UNIVERSIDADE FEDERAL DE SANTA CATARINA

CENTRO TECNOLÓGICO

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

DESENVOLVIMENTO DE UM SISTEMA INDUSTRIAL DE COLETA DE DADOS EM AMBIENTE WEB COM INTEGRAÇÃO A E.R.P.

Trabalho de Conclusão de Curso apresentado para a obtenção do título de Bacharel em Ciências da Computação, pela Universidade Federal de Santa Catarina.

Orientador: Dr. Fernando Ostuni Gauthier

Coorientador: Eng. Paulo Narciso Filho

ACADÊMICO: GUSTAVO LÜCKMANN FABRO

Florianópolis, Agosto, 2003.

DEDICATÓRIA

A meus pais, pelo carinho
e amor incondicional;

A meu irmão, companheiro eterno
de caminhada;

E a Deus, pelo Dom maravilhoso
da vida, sem a qual nada
seria possível.

AGRADECIMENTOS

Agradeço a todos que, de alguma forma, contribuíram para a realização deste trabalho, em especial:

A meus orientadores, pela experiência e orientação sobre as melhores formas de atingir meus objetivos;

À HarboR Informática Industrial, por proporcionar as condições necessárias para o desenvolvimento deste trabalho, em um ambiente de aprendizagem amigável e produtivo;

À Universidade Federal de Santa Catarina e todos os meus professores, pelo conhecimento e experiência transmitidos ao longo do curso;

A todos os meus amigos e amigas, pelo apoio, compreensão e desejo de sucesso;

A minha familia, pelo amor, carinho e pela primorosa educação com a qual alicerçaram toda a minha vida;

E, por fim, a Deus, pela oportunidade de estudar, conhecer as coisas, evoluir... enfim, de viver!

RESUMO

Este trabalho versa sobre o desenvolvimento e a implementação de um sistema industrial de coleta de dados relativos a rastreabilidade e genealogia de forma integrada a um sistema de gestão, para uma indústria de compressores.

O sistema foi dividido em módulos, de forma a organizar e facilitar seu desenvolvimento. Para o processo de desenvolvimento, foi utilizada uma abordagem de engenharia de software adotada por Pressman, consistindo em análise, projeto e implementação. Modelos de dados e diagramas UML foram utilizados para modelagem.

Utilizou-se a tecnologia ActiveX para possibilitar a execução dos módulos em ambiente Web, de forma automática e sem a necessidade de instalação de softwares específicos (apenas o navegador).

Foram criados bancos de dados Access e MS-SQL para o armazenamento temporário dos dados. As transferências de dados de e entre bancos, módulos e a interface com o sistema de gestão SAP R/3 deu-se através do uso de XML. Além disso, a linguagem XML é também aplicada para configuração do hardware e impressão de etiquetas (através de transformação XSL). Todo hardware envolvido tem como interface o padrão RS-232.

O sistema funcionou como o esperado e possibilitou, de forma sumária, a coleta necessária para armazenamento de dados de rastreabilidade e genealogia para a linha de produção de unidades condensadoras e unidades seladas da Embraco S.A., assegurando assim a manutenção dos padrões de qualidades exigidos pela referida indústria.

ABSTRACT

This research concerns the development and implementation of an industrial data acquisition system for genealogy and traceability on a compressor industry, interconnected to an enterprise resource planning software.

The concept and project of the system was split into modules, in order to better organize code construction and interaction. The development process used a software engineering approach based on Pressman, consisting in analysis, project and implementation. The models were built using UML and data model diagrams.

ActiveX technology was used to allow the modules execution in a Web environment in a transparent way, without the need for installation and setup of a specific client program (only the browser).

Access and MS-SQL databases were created for temporary data storage. XML was used for the data transfers to and from databases, modules and SAP R/3 interface. XML was also used for hardware configuration and label printing (the latter through XSL transformation). All hardware involved used as interface the RS-232 standard.

The system worked as expected and allowed, in a summarized way, the necessary data acquisition regarding genealogy and traceability for the production line of condensing and sealed units from Embraco S.A., ensuring the maintenance of the quality patterns demanded by the referred industry.

LISTA DE SIGLAS

ASP	Active Server Pages
ERP	Enterprise Resource Planning
HTML	HyperText Markup Language
HTTP	Hyper Text Transfer Protocol
IIS	Internet Information Server
LCD	Liquid Crystal Display
RFC	Remote Function Call
SAP	Systems, Applications and Products in data processing
SQL	Structured Query Language
UC	Unidade Condensadora
UML	Unified Modeling Language
US	Unidade Selada
XML	eXtensible Markup Language
XSL	eXtensible Stylesheet Language
ZPL	Zebra Programming Language

LISTA DE FIGURAS

SUMÁRIO

1. INTRODUÇÃO

1.1 CONSIDERAÇÕES INICIAIS

A indústria catarinense vem crescendo nos últimos anos. A Embraco S.A., indústria de compressores com sede em Joinville, é um exemplo de empresa em franco processo de expansão, tendo ampliado seus horizontes com vertentes na China, Itália, Eslováquia e Estados Unidos, e sendo atualmente responsável por 25% da produção mundial de compressores.

Na medida que cresce a demanda, contudo, cresce também a concorrência, e acima de tudo a necessidade de produzir cada vez mais e com qualidade. Neste sentido, sistemas computacionais vêm contribuindo de forma cada vez mais presente nos processos produtivos de empresas. Alguns se dedicam a auxiliar e orientar o processo de gestão; outros, o próprio processo produtivo. O desenvolvimento de tais sistemas e, principalmente, a integração entre estes, vê-se cada vez mais salutar e necessária para o aprimoramento e aumento de competitividade destas empresas.

Neste trabalho, será apresentado o estudo, análise, projeto e implementação de um sistema industrial com o intuito de melhorar o processo de produção de um dos setores da empresa Embraco S.A., assegurando a correta montagem e armazenagem de produtos e coletando dados para posteriores análises pelo sistema de gestão existente.

1.2 OBJETIVOS

1.2.1 Objetivo geral

O objetivo geral é o desenvolvimento de um sistema que permita, de forma integrada a um sistema de gestão, automatizar o processo de coleta de dados de genealogia e rastreabilidade para a linha de produção de uma indústria de compressores, certificando a correção da montagem e evitando erros.

1.2.1 Objetivos específicos

Desenvolvimento de um sistema que:

- automatize e colete dados relativos à genealogia e rastreabilidade da produção e paletização;
- integre-se com o SAP R/3, no sentido de recolher informações sobre o que deve ser produzido e paletizado e fornecer os dados que foram coletados e gerados;
- gere etiquetas de códigos de barras para identificação única dos itens produzidos;
- seja robusto, de forma a tolerar os mais diversos tipos de erros e possa funcionar 24 horas por dia, 7 dias por semana;
- seja flexível e de fácil instalação e configuração

1.3 JUSTIFICATIVA

As motivações para este trabalho foram as seguintes:

- importância de um sistema automatizado, que garanta a qualidade no processo de produção de uma indústria;
- experiência de análise, projeto e implementação de um sistema real, complexo e robusto;
- estudo para integração com hardware necessária para o sistema em questão: displays digitais, leitores de código de barras, impressora específica;
- estudo para desenvolvimento e interação com bancos de dados;
- estudo de tecnologia para sistemas Web – controles ActiveX
- aprendizado e uso de tecnologia XML, utilizada para configuração do hardware, impressão de etiquetas e transferência de dados em geral.

1.4 ESTRUTURA

Para atender aos objetivos propostos, este documento foi estruturado da seguinte forma:

O capítulo 1 apresenta as motivações e a justificativa para o estudo e desenvolvimento do trabalho em questão.

O capítulo 2 disserta sobre alguns aspectos específicos do problema, necessários para a compreensão deste como um todo.

No capítulo 3 temos os resultados da etapa de **análise** do problema. Aqui são detalhados o problema apresentado e a especificação de requisitos, juntamente com a solução proposta.

No capítulo 4 é detalhada a etapa do **projeto** do sistema. Aqui é apresentada toda a arquitetura do sistema (física e lógica), bem como definidos e detalhados os modelos estático (diagramas de classe e modelo de dados) e dinâmico (diagramas de estado).

O capítulo 5 traz alguns comentários sobre a **implementação** do sistema: como foi a etapa de codificação, algumas peculiaridades do processo de desenvolvimento e como puderam ser feitos os testes. São também apresentadas aqui as interfaces do sistema com o usuário.

O capítulo 6, por fim, encerra com os resultados obtidos e a conclusão final do trabalho.

2. ASPECTOS ESPECÍFICOS DO PROBLEMA

2.1 EMBRACO S.A.

A Embraco é uma indústria que foi fundada em 1971, com o objetivo de atender a indústria brasileira de refrigeradores. Produzindo compressores inicialmente para a indústria nacional, logo iniciou a exportação e hoje tem fábricas na, China, Eslováquia e Itália, além do Brasil e de um escritório nos Estados Unidos, empregando cerca de 9 mil pessoas e atendendo cerca de 75% do mercado nacional e 25% do mercado mundial.

A Embraco tem, no Brasil, sede em Joinville e uma fábrica de componentes em Itaiópolis. Lá ocorre a produção de componentes eletrônicos e existe uma linha de montagem para as chamadas Unidades Condensadoras e Unidades Seladas (UC / US), utilizadas em sistemas de refrigeração em geral, que veremos a seguir.

2.2 SISTEMAS DE REFRIGERAÇÃO

Um sistema de refrigeração, sumariamente falando, é composto por quatro unidades principais: Um compressor, um condensador, uma válvula de expansão e um evaporador. Uma ilustração de uma unidade de refrigeração pode ser visto a seguir:

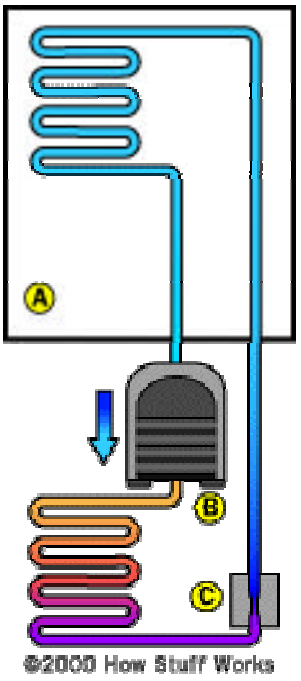

Figura 1: Esquema de um sistema de refrigeração

A Embraco, como fornecedora de equipamentos para a indústria de refrigeração, comercializa diferentes opções para a produção de sistemas de refrigeração.

2.3 UNIDADES CONDENSADORAS / UNIDADES SELADAS

Além de compressores, a Embraco produz também Unidades Condensadoras e Unidades Seladas.

Unidade Condensadora é a associação do compressor com o condensador, já montado, como produto final. Alguns clientes já compram a unidade pronta, enquanto outros se limitam ao compressor e preferem montar a sua própria, com suas características particulares.

Já a Unidade Selada é a associação de um compressor com um condensador e um evaporador e uma válvula de expansão, formando todo o circuito de refrigeração. É chamada de "selada" por não permitir modificações no circuito, estando este já completo e pronto para ser utilizado.

2.4 SISTEMAS DE GESTÃO - ERP

Os chamados Sistemas de Gestão são sistemas que gerenciam a maior parte das etapas do processo de uma empresa. São softwares enquadrados na chamada engenharia de negócios, que têm como objetivo dinamizar e informatizar todo o processo de gestão, oferecendo recursos para análises, planejamento, produção, distribuição, enfim, integram e centralizam informações para uma boa administração.

A SAP (Systems, Applications and Products in Data Processing) é a empresa que faz o atual maior sistema de gestão no mundo, o R/3. Vários módulos deste software são utilizados pela Embraco, para a gestão de seu processo de manufatura. Um destes módulos é o responsável pela administração das informações de rastreabilidade / genealogia.

2.5 RASTREABILIDADE/GENEALOGIA

O SAP, como sistema de gestão, tem um módulo específico para armazenar informações de rastreabilidade e genealogia.

As unidades condensadoras são produtos serializáveis, ou seja, contém um número de série. Da mesma forma, os compressores produzidos pela Embraco possuem, todos, um número de série. Informações sobre genealogia, para o problema em estudo, se referem a saber a quais unidades condensadoras/seladas foram montadas quais compressores, bem como saber em quais paletes estas unidades foram armazenadas depois de produzidas.

Rastreabilidade, por outro lado, é a informação que permite, como o nome sugere, rastrear todo o processo de produção de algum determinado produto. No caso, para as unidades condensadoras/seladas e para a etapa de produção contemplada, as informações relativas à Ordem de produção SAP às quais estas determinadas unidades fazem parte. São também coletadas, para rastreabilidade, informações relativas a data (dia, turno) e onde (linha) as unidades foram produzidas.

Informações de genealogia e restreabilidade são importantes para que se possa rastrear o processo de produção de um produto na procura de informações sobre peças utilizadas, fornecedores, etc. Tais informações são úteis para controle de qualidade, *recalls* que porventura tenham de ser feitos, etc.

3. ANÁLISE

3.1 INTRODUÇÃO

Como Pressman [7] define, a engenharia de sistemas de computador é uma atividade destinada a solucionar problemas. As funções desejadas para o sistema são desvendadas, analisadas e designadas aos elementos individuais do sistema. O engenheiro de sistemas de computador começa com metas e restrições definidas pelo cliente e deriva uma representação da função, desempenho, interfaces, restrições de projeto e estrutura de informações que podem ser atribuídos a cada um dos elementos de sistema.

Ora, para que possa iniciar o processo de desenvolvimento de um sistema é necessário, antes de tudo, a definição concreta do que se espera deste. Esta é a responsabilidade dessa primeira etapa de desenvolvimento.

Iniciamos com uma definição do problema. Durante esta etapa, uma descrição limitada do escopo do software é desenvolvida, os recursos exigidos para se desenvolver o software são previstos e estimativas de custo e de prazo são estabelecidas, sendo efetuada uma análise de riscos. Aqui se deve validar a viabilidade do projeto, em relação aos custos e prazos estabelecidos.

Segundo Pressman, a etapa seguinte consiste na análise e definição dos requisitos de software. Exigências de desempenho ou limitações de recursos são traduzidas em características de projeto do software. Esta etapa é desenvolvida junto ao cliente, e seu resultado servirá de base para o planejamento e para demonstrar, quando da conclusão do trabalho, que os requisitos foram cumpridos.

3.2 DEFINIÇÃO DO PROBLEMA

A fábrica de Itaiópolis da Embraco possuía uma linha de produção para Unidades Condensadoras e Seladas que produzia em uma escala muito baixa. Com a baixa demanda, algumas informações sobre a produção, encontradas nos códigos seriais dos compressores, já eram suficientes para se ter algum controle de qualidade.

Contudo, a empresa está expandindo esta linha de produção, de sorte que as poucas informações disponíveis no número serial já não seriam suficientes para a manutenção da qualidade desejada. O departamento de qualidade da Embraco, seguindo uma série de padrões e normas para assegurar a qualidade da produção, tem como exigência que se colete informações relativas à genealogia e rastreabilidade destes produtos, pelos motivos já expostos. No entanto, a coleta manual destas informações se tornaria inviável, dada a demanda de produção e a possibilidade de erros que tal procedimento acarretaria.

Assim, evidenciou-se a necessidade de investimento em um sistema que permitisse a coleta de informações de produção relativas à rastreabilidade e genealogia, assegurando a correção das informações e já auxiliando o processo de produção, no sentido de prevenir erros de montagem e/ou armazenamento.

3.2.1 Descrição geral do processo

O processo de produção em Itaiópolis é dividido em duas etapas principais : a genealogia e a paletização

Na primeira etapa é feita a genealogia. Através de uma esteira chegam a unidade condensadora e o compressor que nela deve ser montado. O operador deve interagir com o sistema de forma a verificar se a montagem está correta e pode ser autorizada. Obtendo a confirmação, monta a unidade. Uma etiqueta é impressa pelo sistema, que é colada na caixa da unidade e encaminhada para a próxima etapa.

Na segunda etapa, é feita a paletização. Após ter iniciado a ordem respectiva àquelas unidades, o operador comanda a abertura, introdução das unidades e fechamento de cada palete. O sistema faz todas as verificações devidas, registra as informações e autoriza a paletização.

3.2.2 Descrição geral dos módulos de software

As estações de coleta de dados executarão Browsers Internet Explorer, no qual serão executados componentes ActiveX responsáveis pelo funcionamento de diferentes partes do sistema:

Init

Para instalação/inicialização do sistema, o usuário deverá acessar endereço na Intranet, que conterá componente Init.

Este componente realiza conexão com servidor Web para criar/atualizar localmente dados de inicialização do sistema.

Paletização

Controla estação onde estará instalado o PC e terá teclado e leitor de código de barras com fio como interfaces com usuário.

Neste componente serão executadas tarefas de:

- início de ordem, consultando no SAP dados da ordem e do material a ser produzido
- apontamentos de paletização (associação/desassociação de números de série de produtos a paletes)
- encerramento de ordens, confirmando produção e lotes no SAP R/3
- entrada manual de dados de ordens, quando não é possível consultar no SAP R/3

Genealogia

Controla estação onde estarão instalados displays LCD para mensagens para usuário, que terá leitor sem fio para interface com sistema.

