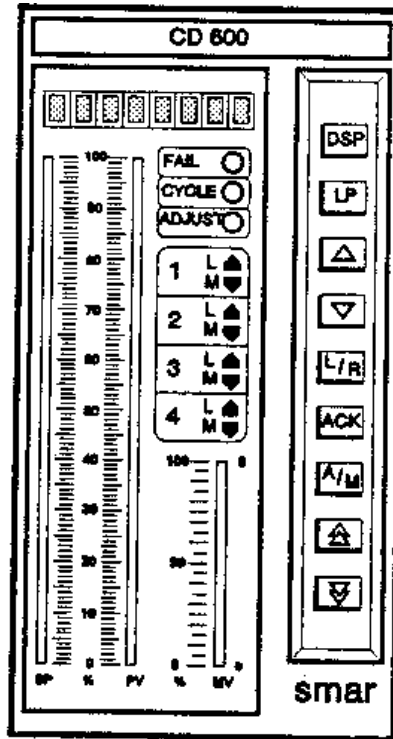
Ele pode normalmente enviar de 1 até 16 sinais de atuação analógicos e ou de 1 até centenas de sinais de saída digital. Em sistemas instalados poderemos ter vários desses controladores instalados em um rack formando uma unidade de controle local.



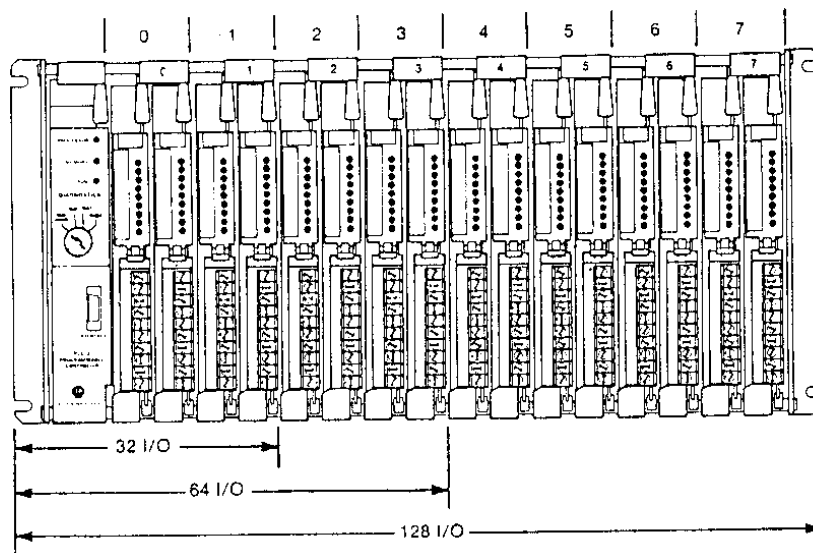
Unidade de Controle Local

Há outro tipo de controlador baseado em microprocessador que se encaixa na descrição de controlador dada acima. É um dispositivo totalmente independente e que através do compartilhamento de tempo controla de 4 a 8 malhas do processo. Ele pode ser programado para fazer uma variedade de tarefas e sua configuração é feita localmente ou remotamente. Normalmente é montado em painel local. Usualmente o mostrador é comum a todas as malhas. O usuário deve então selecionar a malha que deseja supervisionar. Através deste visor ele pode acessar todas as variáveis de processo bem como valores do ponto de operação, saída ou valores das constantes de ajustes das malhas. Ele pode também selecionar operação manual ou automática e mudar os valores da saída e ponto de operação.

Como os fabricantes desse tipo de equipamento tem fornecido um suporte para interfaceamento com a rede de comunicação, ele é considerado como um elemento dos SDCD. Também poderá ser usado como um equipamento de back-up e redundância, visto ter sua interface de operação local.



Controladores lógicos programáveis (CLP) também fazem a função de aquisição e controle. Sua função básica é a de executar tarefas equivalentes a circuitos contadores, temporizadores, e relés encontrando sua maior aplicação em substituição aos painéis de comandos elétricos convencionais( com relés, contadores, etc). Podem também realizar a aquisição de sinais analógicos e executar algoritmos PID realimentando o processo por uma saída analógica correspondente.





## **Sub-sistema de monitoração e operação**

Este subsistema trata especificamente da interface homem-máquina. Por interface homem-máquina entendemos os dispositivos de Hardware que fornecem ao operador um maior controle e um melhor nível de informação sobre a condição de operação da planta, reduzindo o seu esforço através da simplificação dos procedimentos operacionais.

São características normalmente existentes num subsistema de operação e monitoração:

Fornecer ao operador um conjunto de informações sobre o estado de operação da planta, através de um número de estações de operação suficiente para atender todas as variáveis de interesse do processo;

Fornecer ao operador em tempo hábil, informações num formato que evidencie a ocorrência de condições excepcionais de operação (gerência por exceção), para que providência imediatas possam ser tomadas;

Permitir que variáveis de processo sejam agrupadas de maneira que o operador possa realizar uma análise comparativa entre variáveis constituintes de cada grupo;

Permitir ao operador a visualização de informações em detalhe crescente, dependendo do quanto ele precisa saber ou do quanto ele precisa mudar para corrigir as falhas do processo;

Possibilitar o uso simultâneo de várias estações de operação para que todas as funções disponíveis possam ser utilizadas em todas as estações de operação e estas possam ser instaladas em locais diferentes.

Encapsular procedimentos de operação de forma que seja mais segura e veloz a resposta do operador à ocorrência de uma irregularidade na planta. Encapsulamento consiste basicamente da utilização de teclas funcionais. Essas teclas determinam, quando pressionadas, o acionamento de procedimentos de operação, de sorte que toda uma sequência de operações possa ser substituída por apenas uma operação.

Normalmente, os SDCDs utilizam uma filosofia de gerência por exceção, mostrando informações suficientes para o operador saber que tudo corre bem.

Quando as condições saem do normal, maiores detalhes podem ser mostrados ou solicitados. As informações são apresentadas sob a forma de telas gráficas. As telas são claras e sucintas. O acúmulo de informações na tela pode prejudicar a visualização das condições excepcionais.

Janelas (Telas)

As características básicas em termos de janelas (telas) são as seguintes:

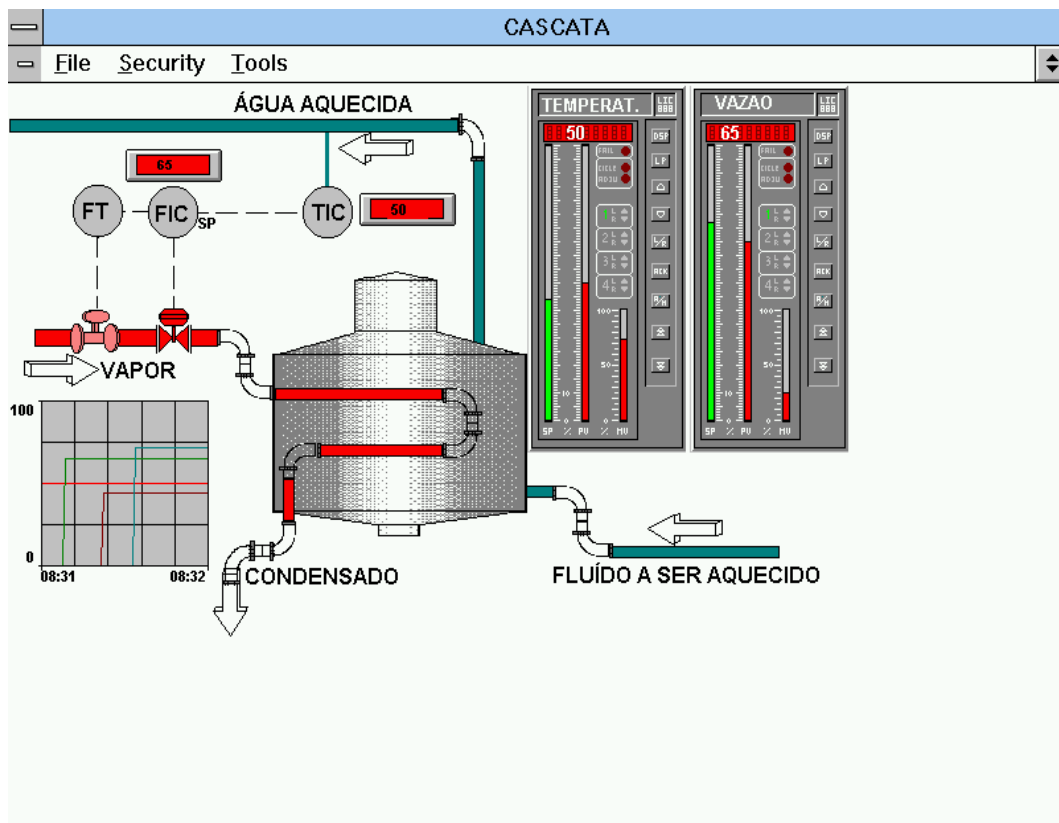
### **Janela de OVERVIEW**

Apresenta de forma bastante simplificada até 300 controladores/indicadores, dispostos em grupos lógicos, arranjados de forma que o operador identifique facilmente as condições de alarme, modo de controle automático ou manual e grandeza dos desvios. Existem diferentes estilos e maneiras de representar as informações nestas telas, entretanto, o tipo mais normal utiliza barras para informar o operador. Consiste num conjunto de barras de desvios alinhadas por uma linha de referência onde as mesmas podem sofrer desvios para cima ou para baixo da linha de referência. O tamanho da barra representará a grandeza do desvio

da variável em relação ao set point ( geralmente configurada para 5 ou 10%). A cor da barra representará as situações de alarme e modo de operação.

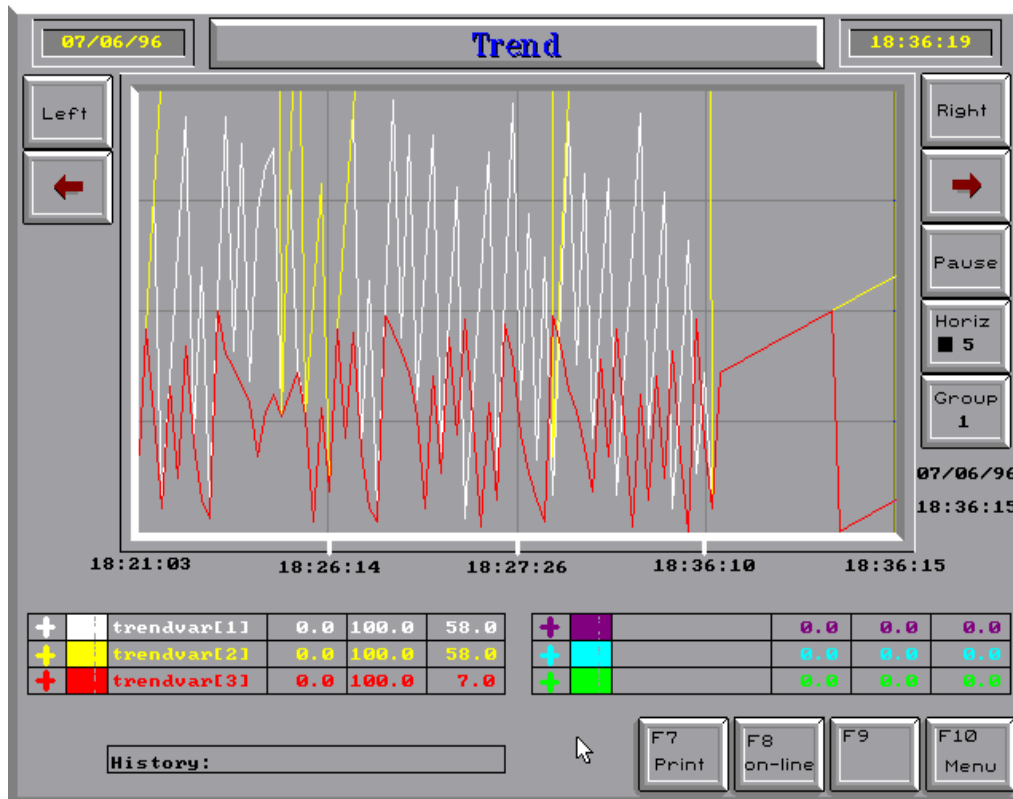
### Janela De Instrumento

Mostra um face plate (frontal) de um instrumento típico de painel (controlador; indicador; botoeira; totalizador; etc...), permitindo ao operador verificar mais em detalhes uma secção da planta que precisa atenção. O operador poderá então monitorar e manipular alguns parâmetros dos controles tais como: set point, transferência automático manual, saída para válvula, etc. Isso cria uma interface de operação bem amigável, porque o operador de painel continua a operar um instrumento convencional.



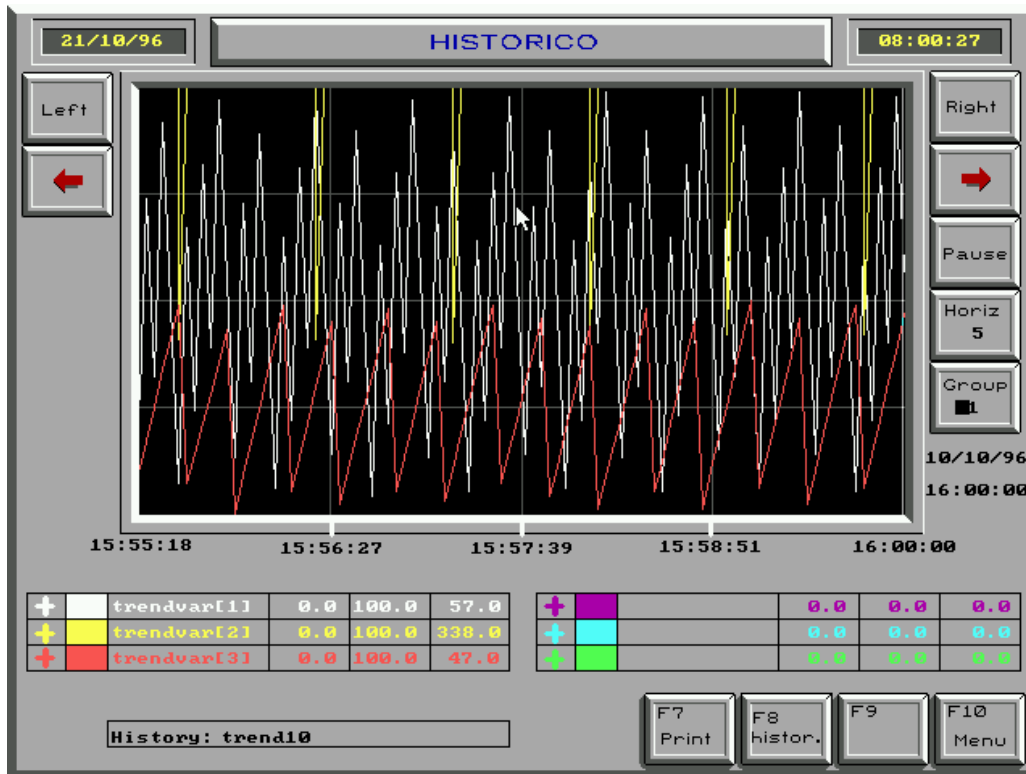
### Janela de tendência tempo-real

Mostra, numa representação gráfica e sempre atualizada, a tendência das variáveis de processo nos últimos 04 a 20 minutos. é desejável que possam ser mostrados simultaneamente os gráficos de tendência de mais de uma variável do processo.



### Janela de históricos

A tendência das variáveis de processo ao longo de períodos maiores tais como, horas, dias e meses são apresentados. Os valores médios nos períodos em questão e o gráfico não são atualizado no tempo. Existem recursos do tipo: cancelar a indicação de variáveis para se estudar separadamente uma ou mais variáveis. Pode dispor de um cursor (linha vertical) que pode ser movimentado pela tela, fornecendo os pontos de intersecção do cursor com as curvas das variáveis em unidades de engenharia da variável naquele ponto.



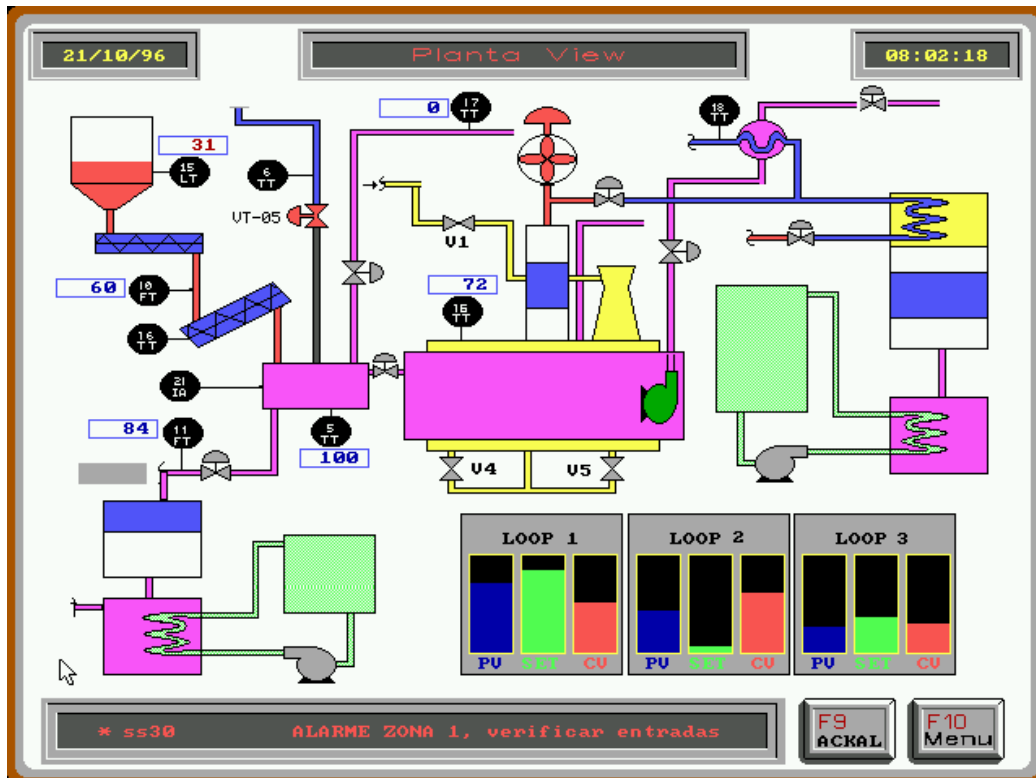
OBS: Atualmente existem no mercado softwares de supervisão que operam em padrões gráficos gerenciados por Sistemas Operacionais baseados em janelas (WINDOWS). Isso significa que não existe nesses sistemas telas fixas como visto anteriormente, bem como a hierarquia de navegação das mesmas.

O usuário poderá na operação abrir as janelas de funções de acordo com suas necessidades tornando o sistema assim muito mais flexível e amigável.

### Janelas de sinóticos de fluxogramas de processo

Mostram graficamente secções de um fluxograma com os valores das variáveis de processo e set points, atualizados continuamente. Os fluxogramas podem apresentar características adicionais que possibilitem um melhor entendimento dos mesmos, tais como indicação de variações de nível e indicação de alarmes, monitoração do trajeto do fluxo pelas tubulações, indicação dos valores das variáveis dinamicamente, etc.

Pode-se inclusive ativar o "faceplate" de um controlador numa região da tela, podendo o operador atuar no mesmo sem sair da tela.



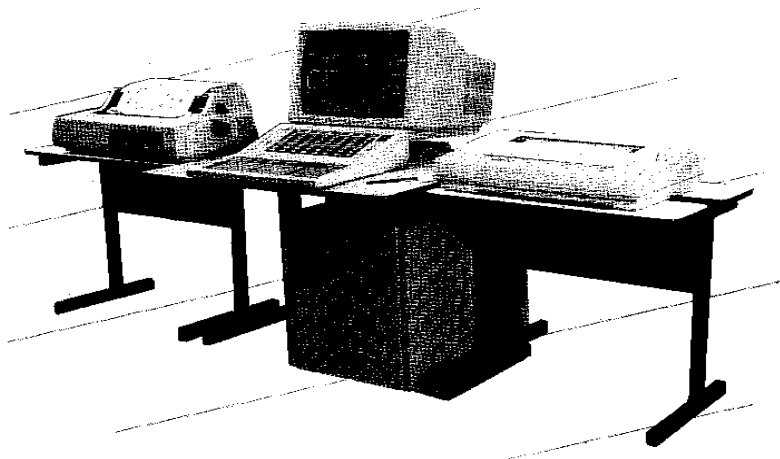
### Componentes básicos de uma estação de operação

É formado por um console de operação composto basicamente de um terminal de vídeo, teclado e impressora.

Neste conjunto é instalado um software de supervisão e controle de processos industriais.

Os arranjos dos consoles são muitas vezes construídos de maneira que várias telas são convenientemente alocadas onde um operador possa observar a operação de várias seções da planta ao mesmo tempo. Podem incluir registradores, chaves críticas, telefone, etc.

É importante que o sistema forneça os dados do processo de maneira rápida e ordenada para o operador da planta. Também é necessário que o operador forneça informações (dados) e comandos ao sistema.



O teclado do operador é um importante aspecto a ser analisado no console. é através deste teclado que o operador pode comandar mudanças do set point, tipo de tela e outros dados da malha de controle. Alguns sistemas usam um teclado como máquina de escrever onde as várias teclas são classificadas e codificadas e desempenham funções específicas no controle do processo. Outros sistemas utilizam um arranjo completamente diferente, onde grupos separados de teclas são arranjados de acordo com sua função. Podem ser codificadas e coloridas para proporcionar maior facilidade de reconhecimento pelo operador. Esse teclado recebe o nome de teclado de operação.

**Soft keys** - atuação de acordo com a descrição em cada tela.

**Teclas de Operação** - atuação como em frontal de painel de equipamentos de controle.

**Teclas Tag List e Process Flow** - atuação em conjunto com as soft keys para chamar as telas desejadas.

**Teclas de Alarme** - seleção e reconhecimento de alarmes de processo.

**Teclas do Usuário** - atuação de acordo com a configuração do usuário.

**Teclas de Função Especial** - configuração de acordo com as necessidades do usuário.

**Teclas Numéricas** - entrada de dados numéricos.

Existe uma tendência de alguns fabricantes do software de supervisórios fornecerem um tipo de vídeo conjugado com um sistema de entrada de dados, que recebe o nome de "touch screen" (toque de tela). Esse sistema consiste de um vídeo o qual recebe uma moldura com emissores de luz infra-vermelha de um lado e elementos fotossensíveis do outro. Isto cria sobre a tela do monitor uma malha invisível de luz infra-vermelha. Quando o operador coloca o dedo sobre a tela, os raios luminosos são bloqueados. A moldura percebe isto e informa as coordenadas da tela ao computador onde a tela foi tocada. Nesses sistema a tela previamente

mostra várias opções de operação e o usuário deve tocar a região da tela demarcada pela moldura da opção (geralmente retângulos).

Quando registros devem ser mantidos documentados e quando a informação registrada deve ser vista por várias pessoas, necessita-se de impressoras como periféricos.

As características mais marcantes que definem o funcionamento de uma impressora são:

Velocidade - geralmente dada em caracteres por segundo (impressoras seriais) ou linhas por minuto (impressoras de linha). Obs. Via de regra quanto mais rápida for a impressão pior será a qualidade do trabalho impresso.

Capacidade da largura da folha - existem de 80 e 132 colunas.

Alimentação do papel-Fricção de papel ou formulário contínuo(tracção)

Geração dos caracteres - Podem ser por cabeça já com os caracteres pré-fabricados(margarida, esfera) ou as matriciais, onde a cabeça é formada por agulhas dispostas em matriz no qual cada caracter corresponde a ativação de um conjunto de agulhas. Cabe salientar que a qualidade de impressão das primeiras é excelente pois o caracter é um bloco impresso contínuo, e no segundo caso será o agrupamento de pontos formando o caracter. Como desvantagem as primeiras são muito lentas e não permitem a impressão de gráficos.

Impressão dos caracteres - Poderão ser por impacto ou não. Os por impactos são mais lentos e ruidosos, sujeitos a maiores danos mecânicos. Porém o resultado final do trabalho é satisfatório para este ambiente de trabalho. Impressoras sem impacto podem ser térmicas, eletrosensitivas, lasers, jato de tinta. E ainda coloridas(p/hard copy).