São executadas tarefas de:

- associação de componentes a produtos (genealogia), através da leitura de suas etiquetas de código de barras
- comanda impressão de etiquetas, a partir de dados do produto consultados no SAP pelo componente de Paletização

Impressão

Componente controlado pela estação de genealogia, reponsável pela impressão de etiquetas. Realiza toda comunicação com impressora Zebra, em linguagem proprietária.

Transferência

Componente responsável pela transferência dos dados coletados ao SAP.

3.3 ANÁLISE E DEFINIÇÃO DE REQUISITOS

3.3.1 Requisitos Gerais

3.3.1.1 Disponibilidade

O aplicativo de coleta de dados deverá rodar 24x7 (24 horas por dia, 7 dias por semana), independente do funcionamento dos servidores SAP ou IIS.

3.3.1.2 Armazenamento definitivo de dados

O sistema armazenará dados de ordens, paletes e produtos apenas enquanto o palete está sendo processado. Depois de fechado o palete, todos os dados são transferidos para SAP R/3.

3.3.2 Requisitos de identificação

Unidades Condensadoras e Compressores

As etiquetas de unidades condensadoras e compressores serão impressas no sistema de Etiquetas, seguindo padrão já existente de 17 dígitos (9 para código do material, 8 para número de série), padrão Code39.

Paletes

As etiquetas de paletes serão impressas no sistema de etiquetas, seguindo mesmo padrão já existente na matriz de Joinville da empresa.

Caixas Individuais

A etiqueta da caixa individual terá leiaute único para todos os modelos de UC/US, e não terá elementos gráficos.

Todas as etiquetas serão impressas em impressora Zebra, de um único modelo.

3.3.3 Requisitos de performance

As operações de paletização ou genealogia não podem ser interrompidas enquanto o sistema estiver conectando ou trocando informações com sistemas complementares ou periféricos.

3.3.4 Restrições de Design

3.3.4.1 Estações de chão-de-fábrica

- Não haverá dispositivo apontador (mouse, touch-screen, etc.)
- Tamanho de tela será 800x600 pixels
- Deverá ser possível ler a tela à distância mínima de 3 metros

3.3.4.2 Plataforma de Servidores

Será utilizada plataforma Microsoft já existente: IIS, SQL Server, ASP.

3.3.5 Outros Requisitos

3.3.5.1 Start-up do Sistema

Para "instalação" do sistema, deverá ser acessada página html da Intranet, de acordo com linha onde o sistema está sendo instalado.

Esta página fará o download automático de componentes ActiveX e arquivos de configuração do sistema para o disco local do PC. O sistema estará pronto para uso.

Para as utilizações subsequentes, serão sempre acessados arquivos html locais.

O sistema fará updates automáticos de componentes do sistema, se necessário. Neste caso será mandatória a existência de conexão com servidor Web.

3.3.5.2 Configuração do Sistema

Serão configuráveis no sistema:

Parâmetros de inicialização

- Identificação da linha
- Parâmetros de comunicação com impressora
- Parâmetros de comunicação com leitores de código-de-barras
- Parâmetros de comunicação com displays
- Parâmetros de funcionamento do aplicativo de transferência

Leiaute da Etiqueta

Será possível a formatação de um leiaute por linha, através de edição de arquivo XSL contendo protocolo ZPL.

3.3.6 Estratégias de Arquitetura e Design

- A estação de controle geral do sistema terá aplicativos rodando sobre browser Microsoft Explorer versão 5.5 ou superior, sem necessitar de instalação de aplicativo cliente. Será necessário configurar o browser para permitir a instalação e execução de ActiveX no mesmo, o que, dependendo do sistema operacional utilizado, pode requerer que um usuário com privilégios de Administrador execute a aplicação pelo menos uma vez em cada PC, para completar a instalação do ActiveX.
- Todos os dados de rastreabilidade e genealogia serão armazenados, consultados e editados no SAP R/3.
- Será utilizado servidor WEB MS-IIS já existente na Matriz/Joinville.
- O operador utilizará os leitores de código de barras para comandar o sistema (não haverá mouse)

3.3.7 Interfaces Externas

3.3.7.1 Interface com Usuário

Todos as funções serão comandadas a partir de leituras de códigos de barras pré-impressos, fixados junto às estações. Somente para escolha das ordens será utilizado teclado.

O operador da paletização utilizará o display do PC como interface. O operador da estação de genealogia terá 2 displays de cristal líquido que apresentarão mensagens de 2x20 caracteres, idênticas nos dois displays.

Não haverá alarmes sonoros ou bloqueio de linha em qualquer situação.

3.3.7.2 Interfaces com Periféricos

Impressoras

Uma impressora Zebra 90 XiIII será conectada ao sistema através de interface serial RS-232. Os dados de etiquetas serão enviados em formato ZPL.

ZPL é acrônimo para Zebra Programming Language, e é uma linguagem proprietária para comunicação com impressoras Zebra.

O sistema monitorará o status da impressora antes da impressão cada nova etiqueta. Em caso de erro, este será mostrado no componente que controla a impressora. O usuário terá ainda a opção de verificar o erro no próprio painel da impressora.

Leitores código de barras

Os leitores de código de barras serão conectados ao sistema através de interface serial RS-232.

Poderão ser utilizados leitores de quaisquer fabricantes, desde que possibilitem a configuração de envio de caracteres de início e fim de mensagem.

LCDs (Displays de cristal líquido)

A comunicação com os LCDs será feita via uma interface RS-232.

Será implementado protocolo do “Display Torre Gradual”. Os dois LCDs serão ligados em cascata, ou seja, a informação enviada ao primeiro display será reproduzida integralmente no segundo display.

3.3.7.3 Interfaces de Software

O sistema terá interfaces RFC com SAP R/3, detalhada em documento específico.

3.4 SOLUÇÃO PROPOSTA

3.4.1 Finalidades

O sistema apresentado tem por finalidades:

- coletar dados de genealogia compressor x UC/US
- coletar dados de rastreabilidade, associando números de série a lotes (embalagens), lotes a ordens de produção;
- gerar etiquetas de identificação de embalagens individuais de produtos, replicando número de série da UC/US;
- através de interfaces com SAP R/3, serão realizados apontamentos de produção e de rastreabilidade, associando lotes a ordens e números de série a lotes, e também dados de genealogia.

3.4.2 Requisitos técnicos

São requisitos técnicos do sistema:

- estação de coleta de dados deverá funcionar mesmo que conexão de rede esteja inativa;
- a coleta de dados será feita através de leitores de código de barras.

3.4.3 Restrições

São restrições do sistema:

- o sistema não coletará dados de lotes de matéria-prima;

- somente serão utilizadas impressoras Zebra do mesmo modelo, leiaute de etiqueta é fixo por tipo de embalagem, não serão impressas imagens nas etiquetas (futuramente poderão ser desenvolvidos drivers de impressão para outros tipos de impressoras);
- o sistema **não contempla** armazenagem e consulta de dados históricos;
- após apontamento dos dados no SAP, todas as transações deverão ser efetuadas diretamente no SAP (desmontagem/rearranjo de palete, desmontagem de unidade condensadora, etc.), o sistema de paletização não fará nenhuma crítica com relação a repetição de peças já produzidas (exceto no momento do apontamento do palete, se o SAP não aceitar o apontamento).

3.4.4 Funcionalidades

No chão-de-fábrica, estações de coleta com arquitetura PC rodarão browser Internet Explorer, onde serão "baixados" do servidor Windows/IIS os componentes de software do sistema, com as seguintes funções:

- interface com usuário;
- interfaces com periféricos (impressora);
- interfaces com SAP R/3;
- salvar localmente dados que não puderam ser integrados com SAP por indisponibilidade de rede.

Funcionalidades do Aplicativo Principal:

a. Início / Encerramento de ordem de produção (interface SAP);

b. Abertura / fechamento de palete;

c. Impressão de etiqueta de embalagem individual. Os dados do material necessários para a impressão das etiquetas individuais (descrição, características de engenharia) serão lidos do SAP e salvos em arquivo local, não sendo lidos novamente até que seja trocada a ordem de produção;

d. Inserção / retirada de peça do palete;

e. Interfaces com SAP para consulta de ordens de produção, dados de material, apontamento de produção e lotes, apontamentos de números de série, apontamentos de genealogia.

Funcionalidades sem rede (Aplicativo Principal):

a. o usuário digitará manualmente a Ordem de Produção;

b. dados serão coletados normalmente e serão armazenados localmente;

c. quando a rede estiver disponível os dados serão enviados ao servidor IIS, que fará os apontamentos no SAP R/3;

d. no caso de haver problemas com a integração dos dados (problemas de consistência de dados digitados), registros serão armazenados no SQL Server e usuário receberá um aviso na sua estação de coleta; através de computadores de escritório será possível acessar via browser os registros problemáticos e corrigí-los;

e. serão utilizados para a impressão de etiquetas os dados do material conforme salvos na última vez em que foram produzidos com a rede disponível. Caso o material esteja sendo produzido pela primeira vez, ou o arquivo de dados local esteja corrompido, não será possível fazer a impressão das etiquetas individuais.

Funcionamento (Aplicativo Genealogia):

a. Após início da ordem de produção na estação de paletização, o sistema consultará no SAP código do compressor que deverá ser montado na UC/US;

b. número de série do compressor deve ser lido, e sistema verifica no SAP se o componente é compatível com a ordem (se está na Lista Técnica do material produzido);

c. número de série do produto final é informado, e sistema associa os dois números de série sem apontar no SAP (banco de dados do sistema);

d. ao apontar produção (Aplicativo Principal), as informações de genealogia serão passadas para o SAP.

Funcionamento sem rede (Aplicativo Genealogia):

a. o usuário informa a ordem de produção e lê os códigos de série do componente e produto acabado (não será feita nenhuma verificação);
b. os dados são salvos localmente e quando a conexão for restabelecida são enviados para o servidor IIS, onde são armazenados em banco de dados SQL Server e aguardarão o processamento em batch dos apontamentos de produção correspondentes;
c. no caso de haver problemas com a integração dos dados (problemas de consistência de dados digitados), o usuário receberá um aviso na sua estação de coleta; através de computadores de escritório será possível acessar via browser os registros problemáticos e corrigí-los.

3.4.4 Interfaces SAP R/3

Será instalada SAP GUI no servidor Web MS-IIS. O aplicativo de coleta de dados fará requisições ao Web Server, que disparará as transações do SAP online. Quaisquer problemas de integração serão informados ao usuário no momento do apontamento.

No caso de registros que foram armazenados localmente por indisponibilidade de rede, serão enviados todos ao servidor, que os processará em Batch. Em caso de erros, o usuário da estação de coleta receberá um alarme. Também será possível acessar os alarmes e editar dados incorretos a partir de qualquer browser.

3.4.5 Periféricos

O leitor de código de barras terá interface via RS-232 com o sistema.

Impressora será conectada através de porta serial RS-232, e controlada por ActiveX instanciado no browser.

3.4.6 Manutenção do sistema

O sistema será instalado automaticamente nos PCs, através do próprio browser. Atividades de configuração das estações de coleta, versões de produção, etc. serão realizadas de qualquer browser, através de aplicativo Web.

Leiautes de etiquetas serão configurados em linguagem proprietária da impressora.

Hardwares adicionais do PC deverão ser instalados no local (placa multiserial, por exemplo).

3.4.7 Benefícios e vantagens da solução

- Arquitetura baseada em browser Internet Explorer reduz custo de manutenção e esforço para instalação e configuração do sistema;
- Não há necessidade de instalação de qualquer software na Estação de Coleta além do sistema operacional e Internet Explorer;
- Controles de produção e rastreabilidade totalmente baseados no SAP R/3;
- Sistema de simples operação.

4. PROJETO

4.1 INTRODUÇÃO

Passada a etapa de análise, cabe à próxima fase projetar o que será desenvolvido em software para satisfazer o problema proposto.

É na etapa de projeto que são definidos e modelados todos os componentes de software que terão de ser desenvolvidos. PRESSMAN [7] nos diz que "a meta do projetista é produzir um modelo ou representação de qualquer entidade que será construída porteriormente". Desta forma, temos que ter como resultado desta fase toda a estrutura do software definida e especificada, de sorte que tenhamos uma base sólida para a etapa de implementação, que garanta a qualidade e a facilidade de manutenção.

Neste sentido, inicialmente modelamos a arquitetura física do sistema (equipamentos de hardware necessários) e a arquitetura lógica (divisão em módulos do sistema, e componentes de cada módulo, bem como a interação entre estes).

Em seguida, foram adotados padrões para os modelos estáticos e dinâmicos do sistema. Para os modelos estáticos, utilizou-se um modelo de dados para a modelagem do dados do banco local, seguido de uma descrição textual de cada elemento da tabela. Foram também modelados os diagramas de classes, segundo o padrão UML para representação das classes.

Por fim, foi modelado o comportamento dinâmico do sistema com diagramas de estado, consoantes com o padrão UML, complementados por uma descrição textual do comportamento de cada funcionalidade.

4.2 ARQUITETURA DO SISTEMA

Após a visão geral do sistema, suas arquiteturas físicas e lógicas são elucidadas a seguir.

4.2.1 Arquitetura física

A arquitetura física apresenta os periféricos exigidos por todo o sistema, bem como sua disposição na linha de produção:

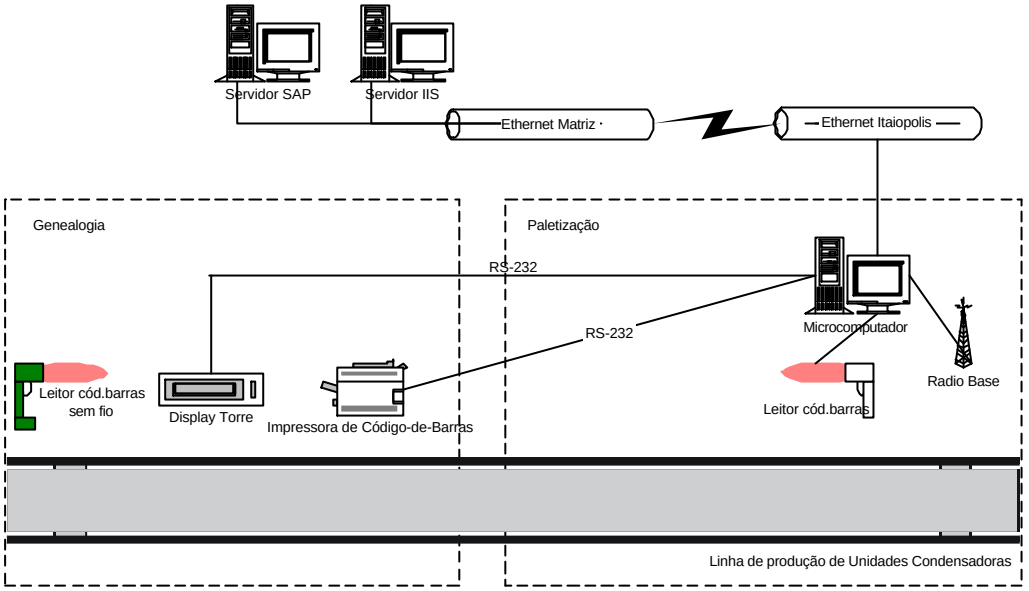

Figura 2: Arquitetura física do sistema

Uma descrição de cada um dos componentes da arquitetura física do sistema segue-se:

- **Microcomputador**

  Computador localizado na linha de paletização, onde devem ser executados os vários módulos do sistema. Deve rodar o sistema operacional Windows, e ter um navegador Internet Explorer 5.5 ou superior. No computador deve ser instalada uma placa multi-serial de forma a expandir sua capacidade para 4 portas seriais

- **Leitor código de barras**

  É um leitor de código de barras, marca HHP, modelo IT3800. Sua comunicação com o computador é feita através da interface RS-232.

- **Leitor código de barras sem fio**

É um leitor de código de barras, com a diferença de que sua comunicação é feita por radiofrequência. Este leitor tem a vantagem de permitir a locomoção em grande amplitude, sem a limitação do fio RS-232.

- **Display Torre**

  É um display digital, de 20 colunas e 2 linhas, marca Wilbor Tech, situado na etapa da genealogia, para mostrar as mensagens do sistema para o usuário

- **Impressora ZebraS400**

  Impressora de código de barras, marca Zebra, modelo S400. A comunicação desta com o microcomputador é feita pela interface RS-232, utilizando-se de uma linguagem própria, chamada ZPL.

- **Servidor SAP**

  É o servidor já existente na empresa, que abriga o sistema ERP

- **Servidor IIS**

  É outro servidor já existente, que fornece os serviços de servidor HTTP (com o Internet Information Server).