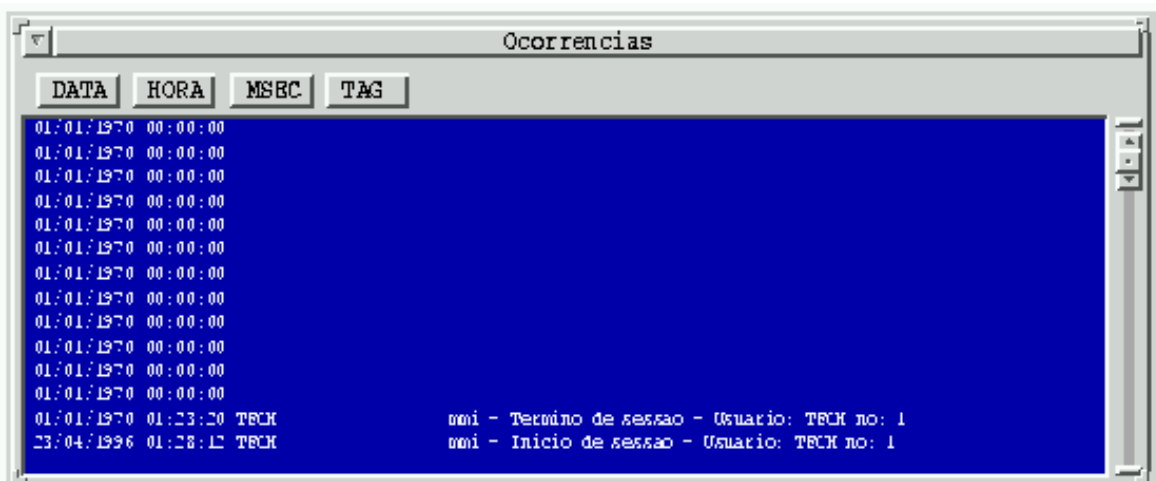
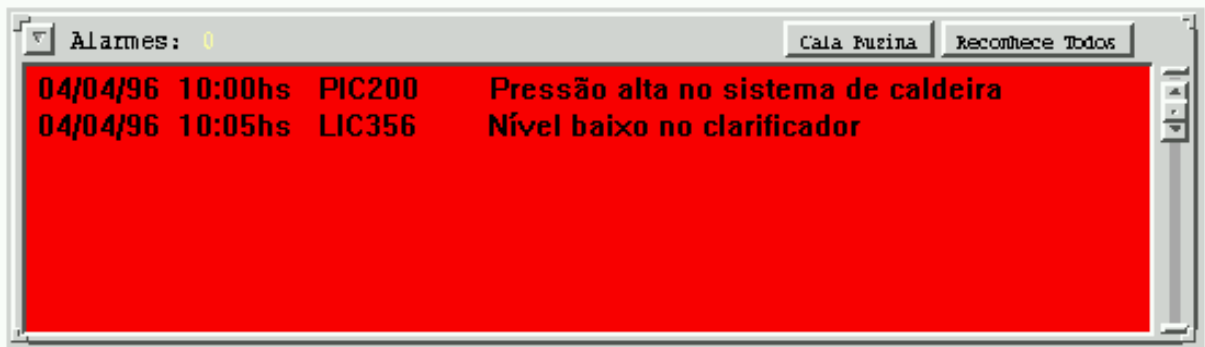
### **Sub-sistema de supervisão e monitoração**

O subsistema de supervisão e otimização consiste de um minicomputador (uso opcional) capaz de executar as funções de supervisão total do sistema, otimização do processo e a geração de relatórios gerenciais.

Suas principais funções e características são as seguintes:

Formatar e indicar condições de alarme nos consoles de vídeo e imprimi-las numa impressora de alarmes, conforme modelo à seguir

Alimentar janela de sumário de alarmes com tabela alfanumérica contendo os alarmes ativos, seus estados, reconhecidos ou não, e sua condição de alarme, crítico ou não, horários de ativação, reconhecimento e desativação. Obs.:Condições de alarmes também podem ser visualizados nas janelas de situação geral, grupo e individual. Quaisquer métodos poderão ser utilizados para notificar o operador da ocorrência de alarmes, como por exemplo: sinais sonoros; simbologia diferenciada; alteração de cores da tela; etc..



Coletar dados através dos subsistemas de controle e aquisição e registrá-los em meios magnéticos, tais como unidades de disco, para mostrá-los instântanea ou posteriormente nos consoles ou imprimi-los nas impressoras. Em termos de relatórios, normalmente estão disponíveis os seguintes:

- . Momentâneo: Emitido a pedido do operador, apresentando as variáveis de processo, seus tags, valores e situação do loop. Ex. Hard-Copy de tela, situação das variáveis, etc...

- . Evento: Emitido na ocorrência de um evento pré-configurado. Ex. Relatório de eventos de alarmes, transferência auto-manual, etc.

- . Periódicos: Emitidos periodicamente, conforme o período pré-configurado. Ex. Situação das variáveis de hora em hora, etc...

Realizar cálculos para atingir um ou mais objetivos de otimização da planta ou de consumo de energia e analisar a performance da planta ou equipamentos.

Gerenciar módulos de batelada, objetivando melhorar e desempenho de várias atividades que teriam que ser realizadas manualmente. Normalmente, o subsistema de supervisão e otimização permite o desenvolvimento de software de aplicação ou mesmo, de software de controle de processos em background, sem interrupção do sistema de controle, facilitando a alteração de configurações de controle, cálculos de performance, equações de balanço material e de energia, etc.



### Componentes básicos do subsistema de supervisão e otimização

O principal elemento deste subsistema é o que chamamos de computador hospedeiro (Host Computer).

Computadores são usualmente divididos em várias classes, com diferentes critérios de classificação. Velocidade, memória principal e custo podem ser usados para classificar os computadores. Tanto a velocidade quanto a memória dependem muito do comprimento da palavra, isto é, o número de bits que um computador pode processar por vez.

Os computadores geralmente são agrupados em quatro classes principais:

microcomputadores: são constituídos por uma única CPU, possuem 8,16, 32 bits de comprimento de palavra. Pode acessar 32 Mbyte ou mais de memória principal.

minicomputadores: são máquinas de 32 ou 64 bits. Podem acessar até 1 Gbytes de memória principal. Trabalham com mais de uma CPU. Possuem alta velocidade de processamento.

Mainframes: são qualificados pelo seu grande tamanho de memória e velocidade. Trabalham com várias CPUs, usualmente são encontrados como computadores centrais de grandes corporações.

Super computadores: são uma classe especial de processadores, frequentemente definidos como os computadores mais rápidos do mundo.

Estes sistemas são otimizados usando os mais recentes avanços em arquiteturas de computadores e hardware. Combinam vários processadores para realizar operações simultâneas, somente encontrados para uso científico.

OBS.: Vários pacotes de software rodam (simultaneamente na mesma estação) as funções de monitoração / operação e supervisão. Pode-se ainda configurar o software para dividir essa carga de trabalho, entre as várias estações que compõe a rede (sistema distribuído).

O computador hospedeiro, quando existe, geralmente consiste de um minicomputador com um tempo de acesso de memória razoavelmente rápido. Suas funções são muitas. Elas podem ser tanto processamento de palavras e dados como aplicações de controle diretas, geração de telas gráficas dinâmicas e ou programas de otimização e coordenação gerais de operação da planta.

Entre programas associados ao hospedeiro, temos programas de otimização, emissão de relatórios periódicos, entre outros mas sempre com a finalidade de fornecer informações de alto nível ao gerente da planta.

## SOFTWARE SUPERVISÓRIO

Os softwares supervisório para automação industrial são produtos que incorporam funções de:

controle supervisório, tais como: comando de atuadores de campo; monitoração de dados de processo (temperatura; nível; etc.); controle contínuo; controle por processamento em lote e controle estatístico;

alarmes de condições e estado de variáveis de processo;

emissão de relatórios,

aquisição de dados (“SCADA”).

soluções para processamento de batelada (“Batch”).

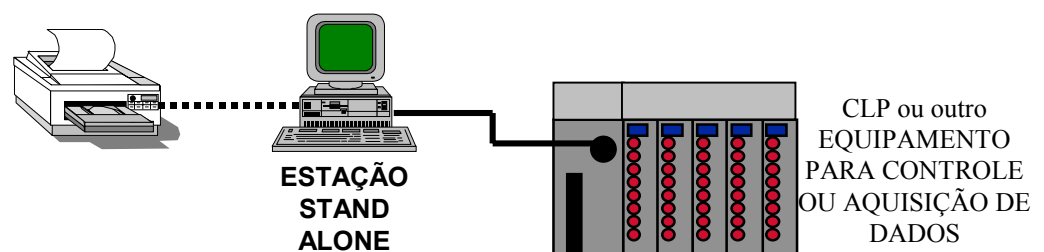
O software deve apresentar em tempo real, os dados de processo e produção tanto aos operadores quanto a outros aplicativos de software.

Dentre as funções mais importantes de software supervisório para automação industrial, tem-se que destacar na aquisição de dados a característica de vínculo bidirecional, onde os dados podem também ser enviados para o chão de fábrica, além de serem lido dos equipamentos de aquisição e controle. Uma característica importante é a capacidade de comunicação do software com equipamentos (hardware) proprietário de vários fabricantes diferentes, conseguido em função do uso de pacotes de software de interface, denominado driver de comunicação e melhor detalhado no texto à seguir..

É importante deixar claro, inicialmente, alguns conceitos importantes relacionado a aplicação dos softwares:

Estação (Nó) ⇒ Uma estação (nó) é qualquer computador que esteja rodando um software supervisório, sendo que uma estação (nó) local é aquela em que você esteja operando ou configurando e uma estação (nó) remota é aquela que é acessada através de um link de comunicação.

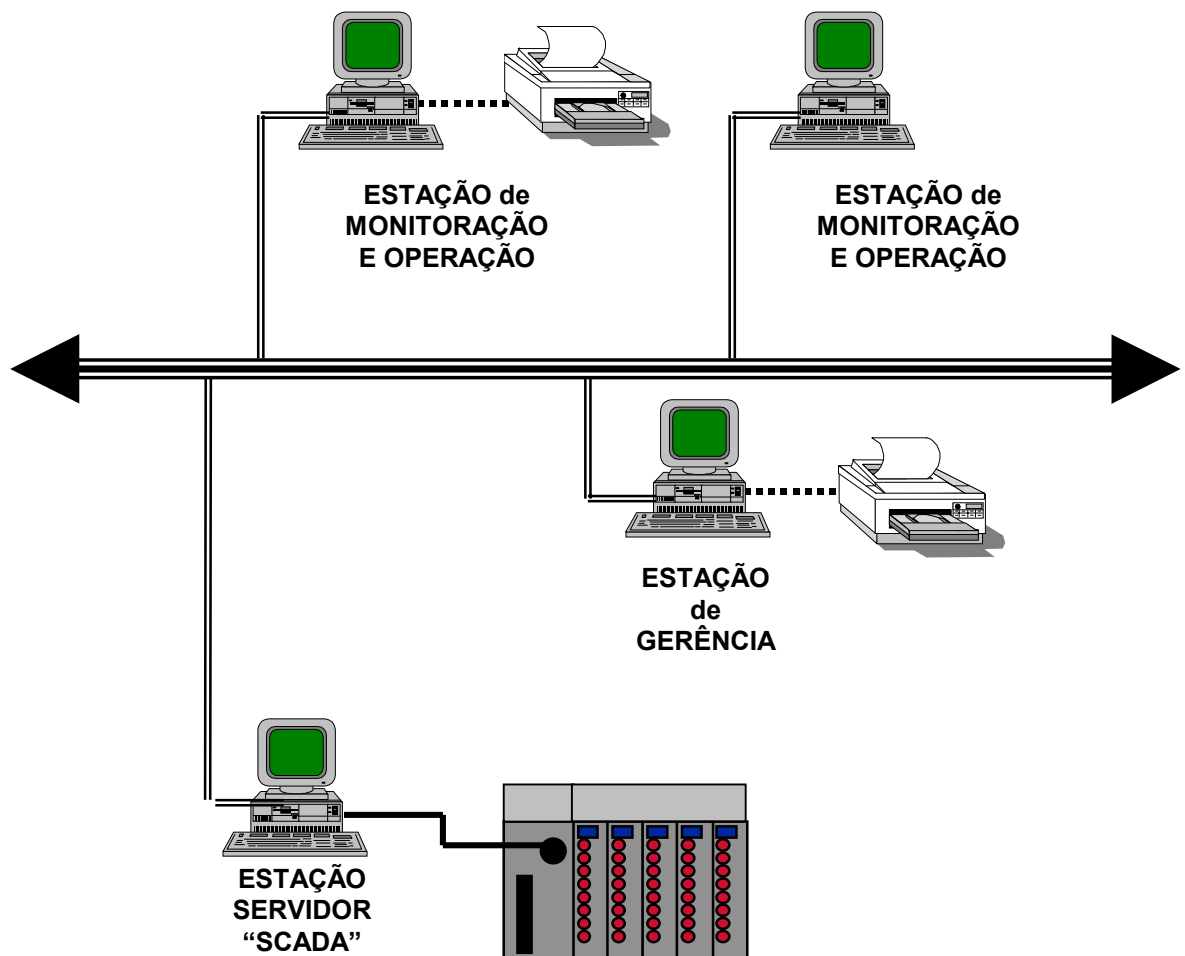
Estação Independente (“Stand Alone”) ⇒ É uma estação que desempenha todas funções de um sistema de supervisão não conectada a uma rede de comunicação.



Estação Servidora de Base de Dados (Servidor “SCADA”) ⇒ É uma estação que executa a função de aquisição de dados

Estação de Monitoração e Operação ⇒ É uma estação que permite que o operador monitore o processo, altere parâmetros do processo, reconhece alarmes e mais algumas tarefas de operação de processo, mas não permite alterar a configuração de telas nem da base de dados.

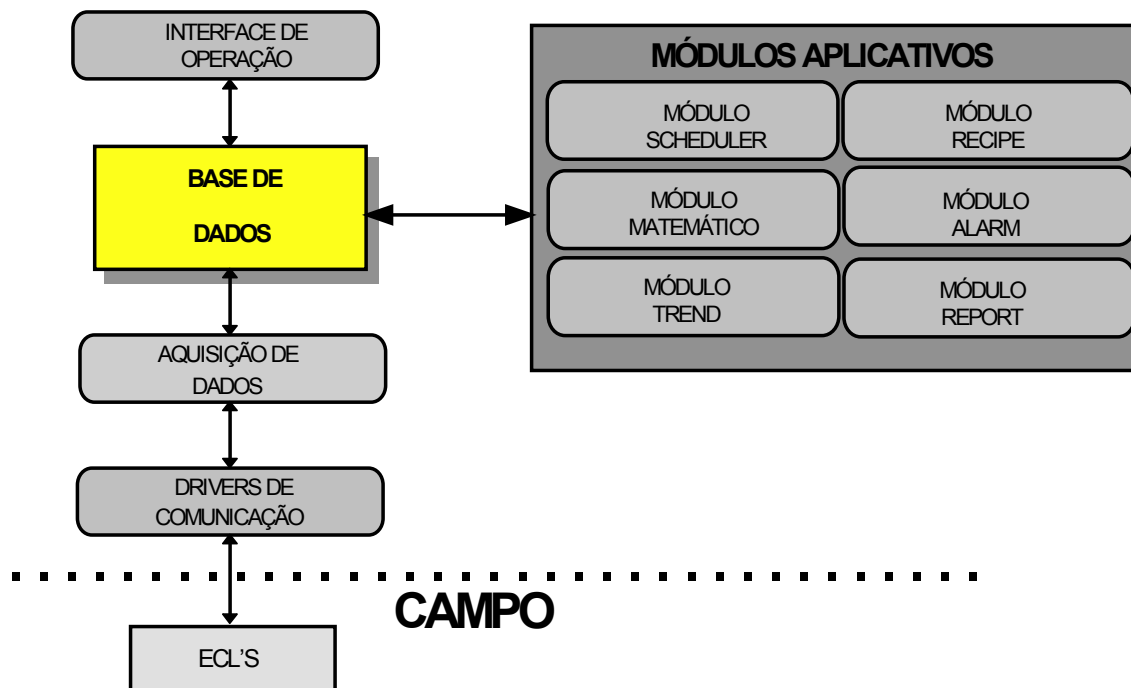
Estação de Gerência ⇒ É uma estação que permite a gerentes, supervisores ou quaisquer outra pessoas terem acesso aos dados de processo, em forma de relatórios; gráficos; telas, sendo que reconhecimentos de alarme, alteração de parâmetros do processo entre outras tarefas de operação não poderão ser realizadas nesta estação



## Arquitetura

Um software de supervisão e controle de sistemas de automação de processos tem como característica fundamental, a capacidade de ser multitarefa, ou seja poder executar vários módulos aplicativos simultaneamente.

Sua arquitetura simplificada poderá ser vista como:



Pela figura podemos notar que o centro do sistema é um grande **banco de dados** onde o mesmo é alimentado pelas informações oriundas do processo via ECL's (Estação de Controle Local). O mesmo é responsável em formatar e organizar esses dados para que outros módulos do sistema as utilizem em suas atividades.

A **interface de operação** é a parte do software supervisão com o qual é desenvolvido telas gráficas, para visualização e comando do processo que o sistema está atuando.

O **driver de comunicação** é um módulo de software capaz de ler e escrever dados nas ECL's (Estação de Controle Local) de campo, sendo que para fabricante de ECL, se faz necessário um driver de comunicação.

A **aquisição de dados** tem capacidade de comunicar-se com os diversos drivers de comunicação, permitindo o interfaceamento destes drivers com a base de dados, lendo e escrevendo na mesma.

Os **módulos aplicativos** são módulos que acessam a base de dados do software, gerando relatórios, gráficos, alarmes, acesso a outros banco de dados, etc..

Um importante conceito na estrutura de um SDCCD é o modo de trabalho denominado configuração, através do qual é montado e alterado, quando necessário, o conjunto de informações que serão utilizadas no funcionamento do Sistema Digital de Controle Distribuído.

O modo configuração tem geralmente as seguintes características:

Possibilidade de definição e de geração de sinóticos de fluxogramas de processo através de linguagens específicas ou de utilização da própria tela para geração dos desenhos, bem como animações respectivas das variáveis representadas em tempo real.

Possibilidade de definição do sequenciamento de operações de processos em batelada.

Note que para realizar essas funções cada fabricante fornecerá um módulo de configuração do sistema que atenda as características gerais acima mencionadas, podendo diferir brutalmente a forma como isso será implementado.

## **BASE DE DADOS**

Denominaremos base de dados ("Database") a um arquivo de dados em memória RAM do computador, contendo todas as variáveis atualizadas em tempo real para alimentação de dados do software supervisorio para automação industrial.

Todo o sistema de supervisão será alimentado com dados vindos de um CP ou ECL. Assim é necessário deixar estes dados disponíveis para que outros módulos (vistos no item anterior) possam utilizá-los. Cada ponto que é lido do CLP ou da ECL, tem seu valor armazenado em uma variável da aplicação, que daí em diante será chamada TAG. TAGs não são somente pontos lidos de equipamentos de campo, mas qualquer variável da aplicação, inclusive as pré-definidas do Sistema Supervisorio, que são chamadas TAGs INTERNOS. O conjunto de TAGs (internos ou da aplicação) ficam disponíveis no DATABASE (BASE DE DADOS), onde todos os módulos vão buscar valores ou alterar valores dos TAGS. O DATABASE funciona como uma conexão entre os módulos da aplicação, onde todos os módulos configurados só comunicam-se com o DATABASE e a execução do sistema ficará semelhante a um sistema multitarefa, cada módulo tem sua vez de ser executado e comunicar-se com o DATABASE, seguindo sempre a mesma sequência de execução dos módulos.

Essa base de dados será criada através de um software de configuração do sistema. Geralmente se dará sob a forma de telas a serem preenchidas informando os dados necessários para a configuração do sistema que se pretenda criar.

Nesse banco de dados encontraremos principalmente as variáveis de processo que pretendemos supervisionar referenciadas por mnemônicos, contendo informações adicionais tipo: range, alarmes, etc.

### Configuração da interface de operação

É onde se irá definir as telas do sistema e as variáveis que integrarão as mesmas. Faremos uma divisão por telas e em cada tela quais os dados geralmente solicitados.

## **TELAS DE GRUPO**

Essa fase da configuração é obtida após que todos os tags estejam definidos. O sistema configurador solicitará quais são os tags e a quais parâmetros eles possuem.

Esses instrumentos, normalmente já são previamente criados pelo sistema cabendo ao usuário definir qual o seu tipo. Os tipos mais comuns podem ser:

Indicação digital

Indicação analógica

Totalização

Controle digital

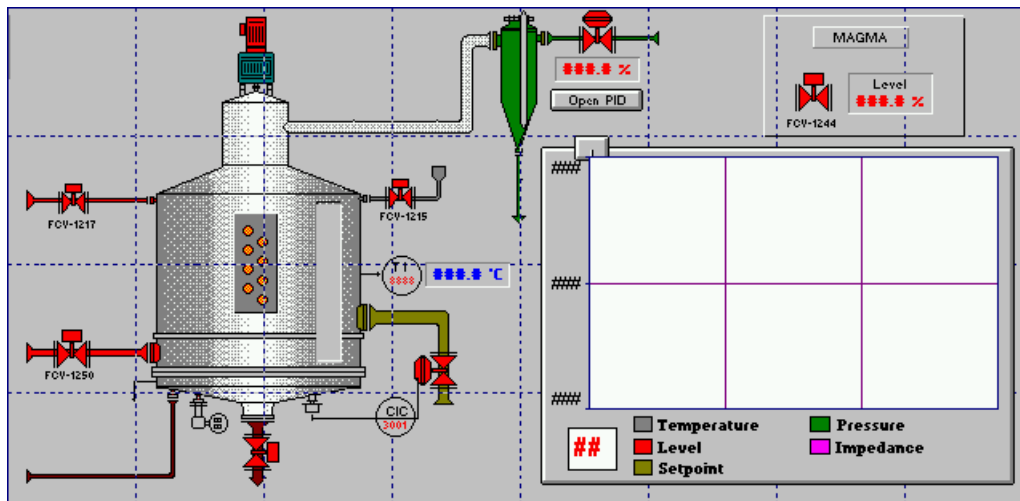
Controle analógico

Registrador

## TELAS DE SINÓTICO

Esta fase de configuração poderemos definir como a mais trabalhosa em termos de configuração. Isso porque teremos que criar um sinótico ou vários do processo em questão.

Como esses sinóticos são um retrato particular para cada planta, não existe então sinóticos pré-configurados, mas sim ferramentas gráficas que possibilitem a criação de qualquer desenho.



A maioria dos sistemas tem implementado softwares bem semelhantes aos softwares específicos na área de CAD, porém com recursos gráficos limitados, mas suficientes para a elaboração dos desenhos.

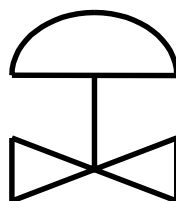
Outros fabricantes criam softwares de importação de arquivos de desenhos gerados diretamente em softwares de CAD. Isso torna o trabalho de criação do desenho mais suave.

Podemos dividir esta fase de configuração em dois principais módulos. Módulo de edição de desenhos e modulo de animação de desenhos.

### Edição de desenhos

O módulo de criação de desenhos permite que seja elaborado um desenho estático que poderá representar qualquer parte do processo.

O modo que os fabricantes permitem esse trabalho é o uso de elementos geométricos primitivos que agrupados convenientemente permitirão a criação do desenho.



Os elementos geométricos primitivos mais comuns são:

Pontos

Linhas

Arcos

Circunferências

Basicamente o processo de construção do desenho será o de escolher os elementos desejados e os fixar em determinada região da tela, como por exemplo caso desejássemos criar uma tela contendo o símbolo de uma válvula de controle com atuador pneumático, teríamos a figura final como abaixo com os seguintes elementos primitivos necessários para sua montagem: retas e arcos.

A inserção desses elementos na tela genericamente é feita da seguinte forma:

A tela é dividida em 2 grandes regiões: seletor de opções (toolbar) e região gráfica. O processo então, é relativamente simples. No menu de opções (toolbar) escolhe-se o elemento a ser editado e após localiza-se em qual região da tela gráfica o mesmo deverá ser inserido.

Como recurso adicional a tela gráfica é mapeada em coordenadas linha x coluna com um indicador mostrando qual a posição que o cursor está ocupando neste momento. Isso permite desenhos com suas dimensões definidas. Poderá inclusive ser ativado um comando que permita se visualizar esse mapeamento em intervalos definidos (grid).

Para inserção dos elementos temos basicamente os seguintes casos:

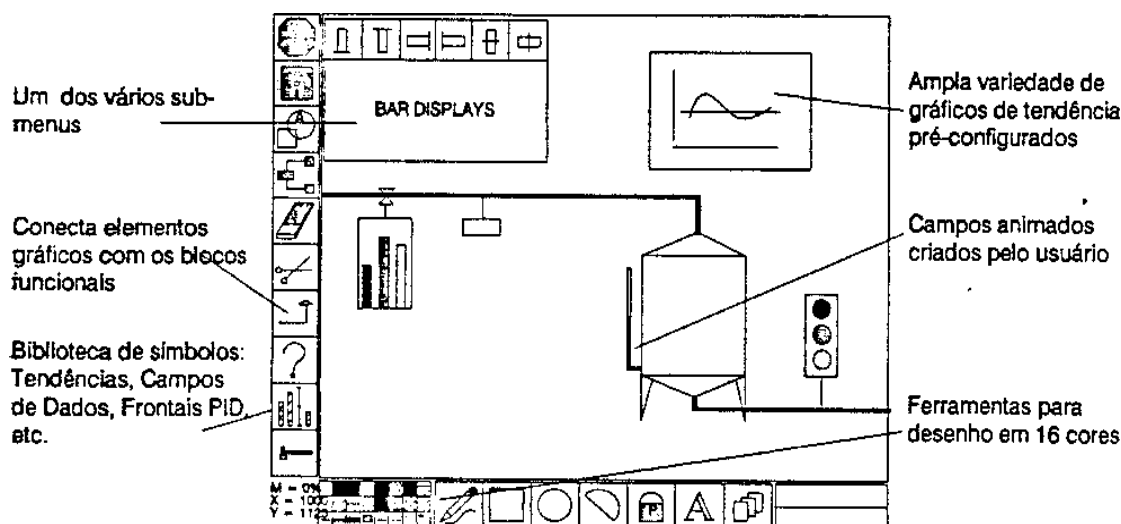
Pontos: Basta se localizar uma coordenada na tela;

Retas: Necessário localizar duas coordenadas na tela;

Arcos: Necessário localizar a coordenada de centro do arco;

Circunferências: Necessário localizar a coordenada de centro;

Quadriláteros: Necessário localizar as coordenadas de canto superior esquerdo e canto inferior direito.



Outros recursos que o sistema oferece na manipulação dos elementos poderão ser:

**Alterar cores:** Permite a escolha pelo menu, a cor que o elemento será desenhado;

**Pintar bloco:** Permite o preenchimento total de um polígono qualquer regular ou não com uma cor qualquer. Necessário somente localizar o cursor no interior do mesmo.

**Figura pré-definida:** Permite a inserção de um bloco de desenho vindo de uma biblioteca de figuras padrões. Essas figuras geralmente serão símbolos adotados pelas normas de instrumentação, como por exemplo: válvulas, tanques, aquecedores, bombas, etc. Para tanto basta escolher a figura desejada e o ponto de inserção no desenho.