4.2.2 Arquitetura lógica

A arquitetura lógica do sistema descreve os vários componentes e o relacionamento entre eles. Um diagrama geral dá uma visão em alto nível do sistema, e um diagrama específico detalha o funcionamento interno dos módulos mais complexos:

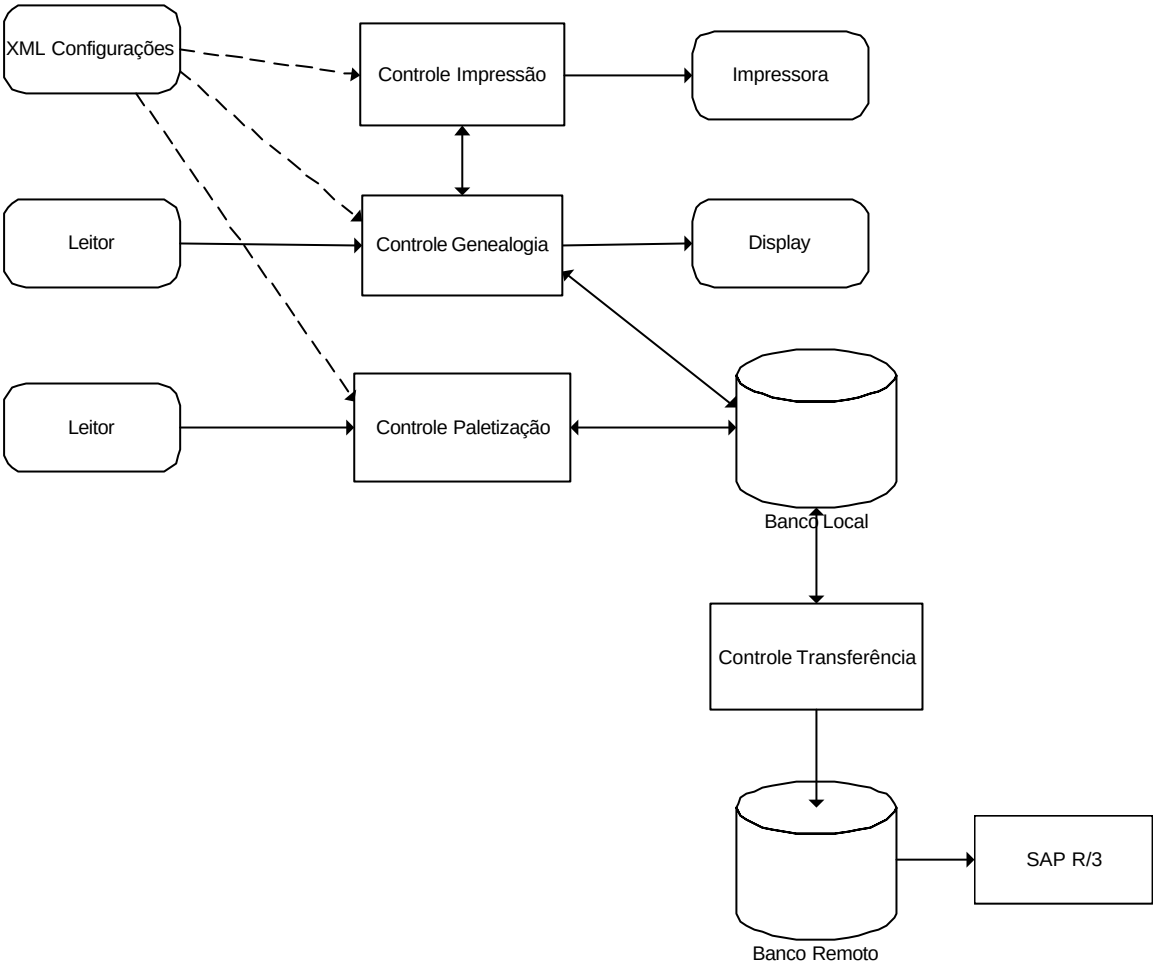

Figura 3: Modelo Lógico geral

O sistema foi construído em forma de módulos. Cada módulo é responsável por uma funcionalidade específica do sistema, e foi encapsulado em um controle ActiveX. Como pode ser observado no diagrama, o banco de dados local e o arquivo XML com as configurações de hardware/software é compartilhado entre vários módulos.

Todos os módulos são executados, simultaneamente, em uma única janela do browser no computador cliente. A comunicação entre eles é feita em sua maior parte através do banco local, com exceção da interação entre os módulos Genealogia e Impressão, que é feita por chamadas diretas de funções.

4.2.2.1 Módulo Impressão

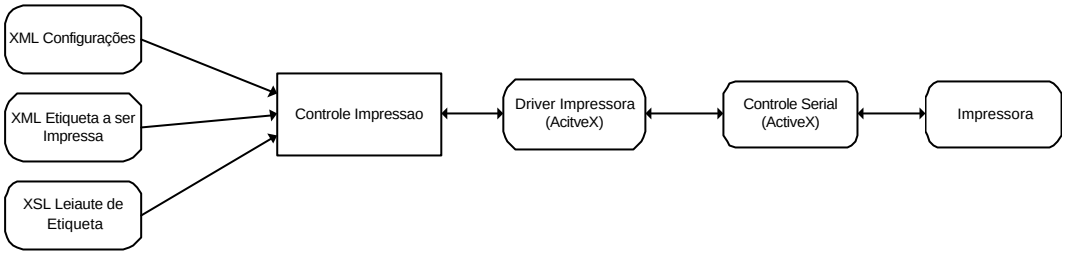

Figura 4: Modelo Lógico Módulo Impressão

A finalidade deste módulo é imprimir as etiquetas de códigos-de-barras a serem impressas nos recipientes das unidades condensadoras. Este controle lê, de um arquivo XML, as configurações respectivas à localização da impressora (porta serial em que esta se encontra) e configurações desta (paridade, baud rate, data bits, caracter de controle e caracter de formatação), e inicializa o driver da impressora com as configurações especificadas. Este controle se comunica com o controle de genealogia, do qual recebe um objeto XML com as informações sobre a etiqueta a ser impressa, bem como um XSL com a formatação a ser aplicada. O controle então repassa estes objetos ao driver para sua impressão. Ele também checa o status da impressora de tempos em tempos, mostrando-o na tela principal.

4.2.2.2 Módulo Genealogia

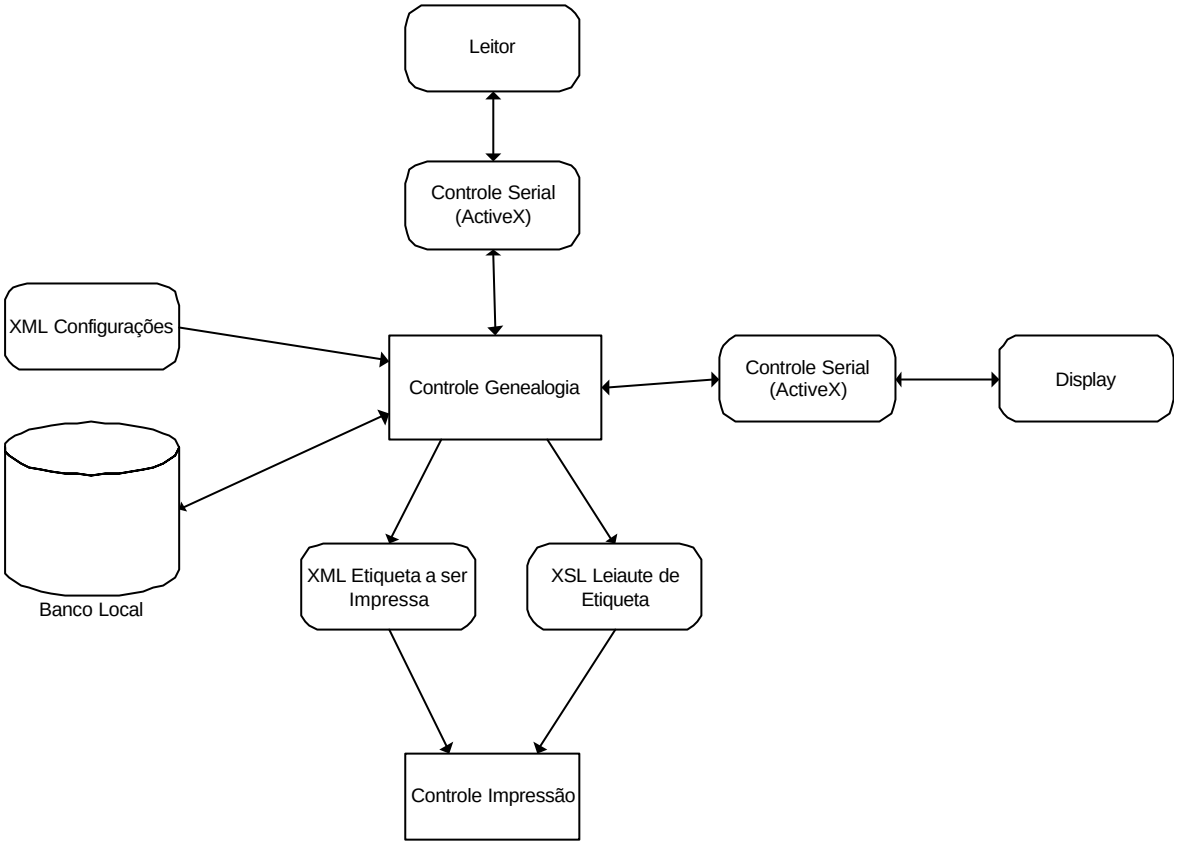

Figura 5: Modelo Lógico Módulo Genealogia

Este módulo é o responsável pela genealogia da montagem das unidades condensadoras. As configurações relativas à localização dos periféricos com os quais se comunica estão em um arquivo XML de configurações, lido por ele. A comunicação com o leitor de código de barras é feita através de um controle ActiveX, também desenvolvido.

O controle executa consultas em um banco local para verificar a validade da montagem que, se aceita, é armazenada neste mesmo banco. As informações do estado atual do sistema, bem como o que o usuário deve fazer são mostradas tanto na interface principal do controle bem como em um display eletrônico (pelo fato da linha de genealogia se situar distante da linha de paletização). Após validar a montagem, o sistema captura do banco os dados relativos ao produto montado, monta um XML com estas informações e o repassa ao controle de impressão para a impressão da etiqueta.

4.2.2.3 Módulo Paletização

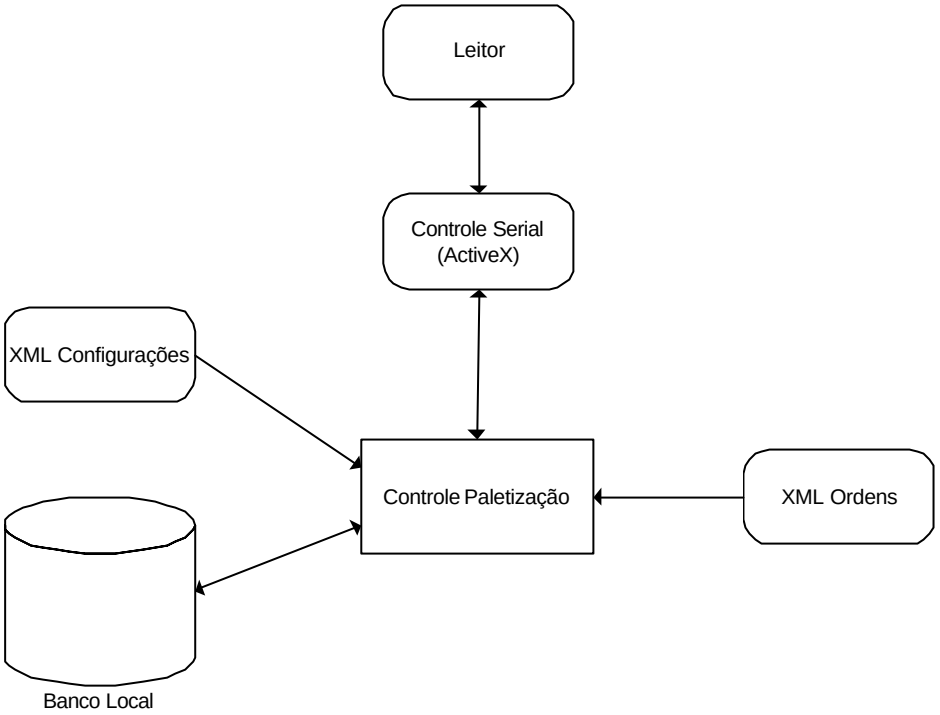

Figura 6: Modelo Lógico Módulo Paletização

O controle de paletização é o que apresenta a máquina de estados mais extensa, e é responsável pelo processo de paletização. Ele lê as configurações relativas ao hardware (leitor) em um arquivo XML. É através do leitor que toda a comunicação com o operador é feita. Foram elaborados códigos para representar os comandos (Abrir Ordem, Fechar Palete, Inserir Produto, Desmontar Palete, etc). Estes comandos, quando lidos, é que vão direcionar o andamento do sistema. É este controle quem atualiza o banco local com as informações sobre as ordens SAP. Quando é lido o comando para iniciar uma nova ordem SAP, ele faz a requisição a uma pagina ASP solicitando os dados das ordens do SAP, obtendo como retorno um XML contendo as ordens e suas informações. Estas informações são então repassadas no banco local.

4.3 MODELO ESTÁTICO

4.3.1 Diagrama de modelo de dados

O diagrama de modelo de dados descreve as tabelas e as relações entre as tabelas de um banco de dados relacional.

Para o sistema em questão, foi feito o diagrama para o banco local, apresentado a seguir:

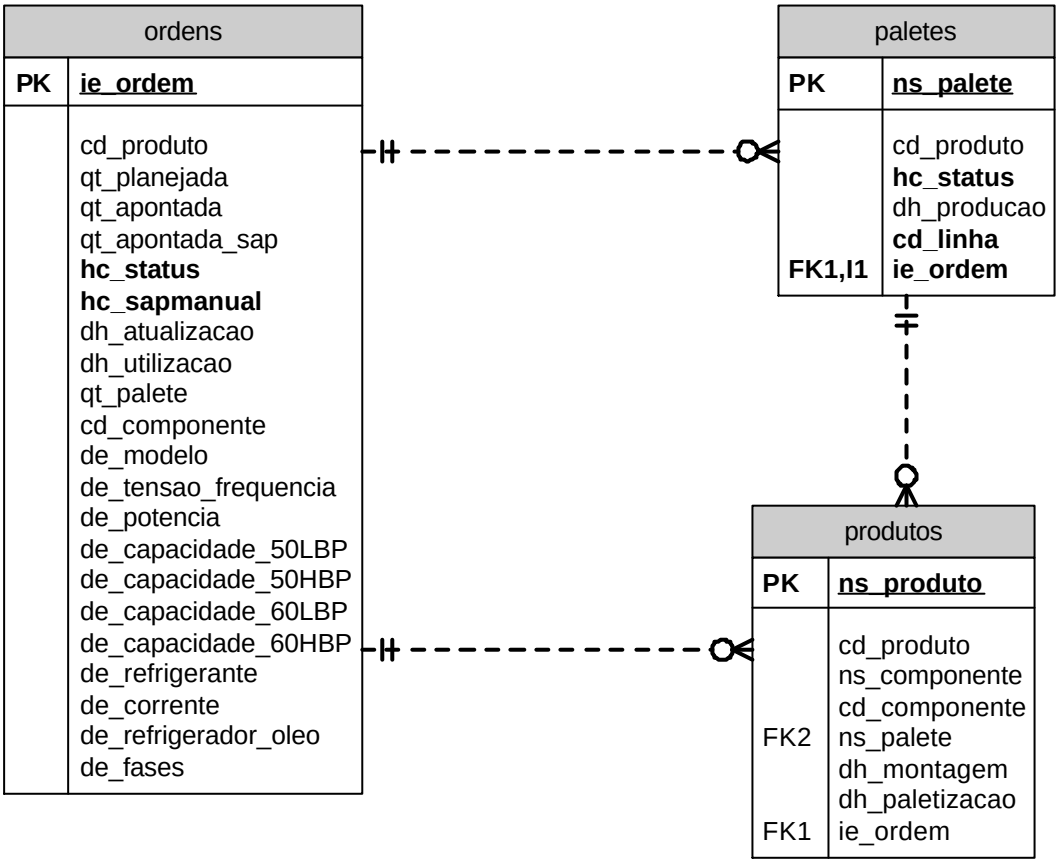

Figura 7: Diagrama de modelo de dados - Banco local

A redundância na chave estrangeira ie_ordem, na tabela produtos, deve-se ao fato da genealogia preceder a paletização. Assim, no momento da genealogia não se têm ainda as informações sobre o palete ao qual aquele produto fará parte, sendo contudo necessário armazenar qual ordem forneceu as especificações para aquela montagem.