**Texto:** Permite a inserção de caracteres de texto no desenho sendo, usados geralmente para legendas, tags de instrumentos, etc. Através de bibliotecas poderá se escolher o tipo de letras dos caracteres e sua escala.

**Mover região:** Permite o deslocamento de uma determinada região da tela para outro ponto.

**Escala :** Permite o aumento ou diminuição de elementos criados, como por exemplo o diâmetro de círculos.

**Rotação:** Permite a rotação de uma figura sobre seu eixo permitindo assim inseri-la de qualquer maneira desejada.

Os sistemas poderão oferecer ainda muitos outros recursos, sendo necessário um estudo particular de cada editor de figuras.

## ANIMAÇÃO DE DESENHOS

O módulo de animação de desenhos consiste em criar alterações no desenho montado anteriormente em função de valores das variáveis da planta que estão integradas no sistema digital.

Isso significa dizer que teremos um menu com os tipos de animações possíveis do desenho. Após inserida a animação no desenho o software solicitará qual será o tag da base de dados que irá comandar a animação.



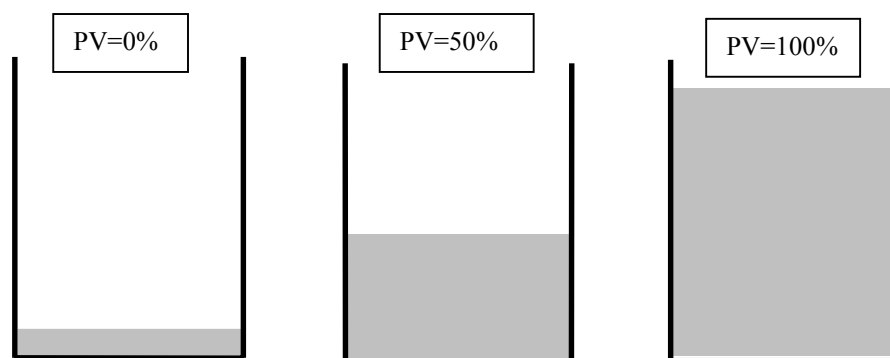
Tomemos por exemplo o desenho de um tanque TQ-01, e uma malha de controle com tag LIC-01. Um efeito interessante que pode ser realizado com a variável nível será o de pintar a região interna do tanque em função do valor da PV do LIC-01. Ou seja, se o valor da PV for 0% o interior do tanque não será pintado. Para 50% de PV teremos 50% do tanque pintados e conseqüentemente 100% de PV teremos todo o interno do tanque pintado.

Para isso ser realizado teríamos as seguintes fases:

Criar o desenho do tanque

Criar o efeito de mudança de nível

Ligar o efeito de animação com a variável PV do LIC-01



Como exemplos de animação que poderemos encontrar nos editores de desenhos poderemos citar:

**Bargraph:** Efeito de uma barra retangular que é preenchida internamente em função de uma variável analógica associada. Pode se escolher a cor de fundo e cor da barra proporcional a variável associada, as dimensões de altura e largura desta barra. A animação se dará no sentido vertical de baixo para cima. Como exemplo desse efeito temos a criação dos faces-plates dos controladores nas telas de grupo, onde temos os bargraphs de PV, SP e MV, proporcionais ao seu valor analógico.

**Valor numérico:** Cria uma janela numérica, que sendo associada com uma variável analógica, retorna seu valor. Esse efeito é usado na tela de grupo onde temos o bargraph dando idéia global do valor da variável e uma janela numérica com seu valor exato.

**Mudança de cor:** Efeito em que uma região retangular da tela pode sofrer mudança de sua cor em relação a cor de fundo. Esse tipo de animação é associada a uma variável digital. Como exemplo poderíamos citar uma animação do tipo em que teríamos o desenho de uma bomba com sua respectiva tubulação. Se o sistema receber o sinal digital informando que a bomba foi ligada, poderemos então mudar a cor no interior da tubulação, simulando assim o caminho que o fluxo estará fazendo pela planta.

**Atuação:** Efeito de se poder mudar o status de um ponto digital ou analógico através de uma janela aberta no sinótico. Isso permite a atuação do ponto sem a necessidade de ter que navegar até a tela de instrumentos. Ex. Ligar uma bomba.

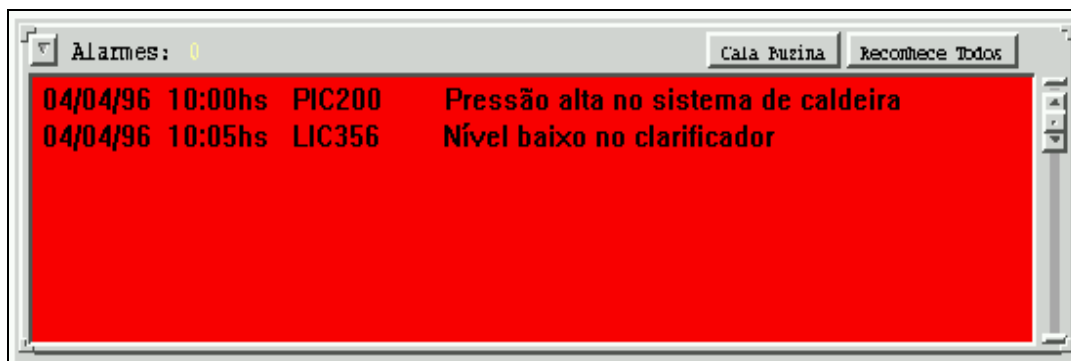
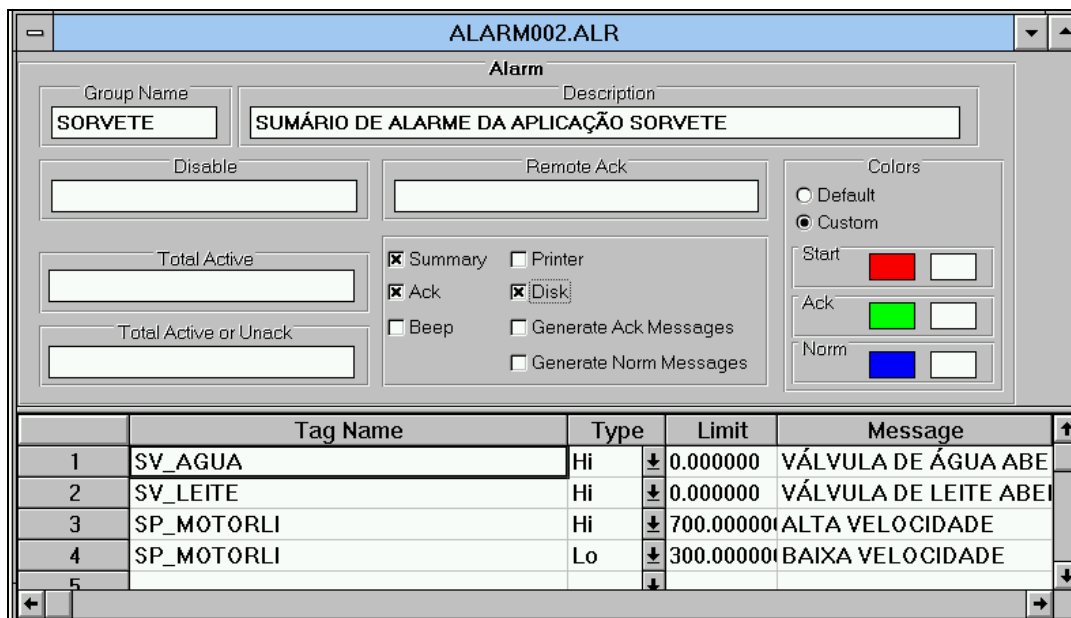
Note que em todas as animações descritas acima, os sistemas permitem escolha de cores, escalas conforme o mais indicado para o sistema que se pretenda configurar.

Configuração dos módulos aplicativos

Os módulos aplicativos do sistema, consistem em pacotes de software, tais como: scheduler ou clock; recipe (receitas); report (relatórios); matemático; trend (tendência); alarm, com o objetivo de otimizar o processo.

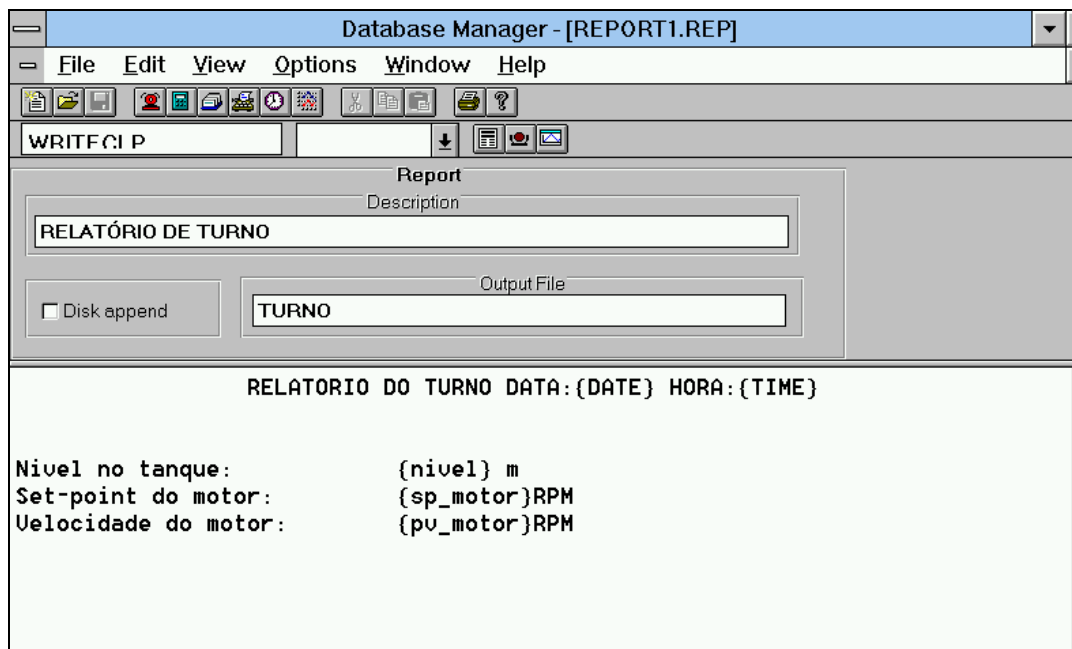
**MÓDULO ALARM**

Neste módulo, é possível definir as mensagens de alarme que o sistema irá emitir, assim como sob quais condições as mesmas serão emitidas. O pacote de software responsável pelo gerenciamento dos alarmes, definirá também, se as mensagens de alarmes serão impressas ou armazenadas em disco e ainda classificará os tags e suas mensagens em grupos de alarmes.



## MÓDULO REPORT (RELATÓRIO)

É possível definir relatórios de eventos, normalmente inerente ao sistema e de dados das variáveis, inclusive valores em função das próprias variáveis. Este módulo geralmente solicitará parâmetros de configuração, do tipo: cabeçalho do relatório, mneumônicos das variáveis que comporão o relatório, sistema de emissão de relatório.



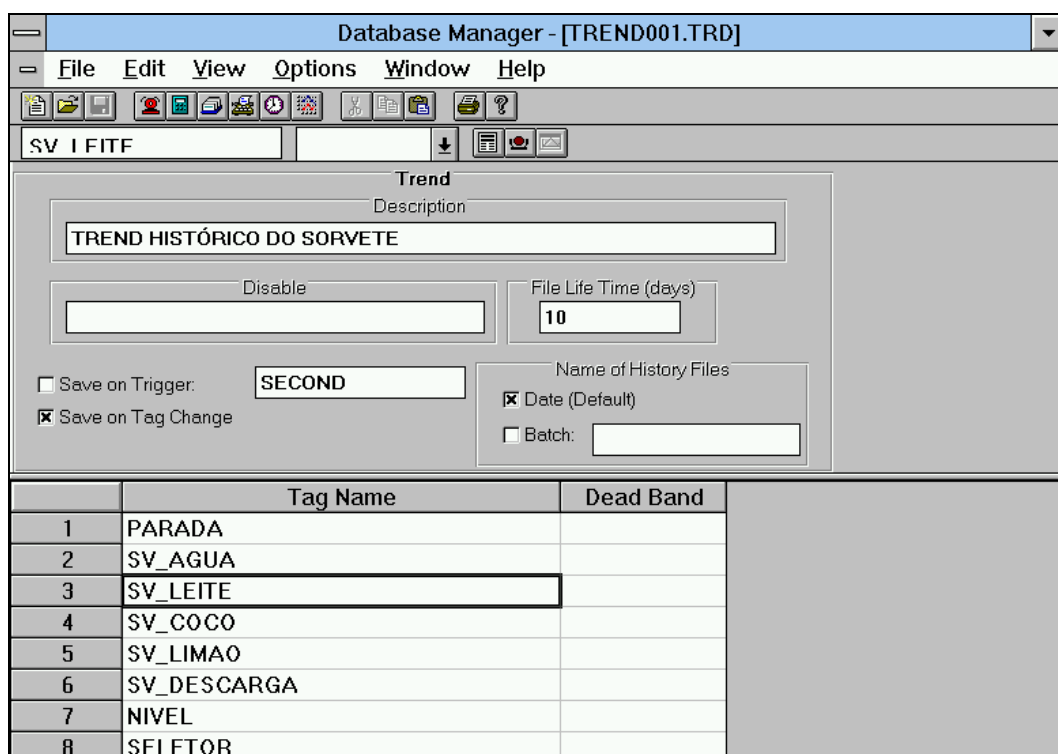
Quanto ao sistema de emissão de relatório poderemos ter a geração de relatórios instantâneos, ou seja o operador via console solicita o relatório com as variáveis naquele instante, ou ainda poderá ser definido a periodicidade automática para a emissão dos mesmos, como por exemplo de hora em hora.

## MÓDULO TREND

Neste item deve-se definir quais serão as variáveis que formaram as telas de histórico. Basta-se então informar qual o mnemônico da variável.

Outro parâmetro que o sistema irá solicitar será a base de tempo de salvamento, que poderá ser de 1 hora ou uma outra quantidade qualquer de tempo, como por exemplo 1 semana, 1 mês, etc..

Obs: Todos os valores das variáveis históricas são armazenados em unidades de disco. Note então, que o sistema cria uma região definida de memória para armazenamento desses valores. Assim sendo, fica o conceito de que ao ser iniciada a aquisição e gravação desses dados em disco, existirá um dado momento que esse espaço de memória será preenchido e então o sistema começa a sobrepor as primeiras informações que foram gravadas. (Sistema de memória FIFO).



A segunda observação, é a de que parece a primeira vista então, que quanto mais alta for a base de tempo, melhor será, pois teremos um maior espaço de tempo registrado. Isso realmente é verdade. A desvantagem é a de que como a quantidade de memória por variável histórica é finita então um sistema com base de tempo alta, terá uma taxa de amostragem também alta, o que poderá prejudicar a precisão do registro histórico da variável numa futura análise.

Para ilustrar melhor o que foi dito imaginemos uma área de memória definida por exemplo de 1000 unidades de memória. Imaginemos também 2 bases de tempo: 1 hora e 1 dia.

- A base de tempo de 1 hora fará aquisições da variável a cada 3,6s ( $3600s/1000$ ).
- A base de tempo de 1 dia fará aquisições de variável a cada 86,4 s ( $86400s/1000$ )

Note então enquanto a primeira base de tempo escolhida lê a variável num determinado intervalo de tempo a segunda o faz num tempo de 24 vezes maior.

Deverá então de acordo com a gerência de processos definir qual o valor ideal de base de tempo.

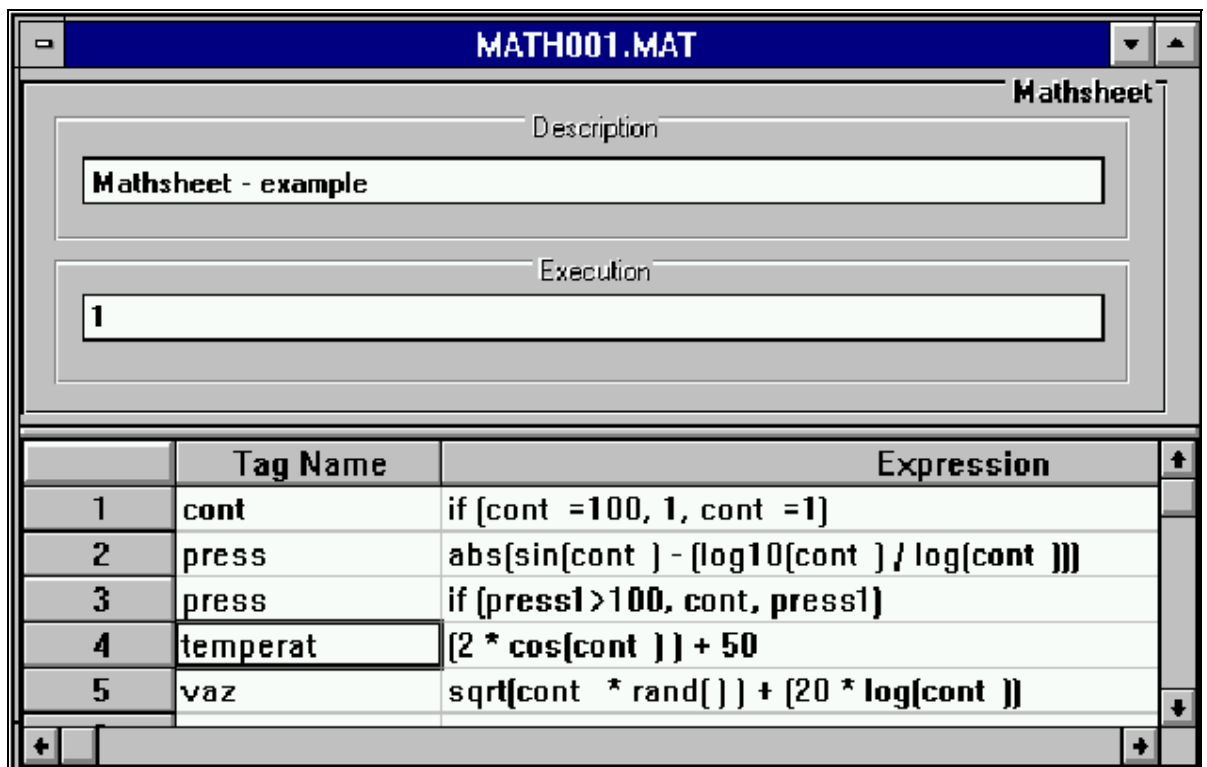
Um recurso adicional a esse sistema será o salvamento dos registros históricos em disquete. Ou seja poderá o sistema automaticamente ir descarregando o banco de dados em disquete. Quando o disquete é totalmente carregado o sistema cria um aviso ao operador para a troca do disquete por outro vazio.

Um detalhe interessante é que os dados armazenados em disco, podem ser migrados facilmente para sistemas de análise de dados (planilhas eletrônicas) sem nenhum inconveniente.

## MÓDULO MATEMÁTICO

Permite que sejam desenvolvidas rotinas lógicas e cálculos matemáticos, necessários para uma aplicação. Este módulo pode ser implementado na própria estação de trabalho, ou é possível também, existir um determinado computador em um nível hierárquico superior ao de uma estação de trabalho, executando esta tarefa, ou seja, recebendo valores oriundos dos níveis mais baixos (ECL' s) e efetuando algoritmos matemáticos com os mesmos. Esse resultado poderá servir para realimentar os níveis mais baixos. O propósito de se usar um computador específico para realizar esse tipo de função é devido ao fato de no nível de controle, temos processadores dedicados em funções de controle de processo em tempo real. Caso tentemos incrementá-los com cálculos avançados os mesmos poderiam perder essa característica, o que seria desastroso.

Geralmente o modo de se implementar tais cálculos será efetivado através do uso de linguagens de alto nível (C, Pascal, Fortran, etc) ou pacotes de softwares utilitários que venham com funções pré definidas, conforme ilustração à seguir.



## MÓDULO RECIPE (RECEITA OU BATELADA “BATCH”)

Este módulo, permite que um determinado computador ou a própria estação de trabalho, possa gerenciar a execução de procedimentos que normalmente seriam efetuados pelo operador do sistema. Esse recurso permite que os procedimentos sejam sempre realizados de forma sempre repetitiva, com uma precisão muito maior e inclusive com maior velocidade do que a mesma operação em modo manual. Esse módulo pode inclusive ser incrementado com rotinas de tomadas de decisão (If  $\Rightarrow$  Then  $\Rightarrow$  Else ) e ainda controle de loop (Do  $\Rightarrow$  While), que fazem que procedimentos complexos possam ser executados eficazmente.

Geralmente o modo de se implementar tais procedimentos será efetivado através do uso de linguagens de alto nível (C, Pascal, Fortran, etc) ou pacotes de softwares utilitários que venham com funções pré-definidas.

```

/*-----*/
/*- Programa executa a parada da planta piloto          -*/
/*- Use as seguintes funcoes para acessar o Banco de Dados do Tech 2000  -*/
/*-          -*/
/*- GetA( "Tag" )      : Le o valor de um tag analogico          -*/
/*- GetD( "Tag" )      : Le o valor de um tag discreto          -*/
/*- SetD( "Tag", value ) : Escreve o valor em um tag discreto (BD) -*/
/*- SetAECL("Tag", value ) : Escreve o valor em um tag analogico (CAMPO) -*/
/*- SetDECL( "Tag", value ) : Escreve o valor em um tag discreto (CAMPO) -*/
/*-          -*/
/*-----*/

#include <stdio.h>
#include <sys/sched.h>
#include <tech_dados.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <tech_log.h>

/* ATIVA MODO DE CONTROLE VIA SINGLE LOOP */

Log_printf(0,"Ativando selecao de controle para single-loop");
SetDECL( "RSL", 1);

```

```
Log_printf(0,"Controle single-loop ativado");
delay (2000);

/* PARANDO A MALHA DE CONTROLE DE PRESSAO TANQUE 2 */

Log_printf(0,"Iniciando procedimento de parada do controle de pressao do tanque 2");
Log_printf(0,"Ajustando set point de controle em modo remoto");
SetDECL( "PIC200#LR", 0);
delay (3000);
Log_printf(0,"Ajustando saida do controle em modo manual e posicionando saida em 0
%");
SetDECL( "PIC200#AM", 1);
SetAECL("PIC200#MV", 0 );
Log_printf(0,"Ativando solenoide de intertravamento da LCV200 na posicao aberta");
SetDECL( "SV200" , 1);

Log_printf(0,"F I M   D E   B A T E L A D A !!!!!");
    }
```

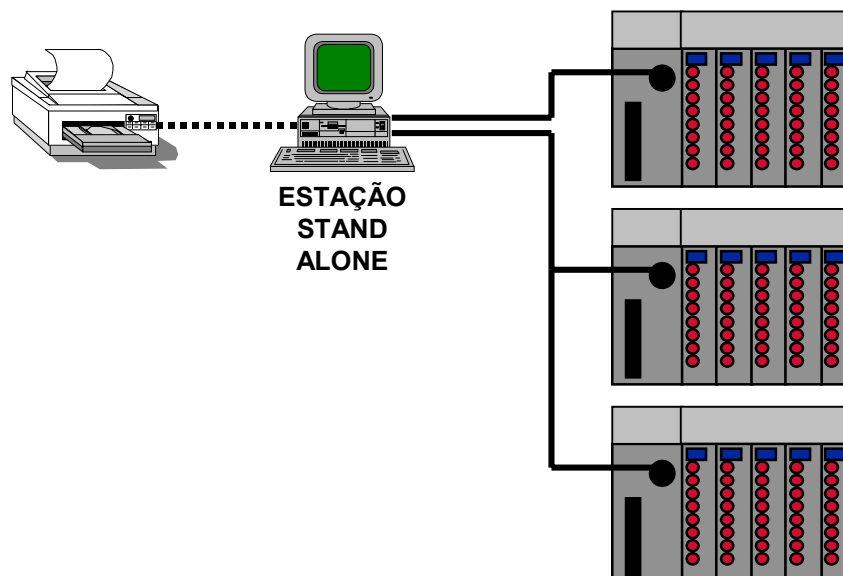
#### Configuração do módulo de comunicação

Esse módulo pode ser dividido na maioria dos softwares de configuração em duas partes:

- Configuração dos canais de comunicação
- Configuração dos endereços das variáveis

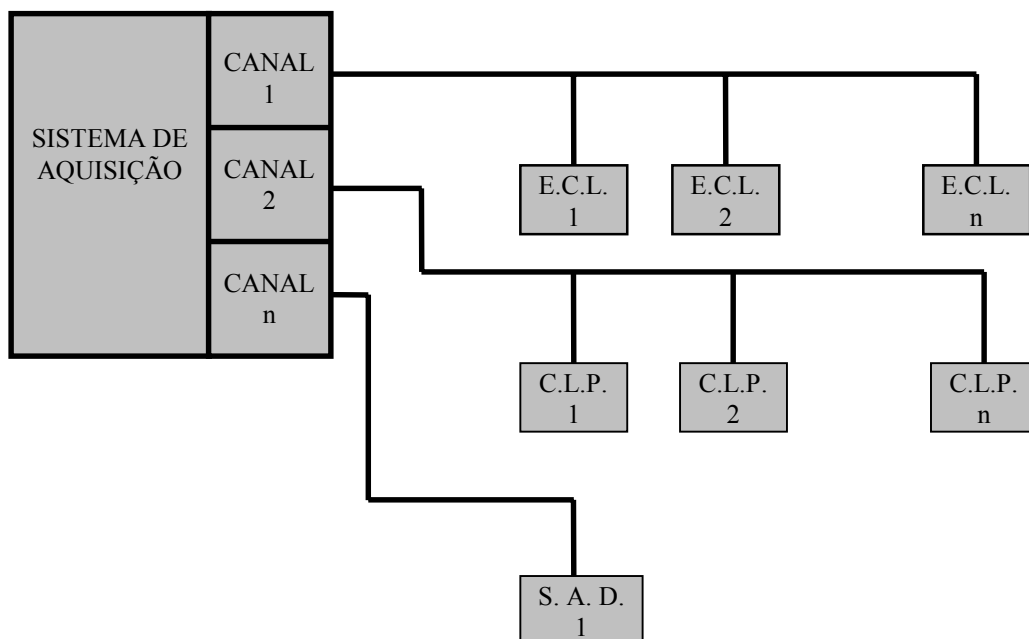
## CONFIGURAÇÃO DE CANAIS DE COMUNICAÇÃO

Num sistema digital distribuído temos vários canais de comunicação com os equipamentos distribuídos pela planta. O objetivo desse módulo é informar ao sistema qual equipamento estará conectado num determinado canal e informar (caso necessário) dados sobre a comunicação do sistema com o equipamento, como por exemplo: velocidade de comunicação (Baud Rate), quantidade de bits de dados, stop bits, paridade, etc.



Note que num canal de comunicação poderemos ter um ou vários equipamentos conectados, porém todos eles terão que ser o mesmo tipo de equipamento. Isso se deve ao fato de que cada equipamento tem seu próprio protocolo de comunicação, então o sistema deve possuir uma biblioteca contendo os vários protocolos dos equipamentos (Drives).





### CONFIGURAÇÃO DOS ENDEREÇOS DAS VARIÁVEIS

Este módulo irá dizer ao subsistema de monitoração e operação de onde virá e/ou irão os valores das variáveis do processo que serão monitorados.

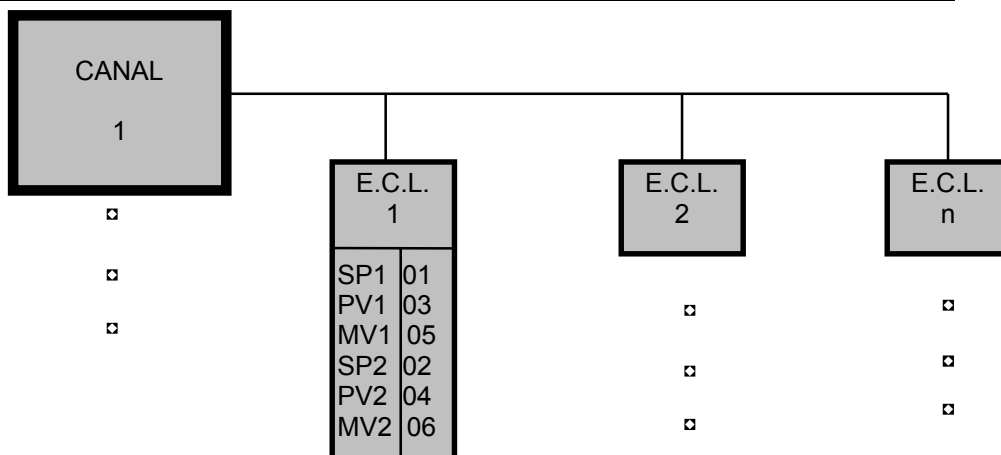
Basicamente consiste em criar-se um LINK entre um endereço da ECL e o tag da base de dados. A esse tag irá se definir o canal, o endereço do equipamento e o valor da memória do equipamento onde se encontra o valor da variável desejada.

Note que este passo é de fundamental importância para o funcionamento correto do subsistema de monitoração e operação, pois caso tenha-se endereçado incorretamente o valor de uma variável as informações apresentadas no console de operação não serão espelho do que está ocorrendo no processo, trazendo implicações desastrosas. É recomendável que após o sistema configurado e instalado, se realize um loop-test das variáveis tanto de aquisição como as de atuações para total confiabilidade do sistema. Suponhamos que as variáveis SP1, PV1 e MV1 se refiram à uma malha de controle com tag TIC2202 e SP2, PV2 e MV2 se refiram à malha com tag FIC2202.

Seus respectivos endereçamentos ficariam conforme o mapa à seguir:

TAG	CANAL	ENDEREÇO DO EQUIPAMENTO	ENDEREÇO DE MEMÓRIA
TIC-2202-SP	1	1	01
TIC-2202-PV	1	1	03
TIC-2202-MV	1	1	05

FIC-2202-SP	1	1	02
FIC-2202-PV	1	1	04
FIC-2202-MV	1	1	06



Para demonstrar a importância desses parâmetros vamos imaginar a seguinte tabela configurada:

TAG	C ANAL	ENDEREÇO DO EQUIPAMENTO	ENDEREÇO DE MEMÓRIA
TIC-2202-SP	1	1	01
TIC-2202-PV	1	1	04
TIC-2202-MV	1	1	06
FIC-2202-SP	1	1	02
FIC-2202-PV	1	1	04
FIC-2202-MV	1	1	05

Note que na malha do TIC-2202 o SP está endereçado corretamente, porém a PV e a MV estão endereçados da malha do FIC2202. Isso significa dizer que no console do operador o mesmo verá essa malha com a PV e MV do FIC2202.

Já na malha do FIC2202 somente a MV está errada, pois está endereçada à MV do TIC2202.

Imaginem caso o sistema não tenha sido checado e o operador inicie a operação da planta: ao selecionar no console o FIC2202 e o mesmo comandar a abertura da válvula do FIC, ele não verá o aumento da vazão do sistema( a válvula está endereçada à MV do TIC) e também não verá que está aumentando na realidade a PV do TIC2202.

As situações que podem ser geradas são inúmeras portanto, todo cuidado é pouco nessa fase de configuração.



## exercícios de configuração do software supervisor unisoft

### Exercício 1

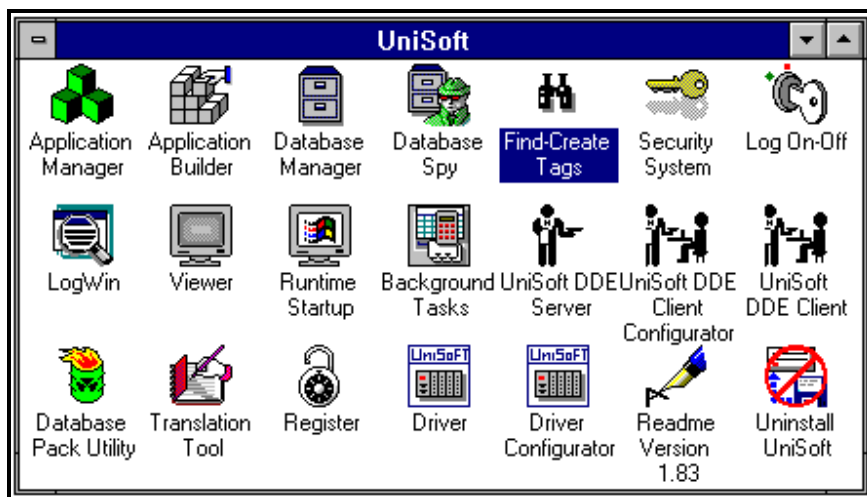
Este tutorial tem como objetivo a construção da simulação de um tanque reservatório contendo um determinado líquido, utilizando-se as funções disponíveis no editor de objetos do **UniSoft**.

É mostrada, também, a possibilidade de captura de um símbolo previamente criado para a criação de uma outra animação. Além disso você poderá usufruir de diversas ferramentas que o **UniSoft** irá lhe proporcionar, para a criação e manipulação de diversos objetos, segundo sua criatividade para confecção de suas telas.

### Grupo **UniSoft**

Uma vez dentro do ambiente Windows, é necessário que você siga determinados passos para começar a sua aplicação no **UniSoft**:

Abra o grupo de arquivos do **UniSoft** gerado pelo software instalador.



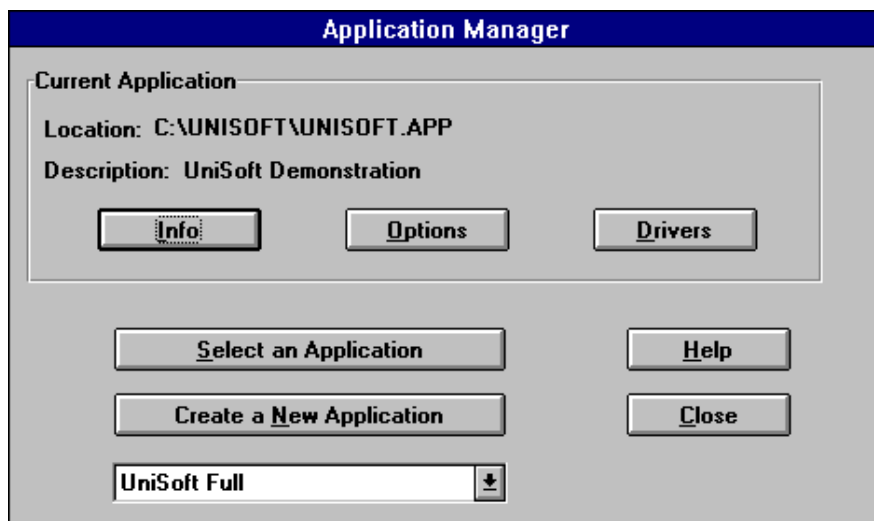
### Grupo de Programas **UniSoft**

Duploclique no ícone APPLICATION MANAGER, para começar sua primeira aplicação seguindo o tutorial.



Ícone do Application Manager

O APPLICATION MANAGER permite selecionar ou criar aplicações entre outras opções (detalhadas mais adiante nos tutoriais).

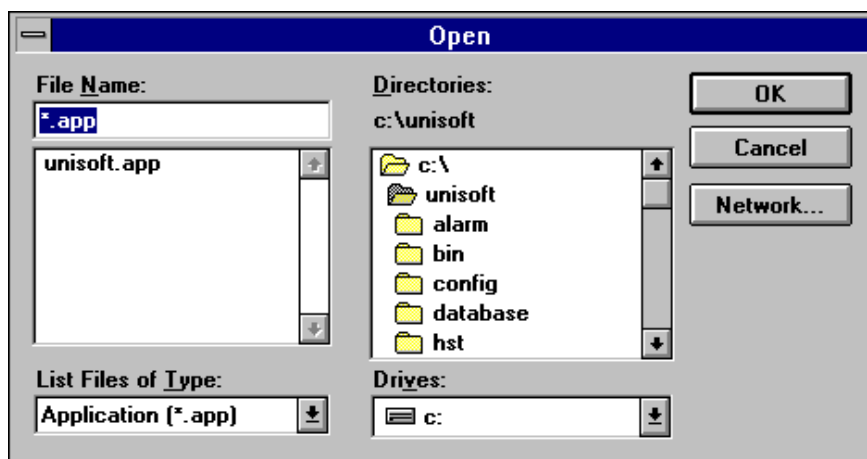


Application Manager

### Selecionando uma Aplicação do *UniSoft*

Selecione na janela do APPLICATION MANAGER a opção **Select an Application**, (para isso clique sobre o botão correspondente, ou digite a letra **S**) para selecionar uma aplicação.

Aberta a janela **Open** no campo **Directories** duploclique sobre o diretório raiz **C:** e, em seguida, sobre o diretório **UniSoft** (ou onde o **UniSoft** tenha sido instalado).



### Open

Duploclique sobre o arquivo UNISOFT.APP no campo **File Name** para a escolha da aplicação.

**NOTA:** Para selecionar uma aplicação você deverá escolher, certamente, entre alguma aplicação já existente. Como partimos do princípio que recém foi instalado o **UniSoft** no seu micro, com a opção de instalação do Demo, a aplicação indicada para você iniciar é UNISOFT.APP ou qualquer que apareça com

a mesma extensão .APP.

A seguir, feche a janela, clicando sobre a opção **Close** ou digitando a letra <C>.

### Criando uma Nova Aplicação

Para criar uma nova aplicação, siga os passos abaixo:

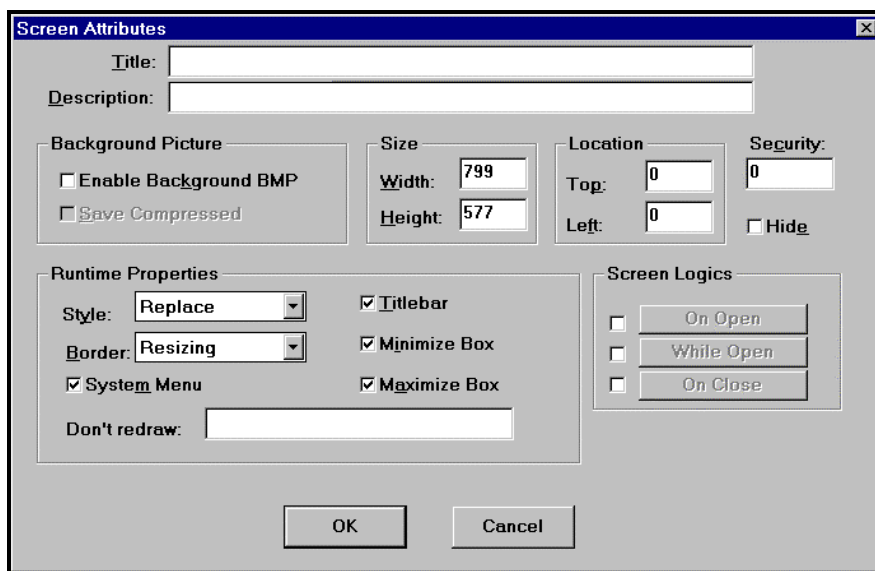
No APPLICATION MANAGER clique na opção **Create a New Application**.

No campo **Application Name** da janela que é aberta, digite um nome para sua aplicação. Este nome deve respeitar as regras de nomenclatura para arquivos no DOS (no máximo 8 caracteres) quando utilizada a versão para Windows 3.11.

Clique no ícone APPLICATION BUILDER, NA TELA PRINCIPAL DO UNISOFT.

No menu principal do APPLICATION BUILDER escolha com o mouse a opção de menu **File + New**. É aberta então a janela **Screen Attributes**. Vá com o mouse ao campo **Title** e digite o nome de uma tela que você deseja (qualquer nome, por exemplo: TUTORIAL TELA1). Já o campo **Description** serve para você inserir a descrição resumida do que se refere essa tela, assim sendo confirme com **Ok**, e pronto, a sua nova tela já esta pronta.

Terminando a confecção da sua tela salve-a. Para isso, no menu principal, clique em **File** e depois em **Save As**, e ela receberá o nome que você digitou no campo **File**. Salve esta tela com o nome **DEMO.SCR** se esta não existir.



### Screen Attributes

Seguidos estes passos, já é possível começar a criar uma primeira aplicação.

**NOTA:** Para selecionar telas no APPLICATION BUILDER basta clicar em **File + Open**. Aberta a janela **Open** basta selecionar no campo **Directories** o diretório raiz **C:/UniSoft/Screen** e por último escolher da lista de arquivos **\*.SCR** o arquivo que você deseja abrir.

### Apresentação do Toolbar



### Toolbar Numerado

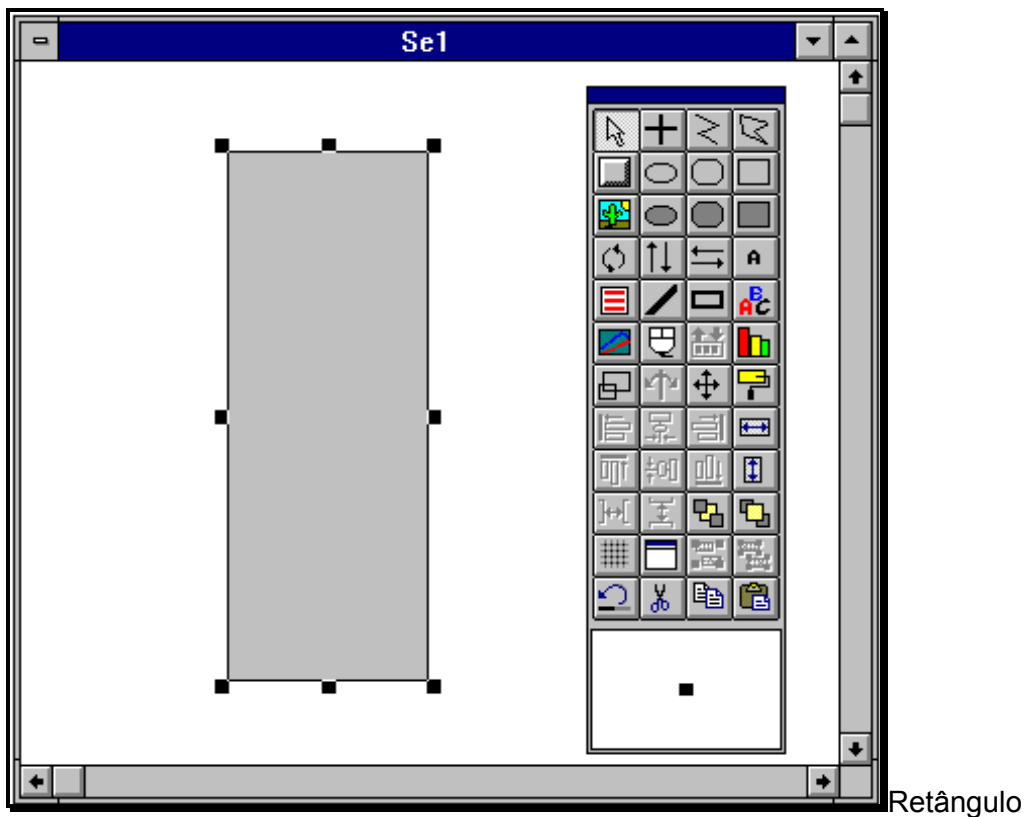
**NOTA:** Para melhor referenciar você nos componentes do Toolbar, além de mostrar a figura do componente é indicado suas coordenadas (entre parênteses) conforme a numeração da figura.

### Criando um bargraph

A animação de um *bargraph* (ou **barra gráfica**), é qualquer polígono fechado que possua, ao fundo, uma barra gráfica que se movimente de acordo com alguma variação ou algum algoritmo definido.

Como primeiro exemplo, supomos que um cliente exija a monitoração de um reservatório contendo um certo líquido. De acordo com um levantamento feito no campo e por uma exigência do cliente, o reservatório deve ser representado por uma figura retangular.





Para começar, desenhe um retângulo:

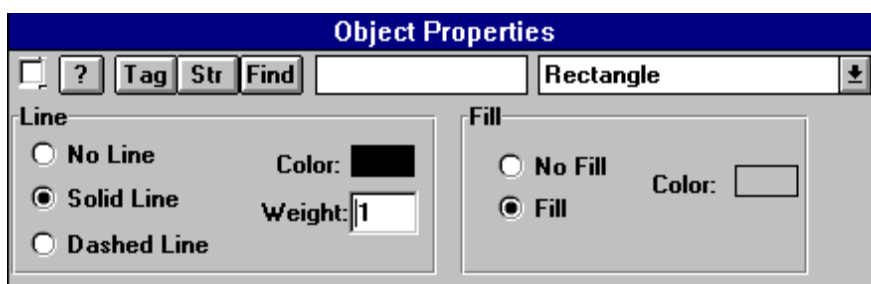


Clique com a tecla esquerda do mouse sobre o ícone de retângulos (D3).

Leve o mouse até o local desejado na tela e clique a tecla esquerda mantendo-a pressionada até definir a forma e o tamanho do reservatório.

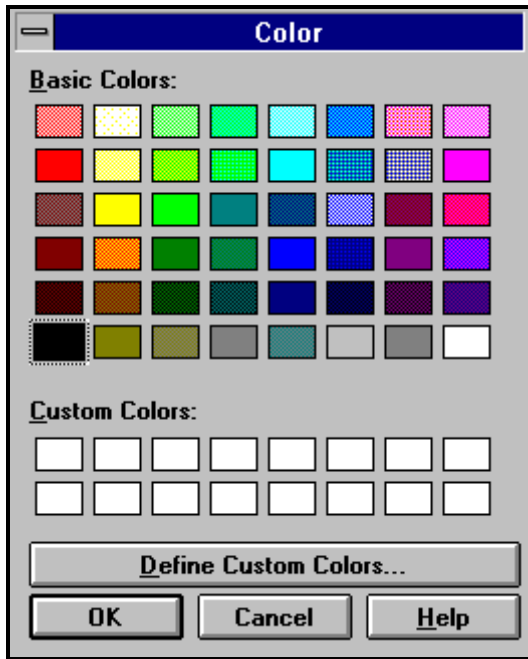
Desta forma, você acabou de criar um objeto.

Duploclique sobre a figura para chamar a janela de propriedades de objetos.



### Object Properties

Para preencher o reservatório com uma cor de fundo, clique no campo de preenchimento **Fill** e escolha a cor desejada, clicando sobre o retângulo colorido. Desta maneira, é aberta uma janela de escolha de cor. Clique a cor desejada e confirme com a tecla **OK**.



### Janela de Escolha de Cor

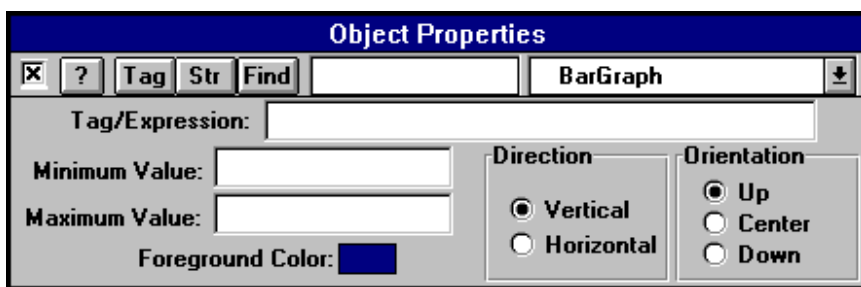
Agora, para definir uma dinâmica de *bargraph*:

Caso a janela **Object Properties** não esteja aberta, duploclique sobre o objeto recém criado.





Clique com a tecla esquerda do mouse sobre o ícone de **Bargraph** (D6).


A janela de atributos apresentar-se-á como abaixo.



### Bargraph

Se de imediato a janela de propriedades **Bargraph** não se apresentar, basta você ir com o mouse até o campo de listagem de objetos (representado por uma pequena flecha para baixo ) e ao clicar, perceberá que nesse campo estão armazenados todos os objetos e comandos que você está utilizando para sua tela os quais você já selecionou no *toolbar*, dessa maneira quando você precisar novamente de uma propriedade de objeto já utilizada ou que você selecionou e não substituiu no momento é só recorrer a essa listagem.

Se você preferir é interessante deixar a janela de propriedades sempre ativada, para isso vá com o mouse até o campo **Check Box**, representado por um quadrado pequeno no canto esquerdo superior de qualquer janela de propriedades  (do nosso caso a do **Retângulo**), e clique com o mouse. Dessa maneira, independente do objeto que você estiver usando do *toolbar*, sua janela estará ativada.

Abra a planilha matemática pelo menu principal **Tools** escolha **Database Manager** e clique no ícone . Ou do menu principal **Tools** clique em **Math** e em ambos os casos crie uma nova planilha clicando em **New**.

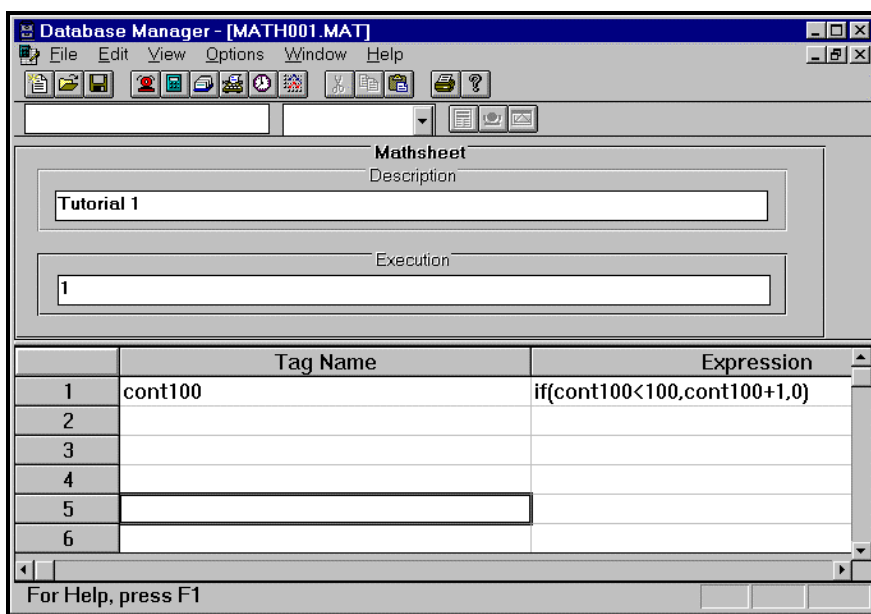
Uma utilidade interessante da planilha matemática é a criação de tags relacionados com simulação de variáveis de supervisão. No nosso caso, a variável de supervisão é o nível do reservatório, que será gerado pelo tag **cont100**.

Nos campos desta nova planilha:

**Description**- coloque um breve comentário para documentação e sua aplicação.

**Execution**- coloque 1, para que sua planilha seja sempre executada.

Na primeira célula da planilha, no campo **Tag Name**, digite **cont100**. Como esta variável não consta ainda no **Application Database** do DATABASE MANAGER (banco de dados), a planilha matemática perguntará se você deseja criá-la. Clique em **YES**, e escolha o tipo *integer* (inteiro) para sua variável, na janela **Tag**. Clique em **OK**. No campo **Expression** coloque **if(cont100<100,cont100+1,0)**.