Segue uma descrição de cada coluna para cada tabela do diagrama:

Tabela Ordens:

Item	Descrição
ie_ordem	Identificador da Ordem
cd_produto	Código de produto para esta ordem
qt_planejada	Quantidade de peças planejada para ordem
qt_apontada	Quantidade já apontada nesta ordem
qt_apontada_sap	Quantidade produzida já cadastrada no SAP
hc_status	Status da ordem: Iniciada, Consultada, Encerrada
hc_sapmanual	Indica se a ordem foi consultada do SAP ou se é ordem criada manualmente
dh_atualizacao	Data / Hora de consulta no SAP ou de criação manual
dh_utilizacao	Data / Hora do último palete produzido nesta Ordem
qt_palete	Quantidade de peças a serem inseridas por palete
cd_componente	Código do componente para ordem
de_modelo	Descrição do modelo sendo produzido
de_tensao_frequencia	Tensão / Freqüência do compressor para ordem
de_potencia	Potência do compressor, em watts
de_capacidade_50LBP	Capacidade de refrigeração da unidade, em 50Hz, baixa pressão
de_capacidade_50HBP	Capacidade de refrigeração da unidade em 50 Hz, alta pressão
de_capacidade_60LBP	Capacidade de refrigeração da unidade em 60 Hz, baixa pressão
de_capacidade_60HBP	Capacidade de refrigeração da unidade em 60 Hz, alta pressão
de_refrigerante	Refrigerante utilizado pelo compressor
de_corrente	Corrente utilizada pelo compressor
de_refrigerador_oleo	Indica se o compressor tem refrigerador de óleo
de_fases	Número de fases do compressor

Tabela Paletes:

Item	Descrição
ns_palete	Número de série do palete
cd_produto	Código de produto para este palete
hc_status	Status do Palete: Aberto, Fechado, Transferindo, Transferido
dh_producao	Data / Hora de produção (do fechamento).
cd_linha	Código da linha que produziu o palete
ie_ordem	Identificador da Ordem

Tabela Produtos:

Item	Descrição
cd_produto	Código do produto

ns_componente	Número de série do componente agregado ao produto
cd_componente	Código do componente agregado ao produto
ns_palete	Número de série do palete onde produto foi embalado
dh_montagem	Data / Hora de montagem do componente ao produto
dh_paletizacao	Data / Hora de paletização
ie_ordem	Identificador da Ordem

4.3.2 Diagramas de classes

Foram então definidas as classes para cada módulo desenvolvido.

4.3.2.1 Módulo Paletização

Será apresentado inicialmente um diagrama simplificado, contendo as classes envolvidas no módulo e seus relacionamentos. Em seguida, um detalhamento da responsabilidade de cada uma delas, bem como uma descrição de cada função e propriedade se segue.

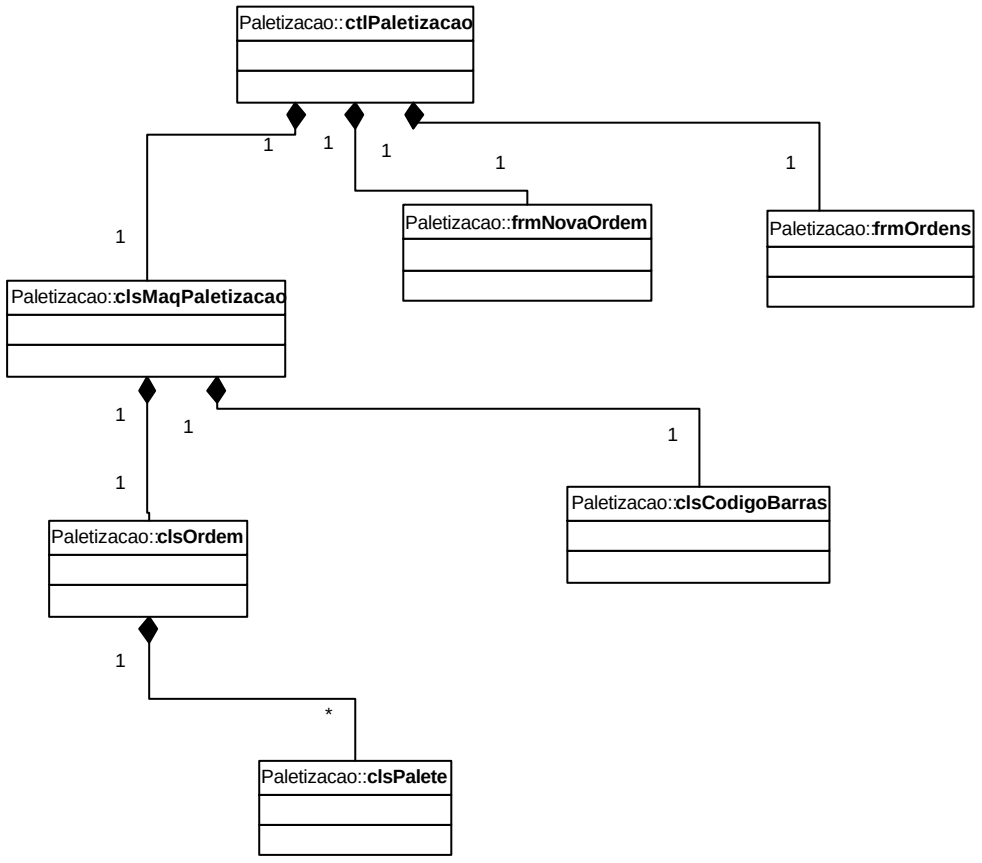

Figura 8: Diagrama de classes simplificado – Módulo Paletização

Classe ctlPaletizacao

Esta classe é a responsável pelo controle ActiveX em si. Sua representação segue:

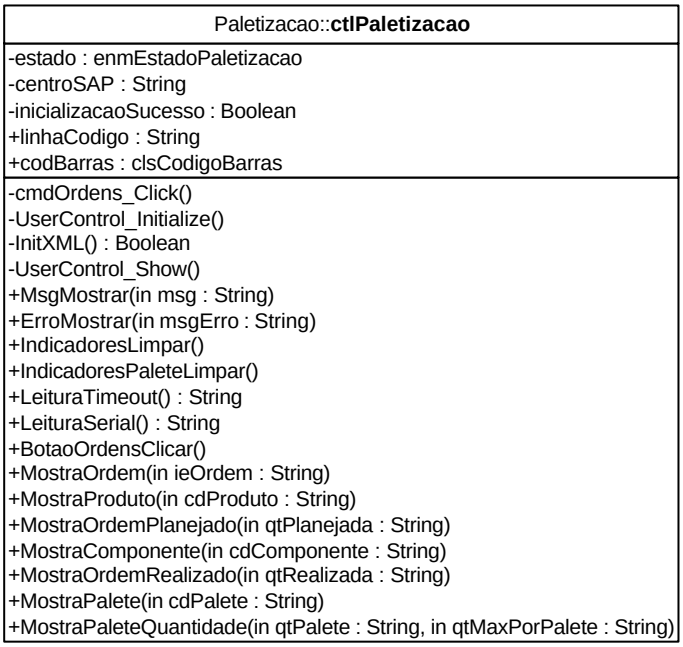

Figura 9: Diagrama de classe – ctlPaletizacao

As propriedades da classe ctlPaletização são:

Atributo	**Descrição**
centroSAP	String para armazenar o centro SAP ao qual pertence a linha de montagem onde se situa o sistema
inicializacaoSucesso	Determina se a inicialização do controle foi efetuada com sucesso ou não
linhaCodigo	Código da linha onde o controle está rodando
codBarras	Objeto que controla a interpretação dos códigos de barras lidos

As funções, seus parâmetros, finalidades e valores de retorno são:

Função: UserControl_Initialize()

Finalidade: Inicializa o controle, suas configurações XML e sua conexão com o banco local

Função: UserControl_Show()

Finalidade: Evento que é disparado quando acaba a carga do controle na tela. Inicializa a máquina de estados, caso tenha ocorrido tudo certo na inicialização do controle.

Função: InitXML()
Finalidade: Lê o arquivo XML principal de configuração e configura controle de leitura da porta serial e configurações do código de barras
Retorna: TRUE, caso tudo se inicialize com sucesso
FALSE, caso contrário

Função: cmdOrdens_Click()
Finalidade: Evento que é disparado quando o botão "Ordens" é clicado. Chama a janela frmOrdens, com o comportamento de "Consulta". Este botão serve para o caso de se desejar fazer apenas uma consulta no SAP, para atualizar as ordens existentes no banco local, por exemplo.

Função: MsgMostrar(msg As String)
Finalidade: Mostra mensagem no label e a escreve na janela de log. Mantém o semáforo verde.
Parâmetros: msg: Mensagem a ser mostrada

Função: ErroMostrar(msgErro As String)
Finalidade: Mostra mensagem de erro no label e a escreve na janela de log. Alterna o semáforo pra vermelho e dorme 2 segundos.
Parâmetros: msgErro: Mensagem a ser mostrada

Função: IndicadoresLimpar()
Finalidade: Limpa todos os indicadores da interface principal (ordem, componente, palete, etc.)

Função: Indicadores PaleteLimpar()
Finalidade: Limpa todos os indicadores da interface relativos ao palete (NS do palete, quantidade)

Função: LeituraTimeout()

Finalidade: Espera leitura da serial até INTERVALO_LEITURA.

Retorna: Leitura em caso de sucesso, vazio em caso de timeout.

Função: LeituraSerial()

Finalidade: Fica aguardando a chegada de uma valor completo (inicio, valor, fim) da serial sem interromper outros processos.

Retorna: O valor lido, assim que este chegar.

Função: MostraOrdem(ieOrdem As String)

Finalidade: Mostra na interface a ordem iniciada.

Parâmetros: ieOrdem: Ordem a ser mostrada

Função: MostraProduto(cdProduto As String)

Finalidade: Mostra na interface o produto lido.

Parâmetros: cdProduto: Código do produto

Função: MostraOrdemPlanejado (qtPlanejada As String)

Finalidade: Mostra na interface a quantidade planejada para a ordem iniciada.

Parâmetros: qtPlanejada: Quantidade planejada para a ordem

Função: MostraComponente (cdComponente As String)

Finalidade: Mostra na interface o código do componente do produto lido.

Parâmetros: cdComponente: Código do componente

Função: MostraOrdemRealizado(qtRealizada As String)

Finalidade: Mostra na interface a quantidade de produtos já paletizados para a ordem.

Parâmetros: qtRealizada: Quantidade já realizada da ordem

Função: MostraPalete(cdPalete As String)

Finalidade: Mostra na interface o código do palete aberto.

Parâmetros: cdPalete: Código do palete

Função: MostraPaleteQuantidade(qtPalete As String, qtMaxPorPalete As String)

Finalidade: Mostra na interface a quantidade já produzida para o palete e o máximo suportado por ele.

Parâmetros: qtPalete: Quantidade já produzida para o palete aberto

qtMaxPorPalete**:** Quantidade máxima para o palete

Classe frmOrdens

Esta classe controla a janela das ordens, responsável por se comunicar com uma página ASP, receber o XML com as ordens vindas do SAP e atualizar o banco local com estas.

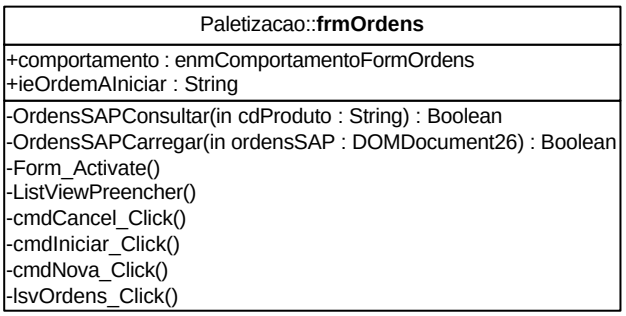

Figura 10: Diagrama de classe – frmOrdens

As propriedades da classe frmOrdens são:

Atributo	**Descrição**
ieOrdemAIniciar	String que armazena a ordem a ser iniciada, escolhida pelo usuário
comportamento	Este é um enumerado que determina o comportamento da janela. Esta janela pode ser chamada em dois contextos: Quando se manda iniciar uma nova ordem (onde o usuário deve escolher uma ordem para ser iniciada), ou quando se quer simplesmente consultar o SAP para atualizar o banco local. Esta variável é definida pelo controle, na hora da chamada do form.

As funções, seus parâmetros, finalidades e valores de retorno são:

Função: OrdensSAPConsultar(cdProduto As String)

Finalidade: Carrega os dados do SAP e os repassa para tratamento local.

Parâmetros: cdProduto: Código do produto a ser pesquisado (necessário para a função do SAP)

Retorna: TRUE, se der tudo certo.
FALSE, caso ocorram problemas na tentativa de solicitar o XML ou este venha vazio.

Função: OrdensSAPCarregar(ordensSAP As DOMDocument26)

Finalidade: Recebe um XML com os dados provenientes do SAP e atualiza o banco local.

Parâmetros: ordensSAP: XML contendo as informações com as ordens do SAP a serem carregadas

Retorna: TRUE, se der tudo certo.
FALSE, caso ocorram problemas.

Função: Form_Activate()

Finalidade: Evento que ocorre toda vez que a form se torna a janela ativa. Tenta consultar as ordens do SAP, e depois carrega os dados do banco local, indicando se houve ou não erro de conexão.

Função: ListViewPreencher()

Finalidade: Consulta ordens no banco de dados local e apresenta numa listagem na janela.

Função: cmdCancel_Click()

Finalidade: Cancela início de ordem.

Função: cmdIniciar_Click()

Finalidade: Coloca em variável global da form o ie_ordem da ordem a ser iniciada pela máquina de estados.

Função: cmdNova_Click()

Finalidade: Abre a janela para entrada de dados de nova ordem manual. Verifica se ordem já não existe na base local e, não existindo, a acrescenta à listagem.

Função: lsvOrdens_Click()

Finalidade: Evento que ocorre no clique de uma ordem na listagem de ordens. Garante que não se permita iniciar uma ordem que já esteja iniciada, habilitando ou não o botão de inicialização.

Classe frmNovaOrdem

Esta classe controla a janela "Nova Ordem", responsável pela entrada de dados manual de uma nova ordem SAP. Esta janela é utilizada, geralmente, caso a conexão com o SAP esteja caída e se deseje trabalhar em uma ordem, cujos valores se têm à mão.

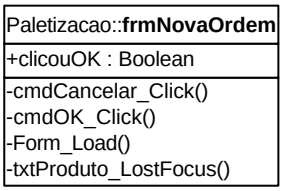

Figura 11: Diagrama de classe – frmNovaOrdem

A classe frmNovaOrdem tem uma propriedade definida, que é:

Atributo	Descrição
clicouOK	Esta variável determina se o usuário saiu da janela clicando no botão OK (ou seja, confirmando a inclusão da nova ordem manual) ou cancelando ou fechando a janela (cancelando, assim, a inclusão).

As funções, seus parâmetros, finalidades e valores de retorno são:

Função: cmdCancelar_Click()

Finalidade: Limpa todos os campos da janela e cancela a inclusão.

Função: cmdOK_Click()

Finalidade: Verifica se algum item não foi preenchido corretamente.Caso tudo esteja correto, fecha a janela e confirma a inclusão.

Função: Form_Load()

Finalidade: Evento que ocorre quando a janela é carregada. Seta o padrão para a variável clicouOK para falso.

Função: txtProduto_LostFocus()

Finalidade: Procura nas ordens locais se existe alguma ordem que descreva as características do produto para o código digitado na caixa de texto txtProduto. Caso exista, já preenche os campos da janela relativos às informações sobre o produto com os valores contidos no banco..

Classe clsMaqPaletizacao

Esta classe é a principal classe do controle, responsável pelo controle e funcionamento da máquina de estados de paletização. O funcionamento de sua principal função, executar(), será vista com um melhor detalhamento no modelo dinâmico. A função principal da classe é, além de controlar a máquina de estados, chamar a validação de entradas e iniciar as transições de estados chamando funções da classe clsOrdem.

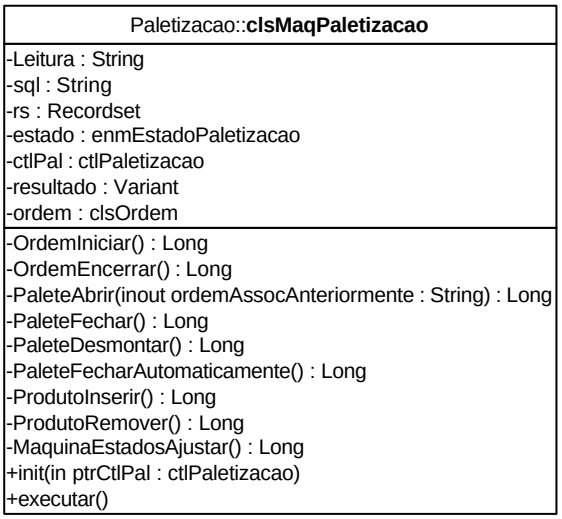

Figura 12: Diagrama de classe – clsMaqPaletizacao

As propriedades da classe clsMaqPaletizacao são:

Atributo	**Descrição**
Leitura	String onde é armazenadas a leitura feita pelo leitor de código de barras
sql	Armazena as instrução SQL
rs	Recordset com as informações provindas do banco de dados
estado	Enumerado que define o estado em que se encontra a máquina de estados. Os valores que podem ser assumidos podem ser vistos no modelo dinâmico
ctlPal	Referência para o controle que está chamando a classe. Necessário para as atualizações da interface
resultado	Armazena o resultado de chamadas de funções
ordem	Objeto da classe ordem que representa a ordem sendo paletizada.

As funções, seus parâmetros, finalidades e valores de retorno são:

Função: OrdemIniciar()

Finalidade: Inicia uma ordem.Carrega a janela de interface, obtém o número da ordem selecionado pelo usuário e manda o objeto ordem iniciar-se.

Retorna: Uma das seguintes constantes:

RESULT_ORDEM_INICIADA

RESULT_ERRO_NENHUMA_ORDEM_SELECIONADA

RESULT_ERRO_ORDEM_INICIAR

Função: Ordem Encerrar()

Finalidade: Manda o objeto ordem encerrar-se.