## Planilha Math

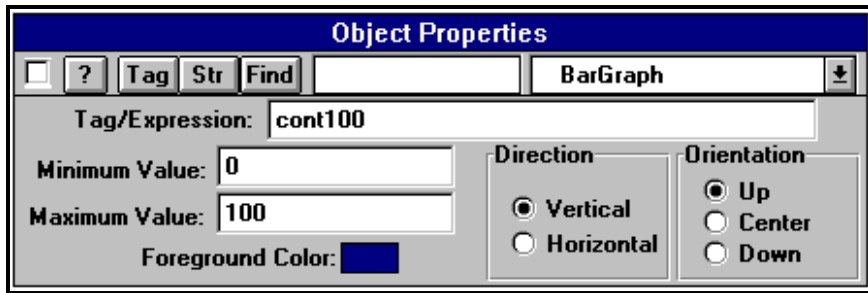
Salve e feche a planilha matemática. Volte para o APPLICATION BUILDER.

Insira a palavra **cont100** no campo **Tag/Expression** da janela **Object Properties** do retângulo. Este foi o tag que anteriormente você definiu na planilha matemática (abordada mais

adiante), de forma a permitir um acréscimo do nível do líquido do tanque de uma em uma unidade.

Defina como máximo o valor 100 no campo **Maximum Value** (100% do reservatório) e como mínimo o valor 0 no campo **Minimum Value** (0%).

Defina a direção da animação do reservatório no campo **Direction**. Como é uma animação real do nível de um tanque, é interessante a escolha da direção Vertical.



### Object Properties definido

Escolha a cor da animação a seu gosto, clicando sobre o retângulo de cores, a seguir, sobre a cor desejada na janela **Color** e confirmando com a tecla **OK**. Caso seja preciso, é possível alterar a cor de animação de dinâmica de barra em função do valor do tag de animação. Suponhamos que seja necessária a troca de cor para que o operador tenha uma visão mais precisa do que ocorre com o processo. Para isso:

Caso a janela **Object Properties** do objeto reservatório esteja desativada, duploclique sobre ele para ativá-la.

Clique sobre o ícone de preenchimento de cor (D7) do *Toolbar*. Desta forma, é ativada a propriedade de mudança de cor.

Clique sobre o campo **Tag/Expression** e digite o nome do tag já definido **cont100**.

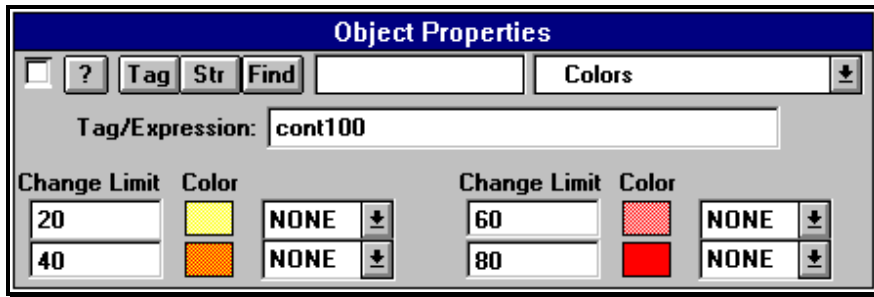
Suponha que seja necessária uma primeira mudança de cor quando o valor de **cont100** atingir **20**. Para isso:

Clique no primeiro retângulo do campo **Change Limit** e digite o valor **20**.

Clique sobre o campo **Color** correspondente.

Dentro da janela **Color**, clique sobre a cor desejada para mudança e confirme com **OK**.

Desta maneira, foi definido um valor limite para a primeira mudança de cor da animação do tanque reservatório. Caso você deseja definir outros limites para mudança de cor, proceda exatamente da mesma forma, simplesmente mudando o valor do limite e a cor desejada.



## Animação de Cores do Reservatório

A sua primeira aplicação está pronta para ser executada. Para testar esta aplicação, existem 2 modos:

1) Para executá-la dentro do APPLICATION BUILDER:

Clique sobre a opção **Tools** do menu principal.

Clique sobre a opção **Run BGTASK** do submenu para ativar o BACKGROUND TASKS, que permite a execução das tarefas de *background*.

Clique novamente sobre a opção **Tools** do menu principal.

Clique, agora, sobre a opção **Execute** do submenu, que permite a visualização da variação do nível do tanque.

Para encerrar a execução, clique outra vez a opção **Tools** do menu principal e, em seguida, a opção **Execute** do submenu.

**NOTA:** Na realidade, quando uma aplicação é executada no campo, a forma de execução é em modo *Runtime*, ou seja, fora do APPLICATION BUILDER.

## 2) Em modo *Runtime*:

Dentro da janela do APPLICATION BUILDER, clique na opção VIEWER do menu **Options** e complete os seguintes campos por enquanto:

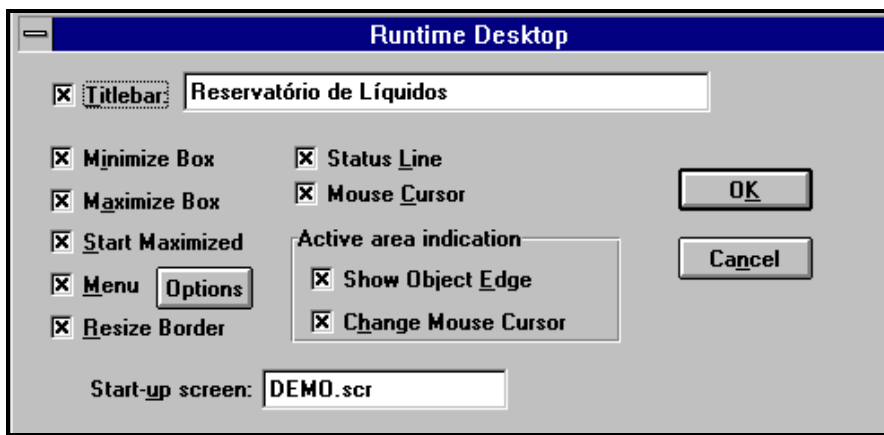
**Titlebar** - Um título para sua aplicação, por exemplo: RESERVATÓRIO DE LÍQUIDOS.

**Start-up Screen** - O nome de sua tela de aplicação feita no APPLICATION BUILDER, por exemplo demo.scr.

Feche esta janela, e de volta ao APPLICATION BUILDER. Clique na opção **Tools**.

Clique sobre a opção **Run BGTASK** do menu **Tools** para ativar o BACKGROUND TASKS, que permite a execução das tarefas.

Clique sobre a opção **Run Viewer** para executar sua aplicação (que, por enquanto, somente mostra uma retângulo na tela).



## Runtime Desktop

Agora é possível ver o aplicativo ser executado em modo de *Runtime*. Uma vantagem de se operar neste modo é a acessibilidade desta tela com as demais e vice-versa. Isto é possível via botão **Menu** disponível no pé direito da página, ou apertando **F11**, ou então apertando sucessivamente as teclas **Alt+Tab** até que apareça a indicação do módulo para onde você necessita se transportar.

Feche a janela do VIEWER, voltando para o APPLICATION BUILDER.

Capturando um Símbolo da Biblioteca

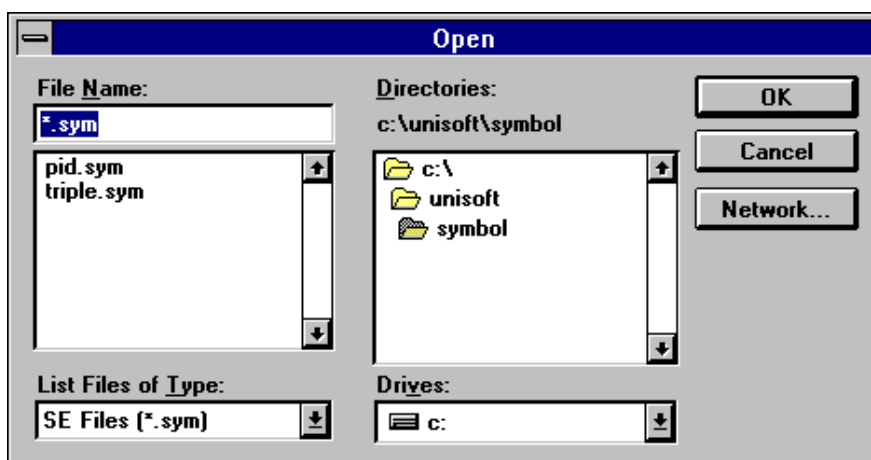
Nesta etapa do tutorial, você aprenderá a capturar um símbolo da biblioteca de símbolos para poder avaliar o uso de objetos prontos para a criação de novas aplicações.

Para proceder a captura, siga os seguintes passos:

Clique a opção **Edit** do menu principal.

Clique a opção **Paste From** do submenu.

Esta opção abre uma janela de símbolos do diretório **UniSoft**, como abaixo:



## Escolhendo um Símbolo

**NOTA:** Em uma nova aplicação, o diretório SYMBOL, estará vazio, portanto, utilize os símbolos do sub-diretório SYMBOL do próprio diretório UNISOFT. Para isto no box “DIRECTORIES” escolha a opção mostrada na figura acima.

Digite o nome **pid.sym** na seção **File Name** ou duploclique sobre seu nome na lista de símbolos.

Clique sobre este objeto e carregue-o até o lugar desejado na tela, mantendo a tecla do mouse pressionada.

O que se quer agora é implementar uma dinâmica sobre este objeto, que é o desenho do frontal de um controlador. Para isso:



### Clique sobre o ícone de retângulo (D3) do *Toolbar*.

Localize o mouse sobre qualquer um dos cantos do retângulo de leitura do controlador, no lado esquerdo do frontal.

Defina um novo objeto retângulo sobre o retângulo do frontal, clicando e mantendo a tecla esquerda do mouse pressionada e movimentando-o até que se tenham retângulos coincidentes e então solte a tecla.

Agora, é necessário definir a dinâmica deste novo objeto, que será feita nos mesmos moldes do objeto *Rectangle* do exemplo anterior:

### Duploclique sobre o objeto retângulo para abrir a sua janela **Object Properties**.



Clique sobre o ícone de dinâmica de barras (D6).

A janela de atributos apresentar-se-á como na figura da mesma forma anterior:

Insira a palavra **cont100** no campo **Tag/Expression**.

Defina como máximo o valor **100** (100% do reservatório) e como mínimo o valor **0** (0%), nos campos **Maximum Value** e **Minimum Value**, respectivamente.

Defina a direção da animação do reservatório na seção **Direction** como **Vertical**.

Escolha a cor da animação a seu gosto, clicando sobre o retângulo de cores, a seguir, sobre a cor desejada na janela de cores e confirmando com a tecla **OK**.

Caso você deseje, é possível a mudança da cor da dinâmica de barra, que será função do valor do nível do reservatório. Para isso:



### **Clique o ícone de preenchimento de cores (D7) do *Toolbar*.**

Desta forma, as propriedades na janela **Object Properties** apresentam-se como antes.

Digite outra vez o tag **cont100** dentro do campo **Tag/Expression**.

Defina um limite de valor para uma primeira troca de cor (valor **10**, por exemplo), no campo **Change Limit** e clique no campo Color correspondente para a escolha da cor de troca, em seguida, clique a cor desejada e confirme com o botão **OK**.

Para promover mais mudanças de cores (mais 3 cores, 4 no máximo), proceda da mesma forma nos campos seguintes.

Para fechar a janela **Object Properties**, simplesmente clique fora dela.

### **Executando a Aplicação**

Para executar a aplicação dentro do APPLICATION BUILDER, como no caso anterior, siga os passos:

Clique sobre a opção **Tools** do menu principal.

Clique sobre a opção **Run BGTASK** do submenu para ativar o BACKGROUND TASKS, que permite a execução das tarefas.

Clique novamente sobre a opção **Tools** do menu principal.

Clique, agora, sobre a opção **Execute** do submenu, que permite a visualização da variação do nível do tanque.

Para encerrar a execução, clique outra vez a opção **Tools** do menu principal e, em seguida, a opção **Execute** do submenu.

### **Executando a Aplicação em modo Runtime**

Para executar a aplicação em modo *Runtime*, proceda da seguinte forma:

Se o VIEWER já estiver sendo executado, salve a tela modificada, e esta será atualizada automaticamente. Caso contrário utilize a opção de menu **Tools + Run BGTASK** e **Tools + Run Viewer** respectivamente.

Como é possível perceber, tanto o tanque reservatório, como o frontal do controlador possuem a mesma dinâmica, o que é de se esperar, visto que ambos foram definidos pelo mesmo tag **cont100**.

### **Agrupando Vários Objetos**



Um recurso muito interessante em se trabalhar com objetos é poder agrupar vários em um único objeto (também chamado de símbolo). Um exemplo de aplicação para isso é quando se deseja acrescentar uma determinada dinâmica com um determinado objeto anteriormente criado, como no nosso caso, onde temos um símbolo de frontal de controlador em conjunto com uma dinâmica de barras (*bargraph*).

Caso se deseje, é possível criar-se um outro símbolo semelhante ao símbolo **pid.sym** capturado anteriormente. Para isso:

Clique a tecla esquerda do mouse sobre um dos objetos do conjunto frontal-nível do tanque (**pid.sym**, por exemplo).

Aperte a tecla <**Shift**>, mantendo-a pressionada, e clique a tecla esquerda do mouse sobre o outro objeto *Rectangle* (a dinâmica de barras, por exemplo).



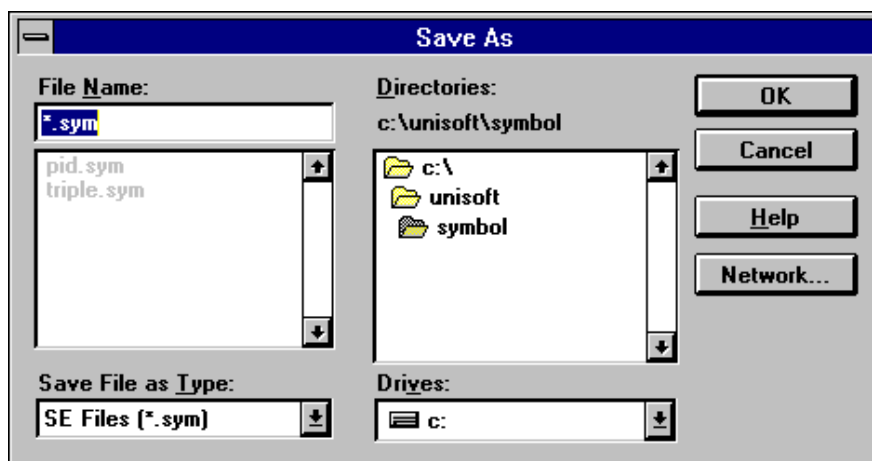
Clique sobre o ícone de agrupamento de objetos (C11).

Agora existe um novo objeto criado a partir de outros dois objetos e que pode ser salvo num único arquivo na biblioteca de símbolos, para tanto:

Clique com a tecla esquerda do mouse sobre este novo objeto.

Clique sobre a opção **Edit** do menu principal e, a seguir, sobre a opção **Copy to** do submenu.

Dentro da janela de cópia de símbolos, digite um nome para o novo símbolo dentro do campo **File Name** (**pid2.sym**, por exemplo).



### Salvando um Novo Símbolo

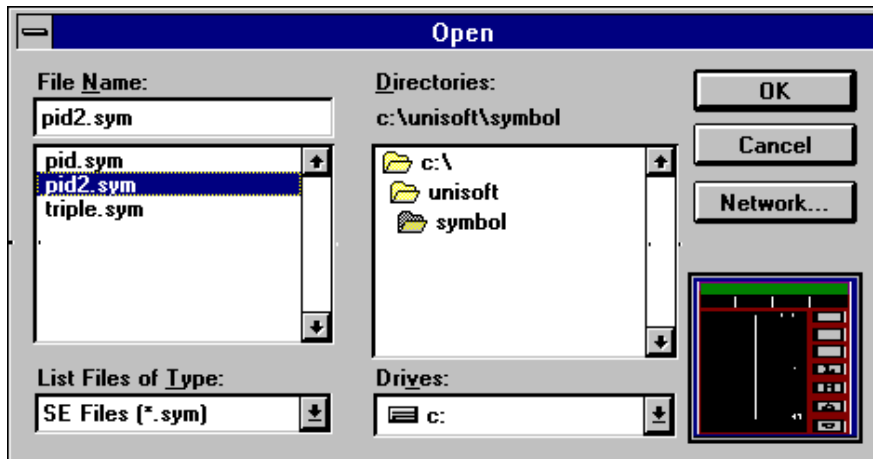
Confirme com um <**Enter**> ou clicando **OK**.

Para verificar a validade do novo símbolo criado, apague-o da tela, clicando sobre ele e teclando <**Delete**>. A seguir, proceda da seguinte maneira para capturar o novo símbolo **pid2.sym**:

Clique sobre a opção **Edit** do menu principal.

Clique sobre a opção **Paste from** do submenu.

Uma vez dentro da janela de captura de arquivos, duploclique sobre o nome **pid2.sym** ou digite seu nome na seção de **File Name**, confirmando a escolha com um <Enter> ou teclando **OK**.



### Selecionando Pid2.sym

Posicione o objeto em algum lugar desejado da tela, clicando com a tecla esquerda do mouse sobre ele e, em seguida, conduzindo-o.

Para executar este aplicativo, proceda da mesma maneira que antes:

### Execução

Clique sobre a opção **Tools** do menu principal.

Clique sobre a opção **Run BGTASK** do submenu para ativar o BACKGROUND TASKS, que permite a execução das tarefas.

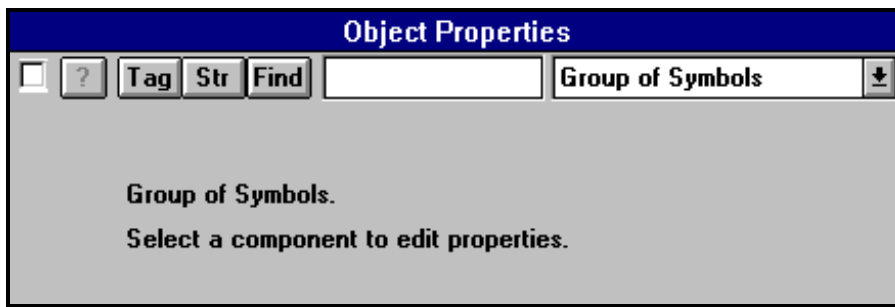
Clique novamente sobre a opção **Tools** do menu principal.

Clique, agora, sobre a opção **Execute** do submenu, que permite a visualização da variação do nível do tanque.

Para encerrar a execução, clique outra vez a opção **Tools** do menu principal e, em seguida, a opção **Execute** do submenu.

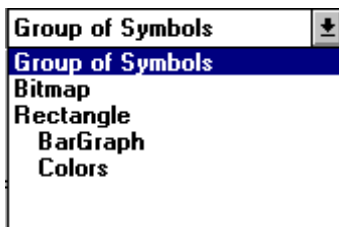
**NOTA:** O aplicativo deve executar da mesma maneira anterior, mostrando que uma vez criado um objeto com suas dinâmicas, ou mesmo acoplando vários objetos, é possível armazená-lo e recuperá-lo como um elemento único, composto de uma série de atributos.

Só para enfatizar a questão dos atributos de objetos, é interessante você duploclicar sobre o objeto recém capturado. No canto superior direito da janela de **Object Properties**.



### Grupo de Símbolos

É possível perceber uma série de propriedades deste objeto, ao clicar sobre a lista no campo superior direito, onde consta **Group of Symbols**. Esta lista fornece a lista de todos os atributos do objeto.



### Atributos do Objeto Agrupado

Para esclarecer, clique sobre a propriedade *bargraph*. A janela vai apresentar todas as características da criação da dinâmica de barras para este objeto, exatamente como criamos anteriormente para o objeto *Rectangle*.

E finalmente você chegou a última etapa do primeiro tutorial, esperamos que já tenha dado para você perceber quantas opções de configuração o **UniSoft** lhe oferece. Parta agora mesmo para o segundo tutorial pois muito mais lhe espera.

### Exercício 2

Este tutorial mostra como criar uma saída de dados na tela através de um *display*, que tem a função de fornecer o valor numérico de uma determinada variável que está sendo monitorada.

Numa segunda etapa, é criado um novo tag contador na planilha matemática para a simulação de nível do reservatório.

### Construindo um Display para Saída de Dados

Em muitos casos é de vital importância que, além da visualização de uma determinada dinâmica, tenha-se o seu valor numérico. No caso do nosso exemplo, é interessante a criação

de um *display* que forneça o valor do nível do reservatório, o que complementa de uma forma importante sua monitoração.

Vamos continuar com a tela DEMO.SCR do **Tutorial 1**, para começar a editar a tela do seu **Tutorial 2**. Mas antes de começarmos, desagrupe o grupo de objetos criado anteriormente e remova o bitmap **pid.sys** (seguindo os passos abaixo), pois não vamos precisar deste objeto.

Para remover o objeto:

Clique sobre o objeto **pid.sym**.

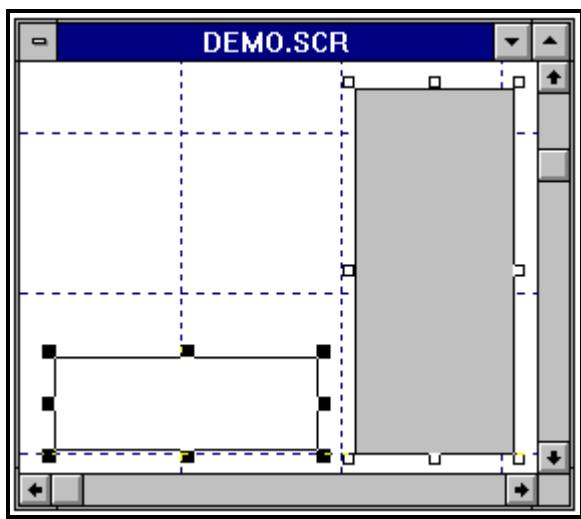
Apague-o, teclando <Delete>.

Primeiramente, é preciso criar o formato do *display*. Uma forma bem rápida e simples é um *display* retangular. Para tanto, proceda da seguinte forma:



Clique com a tecla esquerda do mouse sobre o ícone de retângulos (D3).

Leve o mouse até um lugar desejado na tela e construa um retângulo clicando a tecla esquerda do mouse e mantendo-a pressionada até constituir um retângulo de dimensões coerentes para um *display* (2cm de altura por 4cm de comprimento, por exemplo).



## Display

Duploclique sobre este novo objeto para ativar a janela **Object Properties**.



## janela Object Properties Display

Escolha a cor de fundo do *display* pressionando o retângulo de cores dentro da seção **Fill**, a seguir, na janela **Color**, clicando a cor desejada e confirmando com o botão **OK**.

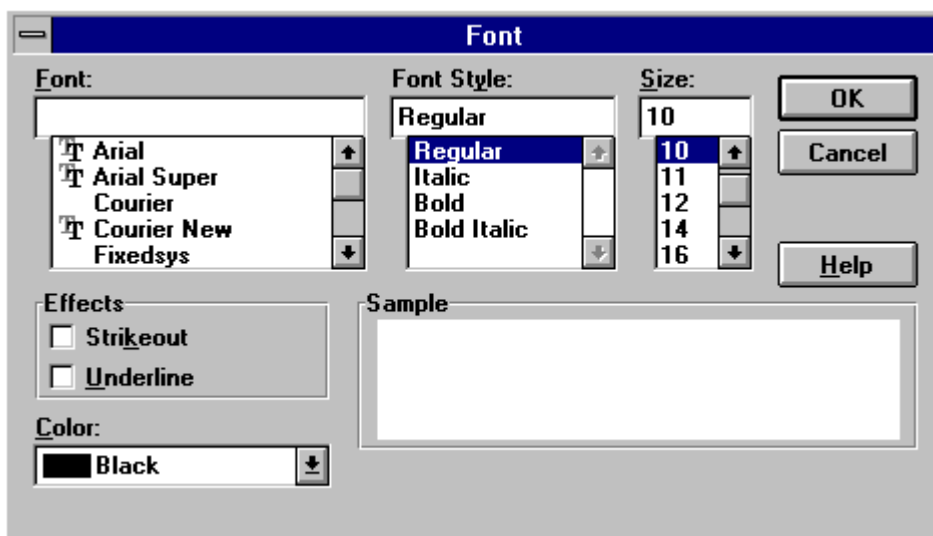
De volta à janela de propriedades, selecione o modo **Fill** da seção **Fill**.

Agora você definiu a parte estática do seu objeto *display*. Falta definir ainda o principal: a saída numérica.

## Modificando Fontes para Saída Numérica



Caso deseje definir um determinado tipo de fonte, clique primeiramente sobre o ícone de definição de tipo de fonte (D5), que abre uma janela para a escolha do tipo e tamanho da fonte a ser utilizada para o *display*.



## Janela para Escolha de Fontes

Criando um String para Saída Numérica



Clique sobre o ícone de texto (D4).

Leve o mouse até a área desejada do seu *display* e clique com a tecla esquerda.

Para definir o número de caracteres de saída do *display* (que no nosso caso é 3, visto que o valor máximo é **100**), digite a mesma quantidade de símbolos **#** do teclado (aqui, digite 3 vezes **#**).

Você acabou de definir uma saída string na sua tela. Falta agora definir com que tag ela se relaciona. Como o que se pretende ler é o nível do tanque reservatório (**cont100**), é preciso relacioná-lo com a saída.

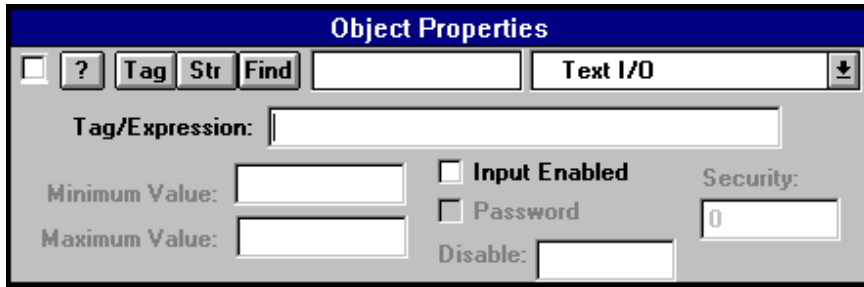
## Habilitando uma Saída de Dados

Duploclique sobre o novo objeto de saída de dados (**###**) para ativar a janela **Object Properties**.