Retorna: Uma das seguintes constantes:

RESULT_ORDEM_ENCERRADA

RESULT_ERRO_ORDEM_ENCERRAR_GENERICO

RESULT_ERRO_OPERACAO_CANCELADA

RESULT_ERRO_LEITURA_INVALIDA

Função: PaleteAbrir(Optional ByRef ordemAssocAnteriormente As String) As Long

Finalidade: Recebe leitura de palete, valida e manda o objeto ordem abrir o palete.

Parâmetros: ordemAssocAnteriormente: Variável na qual será gravado o número da ordem à qual está associado o palete no banco de dados. Necessário para

que o usuário saiba a qual ordem aquele palete já está associado, caso ocorra este tipo de erro na hora da abertura (Palete associado à outra ordem).

Retorna: Uma das seguintes constantes:
RESULT_PALETE_ABERTO
RESULT_ERRO_PALETE_ORDEM_INCORRETA
RESULT_ERRO_PALETE_TRANSFERENCIA
RESULT_ERRO_PALETE_ABRIR_BANCO
RESULT_ERRO_PALETE_ABRIR_GENERICO
RESULT_ERRO_LEITURA_INVALIDA
RESULT_ERRO_TIMEOUT

Função: PaleteDesmontar()

Finalidade: Desmonta um palete aberto. Solicita confirmação para tanto.

Retorna: Uma das seguintes constantes:
RESULT_PALETE_DESMONTADO
RESULT_ERRO_PALETE_DESMONTAR_GENERICO
RESULT_ERRO_PALETE_DESMONTAR_BANCO
RESULT_ERRO_OPERACAO_CANCELADA
RESULT_ERRO_LEITURA_INVALIDA
RESULT_ERRO_TIMEOUT

Função: PaleteFechar()

Finalidade: Fecha um palete aberto, solicitando confirmação da leitura do código de barras do palete.

Retorna: Uma das seguintes constantes:
RESULT_PALETE_FECHADO
RESULT_ERRO_CONFIRMACAO_PALETE
RESULT_ERRO_PALETE_FECHAR_BANCO
RESULT_ERRO_TIMEOUT
RESULT_ERRO_PALETE_FECHAR_GENERICO

Função: PaleteFecharAutomaticamente()

Finalidade: Fecha um palete aberto, sem solicitar confirmações.

Retorna: Uma das seguintes constantes:

RESULT_PALETE_FECHADO

RESULT_ERRO_PALETE_FECHAR_BANCO

RESULT_ERRO_PALETE_FECHAR_GENERICO

Função: PaleteFechar()

Finalidade: Comanda o fechamento de um palete, solicitando confirmação da leitura do código de barras do palete.

Retorna: Uma das seguintes constantes:

RESULT_PALETE_FECHADO

RESULT_ERRO_CONFIRMACAO_PALETE

RESULT_ERRO_PALETE_FECHAR_BANCO

RESULT_ERRO_TIMEOUT

RESULT_ERRO_PALETE_FECHAR_GENERICO

Função: ProdutoInserir()

Finalidade: Recebe leitura de produto, valida o código lido e manda o objeto ordem inserir um produto no palete aberto.

Retorna: Uma das seguintes constantes:

RESULT_PRODUTO_INSERIDO

RESULT_ERRO_PRODUTO_INEXISTENTE

RESULT_ERRO_PRODUTO_SEM_COMPONENTE

RESULT_ERRO_MONTAGEM_INCORRETA

RESULT_ERRO_PRODUTO_JA_INSERIDO

RESULT_ERRO_PRODUTO_JA_INSERIDO_OUTRO_PALETE

RESULT_ERRO_PRODUTO_INCORRETO

RESULT_ERRO_PALETE_CHEIO

RESULT_ERRO_PRODUTO_INSERIR_BANCO

RESULT_ERRO_PRODUTO_INSERIR_GENERICO

RESULT_ERRO_LEITURA_INVALIDA

Função: ProdutoRemover()

Finalidade: Remove um produto no palete aberto.

Retorna: Uma das seguintes constantes:

RESULT_PRODUTO_REMOVIDO

RESULT_ERRO_PRODUTO_EM_OUTRO_PALETE

RESULT_ERRO_PRODUTO_REMOVER_BANCO

RESULT_ERRO_PRODUTO_REMOVER_GENERICO

RESULT_ERRO_TIMEOUT

Função: MaquinaEstadosAjustar()

Finalidade: Esta função é chamada toda vez que a máquina é iniciada. Ela verifica em que estado foi deixado o sistema, quando desligado/resetado/etc., e ajusta a maquina de estados para o estado correto, preenchendo os objetos Ordem e Palete corretamente e mostrando os valores respectivos no controle.

Retorna: Uma das seguintes constantes:

RESULT_ESTADO_PALETE_ABERTO

RESULT_ESTADO_ORDEM_INICIADA

RESULT_ESTADO_INICIAL

RESULT_ERRO_ORDEM_INICIAR

RESULT_ERRO_PALETE_ABRIR

Classe clsOrdem

Essa classe é a responsável pela gerência das ordens SAP.

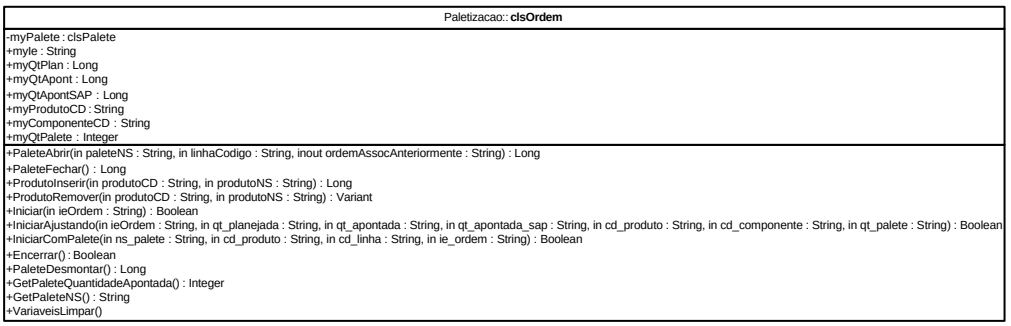

Figura 13: Diagrama de classe – clsOrdem

As propriedades da classe clsOrdem são:

Atributo	Descrição
myIe	Número da ordem
myQtPlan	Quantidade planejada para aquela ordem
myQtApont	Quantidade já apontada para a ordem
myQtApontSAP	Quantidade já apontada para a ordem, vinda do SAP
myProdutoCD	Código do produto para a ordem
myComponenteCD	Código do componente para a ordem
myQtPalete	Quantidade de produtos por palete

Função: PaleteAbrir(paleteNS As String, linhaCodigo As String, Optional ByRef ordemAssocAnteriormente As String)

Finalidade: Manda o objeto palete abrir-se.

Parâmetros: nsPalete: Número serial do palete a ser aberto

LinhaCodigo: Código da linha onde o palete será preenchido

ordemAssocAnteriormente - Ordem à qual está associado o palete no banco de dados. Necessário para que o usuário saiba a qual ordem aquele palete está associado, caso não seja à ordem que está atualmente aberta.

Retorna: O resultado da função abrir() do palete

Função: PaleteFechar()

Finalidade: Manda o objeto palete fechar-se.

Retorna: O resultado da funcao fechar() do palete

Função: ProdutoInserir(produtoCD As String, produtoNS As String)

Finalidade: Manda o objeto palete inserir um produto.

Parâmetros: produtoCD: Código do produto a ser inserido

ProdutoNS: Número serial do produto a ser inserido

Retorna: O resultado da função produtoInserir() do palete

Função: ProdutoRemover(produtoCD As String, produtoNS As String)

Finalidade: Manda o objeto palete remover um produto.

Parâmetros: produtoCD: Código do produto a ser removido

ProdutoNS: Número serial do produto a ser removido

Retorna: O resultado da função produtoRemover() do palete

Função: Iniciar(ieOrdem As String)

Finalidade: Marca uma ordem como iniciada no banco local.

Parâmetros: ieOrdem: O número da ordem a ser iniciada

Retorna: TRUE se deu tudo certo
FALSE, caso contrário

Função: IniciarAjustando(ieOrdem As String, qt_planejada As String, qt_apontada As String, qt_apontada_sap As String, cd_produto As String, cd_componente As String, qt_palete As String)

Finalidade: Inicia o objeto Ordem com os valores passados como parâmetro Essa funçao deve ser usada quando uma ordem foi deixada como ‘iniciada' no banco, sem ter sido encerrada.

Parâmetros: ie_ordem: número da ordem
qt_planejada: quantidade planejada
qt_apontada: quantidade apontada
qt_apontada_sap: quantidade apontada pelo sap
cd_produto: código do produto
cd_componente: código do componente
qt_palete: quantidade máxima de itens no palete

Retorna: TRUE se deu tudo certo
FALSE, caso contrário

Função: IniciarComPalete(ns_palete As String, cd_produto As String, cd_linha As String, ie_ordem As String)

Finalidade: Preenche os atributos do objeto Ordem retirando do banco os valores de uma ordem previamente iniciada. Preenche os atributos do objeto Palete de acordo com os valores fornecidos.

Parâmetros: ns_palete: Número serial do palete
cd_produto: Codigo do produto
cd_linha: Codigo da linha
ie_ordem: Nr da ordem

Retorna: TRUE se deu tudo certo
FALSE, caso contrário

Função: Encerrar()
Finalidade: Marca uma ordem como encerrada no banco local.
Retorna: TRUE se deu tudo certo
FALSE, caso contrário

Função: PaleteDesmontar()
Finalidade: Manda o objeto palete desmontar-se.
Retorna: O resultado da funcao desmontar() do palete.

Função: GetPaleteQuantidadeApontada()
Finalidade: Retorna a quantidade apontada do palete atualmente aberto pertencente à ordem.
Retorna: Inteiro com a quantidade apontada do palete aberto

Função: GetPaleteNS()
Finalidade: Retorna o número de série do palete atualmente aberto
Retorna: String com o número de série

Função: VariaveisLimpar()
Finalidade: Limpa as variáveis da ordem

Classe clsPalete

Essa é a classe responsável pelo controle do palete.

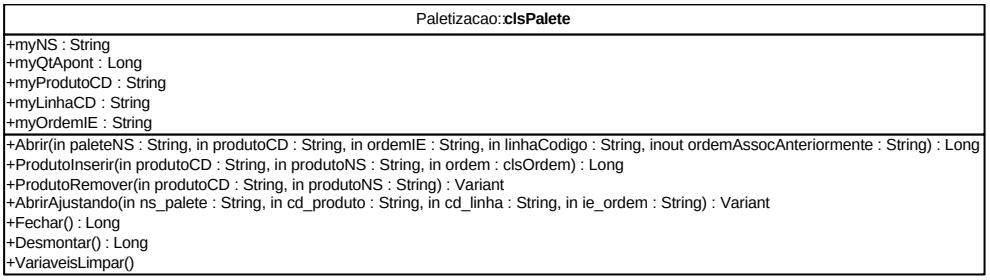

Figura 14: Diagrama de classe – clsPalete

Função: Abrir(paleteNS As String, produtoCD As String, ordemIE As String, linhaCodigo As String, Optional ByRef ordemAssocAnteriormente As String)

Finalidade: Faz as verificações no banco e abre ou reabre o palete, se tudo estiver certo

Parâmetros: paleteNS: Numero serial do palete a ser aberto
produtoCD: Codigo do produto que deve ser colocado no palete
ordemIE: A ordem para a qual o palete esta sendo preenchido
linhaCodigo: Codigo da linha onde o palete sera preenchido
ordemAssocAnteriormente: Ordem à qual está associado o palete no banco de dados. Necessário para que o usuário saiba a qual ordem aquele palete está associado, caso nao seja à ordem que está atualmente aberta.

Retorna: Uma das seguintes constantes:
RESULT_PALETE_ABERTO
RESULT_PALETE_ABERTO_CRIADO_NOVO
RESULT_ERRO_PALETE_ORDEM_INCORRETA
RESULT_ERRO_PALETE_TRANSFERENCIA
RESULT_ERRO_PALETE_ABRIR_GENERICO
RESULT_ERRO_PALETE_ABRIR_BANCO

Função: ProdutoInserir(produtoCD As String, produtoNS As String, ordem As clsOrdem)

Finalidade: Insere produto no palete. Faz todas as verificações necessárias no banco de dados.

Parâmetros: produtoNS - Número serial do produto a ser inserido
produtoCD - Código do produto a ser inserido

ordem - Ordem para o produto a ser inserido.

Retorna: Uma das seguintes constantes:

RESULT_PRODUTO_INSERIDO

RESULT_ERRO_PRODUTO_INEXISTENTE

RESULT_ERRO_PRODUTO_SEM_COMPONENTE

RESULT_ERRO_MONTAGEM_INCORRETA

RESULT_ERRO_PRODUTO_JA_INSERIDO

RESULT_ERRO_PRODUTO_JA_INSERIDO_OUTRO_PALETE

RESULT_ERRO_PRODUTO_INCORRETO

RESULT_ERRO_PALETE_CHEIO

RESULT_ERRO_PRODUTO_INSERIR_GENERICO

RESULT_ERRO_PRODUTO_INSERIR_BANCO

Função: ProdutoRemover(produtoCD As String, produtoNS As String)

Finalidade: Remove produto do palete.

Parâmetros: produtoNS - Número serial do produto a ser removido

produtoCD - Código do produto a ser removido

Retorna: Uma das seguintes constantes:

RESULT_PRODUTO_REMOVIDO

RESULT_ERRO_PRODUTO_EM_OUTRO_PALETE

RESULT_ERRO_PRODUTO_REMOVER_BANCO

RESULT_ERRO_PRODUTO_REMOVER_GENERICO

Função: AbrirAjustando(ns_palete As String, cd_produto As String, cd_linha As String, ie_ordem As String)

Finalidade: Inicia o objeto Palete com os valores retirados do banco, já fornecidos pelos parâmetros. Calcula também a quantidade apontada para o palete, verificando o número de produtos que estão associados com ele.

Essa função deve ser usada quando um palete foi deixada como ‘aberto' no banco, sem ter sido fechado.

Parâmetros: ns_palete: número serial do palete

cd_produto: código do produto
cd_linha: código da linha
ie_ordem: número da ordem

Retorna: TRUE se deu tudo certo
FALSE, caso contrário

Função: Fechar()
Finalidade: Fecha o palete
Retorna: Uma das seguintes constantes:
RESULT_PALETE_FECHADO
RESULT_ERRO_PALETE_FECHAR_GENERICO
RESULT_ERRO_PALETE_FECHAR_BANCO

Função: Desmontar()
Finalidade: Desmonta o palete. Exclui do banco local seus dados e desassocia os produtos a ele previamente associados.
Retorna: Uma das seguintes constantes:
RESULT_PALETE_DESMONTADO
RESULT_ERRO_PALETE_DESMONTAR_GENERICO
RESULT_ERRO_PALETE_DESMONTAR_BANCO

Função: VariaveisLimpar()
Finalidade: Limpa as variáveis do objeto palete

4.3.2.2 Módulo Genealogia

Um diagrama simplificado do controle genealogia apresenta-se abaixo. Em seguida, serão apresentados os diagramas de cada classe em particular. A classe clsCodigoBarras, também utilizada no módulo paletização, faz parte de classes de uso comum entre os módulos, e será descrito no item 3.3.2.4.

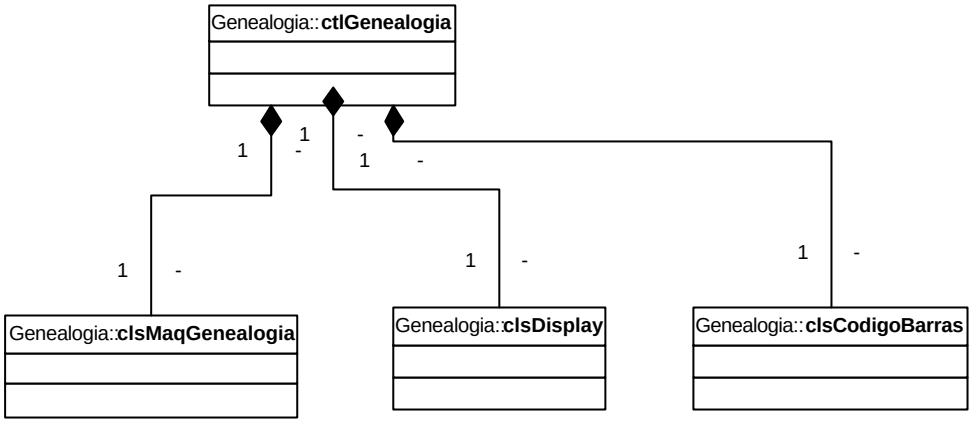

Figura 15: Diagrama de classes Módulo Genealogia simplificado

Classe ctlGenealogia

Essa é a classe responsável pelo controle em si. Sua representação UML se segue:

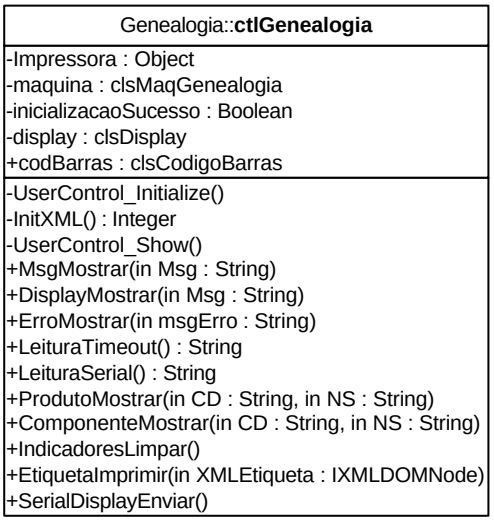

Figura 16: Diagrama de classe – ctlGenealogia

As propriedades da classe ctlGenealogia são:

Atributo	**Descrição**
Impressora	Referência para o controle de impressão, cujos métodos serão chamados para impressão de etiquetas
maquina	Máquina de estados do módulo de genealogia
inicializacaoSucesso	Indica se a inicialização das configurações do controle deu-se com sucesso
display	Controlador dos dados enviados para o display digital
codBarras	Código do produto para a ordem

Função: UserControl_Initialize()

Finalidade: Inicialização do controle. Inicializa máquina de estados, configura controle de leitura e inicia o timer.