Clique com a tecla esquerda do mouse sobre o ícone de saída de dados (C6).

A janela **Object Properties** da saída numérica deve apresentar-se como na figura seguinte.



### Objetct Properties Saída Numérica

Leve o mouse até o espaço **Tag/Expression** e defina o nome do tag que vai ser lido, neste caso é o tag **cont100**.

Clique fora da janela de propriedades para desativá-la.

### Agrupando os Objetos

Agora Selecione o objeto “###” (de saída numérica) , pressione a tecla **<Shift>** e selecione o retângulo (*display*).



Clique sobre o ícone de agrupamento (11C).

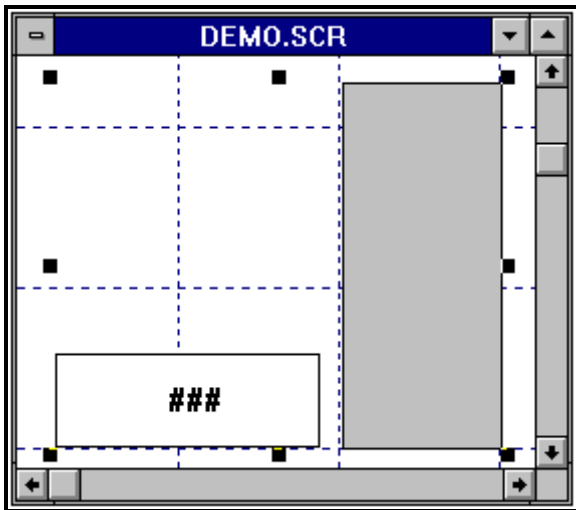
Você acaba de agrupar estes dois objetos transformando-os em um grupo de símbolos.

Selecione este novo grupo de símbolos, pressione a tecla **<Shift>** e selecione o retângulo maior (tanque reservatório).





Clique sobre o ícone de agrupamento (11C).

Agora você agrupou todos os seus objetos em um único grupo de símbolos.



Display do Reservatório

**NOTA:** Para relacionar entrada/saída de dados do Toolbar  (C11) com texto  (D4), este não deve estar agrupado.

### Executando DEMO.SCR

Criado o display para leitura de nível do reservatório, é possível executar a aplicação, seguindo os mesmos passos anteriores:

Clique sobre a opção **Tools** do menu principal.

Clique sobre a opção **Run BgTask** do submenu, caso não esteja ativada, para ativar o BACKGROUND TASKS, que permite a execução das tarefas.

Clique novamente sobre a opção **Tools** do menu principal.

Clique, agora, sobre a opção **Execute** do submenu, que permite a visualização da variação do nível do tanque.

Para encerrar a execução, clique outra vez a opção **Tools** do menu principal e, em seguida, a opção **Execute** do submenu.

### Em modo Runtime

Para executar a aplicação em modo **Runtime**, proceda da seguinte forma:

Se o VIEWER estiver em execução, salve a tela modificada, que esta será atualizada automaticamente no módulo de *runtime*. Caso contrário utilize a opção de menu **Tools + Run BgTask** e **Tools + Run Viewer** respectivamente.

#### Criando um tag contador

Agora que você está familiarizado com a criação de uma dinâmica de barras, captura de símbolos e saída de dados; é interessante conhecer como implementar um algoritmo matemático via planilha matemática.

Uma utilidade interessante da planilha matemática é a criação de tags relacionados com simulação de variáveis de supervisão. No nosso caso, a variável de supervisão é o nível do reservatório, que é gerado pelo tag **cont100**.

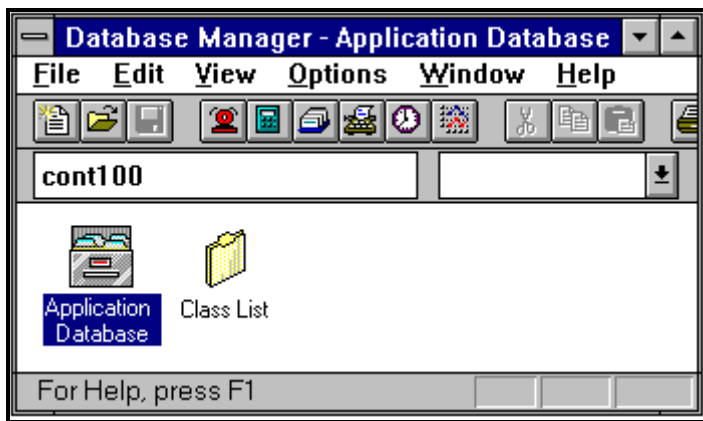
O que se quer agora é a monitoração da porcentagem de líquido no reservatório. O tag **cont100** deve ser visível ao usuário, mostrando 'x' unidades no display.

Abrindo o Database Manager

Clique a opção **Tools** do menu principal.

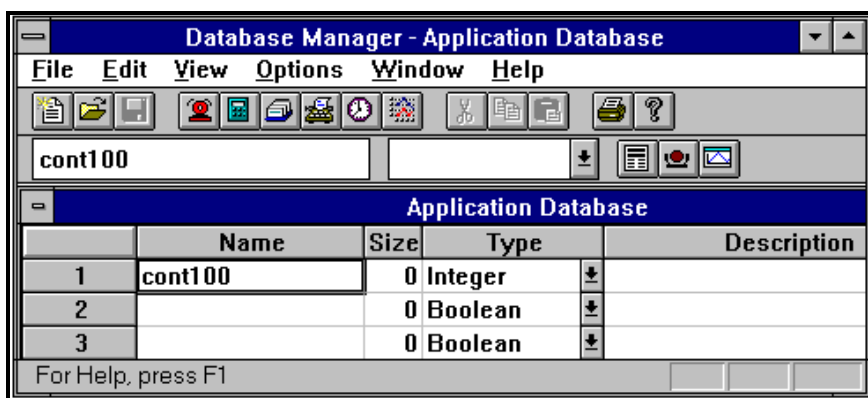
Clique a opção **Database Manager** do submenu.

Você abriu o DATABASE MANAGER, onde são manipulados os tags envolvidos no aplicativo. Na realidade, a planilha a ser usada para a definição do novo tag é a do **Application Database**. Esta planilha está sempre aberta dentro do DATABASE MANAGER, seja na forma maximizada ou na forma de ícone. Caso ela esteja minimizada, maximize ela, clicando sobre o ícone correspondente.



### Application Database Minimizado

Uma vez maximizada, a planilha apresenta-se como abaixo.



### Application Database Maximizado

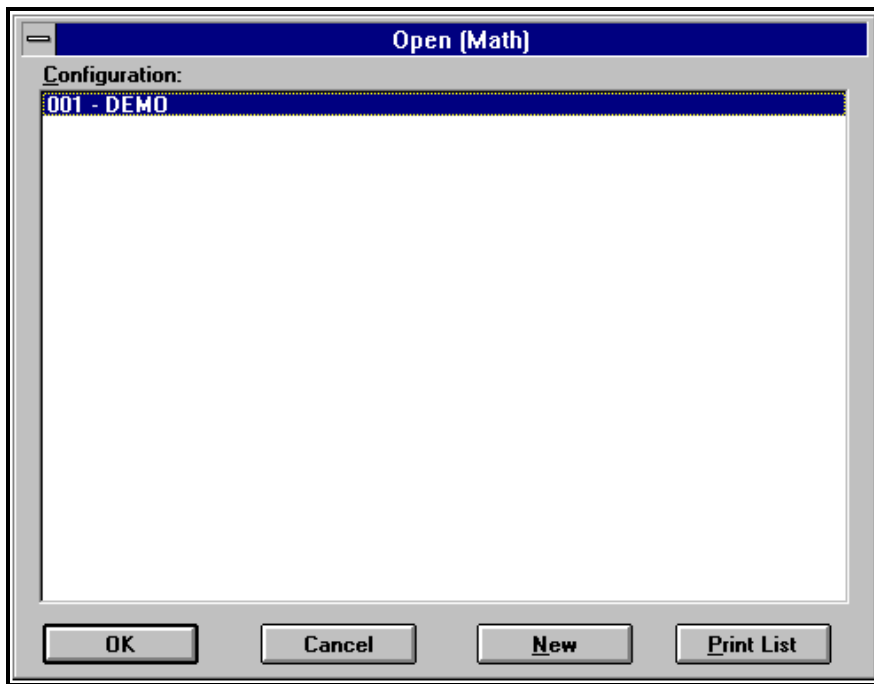
### Criando um Algoritmo Matemático



Uma vez declarado o tag contador **cont100** (no caso dele ainda não existir), é preciso implementar o algoritmo incrementador do nível do tanque.

Clique a tecla esquerda do mouse sobre o ícone de *Math* no *TaskBar*.

Estando dentro da janela **Open (Math)**, abra a planilha criada no **Tutorial 1** (001 - DEMO), clicando sobre o botão **OK**.



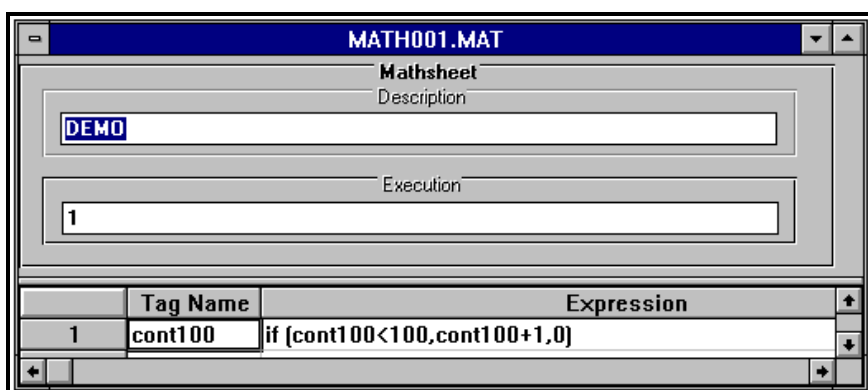
## Open Math

Dentro da planilha **001 - DEMO**:

Leve o mouse ao campo **Tag Name** num espaço disponível e clique com a tecla esquerda para confirmar a posição.

Aperte a tecla **Tab** para deslocar-se ao campo **Expression**.

Digite a expressão: **if(cont100<100,cont100+1,0)**, como deveria ser feito já no **Tutorial 1**.



## Planilha Matemática

A semântica desta expressão é bastante imediata: Enquanto o valor de **cont100** for menor do que **100**, o contador incrementa em **1** unidade, caso contrário, a variável é igualada a zero.

Antes de sair da planilha DATABASE MANAGER, salve a declaração do novo tag, para que ele possa ser executado. Para isso, simplesmente:



Clique sobre o ícone de gravação.

**NOTA:** Caso o ícone de gravação esteja **desabilitado**, clique com o mouse em uma célula 'em branco' da planilha, e ele será habilitado.

Para voltar à tela do APPLICATION BUILDER:

Tecla <Alt> e, mantendo-a pressionada, clique <Tab> sucessivamente até que apareça a indicação de APPLICATION BUILDER e, então, solte ambas as teclas.

## Mudando o nome de Tags

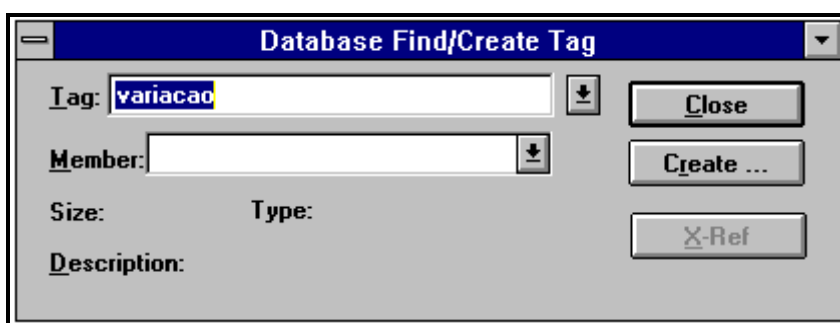
Se você achar que **cont100** não é um bom nome para a variável de variação de nível do tanque, e desejar trocar seu nome para **variação**, você deve seguir os passos abaixo:

Como o tag **cont100** é único em seu banco de dados (**Application Database**) você deve seguir os passos para substituí-lo:

Clique em **Tools** no menu principal.

Clique em DATABASE FIND/CREATE TAGS para criar o novo tag (**variacao**).

Digite o nome **variacao** no campo **Tag** e confirme clicando em **Create**.



## Database Find/Create Tag

Agora você tem em seu banco de dados (**Application Database**) esta variável.

**NOTA:** Caso você já tivesse **variacao** declarada no **Application Database**, bastaria seguir somente os passos abaixo, deixando de lado os passos

anteriores.

Duploclique no grupo de objetos.

Na janela **Object Properties**, clique no botão **Tag**.

Na janela que é aberta, no campo **New Tag**, digite **variacao** (novo nome para o Tag).



### Replace Tags

O resultado final destas operações é que todos os objetos deste grupo, que utilizavam a variável **cont100** agora passarão a utilizar a variável **variacao**.

Para a aplicação ser executada com a nova variável, deve-se mudar as rotinas da planilha matemática criada anteriormente. Onde existia a variável **cont100** deve ser mudado para a nova.

**NOTA:** Caso você selecionasse somente um objeto (o retângulo, por exemplo) e clicasse no botão **Tag** (para trocar a variável **cont100** para **variacao**), esta troca só aconteceria a variável deste objeto. Os demais objetos, pertencentes a sua aplicação, continuariam utilizando a variável **cont100**, gerando um erro na execução de sua aplicação.

### Executando DEMO.SCR

Agora já é possível verificar a execução do aplicativo, uma vez que o tag contador foi substituído nas dinâmicas correspondentes.

Desative a janela **Object Properties** clicando com o mouse fora da janela.

Para executar o aplicativo dentro do APPLICATION BUILDER, proceda da mesma maneira já explicada anteriormente:

Clique sobre a opção **Tools** do menu principal.

Clique sobre a opção **Run BgTask** do submenu, caso não esteja ativada, para ativar o BACKGROUND TASKS, que permite a execução das tarefas.

Clique novamente sobre a opção **Tools** do menu principal.

Clique, agora, sobre a opção **Execute** do submenu, que permite a visualização da variação do nível do tanque.

Para encerrar a execução, clique outra vez a opção **Tools** do menu principal e, em seguida, a opção **Execute** do submenu.

**PARABÉNS:** Este é a primeira aplicação que teve uma saída de display junto com a animação gráfica do tanque reservatório.

### Exercício 3

Este tutorial é, em realidade, um complemento do anterior, visto que será feita uma melhoria no algoritmo matemático da variação de nível para que possa ser implementado dois botões, um que controla a subida e outro a descida de líquido do reservatório, e um terceiro botão que controla a permissão para fechar e abrir a bomba que abastece o reservatório.

Em conjunto com isto, será criada uma inter-relação entre duas telas, uma conterà somente o tanque reservatório, e a outra a monitoração do tanque através dos botões de comando.

### Criando um Novo Tag na Planilha Matemática

O que se pretende, agora, é criar um algoritmo que permita o enchimento e esvaziamento do tanque. Para isso, é necessária a presença de uma variável auxiliar que permita verificar se o nível do líquido está subindo ou descendo (na realidade, é um *flag* de direção).

O que esta variável faz é o seguinte: uma vez que o nível atingiu o valor máximo (100) ela é zerada (resetada) e quando o nível chegar a zero, ela será setada.

Imagine esta nova variável (tag) como um relê que liga uma bomba para encher o reservatório. É fácil de imaginar que quando o nível do tanque atingiu seu valor máximo, a bomba precisa estar em *OFF*, ao passo que enquanto estiver abaixo deste valor, precisa estar em *ON*.

Para criar esta nova variável, vá até planilha matemática (*Math*) da seguinte maneira:

Abrindo a Planilha Matemática (*Math*)

Clique sobre a opção **Tools** do menu principal do APPLICATION BUILDER.

Clique sobre a opção **Math** do submenu.

Na janela de **Open (Math)**, clique na planilha **001-DEMO** (ou alguma outra descrição), criada no **Tutorial 1**.

Agora que você está dentro da planilha já editada, duploclique sobre a variável **variacao**.

Tecler duas vezes seguidas **<F5>**, para dar dois espaços antes desta expressão (A expressão deve ir dois espaços para frente).

Podemos supor que você erre algo e só perceba após gravar sua planilha ou mesmo crie uma ou mais linhas além das que você desejava e agora deseje apagar. Dessa maneira leve o mouse até o número dessa linha que você deseja apagar e clique com o botão esquerdo.

Perceba que a linha ficará selecionada, agora então vá com o mouse no ícone **Cut** representado por uma pequena tesoura fixada no menu principal da janela e perceba que clicando nela a linha desaparecerá. Este processo pode ser realizado em qualquer módulo do **UniSoft** com exceção do **Application Database**.

No primeiro espaço novo criado, insira no campo **Tag Name** o nome da nova variável como **relay**, por exemplo.

Como é de se esperar, é aberta a janela abaixo contendo um aviso de inexistência desta variável no banco de dados.

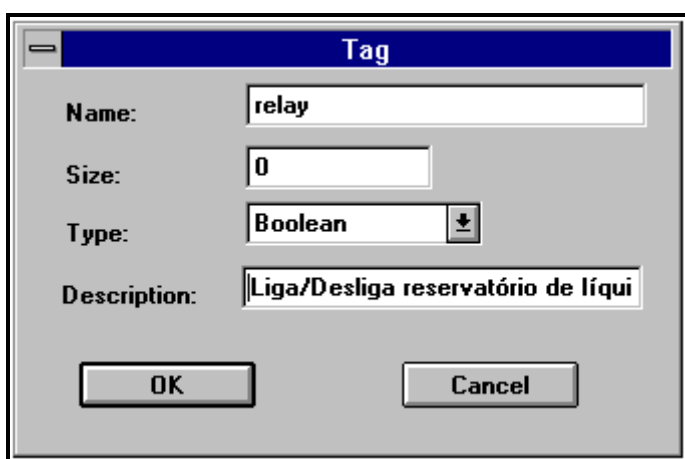


### Create Tag

Para isso:

Clique **YES** para confirmar a criação da nova variável.

Ao abrir a janela de declaração de variável, simplesmente mude o seu tipo de inteiro (**Integer**) para booleano (**Boolean**), clicando sobre o campo **Type** e, posteriormente, sobre a opção **OK**.



### Tipo do Tag

Digite a primeira expressão de **relay** no campo **Expression: if(variacao=100,0)**

Esta expressão determina o valor **OFF** de **relay**, quando o nível atingir o valor **100**. É quando o reservatório deverá ser esvaziado.

Vá ao espaço seguinte, duploclicando a tecla esquerda do mouse sobre o campo **Tag Name**.

Digite **relay** outra vez.

Digite a segunda expressão de **relay** no campo **Expression**: **if(variacao <=0,1)**

Esta expressão determina o valor **on** de **relay**, quando o nível atingir o valor **0**. A partir daí, o reservatório é reabastecido pela bomba.

Agora, é necessária uma outra modificação na expressão de **variacao**, para que haja a animação de esvaziamento/enchimento. Assim:

Duploclique sobre o campo **Tag Name** do tag **variacao** para confirmar a posição.


Altere a sua expressão para:

**if ( (variacao<100) and (relay), variacao+1,variacao-1)**. Desta forma, a planilha apresenta-se como demonstrada a seguir:

MATH001.MAT		
	Tag Name	Expression
1	relay	if [variacao=100,0]
2	relay	if [variacao<=0,1]
3	variacao	if [ (variacao<100) and [relay], variacao+1, variacao-1]
4		

### Planilha Matemática

É através desta primeira expressão que é feito o esvaziamento do tanque, toda vez que ele chega a seu nível máximo.

**IMPORTANTE:** Salve esta planilha, confirmando as mudanças com um clique fora dos espaços de expressões e um clique sobre o ícone de gravação  do *Taskbar*.

Voltando ao Application Builder

Aperte permanentemente **<Alt>** e tecla **<Tab>** sucessivamente até encontrar o nome deste ambiente e soltando as duas teclas.

Execute novamente o aplicativo dentro do APPLICATION BUILDER, conforme os passos anteriores de execução:

Executando o Aplicativo:

Clique sobre a opção **Tools** do menu principal.

Clique sobre a opção **Run BGTASK** do submenu para ativar o BACKGROUND TASKS, que permite a execução das tarefas.

Clique novamente sobre a opção **Tools** do menu principal.

Clique, agora, sobre a opção **Execute** do submenu, que permite a visualização da variação do nível do tanque.

Para encerrar a execução, clique outra vez a opção **Tools** do menu principal e, em seguida, a opção **Execute** do submenu.

Como você pode notar, seu reservatório enche até o nível máximo (100) e esvazia automaticamente.

**IMPORTANTE:** Para cada modificação que for feita na planilha matemática, depois de salva, será atualizada automaticamente nos módulos de runtime. Caso você experimente alguns problemas na configuração on-line, desative o BACKGROUND TASKS (com as teclas <Alt + F4>) e ative-o novamente, clicando duas vezes no menu principal **Tools** submenu **Run BGTASK**, pois você tem uma planilha modificada.

### **Criando um Botão ON/OFF para a Bomba do Tanque**

Uma maneira interessante de incrementar a nossa animação é possibilitar ao operador a escolha de esvaziar ou encher o tanque, caso o nível do líquido esteja dentro dos limites aceitáveis (entre **0** e **100**).

A primeira coisa que você deve fazer é a implementação de um botão, que simularia um comando para a abertura e fechamento de um relê que liga e desliga a contactora da bomba que enche o reservatório. Permitindo ou não que o tanque seja enchido ou esvaziado.

Primeiramente, é preciso criar um pano de fundo para o botão:

Clique sobre o ícone de retângulos (D3) do *Toolbar*.

Escolha um lugar conveniente da tela para posicionar o botão, clique a tecla esquerda do mouse e, mantendo-a pressionada, crie o formato de um botão (3cm de largura por 3cm de altura, por exemplo) e depois solte a tecla.

Duploclique sobre o novo objeto para ativar a janela **Object Properties** e escolha a cor do botão, clicando sobre a área retangular **Color** do campo **Fill**, clicando a cor desejada e confirmando com um **OK**.

Clique a opção **Fill**, para preencher o retângulo.

Uma vez definida a posição e o formato do botão, é preciso permitir que ele funcione. Uma forma bem simples de imaginar seu funcionamento é pensar que ele deve mudar o estado da variável **relay**. Caso esta variável esteja em 1 (bomba ligada) e o botão for apertado, **relay** deve ir para 0, o que provoca o desligamento da nossa bomba imaginária e não permitirá conseqüentemente que o reservatório seja abastecido ou esvaziado.

### **Dinâmica de Command**

A dinâmica que permite este tipo de comportamento é a dinâmica de command. Para implementá-la:

Duploclique sobre o objeto para ativar a janela de **Object Properties**, que agora possui também a propriedade de command.



Clique sobre o ícone de dinâmica de command (B6) do Toolbar.

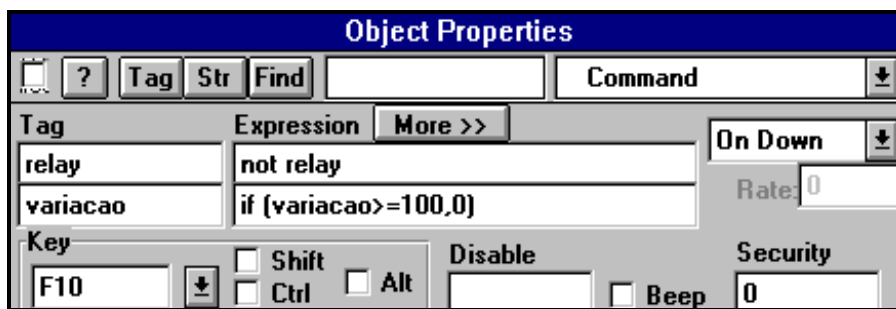
Deste modo, você tem como uma nova propriedade pertencente a este objeto, denominada **Command**.

Clique sobre o campo **Tag** e digite a variável **relay**.

Clique sobre o campo **Expression** e digite a expressão **not relay**.

Clique sobre o campo **Tag** e digite **variacao**.

Clique sobre o campo **Expression** e digite **if (variacao>=100,0)**.



### Dinâmica de Mouse

Como o tag **relay** é do tipo boleano, toda vez que o botão for clicado, a dinâmica **Command** faz com que seu valor seja invertido. Por exemplo, se **relay** estiver em *On* (valor 1), ao clicar o botão, ele passa para *Off* (valor 0).

Falta definir uma tecla auxiliar deste botão:

Clique sobre campo **Key** desta janela e escolha **F10**, por exemplo.

Agora, é possível mudar o valor de **relay**, tanto clicando sobre o novo botão ou teclando **<F10>** pelo teclado.

É interessante, agora, criar um texto **F10** e agrupá-lo ao botão:

Clique sobre o ícone de fontes (D5) para definir um tipo e tamanho coerente de fonte, lembrando que além deste texto, vai ser colocado o texto de estado sobre o botão.

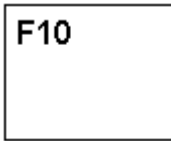
Escolha o tipo e tamanho desejado de fonte e, em seguida, confirme com um **OK**.





Clique sobre o ícone de texto do *Toolbar* e escreva **F10**.

Leve o texto até o canto superior esquerdo do botão, clicando com a tecla esquerda do mouse sobre ele e mantendo-a pressionada até localizá-lo conforme a figura.

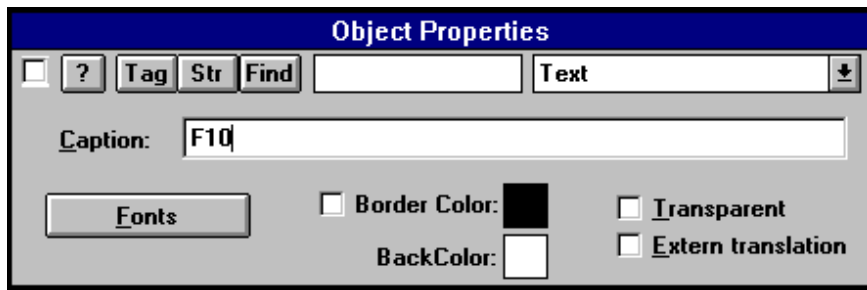


### Tecla de Atalho

Caso deseje, faça o fundo do texto ficar transparente em relação ao botão, para isso:

Duploclique sobre o novo objeto texto para ativar a sua janela **Object Properties**.

Clique sobre a opção **Transparent**, para que a cor de fundo do texto seja a mesma do botão.



### Object Properties do Relê

Seria interessante agrupar o texto com o objeto botão, para isso:

Clique sobre o botão (fora do texto).

Tecele <**Shift**> e mantenha pressionada até clicar sobre o texto.



Clique sobre o ícone de agrupamento de objetos (C11) no *Toolbar*.

Agora, tem-se um único objeto que agrupa um botão com um texto. É interessante implementar uma mudança de texto que permita ao operador saber para que estado o botão leva **relay**, caso seja clicado. Para isso, serão criados dois objetos texto (um indicando o estado *Off* e outro, indicando o estado *On*). Para isso:



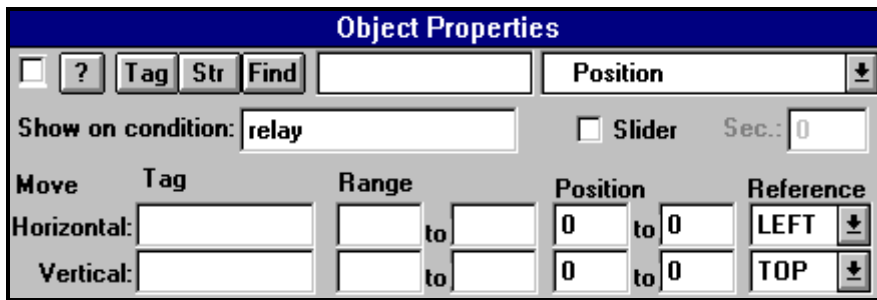
Clique sobre o ícone de texto (D4) e escreva *On*.

Duploclique este texto, para ativar a sua janela **Object Properties**.



Clique sobre o ícone de posição (C7) do *Toolbar* para adicionar esta propriedade ao objeto texto.

No campo **Show on Condition**, digite a condição **relay**.

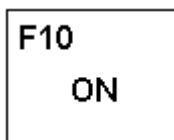


### Show on Condition

Isto quer dizer que toda vez que **relay** estiver em *On*, aparece este texto.

Agora, falta localizar o texto sobre o botão:

Carregue o texto até o botão, clicando sobre ele e mantendo a tecla esquerda do mouse pressionada até localizá-lo, de preferência, numa posição central inferior.



Botão Relê Ligado

Para fazer com que o fundo do texto seja transparente em relação à cor do botão:

Duploclique sobre o objeto texto para ativar a janela **Object Properties**.

Clique sobre a subjanela de propriedades no canto superior direito e, em seguida, clique a propriedade **Text**.

Clique no campo **Transparent**.

Da mesma forma, que com o texto **On**, proceda para o outro texto *Off*.



Clique sobre o ícone de texto (D4) e escreva *Off*.

Duploclique sobre este objeto texto para ativar a sua janela **Object Properties**.



Clique sobre o ícone de posição (C7) do *Toolbar*.

Clique no campo **Show on condition** e escreva **not relay**.

Isto quer dizer que toda vez que **relay** estiver em **off**, aparece este texto.

Agora, falta localizar o texto sobre o botão:

Carregue o texto até o botão, clicando sobre ele e mantendo a tecla esquerda do mouse pressionada até localizá-lo coincidentemente sobre o texto **On**.

Para fazer com que o fundo texto seja transparente em relação à cor do botão:

Duploclique sobre o objeto texto para ativar a janela **Object Properties**.

Clique sobre a subjanela de propriedades no canto superior direito e, em seguida, clique a propriedade **Text**.

Clique no campo **Transparent**.

O que acontece aqui é uma troca de textos sobre o botão, que depende do estado do tag **relay**.

Agora que você tem monitorado o estado do tanque (*On/Off*), você pode agrupar estes objetos em um único grupo de símbolos. Para isso:

Selecione o retângulo que você já agrupou com a tecla de atalho <**F10**>, pressionando a tecla <**Shift**> clique no texto *On* e *Off*.



Clique no ícone agrupamento de objetos (C11) no *Toolbar*.

Botões Enchendo / Esvaziando

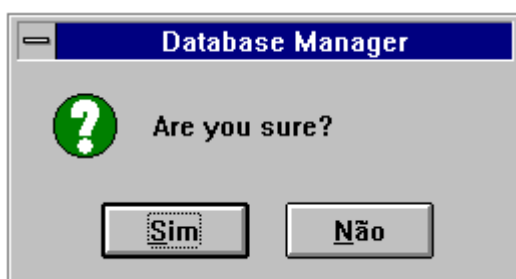
Agora que seu tanque reservatório conta com um meio externo para decidir quando pode encher ou esvaziar, falta implementar dois botões para estes comandos.

Para isso:

A primeira coisa que você tem a fazer é abrir a planilha matemática que você estava usando através do menu principal **Tools**, e no submenu **Math**.

Escolha então a planilha **001 - DEMO**, que você vinha utilizando.

Delete todas as três células que você havia criado. Para isso, clique no botão no canto direito de cada célula, selecionando a célula, clique no menu **Edit**, e no submenu seguinte **Cut**. Uma mensagem aparecerá perguntando se você quer realmente efetuar a remoção. Clique em **Yes**, e faça isso para todas as quatro células.



Mensagem para Deleção de Células

Agora digite na primeira célula a nova expressão :

**Relay      If (variação $\geq$ 100,0).**

E na segunda célula : **Relay      If (variação $\leq$ 0,1).**

Conforme a figura abaixo:

MATH001.MAT		
	Tag Name	
1	relay	if [variacao $\geq$ 100,0]
2	relay	if [variacao $\leq$ 0,1]
3		

### Novas Expressões na Planilha Matemática

A planilha matemática cuidará unicamente do limite da variação de líquido no tanque reservatório. Caso o nível do reservatório exceda 100 unidades o relê é desativado (*Off*), e caso seja inferior a 100 o relê está sempre ativado (*On*).

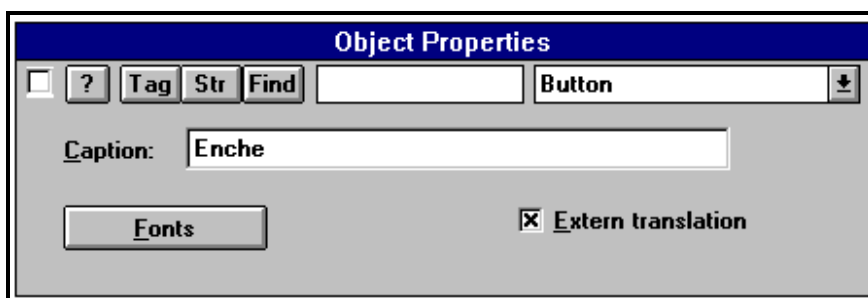
**NOTA:** Aquelas expressões, utilizadas no **Tutorial 2**, para controle do tag **variacao** não serão mais necessárias na planilha, pois vamos partir agora para a dinâmica de botões.



Clique no ícone de botões (A2) no *Toolbar*.

Clique em uma parte da tela e com o mouse pressionado dimensione seu botão no tamanho desejável (1 x 3 cm).

Duploclique no objeto recém criado. Aberta a janela **Object Properties** no campo **Caption** escreva **Enche**.

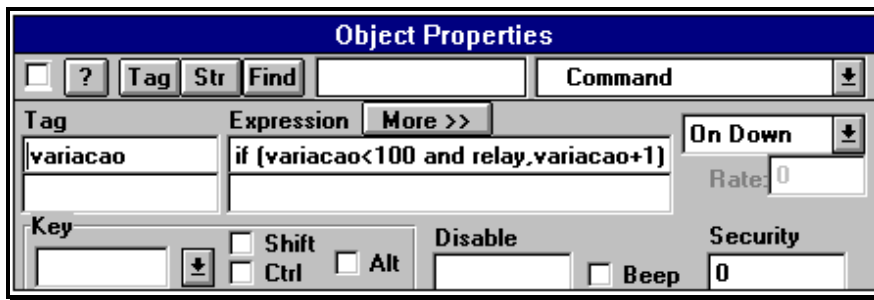


### Object Properties do Objeto Botão



Clique no ícone de dinâmica de mouse (B6).

Na janela que se abre, digite no campo **Tag** a variável **variacao**, e no campo **Expression**: **if (variacao<100 and relay, variacao+1)**.



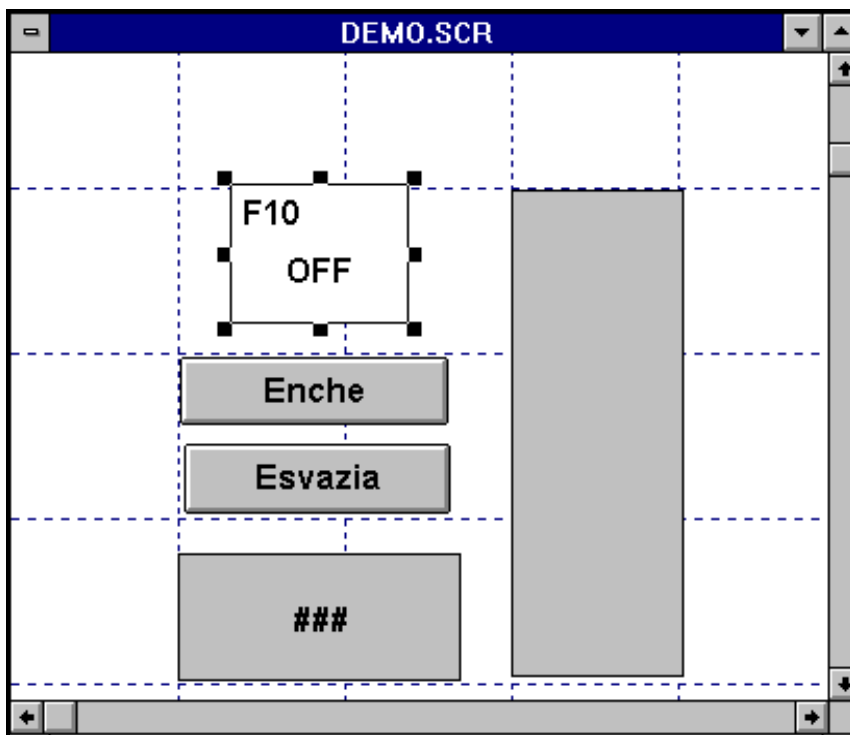
### Dinâmica de Mouse

Faça isso novamente para o outro botão **Esvazia**.

**OBS:** Não se esqueça de que no campo **Expression** o tag **variacao** deve ser decrementado. Então:

**Expression:** **if (variacao<100 and relay, variacao-1)**.

Agora você tem um tanque reservatório monitorado pelo usuário, conforme a figura abaixo:



### Tanque Reservatório

#### Executando o Aplicativo no runtime

Clique sobre a opção **Tools** do menu principal.

Clique sobre a opção **Run BGTASK** do submenu para ativar o BACKGROUND TASKS, que permite a execução das tarefas.

Clique novamente sobre a opção **Tools** do menu principal.

Clique, agora, sobre a opção **Run Viewer** do submenu, que permite a visualização da tela com dinâmica de mouse e botões.

Para encerrar a execução, clique no menu principal do VIEWER em **File** e no submenu em **Exit**. Você retornará ao APPLICATION BUILDER.

**IMPORTANTE:** É necessário que a aplicação seja executada em modo VIEWER, pois você utiliza dinâmica de botões e mouse, e ambos necessitam interação com o usuário.

Como você pode notar, o tanque reservatório só é abastecido em 1 unidade quando acionado o botão **Enche**, ou esvaziado em 1 unidade, quando acionado o botão **Esvazia**.

Quando o nível de líquido chega ao máximo de 100 unidades o relê é desligado, não permitindo que as dinâmicas dos botões atuem na bomba do reservatório. Para este quadro se reverter é necessário então clicar sobre o relê, e este esvazia totalmente o tanque, iniciando o nível do tanque novamente do nível 0 (zero).

**NOTA:** Caso você deseje que o nível do tanque encha ou esvazie em mais unidades, basta mudar nos botões as expressões **variacao+1** para **variacao+10**.

### Criando uma nova tela

Uma vez que criamos um aplicativo, onde é monitorado o nível de um tanque reservatório através do frontal de um controlador, seria interessante um monitoramento do próprio tanque. Para tornar a situação o mais próxima do real, o esquema do tanque poderia estar numa outra tela, que seria acessada através de um botão nesta tela atual.

Primeiramente, é preciso criar uma nova tela que conterá o objeto daquele nosso tanque criado anteriormente. Para isso:

Salve a tela DEMO.SCR, clicando na opção **File** do menu principal e, em seguida, sobre a opção **Save** do submenu.

Clique no retângulo que representa o reservatório. Pressione junta as teclas **<Ctrl + C>**. Agora você capturou na memória o reservatório, tecle **<Delete>**.

Crie uma nova tela, clicando outra vez sobre a opção **File** do menu principal e em seguida, sobre a opção **New** do submenu.

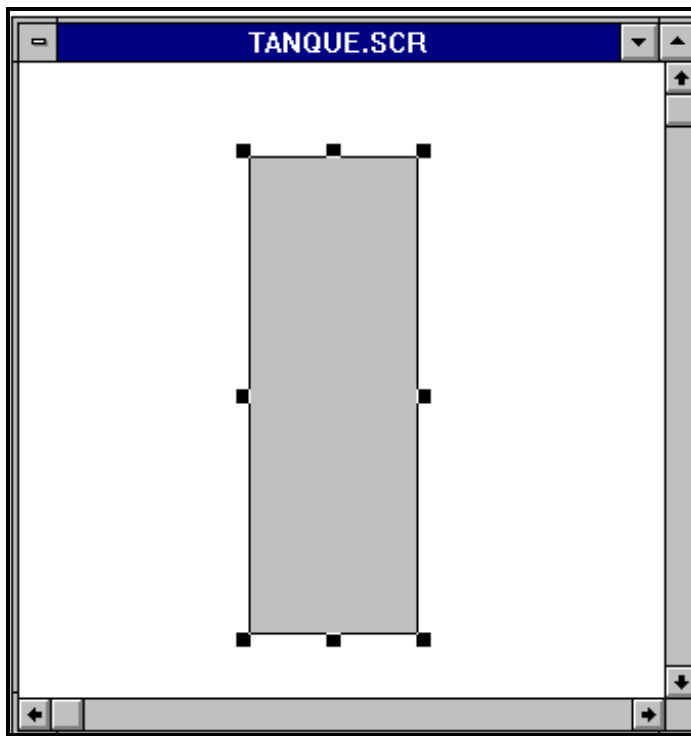
Ao abrir a janela de atributos da tela (**Screen Attributes**), vá ao campo **Title** e defina um título para esta nova tela (TANQUE.SCR, por exemplo) e confirme com um **OK**.

Uma vez criada a tela, é possível inserir o tanque da tela TANQUE.SCR:

Pressione **<Ctrl + V>**, e o objeto será inserido nesta nova tela.

Posicione o objeto no lugar desejado.

Salve a tela no menu principal **File**, no submenu **Save as...**, e digite o nome TANQUE.SCR.



## Nova Tela de Animação

### Criando uma dinâmica de botão

Uma vez criadas as duas telas do nosso aplicativo (DEMO.SCR e TANQUE.SCR), é importante um mecanismo de interação entre as duas, de modo que a tela que contém o esquema do reservatório possa ser acessada pela tela que contém o controlador e vice-versa.

Para começar, abra novamente a tela DEMO.SCR, procedendo da seguinte forma:

Clique a opção **File** do menu principal.

Clique a opção **Open** do submenu.

Duploclique sobre o nome do arquivo DEMO.SCR na lista de arquivos de tela.

Agora, dentro da primeira tela de exemplo, implementaremos um botão, cuja finalidade será de chamar a segunda tela em modo *runtime*.

Como primeiro passo, você deve criar um objeto que é o desenho do botão em si. Para isso:

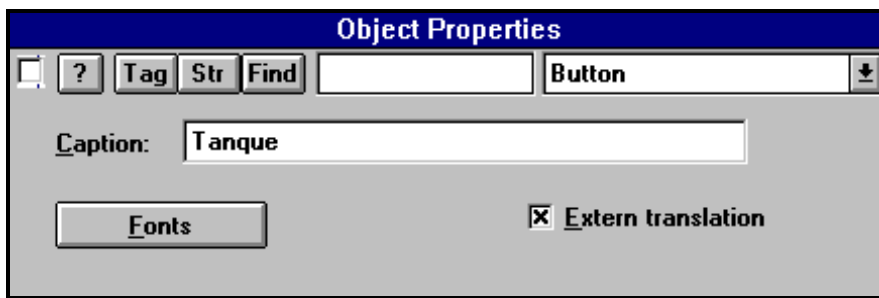


Clique sobre o ícone de botões (A2) do *Toolbar*.

Leve o mouse até um canto desejado da tela (o canto inferior direito, por exemplo) e pressione a tecla esquerda, mantendo-a pressionada para definir um tamanho coerente de botão (2cm de altura por 2.5cm de comprimento, por exemplo). Solte a tecla assim que estiver definido o botão.

Duploclique o objeto para chamar a janela **Object Properties**.

Dentro do campo **Caption**, digite um texto que sugira a troca de tela (**Tanque**, por exemplo).



### Título do botão para troca de tela

É interessante, neste caso, declarar uma tecla auxiliar que execute a mesma função (<F1>, por exemplo). Para isso:

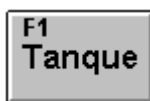


Clique sobre o ícone de tipo de fonte (D5) e defina um tipo desejado. Lembre-se que a fonte não pode ser muito grande. Depois, confirme com um **OK**.



Clique sobre o ícone de textos (D4) do Toolbar.

Clique sobre o objeto botão e escreva **F1** no canto superior esquerdo.



Botão para acessar a segunda tela

Para tornar o fundo do texto transparente em relação ao botão, simplesmente:

Duploclique o objeto-texto para ativar a sua janela **Object Properties**.

Clique na opção **Transparent**.

É interessante juntar estes dois textos num único objeto:

Clique sobre o objeto texto **F1**.

Aperte a tecla <Shift> e mantenha-a apertada para clicar sobre o objeto botão.





Una os objetos, clicando o ícone de agrupamento de objetos (C11) no Toolbar.

Falta definir a dinâmica de botão para a mudança de tela.

O **UniSoft** possui uma função específica para troca de tela: é a função **Open**. É esta função que vai permitir que a tela 1 acesse a tela 2. Para isso:

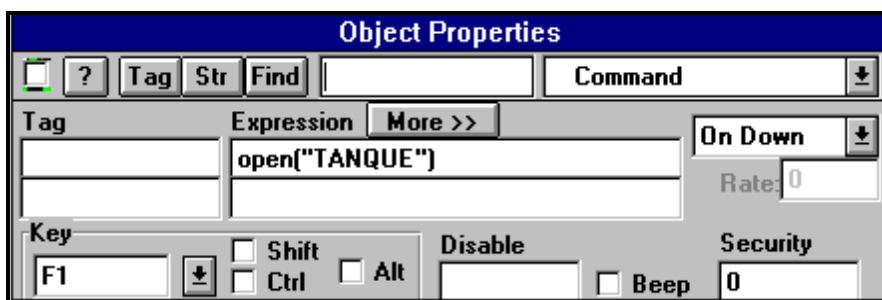
Duploclique sobre o objeto botão de troca de tela para ativar a janela **Object Properties**.

Clique sobre o ícone de dinâmica de mouse/botão (B6) do *Toolbar*, para ativar esta propriedade.

Deixe o campo **Tag** em branco.

Clique sobre o campo **Expression** e escreva **open("TANQUE.SCR")**, que é o nome do arquivo da **segunda tela** que contém o reservatório.

No campo **Key** escolha **F1** para ser sua tecla de atalho.



### Comando para troca de telas

Com isso, é possível a primeira tela (DEMO.SCR) acessar a segunda tela (TANQUE.SCR) em modo *runtime*. Porém, é necessário que haja a possibilidade da segunda tela voltar à primeira. Para isso, é preciso implementar-se um botão na segunda tela, da mesma forma que foi implementado na primeira.

Salve DEMO.SCR:

Clique sobre a opção **File** do menu principal.

Clique sobre a opção **Save** do submenu.

Agora carregue TANQUE.SCR outra vez:

Clique sobre a opção **File** do menu principal.

Clique sobre a opção **Open** do submenu.

Digite o nome do arquivo TANQUE.SCR ou duploclique sobre o seu nome na lista de arquivos disponíveis.

Agora está carregada a nossa segunda tela. Para implementar o botão, proceda da mesma maneira anterior:



Clique sobre o ícone de botões (A2).

Leve o mouse até um canto desejado da tela (o canto inferior esquerdo, por exemplo) e pressione a tecla esquerda, mantendo-a pressionada para definir um tamanho coerente de botão (2cm de altura por 2.5cm de comprimento, por exemplo). Solte a tecla assim que estiver definido o botão.

Duploclique o objeto para ativar a janela **Object Properties**.

Dentro do campo **Caption**, digite um texto que sugira a troca de tela (**Volta**, por exemplo).

É bom declarar uma tecla auxiliar que execute a mesma função (**F1**, por exemplo, como na primeira tela). Para isso:

Clique sobre o ícone de tipo de fonte (D5) e defina um tipo desejado. Lembre-se que a fonte não pode ser muito grande. Depois, confirme com um **OK**.



Clique sobre o ícone de textos (D4) do *Toolbar*.

Clique sobre o objeto botão e escreva **F1** no canto superior esquerdo.

Para tornar o fundo do texto transparente em relação ao botão, simplesmente:

Duploclique o objeto texto para ativar a sua janela **Object Properties**.

Clique na opção **Transparent**.

É interessante juntar estes dois textos num único objeto:

Clique sobre o objeto **F1**.

Aperte a tecla **<Shift>**, mantendo-a apertada e clique sobre o objeto botão.



Clique sobre o ícone de agrupamento de objetos (C11) no *Toolbar*.



Botão para voltar à tela de monitoração

Falta definir a dinâmica de botão para a mudança de tela.

Da mesma maneira que é feito na primeira tela, aqui também é utilizada a função *open* para a troca de telas. É esta função que vai permitir que se acesse a tela do **DEMO** a partir da tela do **RESERVATORIO**:

Duploclique sobre o objeto botão para ativar a janela **Object Properties**.

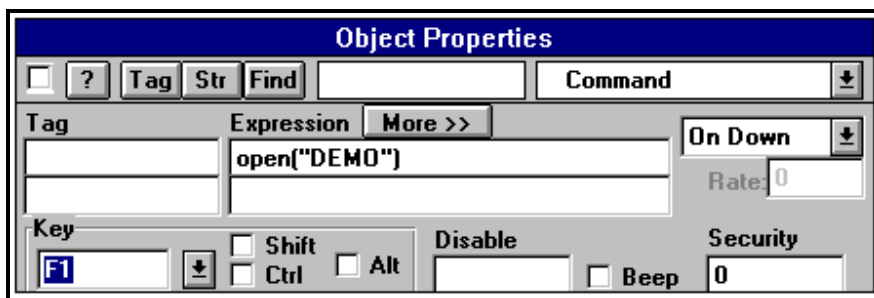


Clique sobre o ícone de dinâmica de mouse/botão (B6) do *Toolbar*, que ativa a propriedade **Command**.

Deixe o campo **Tag** em branco.

Clique sobre o campo **Expression** e escreva **open ("DEMO")**, que é o nome do arquivo da tela 1.

No campo **Key** escolha **F1** para ser sua tecla de atalho.



### Comando para trocar de tela

Com isso, é possível a segunda tela (TANQUE.SCR) acessar a primeira tela (DEMO.SCR) em modo *Runtime*.

Salve TANQUE.SCR:

Clique sobre a opção **File** do menu principal.

Clique sobre a opção **Save** do submenu.

Executando sua aplicação

Uma vez definidos os botões para as duas telas, é possível executar o aplicativo em modo *runtime*, mesmo porque só é possível executar um aplicativo com dinâmica de botão em modo *runtime*:

Volte ao **Program Manager** apertando **<Alt>**, mantendo-a pressionada e teclando **<Tab>** sucessivamente, até que apareça sua indicação.

Dentro do VIEWER, feche a atual tela, clicando sobre a opção **File** do menu principal e, em seguida, sobre a opção **Close** do submenu.

Abra a versão atualizada da tela demo, clicando sobre a opção **File** do menu principal e, em seguida, sobre a opção **Open** do submenu.

Dentro da janela **Open**, duploclique, dentro do campo **File Name**, sobre o nome **demo.scr**, para carregar esta tela.

Ou, dentro do APPLICATION BUILDER, clique em **Tools** do menu principal.

**Run BGTask** no submenu.

**Run Viewer** neste mesmo submenu.

Desta maneira, você está acessando a execução da primeira tela (**demo.scr**). Para acessar a segunda tela, clique o botão recém-criado, que permite seu acesso. Nesta segunda tela, é possível verificar a animação do tanque em si. Para voltar ao frontal de controle, simplesmente clique sobre o botão recém criado que permite o acesso à primeira tela.

#### Exercício 4

Neste tutorial, você aprenderá a implementar uma dinâmica de alarme na sua aplicação. Esta dinâmica é de suma importância para casos em que esteja-se supervisionando algum processo que possa acarretar algum perigo em determinadas condições ou casos em que algum processo possa sofrer algum prejuízo ou mal funcionamento.