Função: InitXML()

Finalidade: Lê o arquivo XML principal de configuração e configura controle de leitura da porta serial e configurações do código de barras

Retorna: Uma das seguintes constantes:
RESULT_INICIALIZACAO_XML_OK
RESULT_ERRO_CARGA_XML
RESULT_ERRO_INICIALIZACAO_DISPLAY
RESULT_ERRO_INICIALIZACAO_LEITOR
RESULT_ERRO_INICIALIZACAO_IMPRESSORA

Função: UserControl_Show()

Finalidade: Inicializa maquina de estados, caso tenha ocorrido tudo certo na inicialização do controle

Função: MsgMostrar(msg As String)

Finalidade: Mostra mensagem no label e a escreve na janela de log. Mantém o semáforo verde.

Parâmetros: msg: Mensagem a ser mostrada

Função: DisplayMostrar(msg As String)

Finalidade: Mostra mensagem no display digital. Corta mensagem para 20 caracteres (máximo que o display suporta para uma das linhas).

Parâmetros: msg: Mensagem a ser mostrada

Função: ErroMostrar(msgErro As String)

Finalidade: Mostra mensagem de erro no label e a escreve na janela de log. Alterna o semáforo pra vermelho e dorme 2 segundos.

Parâmetros: msgErro: Mensagem a ser mostrada

Função: ProdutoMostrar(CD As String, NS As String)

Finalidade: Mostra na janela do controle o código e o numero de serie do produto.)

Parâmetros: CD - Código do produto

NS - Numero de serie do produto

Função: ComponenteMostrar(CD As String, NS As String)

Finalidade: Mostra na janela do controle o código e o numero de serie do componente.)

Parâmetros: CD - Código do componente

NS - Numero de serie do componente

Função: LeituraTimeout()

Finalidade: Espera leitura da serial até INTERVALO_LEITURA.

Retorna: Leitura em caso de sucesso, vazio em caso de timeout.

Função: LeituraSerial()

Finalidade: Fica aguardando a chegada de uma valor completo (inicio, valor, fim) da serial sem interromper outros processos.

Retorna: O valor lido, assim que este chegar.

Função: IndicadoresLimpar()

Finalidade: Limpa todos os indicadores da interface principal (produto, componente, mensagem, etc.)

Função: EtiquetaImprimir(XMLEtiqueta As IXMLDOMNode)

Finalidade: Envia para impressora comando para imprimir etiqueta

Parâmetros: XMLEtiqueta: XML contendo informações da etiqueta a ser impressa

Classe ctlDisplay

Esta classe é a responsável pelo correto envio de mensagens para o Display Digital. Os dados em si são enviados pelo ActiveX de controle serial, mas esta classe dá a

formatação necessária para o display (caracteres de controle de posicionamento de cursor, limpeza de linha, etc.) O diagrama UML de classe assim se apresenta:

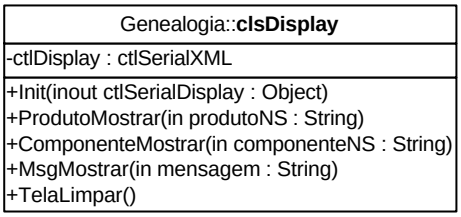

Figura 17: Diagrama de classe – clsDisplay

A classe display como atributos:

Atributo	**Descrição**
ctlDisplay	Referência para o controle de porta serial, através do qual serão enviados os comandos para a apresentação no display

Função: ProdutoMostrar(CD As String, NS As String)

Finalidade: Mostra, no display, informações sobre o produto lido, na posição superior esquerda.

Parâmetros: CD - Código do produto

NS - Numero de serie do produto

Função: ComponenteMostrar(CD As String, NS As String)

Finalidade: Mostra, no display, o serial do componente lido, na posição superior direita.

Parâmetros: CD - Código do componente

NS - Numero de serie do componente

Função: MsgMostrar(mensagem As String)

Finalidade: Mostra, no display, uma mensagem na segunda linha.

Parâmetros: mensagem: Mensagem a ser mostrada

Função: TelaLimpar()

Finalidade: Limpa a tela do display

Classe clsMaqGenealogia

Esta é a principal classe do módulo de genealogia, responsável pelo controle da máquina de estados do mesmo. O funcionamento detalhado da máquina em si (função executar) será descrito no diagrama dinâmico.

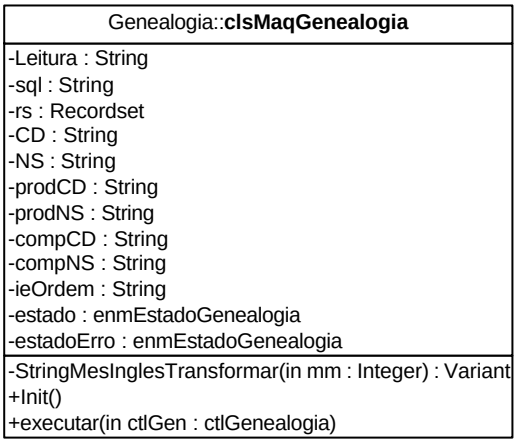

Figura 18: Diagrama de classe – clsMaqGenealogia

Atributo	**Descrição**
leitura	String que armazena o valor da leitura da serial
sql	Armazena instrução SQL
rs	Armazena os resultados dos queries SQL
CD	Código lido pelo leitor de código de barras, genérico
NS	Numero serial lido pelo leitor, genérico
prodCD	Código do produto
prodNS	Número serial do produto
compCD	Código do componente
compNS	Número serial do componente
ieOrdem	Número da ordem do produto sendo montado, de onde serão tiradas as informações para a impressão da etiqueta
estado	Define o estado da maquina de estados
estadoErro	Armazena o estado anterior a um erro do tipo genérico

Função: StringMesInglesTransformar(mm As Integer)

Finalidade: Converte um mês no formato numérico para o formato mmm em inglês. Esta função é necessária pois a representação do mês, nas etiquetas, é escrita em inglês.

Parâmetros: mm - Mês, representado por um número de 01 a 12

Retorna: - "", se o número for inválido

- string com representação "mmm" do número, em inglês, caso seja um número válido

Função: Init()

Finalidade: Inicializa a maquina de estados.

Função: executar(ctlGen As ctlGenealogia)

Finalidade: Método principal da maquina de estados, que é executado ininterruptamente e gerencia as chamadas de leituras, display e transições entre estados.

Parâmetros: ctlGen**:** Uma referencia ao controle do ctlGenealogia que instanciou a máquina.

4.3.2.3 Módulo Impressão

Este é o módulo responsável pela impressão das etiquetas e visualização do status da impressora.

O diagrama de classe do módulo assim se apresenta:

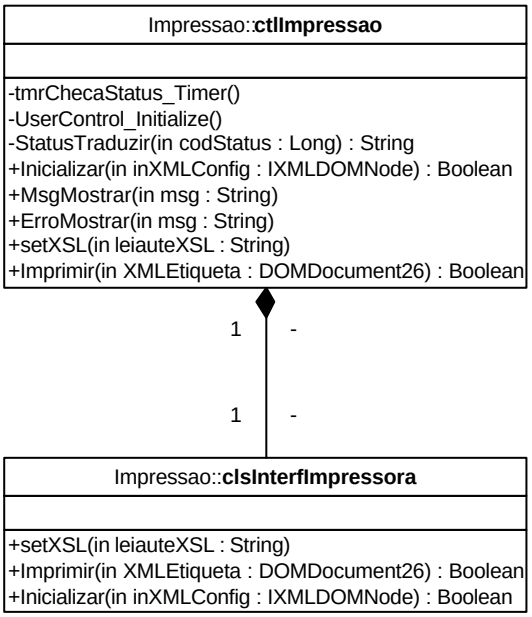

Figura 19: Diagrama de classes Módulo Impressão

Classe ctlImpressao

Este é o controle propriamente dito. A descrição de suas funções e funcionalidades se segue:

Função: Inicializar(inXMLConfig As IXMLDOMNode)
Finalidade: Inicializa o driver da impressora
Parâmetros: inXMLConfig - nó XML contendo as configurações.
Tags esperados no XML:
PORTA_SERIAL – indicando a porta serial
BAUD_RATE – indicando a baud rate
PARITY – indicando a paridade
DATA_BITS – indicando os bits de dados
STOP_BITS – indicando os stop bits
Retorna: - TRUE se tudo deu certo
- FALSE caso contrário (XML com erro, driver não conseguindo inicializar)

Função: MsgMostrar(msg As String)
Finalidade: Mostra mensagem na tela principal do controle, em preto
Parâmetros: msg: Mensagem a ser mostrada

Função: ErroMostrar(msgErro As String)
Finalidade: Mostra mensagem na tela principal do controle, em vermelho.
Parâmetros: msgErro: Mensagem a ser mostrada

Função: setXSL(leiauteXSL As String)
Finalidade: Configura no driver a formatação da etiqueta a ser impressa
Parâmetros: leiauteXSL - XSL com a formatacao da etiqueta

Função: Imprimir(XMLEtiqueta As DOMDocument26)
Finalidade: Envia o comando de impressão para o driver da impressora
Parâmetros: XMLEtiqueta - Um XML contendo a etiqueta a ser impressa

Retorna: - TRUE se conseguiu enviar a impressao para o driver

- FALSE caso nao tenha conseguido inicializar o driver (porta aberta, por exemplo)

Classe clsInterImpressora

Esta classe é simplesmente a interface do controle para outros controles (no caso, para o controle ctlGenealogia). Ele repassa as requisições para o controle ctlImpressão.

4.3.2.3 Driver da Impressora

O controle de impressão faz uso de um driver para comunicar-se com a impressora. O driver, também implementado em ActiveX, foi modelado com uma única classe, principal, a ctlZebraS400Driver.

Classe ctlZebraS400Driver

O diagrama UML da classe pode ser visto a seguir:

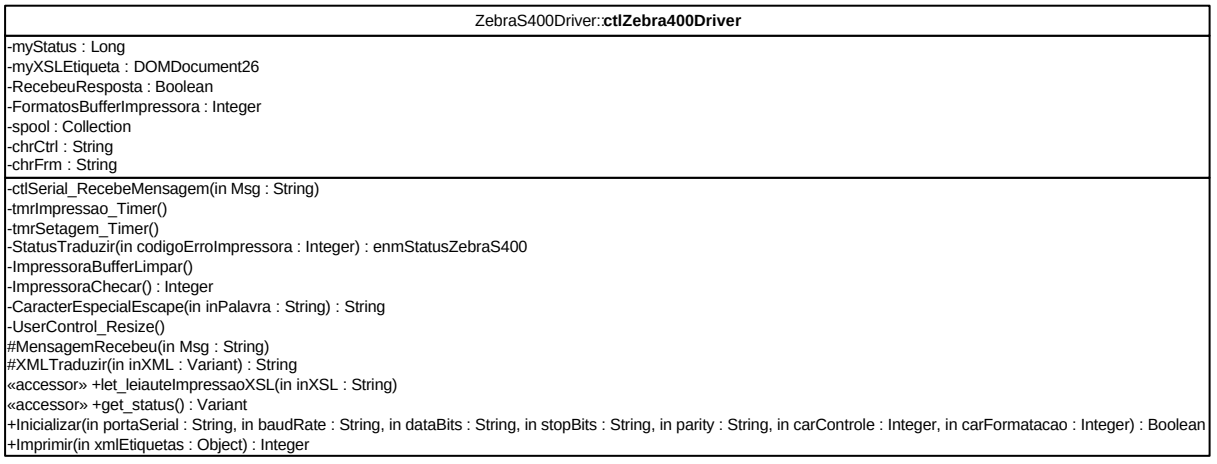

Figura 20: Diagrama de classe – Driver Impressora

A classe ctlZebraS400Driver apresenta os seguintes atributos:

Atributo	Descrição
myStatus	Armazena o código de status retornado pela impressora
myXSLEtiqueta	Armazena o leiaute de impressão, em XSL

RecebeuResposta	Indica se a informação completa sobre o status da impressora já foi recebida pela porta serial
FormatosBufferImpressora	Indica quantos formatos estão presentes no buffer da impressora
spool	Coleção que implementa uma fila de impressão; aqui ficam armazenadas as etiquetas, já formatadas, que aguardam impressão
chrCtrl	Armazena qual caracter será utilizado como caracter de controle
chrFrm	Armazena qual caracter será utilizado como caracter de formatação
leiauteImpressaoXSL	Propriedade que deve ser definida com o leiaute, em formato XSL, que deve ser aplicado sob o XML para fazer a transformação deste na linguagem ZPL a ser enviada para a impressora
status	Propriedade somente leitura que fornece o atual status da impressora

As funções do driver de impressão e sua descrição seguem:

Função: ctlSerial_RecebeMensagem(ByVal Msg As String)

Finalidade: Evento que ocorre quando o controle serial recebe uma string. Chama a função MensagemRecebeu

Parâmetros: Msg – string contendo a mensagem recebida pelo controle serial

Função: tmrImpressao_Timer()

Finalidade: Timer que, de tempos em tempos, checa se há algo para ser impresso no spool. Caso exista, checa status da impressora pra ver se ela está apta a imprimir. Sendo este o caso, comanda a impressão e retira o item impresso do spool.

Função: tmrSetagem_Timer()

Finalidade: Testa o status da impressora e a inicializa. Caso não consiga, tenta novamente.

Função: StatusTraduzir(codigoErroImpressora As Integer)

Finalidade: Pega os códigos de retorno da impressora e os converte para status em um padrão

Parâmetros: codigoErroImpressora - Códigos da impressora especifica do driver. Neste caso, sao os retornos possíveis da funcao ImpressoraChecar

Retorna: Enumerado contendo o status da impressora. Todos os valores de erro sao negativos

Função: ImpressoraBufferLimpar()

Finalidade: Limpa o buffer da impressora

Função: ImpressoraChecar()

Finalidade: Verifica status da impressora, enviando comando '~HS' e aguardando resposta pela serial até TimeoutResposta

Retorna: número de formatos no buffer ou código de erro:

-1: sem papel

-2: buffer lotado

-3: formato etiqueta incompleto

-4: memória RAM corrompida

-5: Temperatura baixa

-6: temperatura alta

-7: head up

-8: ribbon out

-96: erro ao verificar formatos no buffer

-97: erro ao decodificar a resposta da impressora

-98: sem resposta

-99: erro geral ao checar status

Função: CaracterEspecialEscape(inPalavra As String)

Finalidade: Substituir caracteres que não podem ser impressos na impressora Zebra S400 por caracteres que podem

Parâmetros: inPalavra - string a ser substituída

Retorna: string que pode ser impressa pela impressora

Função: UserControl_Resize()

Finalidade: Ajusta o tamanho do user control para o mesmo tamanho da figura

Função: XMLTraduzir(inXML)

Finalidade: Traduz escapes de XML. Caso encontre um escape "\\a" + um número de caracter, substitui pelo caracter definido

Parâmetros: inXML – XML a ser traduzido

Retorna: String com XML traduzido

Função: Inicializar (portaSerial As String, baudRate As String, dataBits As String, stopBits As String, parity As String, carControle As Integer, carFormatacao As Integer)

Finalidade: Inicializa o controle, abre a porta serial, liga o timer para configurar a impressora, se tudo certo habilita a impressão. Caso a porta serial nao consiga ser aberta, seta o status de erro e sai da função.

Parâmetros: portaSerial: Nr da porta serial da impressora
baudRate: baudRate a ser utilizada. Ex: 9600
dataBits: Ex: 8
stopBits: Ex: 1
parity: Paridade (N, E)
carControle: Código ASCII do caracter a ser utilizado para controle (que é, normalmente, ~)
carFormatacao: Codigo ASCII do caracter a ser utilizado para formatacao (normalmente, ^)

Retorna: TRUE se conseguiu abrir a porta. Não garante que a impressora esteja on-line, pronta pra imprimir
FALSE caso nao tenha conseguido abrir a porta.