#### Configurando um alarme

Uma situação interessante para o nosso exemplo é gerar um alarme caso uma das variáveis do tanque atinja um valor fora do estipulado.

Para começar, implementemos um alarme de nível para o tanque

De uma certa forma, a concepção de um alarme é bem direta, bastando o operador definir com qual tag do processo deverá se relacionar. No nosso caso em particular, o tag a ser monitorado pelo alarme será o de nível. Caso você deseje criar alarmes para as demais variáveis do processo, fica como um exercício, visto que o procedimento é muito semelhante.

Para começar, defina uma saída de alarme na tela DEMO.SCR:

Clique sobre o ícone de alarmes (A5) do *Toolbar*.

Dimensione a saída para um alarme, clicando a tecla esquerda do mouse e mantendo-a pressionada até que apareçam todas as saídas do alarme, como: data (**DD/MM/YY**), hora (**HH/MM/SS**), nome do tag (**TTT...**) e mensagem de alarme (**MMM...**).

**DD/MM/YYYY HH:MM:SS \* TTTTTTTTTTTT MMMM**

#### Saídas de Alarme

Duploclique sobre o novo objeto para ativar a janela **Object Properties**.

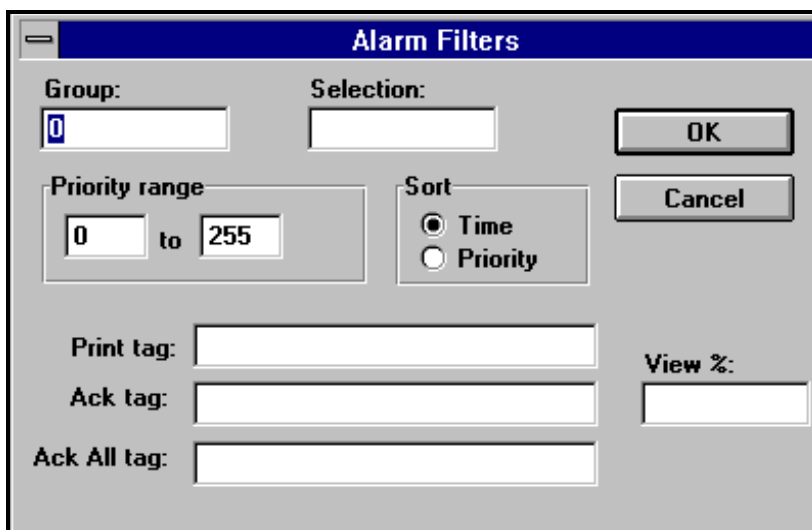
No campo **TAG** digite o valor **14**, e no campo **MESSAGE**, digite o valor **30**, para determinar o número de caracteres para a apresentação do nome do tag e da mensagem de alarme, respectivamente.

Caso deseje mudar a cor de fundo da saída de alarme, clique no campo **WIN** e, em seguida, escolha a cor desejada, clicando **OK** para confirmar.



### Object Properties do Alarme

Dentro dessa janela clique a opção **Selection**, será aberta a janela **Alarm Filters**, dentro dessa nova janela vá ao campo **Group**, este deve permanecer com o valor zero **0**, que significa que estão habilitados todos os grupos, ou planilhas de alarme (tratados adiante). Como, na realidade, só vamos esquematizar uma saída de alarme, teremos apenas um alarme.



### Filtro de Alarme

Feche a janela, clicando **Ok**.

Relacionando um Tag a um alarme

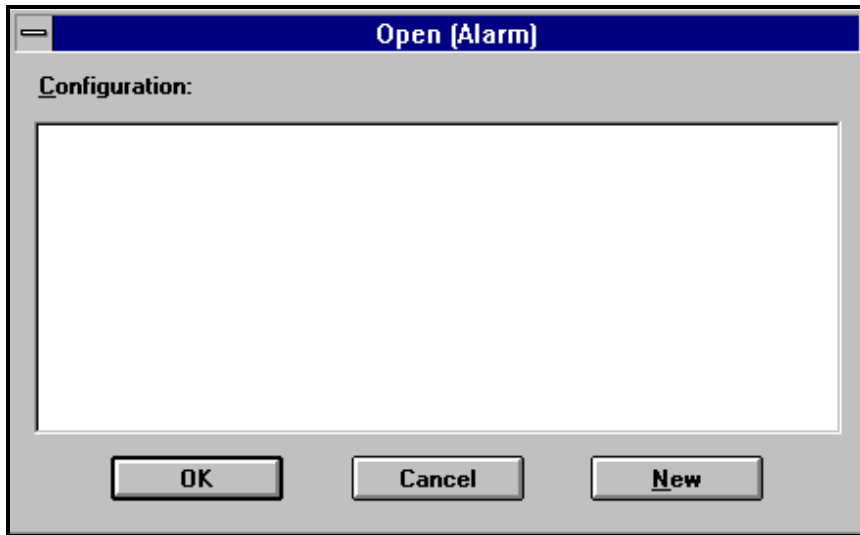
Agora, para identificar um tag com o qual se relaciona a mensagem de alarme, vá ao gerenciador de banco de dados:

Ative o APPLICATION DATABASE, clicando sobre ele, caso esteja maximizado ou duploclicando seu ícone, caso esteja minimizado.

Primeiramente, abra um planilha de alarme:



Clique sobre o ícone de alarmes do *Taskbar* e, em seguida, clique sobre o botão **New**, para a criação de uma nova planilha de alarme.

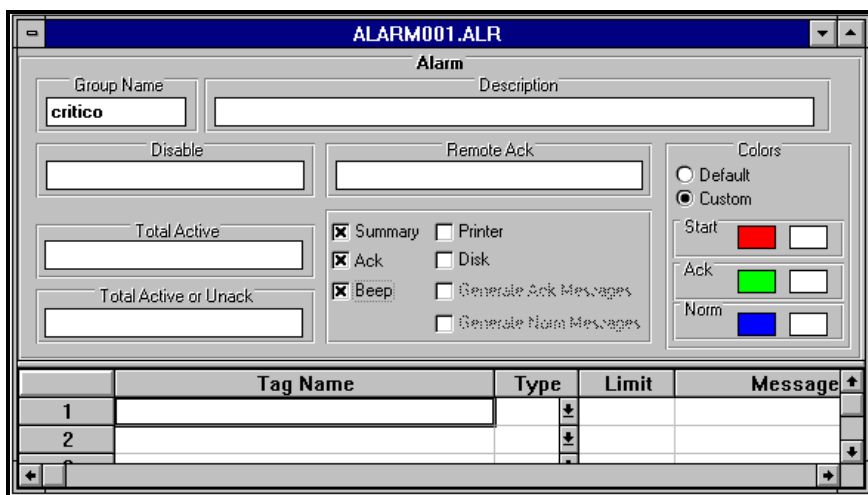


### Janela para abrir uma planilha de alarme

Dentro da planilha, defina um nome de grupo no campo **Group Name** (nome **crítico**, por exemplo).

Clique sobre os campos **Summary**, **Ack** e **Beep**, para permitir emissão de alarme na tela, reconhecimento e sinal sonoro, respectivamente.

Salve esta planilha clicando em **File +Save** e depois feche-a.



### Planilha de Alarme

Duploclique sobre um novo espaço no campo **Tag Name** e digite o nome **ack**, que é uma nova variável usada para reconhecimento de alarme (discutido adiante).

Application Database				
	Name	Size	Type	Description
1		0	Boolean	
2	variacao	0	Integer	
3	tank	2	Class: tank	
4	relay	0	Boolean	Liga/Desliga reserva
5	ack	0	Boolean	
6	i	0	Integer	Indexador da variave

## Variável de Reconhecimento do Alarme

Clique sobre o tag **VARIACAO**, de modo que ele fique selecionado na janela **Tag Name**.



Abra o configurador de alarme, clicando sobre o ícone correspondente ao lado do campo **Members**.

No campo **Vetor Index**, indique o valor do elemento do vetor **tank**, no caso, elemento **1**.

No campo **Remote Ack tag**, digite o nome do tag de reconhecimento **ack**, criado anteriormente.

Clique sobre o campo **Alarms Enabled**, para habilitar alarmes.

Leve o mouse, agora, para o campo de definição de alarmes e localize o nosso alarme, por exemplo na prioridade **Hi**, supondo que o alarme não é tão crítico.

No campo **Limit**, estabeleça um limite **80**, por exemplo, ou seja, cada vez que o nível do tanque atingir o valor **80**, é gerado um alarme.

No campo **Group**, digite o número **0**, criado anteriormente.

No campo **Priority**, deixe como prioridade **0**, já que é o único alarme.

Alarms - Integer/Real Type				
<input checked="" type="checkbox"/> Alarms Enabled	Array Index: 1	<input type="button" value="Close"/>		
Remote Ack tag: ack	<input type="checkbox"/> Translation Enabled			
Dead Band Value: 0.				
<input type="checkbox"/> HiHi	Limit	Message	Group	Priority Selection
<input checked="" type="checkbox"/> Hi	80	Nível de Líquido em Excess	0	0
<input type="checkbox"/> Lo				
<input type="checkbox"/> LoLo				
<input type="checkbox"/> Rate				
<input type="checkbox"/> Deviation+				
<input type="checkbox"/> Deviation-				
Deviation Setpoint:		Deviation Dead Band:	0.	

## Configurando o Alarme

Confirme a implementação no botão **Close**.

## Criando um Botão de Reconhecimento de Alarme

Uma vez criada a nossa condição de alarme, é interessante ter-se uma forma de reconhecer o alarme gerado. Uma maneira de se fazer isso é implementando um botão de reconhecimento.

Para ilustrar o nosso exemplo, suponha que este botão encontra-se próximo ao tanque monitorado, ou seja, o reconhecimento é feito por algum operador no campo. Para isso:

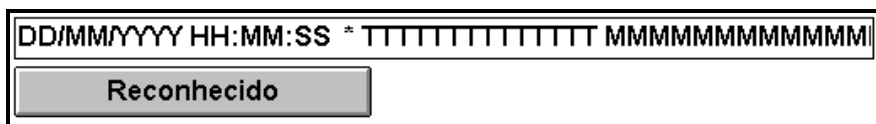


Dentro do TANQUE.SCR, clique sobre o ícone de botão (A2) do *Toolbar*.

Leve o mouse até um lugar desejado para localizar o botão e, então, clique a tecla esquerda do mouse, mantendo-a pressionada para definir a forma, para, em seguida, soltá-la.

Duploclique sobre o objeto-botão para ativar a sua janela **Object Properties**.

No campo **Caption**, defina um texto para este botão (**Reconhecido**, por exemplo). Conforme a figura abaixo:

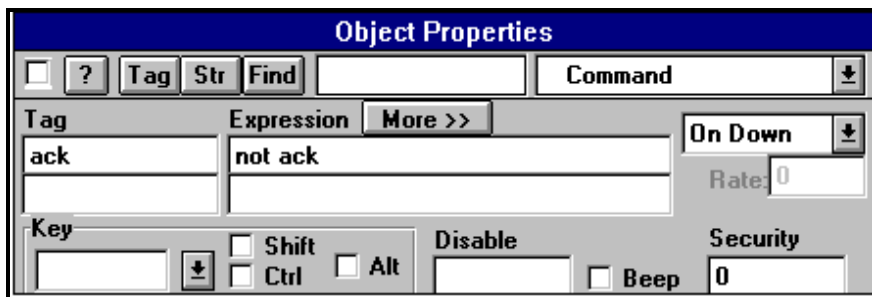


## Botão de Reconhecimento do Alarme



Ainda com a janela **Object Properties** ativada, clique sobre o ícone de dinâmica de mouse/teclado (B6) do *Toolbar*.

No campo **Tag**, digite o nome do tag de reconhecimento **ack** e no campo **Expression**, digite uma expressão do tipo **not ack**, ou seja, o valor de **ack** é invertido toda vez que o botão de reconhecimento for clicado.



## Object Properties do Botão de Reconhecimento

Caso deseje, defina uma tecla auxiliar para este botão, a qual é definida no campo **Key**. Em seguida, defina o texto correspondente a esta tecla para ser colocado sobre o botão (como foi feito para o botão *On/Off*, no DEMO.SCR).

Salve a tela, clicando a opção **File** do menu principal e, em seguida, a opção **Save** do submenu.



**IMPORTANTE:** Salvar a tela é essencial, toda vez que é concluída uma mudança.

Executando Aplicação dentro do Application Builder

Dentro do APPLICATION BUILDER clique em **Tools** no menu principal, e em **Run BGTask** no submenu.

Clique em **Tools** no menu principal, e em **Run Viewer**, para executar sua aplicação.

Executando a Aplicação fora do Application Builder

Volte ao **Program Manager** apertando <Alt>, mantendo-a pressionada e teclando <Tab> sucessivamente, até que apareça sua indicação.

Duploclique no módulo **Run BGTask** no grupo de programas **UniSoft**.

**IMPORTANTE:** Não se esqueça de rodar **Run BGTask** antes do módulo VIEWER. Pois, é esse módulo que ativa todas as planilhas da aplicação.

Duploclique no módulo **Run Viewer**, para executar sua aplicação.

**AVISO:** Para executar um aplicativo que utiliza as dinâmicas de botões e mouse, utilize o modo *runtime* dentro do APPLICATION BUILDER ou diretamente o VIEWER.

Para que a execução esteja correta, é preciso que ocorra um sinal de alarme, via *beep* e mensagem na tela, toda vez que o nível do tanque exceder o valor **80**.

Quando isso ocorrer você pode clicar no botão de reconhecimento do alarme e este se mostrará com uma cor diferente (para melhor visualização que a mensagem de alarme foi reconhecida pelo operador). Caso o nível do líquido seja esvaziado a mensagem mostrar-se-á com uma terceira cor (para melhor visualização que o nível não é mais crítico). Perceba que a mensagem, apesar de mudar de cor, somente desaparece quando o nível do tanque reservatório zerar novamente.

Você pode alterar estas cores dos níveis de reconhecimento do alarme na planilha de alarme.

Como exercício, tente implementar dinâmicas de alarme também para temperatura e pressão do tanque. Estipule um valor máximo (ou mínimo) para o acionamento do alarme para os dois casos.

## Exercício 5

Uma maneira bastante didática e elegante de se visualizar o andamento de um processo é armazenar o seu comportamento é através de gráficos. Estes gráficos podem ser visualizados **on-line** ou através de leitura de arquivos (**history**).

Assim, será apresentado neste tutorial a criação de um gráfico de tendência para a sua aplicação do tanque reservatório.

Criando a saída gráfica

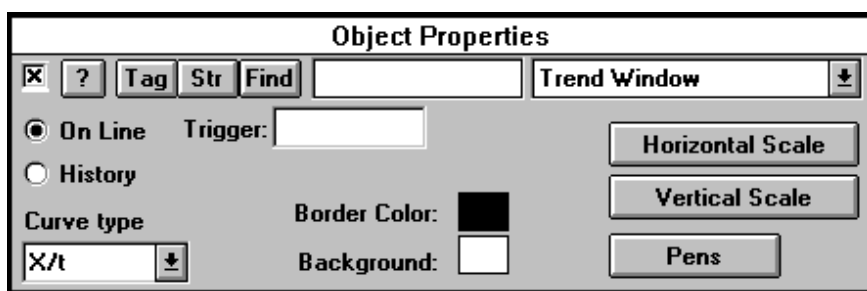
É possível dispor uma saída gráfica para a leitura das variáveis do tanque reservatório. Para isso:

Dentro da tela demo.scr, clique sobre o ícone de trend (A6) do *Toolbar*



Posicione o mouse num lugar conveniente da tela (supondo ser um dos quatro cantos do gráfico), clique a sua tecla esquerda e, mantendo-a pressionada, crie um objeto em torno de 15 cm de largura por 5 cm de altura.

Duploclique sobre este novo objeto para ativar a sua janela **Object Properties**.



### Object Properties do Gráfico

Clique sobre a opção **On Line**.

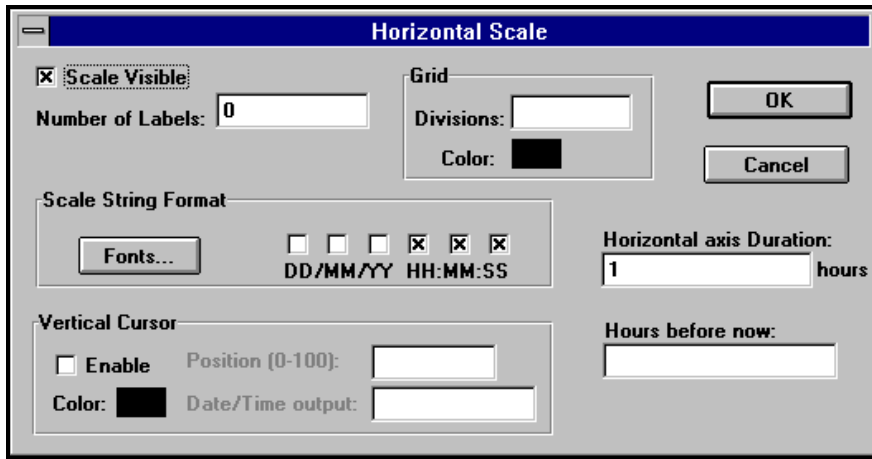
No campo **Curve Type**, selecione a opção **X/t** que permite o monitoramento das saídas em função do tempo.

Caso deseje, é possível modificar as cores da borda e de fundo da saída de trend via campos **Border Color** e **Background**, respectivamente.

Dentro do campo **Trigger**, digite o tag pré definido **second**, que permite um gatilho a cada segundo para a leitura dos dados.

Clique no botão **Horizontal Scale**.

Dentro da janela de definição da escala horizontal, apresentada na página seguinte.



### Horizontal Scale do Gráfico

Dentro do campo **Number of Labels**, defina **3** labels, que representam a saída temporal das variáveis a serem traçadas.

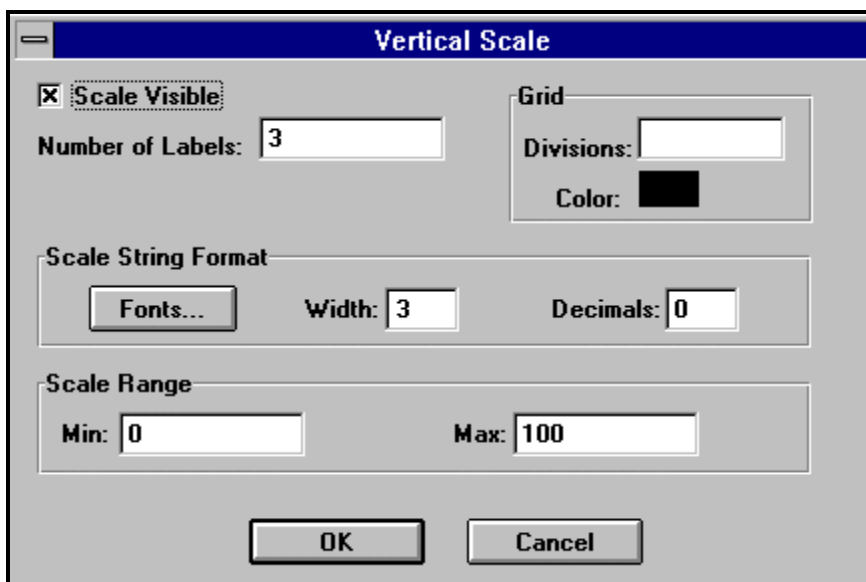
No campo **Scale String Format**, deixe as opções **HH**, **MM** e **SS** selecionadas, para a apresentação no formato de horas, minutos e segundos, respectivamente. Caso deseje, mude o formato da fonte via botão **Fonts...**

Dentro do campo **Horizontal Axis Duration**, digite o valor **0.025**, que define uma duração de **1.5 minuto** para o gráfico ( $1/40$  de 1 hora).

Confirme todas estas atribuições com um clique em **OK**.

De volta à janela **Object Properties**, clique sobre o botão **Vertical Scale**.

Dentro da janela de definição da escala vertical:



### Vertical Scale do Gráfico

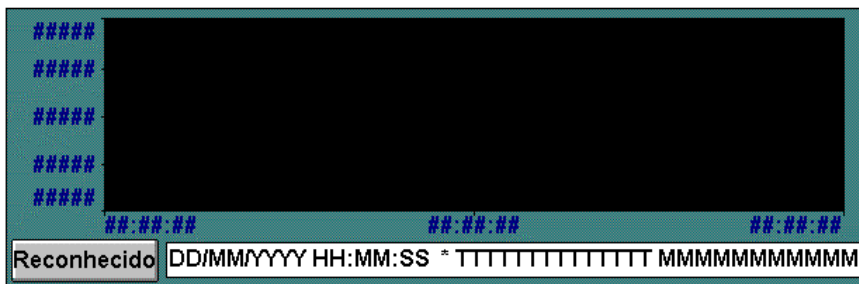
No campo **Number of Labels**, digite **3**, por exemplo, o que permite a impressão de três valores na escala vertical (No nosso caso: **0**, **50** e **100**).

No campo **Scale String Format**, digite o valor **0** no campo **Decimals** e a largura de caracteres **3** em **Width**. Caso deseje, altere a fonte via botão **Fonts...**

Dentro do campo **Scale Range**, digite **0** para o campo **Min** e **100** para o campo **Max**.

Confirme as atribuições clicando em **Ok**.

Seu Gráfico deve ter ficado semelhante ao da figura abaixo:

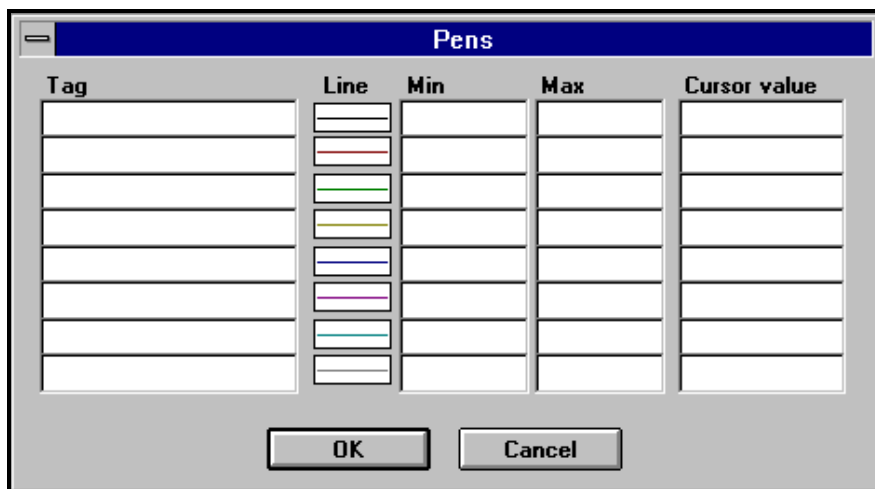


### Gráfico da Aplicação

Agora, é interessante definir as variáveis que desejamos monitorar na saída gráfica:

Dentro da janela **Object Properties** do gráfico, clique sobre o botão **Pens**.

Na janela **Pens**:



### Pens de Plotagem

Digite o nome **VARIACAO..**

Por *default*, já é definida uma cor para a saída gráfica desta variável no campo **Line**, mas é interessante definir uma cor coincidente com aquela do frontal do controlador. Para isso:

Clique sobre o retângulo **Line** correspondente e dentro do campo **Color Selection** escolha a cor coincidente com aquela de animação de nível do frontal do controlador e confirme com um **Ok..**

No campo **Min**, digite o valor **0** e no campo **Max**, o valor **100**.

Não se importe com os demais zeros depois da vírgula aqui só representamos a quantidade de casas decimais disponíveis para seu uso.

Salvando a Nova Aplicação

Confirme as atribuições com um **Ok**.

Antes de rodar seu aplicativo, vá até a sua planilha matemática clicando na opção **Tools** e na opção **Math** do submenu. Dentro da janela **Open(Math)**, abra a sua planilha DEMO.

No campo **Execution** digite **1**. Depois altere a matemática de sua planilha conforme a figura abaixo.

MATH001.MAT		
	Tag Name	Expression
1	relay	if {tank[1].level=100,0}
2	relay	if {tank[1].level<=0,1}
3	tank[1].level	if {tank[1].level<100 and {relay}, tank[1].level+1, tank[1].level-1}
4	tank[1].pressure	tank[1].level/3+6
5	tank[1].temperature	tank[1].level/2+4
6		

Nova matemática

Salve a sua alteração clicando no menu **File** e **Save** no submenu. Feche esta planilha e volte ao APPLICATION BUILDER.

Salve a aplicação clicando em **File** e no submenu **Save**.

### Executando sua Aplicação

Volte ao **Program Manager** apertando **<Alt>**, mantendo-a pressionada e teclando **<Tab>** sucessivamente, até que apareça sua indicação.

Duploclique no ícone de BACKGROUND TASKS para ativá-lo.

Duploclique no ícone do VIEWER para ativá-lo. Feche a atual tela, clicando sobre a opção **File** do menu principal e, em seguida, sobre a opção **Close** do submenu.

Abra a versão atualizada da tela **demo**, clicando sobre a opção **File** do menu principal e, em seguida, sobre a opção **Open** do submenu.

Dentro da janela **Open**, duploclique, dentro do campo **File Name**, sobre o nome DEMO.SCR, para carregar esta tela.

A execução é, de uma certa forma, bem direta, permitindo um monitoramento **On Line** das três variáveis que são alteradas no frontal do controlador, criado nos tutoriais anteriores.

Parabéns você chegou ao final do sétimo tutorial, com toda certeza você agora se encontra familiarizado com o produto **UniSoft** e pode assim partir para estudar individualmente os detalhes de cada módulo do **UniSoft**, estando apto a desenvolver suas aplicações de acordo com sua criatividade e necessidades.

Conserve, se possível, as aplicações dos tutoriais para que você possa, pelo menos na primeira fase, ter exemplos para espelhar-se ao criar novas aplicações.

ANEXO: Capítulo 6 do Manual UNISOFT