Função: Imprimir(xmlEtiquetas As Object)

Finalidade: Recebe um XML com uma ou mais etiquetas, aplica a formatação XSL definida na variavel leiauteXSL e joga no spool para impressão.

Parâmetros: xmlEtiquetas - XML contendo informações da(s) etiqueta(s) a ser(em) impressa(s)

Retorna: Uma das seguintes constantes:

RESULT_ENVIO_IMPRESSAO_OK

RESULT_ERRO_XML

RESULT_ERRO_XSL

4.3.2.4 – Classes de uso comum

Classe clsCodigoBarras

Esta classe, utilizada pelos módulos genealogia e paletização, é a responsável pela validação dos códigos de barras lidos referentes a produtos, componentes e paletes. Ela se baseia num nó específico do arquivo XML de configurações, o <CODIGO-BARRAS>.

O nó "<CODIGO-BARRAS>" deve conter um ou mais nós "<CONFIG>", com um atributo "LEN" que especifica o tamanho (em números de caracteres) aceito por aquela configuração. Dentro de cada configuração, podemos ter dois tipos de separações possíveis: Por posições fixas ou por caracter separador.

Quando a configuração é por posições fixas, devem ser especificados a posição em que se encontram o código e o número serial dentro do código "bruto", para que a classe possa interpretar este código corretamente e devolver os valores de código e número serial para os objetos que o requisitarem.

Mas a configuração também pode ser feita pelo uso de um caracter separador; neste modo, o caracter deve ser especificado no XML, bem como em que "campos" estarão as informações relativas ao código e ao numero serial desejado.

Um exemplo de arquivo de configuração XML pode ser visto no apêndice II.

A representação UML da classe pode ser vista a seguir:

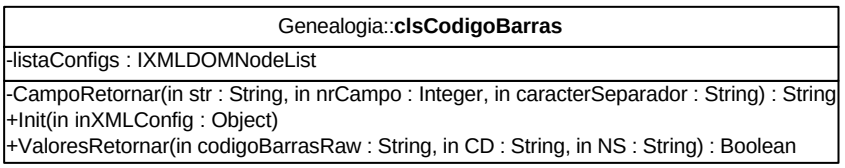

Figura 21: Diagrama de classe – clsCodigoBarras

As funções e sua descrição seguem:

Função: CampoRetornar(str As String, nrCampo As Integer, caracterSeparador As String)

Finalidade: Descobre, em uma determinada string contendo campos separados por um caracter separador, o campo de número especificado e retorna o seu valor.

Pârametros: str - A string com os campos e caracteres separadores

nrCampo - Número do campo do qual deseja-se obter o valor, partindo de 1.

Ex: 1 2 3

1000001#300#7543

caracterSeparador - Define o caractere usado para separar os campos.

Retorna: - "" caso o nr do campo desejado nao exista na string fornecida

- O valor do campo, caso este seja encontrado com sucesso

Função: Init (inXMLConfig As Object)

Finalidade: Inicializa o objeto, pegando do objeto XML os tipos de valores que devem ser aceitos e salvando-os na lista de configurações possíveis listaConfigs

Pârametros: inXMLConfig – XML contendo as configurações permitidas

Função: ValoresRetornar(codigoBarrasRaw As String, CD As String, NS As String)

Finalidade: Lê do arquivo XML as configurações possíveis de posições de NS e CD no código de barras, e descobre, baseado no tamanho do código e na presença ou não de unidade separadora, de qual configuração aquele código se trata. Preenche NS e CD baseado nessas posições.

Pârametros: codigoBarrasRaw - O valor bruto, completo, do codigo de barras

CD - É preenchido com o código do produto/componente

NS - É preenchido com o número serial do produto/componente

Retorna: - TRUE se achou a configuração do código e ele estava correto de acordo com essa configuração

- FALSE caso não se tenha achado uma configuração valida no XML de configuração, ou o código não esteja de acordo com a configuração.

4.4 MODELO DINÂMICO

4.4.1 Módulo Genealogia

4.4.1.1 Funcionamento:

Descreveremos a seguir o funcionamento dinâmico do sistema para as duas principais operações do módulo.

Montando Produto

Pré-requisitos: (nenhum)
Operação Normal:

a. Sistema mostra mensagem "Leia componente" na tela e display e aguarda leitura
b. Operador deverá ler componente
c. Sistema mostrará na tela e display número de série do componente lido e solicitará leitura do produto: "Leia produto"
d. Operador deverá ler produto
e. Sistema verifica na tabela local de cadastro de produto se montagem está correta
f. Sistema comanda impressão de etiqueta de caixa individual
g. Sistema registra associação de componente x produto e apresenta mensagem "Produto montado"

Exceção 1: Operador lê etiqueta do produto antes da etiqueta do componente

a. Sistema identifica que etiqueta lida é de produto, através do código do produto (dado disponível na ordem)
b. Sistema apresentará na tela código do produto lido e solicitará a leitura do componente

Exceção 2: Associação de componente x produto incorreta

a. Mensagem de erro "Montagem incorreta"

Exceção 3: Não existe cadastro local de montagem de produto x componente (acontecerá quando a ordem SAP do produto não foi carregada localmente)

a. Operador deverá reconsultar ordens SAP
b. Se ordem SAP para tal produto não estiver liberada, operador deverá criar uma ordem manualmente, com os dados de genealogia e impressão de etiquetas.

Exceção 4: Componente já está montado em outro produto (no mesmo palete)

a. Mensagem de erro, operador deverá desmontar outro produto

Exceção 5: Produto já está montado com outro componente

a. Sistema solicita confirmação de desmontagem do componente anterior
b. Sistema exclui registro anterior de montagem

Desmontando Produto

Pré-requisito: Não deve haver produto sendo montado

Operação Normal:

a. Operador lê comando "Desmontar produto" com pistola
b. Sistema solicita leitura da etiqueta do produto
c. Sistema apaga do banco de dados local dados de montagem do produto

Exceção 1: Produto não existe

a. Mensagem de erro

Exceção 2: Cancelamento / Timeout de leitura

a. Mensagem de cancelamento / timeout

4.4.1.2 Estados:

O diagrama de estados ilustra o funcionamento da máquina de estados:

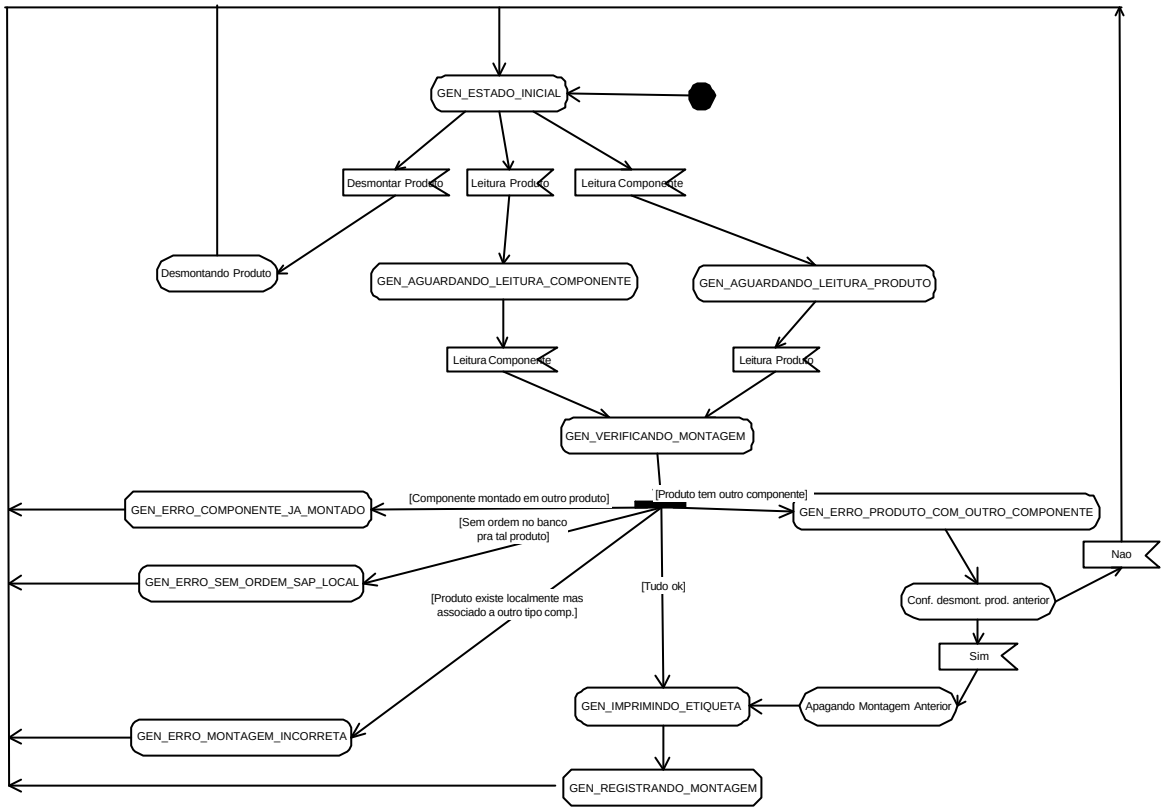

Figura 22: Diagrama de estados – Módulo Genealogia

A seguir, será descrito o que significa cada estado:

GEN_ESTADO_INICIAL

É o único “Estado estável” do sistema. Daqui o controle aguarda uma entrada por parte do usuário, através da leitura de algum código de barras. As entradas podem ser o comando Desmontar Produto ou um código de algum produto ou componente.

GEN_AGUARDANDO_LEITURA_COMPONENTE

É o estado em que o sistema entra após detectar que a leitura trata-se de um componente. Solicita a leitura do produto.

GEN_AGUARDANDO_LEITURA_PRODUTO

É o estado em que o sistema entra após detectar que a leitura refere-se a um produto. Solicita a leitura do componente.

GEN_VERIFICANDO_MONTAGEM

Após a correta leitura do produto e do componente, é o estado que o sistema verifica se a montagem está correta, de acordo com as ordens existentes e as informações no banco.

GEN_IMPRIMINDO_ETIQUETA

Depois de verificada a correção na montagem, é o estado onde o sistema solicita, do banco, as informações relativas ao produto sendo montado, gera um XML com estas informações e comanda a impressão da etiqueta para o controle de impressão.

GEN_REGISTRANDO_MONTAGEM

É o estado onde o sistema registra, no banco local, a associação componente X produto montada.

GEN_ERRO_PRODUTO_COM_OUTRO_COMPONENTE

É um estado de erro quando o produto lido já está cadastrado no banco local. Neste caso, o sistema solicita confirmação para a desmontagem do produto anterior e, dependendo da resposta desta, pode voltar ao estado inicial ou desmontar o produto anterior e se encaminhar para o próximo estado.

GEN_ERRO_COMPONENTE_JA_MONTADO

É o estado de erro quando o componente sido lido já foi montado, ou seja, já está associado a outro produto no banco local. Neste caso, o sistema volta para o estado inicial.

GEN_ERRO_SEM_ORDEM_SAP_LOCAL

Estado de erro quando não é localizada uma ordem de produção para aquele tipo de produto e componente. Volta ao estado inicial.

GEN_ERRO_MONTAGEM_INCORRETA

Ocorre quando a montagem é verificada como incorreta. Existe ordem para aquele tipo de produto, mas foi lido um componente incorreto. Volta para o estado inicial.

4.4.2 Módulo Paletização

4.4.2.1 – Funcionamento

Iniciando Ordem

Pré-requisitos:

a. não deve haver ordem ou palete iniciados

Operação Normal:

a. Operador lê comando "Iniciar Ordem" com pistola
b. Sistema faz chamada http para Web Server, solicitando dados de ordens liberadas e que ainda não tenham confirmação final
c. Servidor devolve XML com dados das ordens
d. Lista de ordens é apresentada para operador, que escolhe com as setas do teclado
e. Dados de todas as ordens disponíveis no SAP são registrados localmente (eventualmente podem ser necessários para estação de Genealogia imprimir etiqueta, ou para que usuário selecione ordem caso não haja conexão com SAP no momento)
f. Ordem selecionada é marcada localmente como iniciada e seus dados são apresentados na tela do aplicativo

Exceção 1: Servidor SAP não está disponível

a. Na janela de escolha das ordens será apresentada mensagem de falha na conexão
b. Operador poderá optar entre escolher uma ordem que já havia sido registrada localmente ou criar nova ordem, entrando manualmente com todos os dados necessários
c. Ordem será iniciada localmente

Exceção 2: Cancelamento da operação

a. Nenhuma ordem é iniciada

Consulta de Ordens

Haverá casos onde dados do produto não estão disponíveis para que estação de Genealogia comande impressão (quando ordem que está sendo montada ainda não foi registrada localmente).

O operador terá opção de carregar do SAP os dados de ordens, clicando no botão "F12 – Ordens". Será mostrada tela similar ao de início de ordem, com os dados já carregados localmente. O operador terá ainda a opção de criar uma nova ordem, digitando os dados.

Abertura de palete

Pré-requisitos:

a. Deve haver ordem aberta, nenhum outro palete deve estar aberto

Operação Normal:

a. Operador lê comando "Abrir Palete" com scanner
b. Sistema solicita leitura da etiqueta do palete
c. Se não existe registro local do palete, novo palete (vazio) é criado

Exceção 1: Palete já foi fechado e ainda não foi transferido

a. Verifica se palete fechado pertence à ordem que está aberta
b. "Reabre" palete, recarregando do banco local dados do palete já fechado
c. Mostra ao operador dados do palete reaberto

Exceção 2: Palete já foi fechado e transferido para SAP

a. Será verificado no banco de dados local se palete já foi produzido nos últimos 2 dias (configurável).
b. Em caso afirmativo, sistema não permitirá abertura do palete

Exceção 3: Cancelamento ou Timeout de leitura de palete

a. Mensagem de cancelamento / timeout, palete não é aberto

Inserindo produto no palete

Pré-Requisitos:

a. Palete deve estar aberto

Operação normal:

a. Operador lê etiqueta do produto (identificado pelo número de caracteres)

b. Código de material e número de série do produto são armazenados no banco de dados local

c. Nova quantidade no palete e na Ordem são apresentados para o operador

d. Se quantidade máxima do palete foi atingida, sistema dispara fechamento do palete automaticamente

Exceção 1: Produto não tem componente agregado (genealogia)

a. Mensagem de erro, produto não é inserido

Exceção 2: Produto já está no palete

a. Mensagem de erro, produto não é inserido

Exceção 3: Quantidade máxima de produtos no palete já foi atingida

a. Mensagem de erro, produto não é inserido

b. Fechamento de palete disparado

Exceção 4: Produto não é igual ao especificado na Ordem

a. Mensagem de erro, produto não é inserido

Exceção 5: Produto tem componente agregado diferente do especificado na ordem (ou em ordem anterior do mesmo material, no caso do SAP não estar disponível no momento de abertura da ordem)

a. Mensagem de erro, produto não é inserido

Retirando produto do palete

Pré-Requisitos:

a. Deve haver palete aberto

Operação Normal:

a. Operador lê comando "Retirar" com scanner

b. Operador lê etiqueta do produto

c. Produto é desassociado do palete no banco de dados local

d. Novas quantidades no palete e na ordem são apresentadas ao operador

Exceção 1: Produto não estava no palete

a. Mensagem de erro

Exceção 2: Cancelamento / Timeout de leitura (operador não leu etiqueta do produto)

a. Mensagem de cancelamento / timeout

Fechamento do palete

Pré-Requisitos:

a. Deve haver palete aberto

Operação Normal:

a. Sistema solicita leitura da etiqueta do palete para confirmação
b. Usuário lê etiqueta do palete
c. Sistema registra localmente que o palete está fechado, pronto para apontamento no SAP
d. Sistema mostra que não há palete aberto para usuário

Exceção 1: Leitura de confirmação da etiqueta do palete incorreta

a. Mensagem de erro

Exceção 2: Cancelamento / Timeout de confirmação

a. Mesagem de cancelamento / timeout

Desmontagem de palete

Pré-Requisitos:

a. Deve haver palete aberto

Operação Normal:

a. Operador lê comando "Desmontar palete" com pistola
b. Sistema solicita confirmação de desmontagem
c. Sistema exclui dados do palete do banco de dados local
d. Sistema mostra que não há palete aberto para usuário
e. Nova quantidade na ordem é apresentada ao operador

Exceção 2: Cancelamento / Timeout de confirmação

a. Mensagem de cancelamento / timeout

Encerrando Ordem

Pré-requisitos:

a. deve haver ordem iniciada, não deve haver palete aberto

Operação Normal:

a. Operador lê comando “Encerrar Ordem” com pistola

b. Sistema solicita confirmação do encerramento

c. Dados da ordem são apagados do banco local

d. Sistema deixa de mostrar dados da ordem para operador

4.4.2.2 – Estados

Assim se apresentou o diagrama de estados para o módulo de paletização:

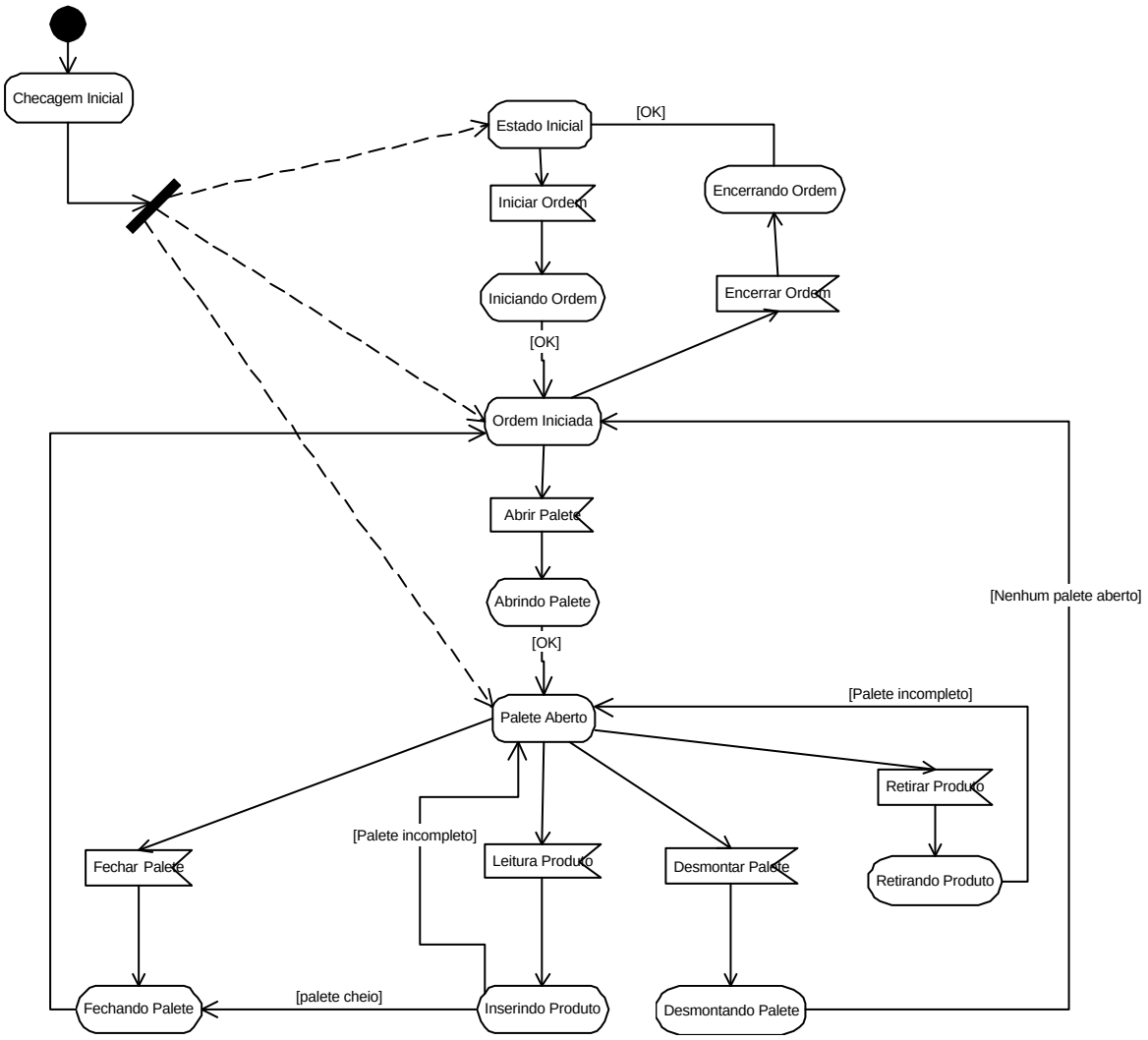

Figura 23: Diagrama de estados – Módulo Paletização

A máquina de estados do módulo de paletização apresenta um estado de ajuste e três estados que podemos caracterizar como “estados estáveis”.

Estados estáveis são estados que não “expiram” ou apresentam um tempo máximo, após o qual retornam. São estados que permanecem até que alguma ação por parte do usuário os modifique de alguma forma.

Já o estado de ajuste é necessário na inicialização do sistema. Este controle tem a capacidade de detectar o estado em que o sistema se encontrava antes de uma queda de luz, ou um fechamento acidental da janela do browser, por exemplo. Tal recurso é necessário pela característica de tolerância a falhas e zelo pela integridade do banco de dados que um sistema industrial requer. Uma descrição de cada estado se segue:

Checagem Inicial

É o estado em que o sistema faz uma checagem no banco, a fim de verificar, inicialmente, se algum palete foi deixado aberto e, em seguida, se alguma ordem ficou com status de “Iniciada”. Caso alguma destas premissas seja verdadeira, ele inicializa os objetos ordem e palete com os valores apropriados e leva a máquina para o estado apropriado.

Estado Inicial

É o estado em que o sistema se encontra quando não há ordem iniciada, nem palete aberto. Daqui o usuário pode comandar o início de uma Ordem. Ao fazê-lo, o sistema busca do SAP as ordens atuais e abre a janela para escolha da ordem a ser iniciada.

Ordem Iniciada

É quando a ordem foi corretamente escolhida pelo usuário, e pôde ser iniciada com sucesso. Carrega os seus valores do banco e aguarda a abertura de um dos paletes.

Palete Aberto

É quando um novo palete foi aberto ou reaberto. Daqui podem ser dados vários comandos (Fechar Palete, Desmontar Palete, Retirar Produto) ou pode ser lido um produto. A cada comando, uma função que o representa é chamada, gerando as chamadas para os objetos ordem e palete, de acordo com o necessário, e configurando a máquina para o novo estado apropriado, se for o caso. Sendo reconhecida a leitura de um produto, são feitas as verificações de praxe e, tanto em caso de sucesso como de erro, o sistema volta ao estado Palete Aberto.

5 IMPLEMENTAÇÃO

5.1 INTRODUÇÃO

A implementação é a fase de codificação propriamente dita de tudo o que foi planejado na etapa de projeto. Além disso, é a fase onde se fazem testes, depuração, se geram arquivos de configuração para, finalmente, instalar e configurar o sistema no ambiente de destino.

Inicialmente serão tecidos comentários gerais sobre como foi o processo de implementação de cada módulo, as dificuldades encontradas e os softwares que tiveram que ser desenvolvidos para que se pudessem efetuar os testes.

Será apresentada também a interface de cada módulo com o usuário, em seus elementos visuais e mecânicos.

5.2 DESENVOLVIMENTO E TESTES

O desenvolvimento foi dividido com base nos módulos estabelecidos pelo projeto. Inicialmente, foi desenvolvido o módulo Init e de Genealogia, que são razoavelmente mais simples. Depois se passou para o módulo de Impressão – e conseqüentemente para o desenvolvimento do driver de impressora – (do qual o módulo genealogia dependia) para, finalmente, o módulo de paletização. Os módulos Init e de transferência foram desenvolvidos por outro membro da equipe da HarboR.

Os testes foram executados durante o desenvolvimento de cada módulo. Cabe ressaltar aqui que o ciclo de desenvolvimento não se deu de forma tão linear. Em verdade, a base de cada módulo foi desenvolvida na seqüência descrita acima, mas após a primeira seqüência de testes, as alterações e correções foram sendo feitas à medida que se viu necessário, para cada módulo.

5.2.1 Preparação Inicial

Antes de iniciar-se o desenvolvimento de cada módulo, uma estrutura foi montada no sentido de convenções de programação e localizações de código-fonte.

Inicialmente, como o projeto inclui módulos que utilizam o mesmo banco de dados local, o mesmo arquivo de configuração XML local, etc, ficou evidente que a criação de arquivos para uso comum seria salutar. Desta forma, os arquivos de uso comum ficaram dispostos em um diretório específico, e todos os módulos os incluíram.

Todas as constantes foram inclusas em um único arquivo, assim como os nomes de diretórios locais. Assim, por exemplo, nomes de tags XML, resultados de chamadas de funções, valores de comandos em código de barras e outros valores ficaram registrados neste arquivo.

Com a possibilidade de exportação deste software para outro país (provavelmente a indústria da Embraco na Eslováquia), o uso de mensagens com a língua portuguesa ao longo do software tornaria difícil o processo de tradução. Assim, todas as mensagens do sistema foram armazenadas em constantes, e reunidas num único arquivo. Tornar-se-á fácil, desta forma, se necessária, a tradução para outro idioma do sistema como um todo.

No mesmo sentido, as classes para conexão com banco de dados e para interpretação de leituras de código de barras, ambas de uso comum, foram dispostas no diretório apropriado.

Por fim, foi criado um diretório para cada módulo, sendo os itens específicos de cada um armazenados nestas localizações.

5.2.2 Genealogia

O módulo de genealogia foi o primeiro a ser desenvolvido. Partindo-se da máquina de estados projetada, iniciou-se o desenvolvimento da classe da máquina, e a partir dela dos componentes do módulo. Aqui já se iniciaram os testes com o banco local e com os controles de leitura da porta serial. Como não se dispunha de um leitor de código de barras, foi desenvolvido um emulador do mesmo para que se pudessem fazer os testes. O emulador rodava no mesmo micro em que foi feito o desenvolvimento, através de um cabo ligado às duas portas seriais do computador.

Para o desenvolvimento da funcionalidade de comunicação com o display digital, foi necessário o estudo de sua documentação, para o conhecimento de seu protocolo de comunicação, comandos de posicionamento de cursor, limpeza de mostrador, etc.

Após os testes de verificação da correta transição entre os estados da máquina e corretas gravações e consultas no banco, ficou pendente o comando de impressão de etiquetas. A partir daí, iniciou-se o trabalho no módulo de impressão para que, depois, fosse completado o desenvolvimento da funcionalidade pendente.

5.2.3 Driver Impressora

O módulo de impressão depende do ActiveX com o driver da impressora, motivo pelo qual este foi desenvolvido primeiro. O estudo da documentação da impressora em questão também foi realizado aqui, para que se obtivesse o conhecimento necessário sobre os caracteres de controle, de formatação, comandos para verificação do status da impressora, etc. Como também não se dispunha de uma impressora Zebra S 400 para que se pudessem fazer os testes, um emulador da mesma, já disponível na HarboR, foi utilizado.

5.2.4 Impressão

Estando desenvolvido o driver, o módulo de impressão foi de relativa fácil codificação. Para a fase inicial de testes, como a parte do módulo genealogia responsável pelo envio do objeto XML com as informações relativas à etiqueta não estava ainda pronta, foi utilizado um XML criado em um arquivo local, à mão. Após a verificação do correto funcionamento deste, partiu-se para o restante do módulo Genealogia para, em seguida, iniciar-se o desenvolvimento da Paletização.

5.2.5 Paletização

Tomando como base a classe relativa à máquina de estados já desenvolvida no módulo de Genealogia, iniciou-se o módulo de paletização. Foi desenvolvida inicialmente toda a lógica da máquina de estados e das classes auxiliares para, depois, a parte de consulta de ordens. Como não havia um servidor SAP disponível na HarboR, em tempo de desenvolvimento as consultas foram feitas a um arquivo editado manualmente, fixo, disponibilizado no servidor IIS instalado na empresa.

5.3 INTERFACE COM USUÁRIO

5.3.1 Paletização

A interface com o usuário para o módulo paletização é feita de duas formas: através das janelas do controle no Windows, dentro da janela do browser, e através do leitor de código de barras.

A janela principal do módulo, isolado, pode ser aqui visualizada:

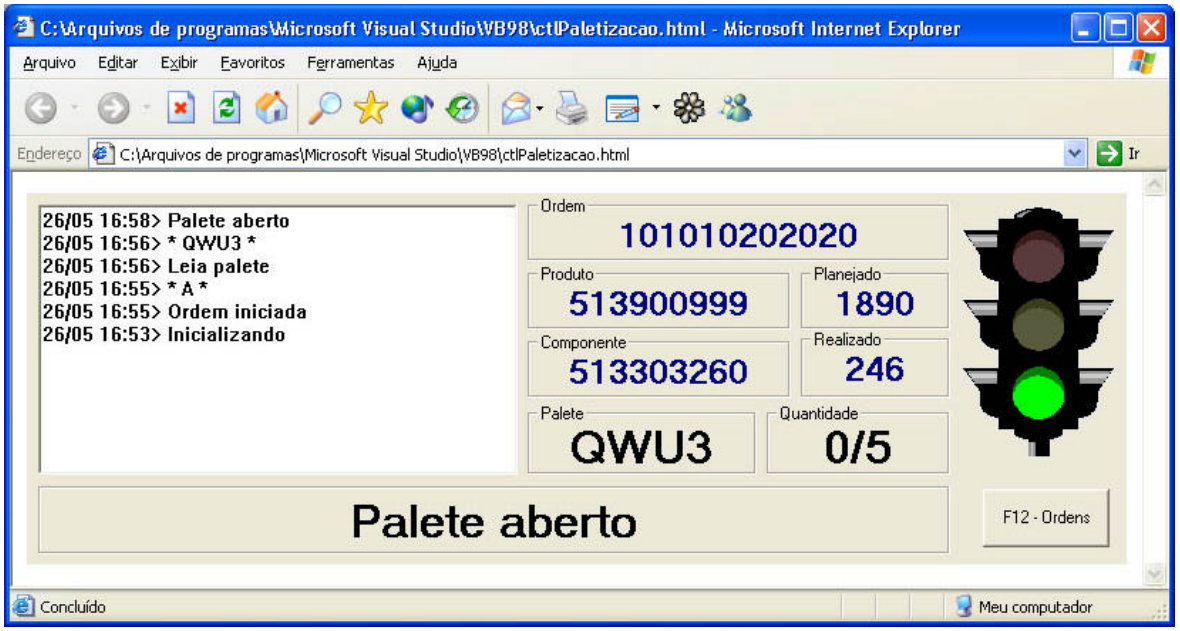

Figura 24: Tela principal do Módulo Paletização

Na parte esquerda, como pode ser visto, existe um campo que serve como "log" das ultimas operações do sistema. Campos de texto indicam a ordem atualmente aberta (se for o caso), o código de produto para aquela ordem, o código de componente para aquela ordem e a quantidade planejada e a já realizada. Além disto, quando é aberto um palete é mostrado o número serial do palete e a quantidade de produtos já inserida, bem como a capacidade daquele palete.

No grande campo embaixo são mostradas as mensagens para o usuário, como "Leia Palete", "Confirma desmontagem?", etc.

Há também um semáforo na parte direita, que indica o status atual do sistema. Caso ocorra algum tipo de erro, o semáforo muda pra vermelho pra indicar o erro. Se este for recuperável, ele volta para o verde, indicando que o usuário pode continuar interagindo com o sistema.

Uma amostra de um estado de erro pode ser visto na figura 21:

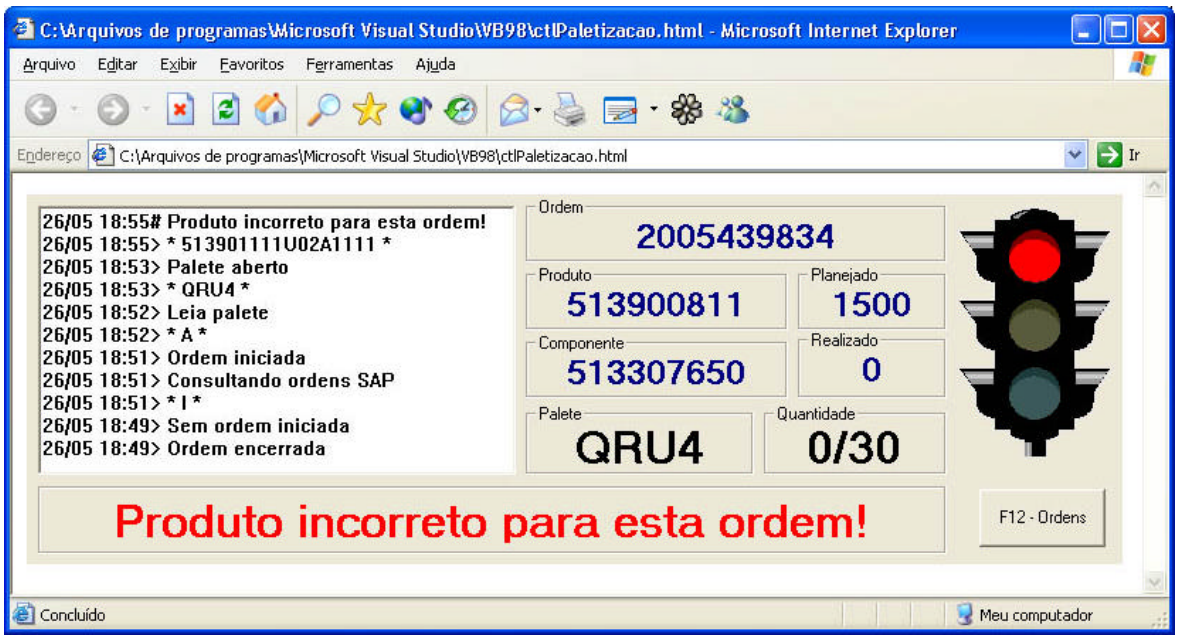

Figura 25: Tela principal paletização: Exemplo de estado de erro

Para a funcionalidade de consulta das ordens SAP existe o botão “F12-Ordens”. Este botão carrega a janela responsável pela captura, carregamento e seleção das ordens, que pode ser visualizada abaixo:

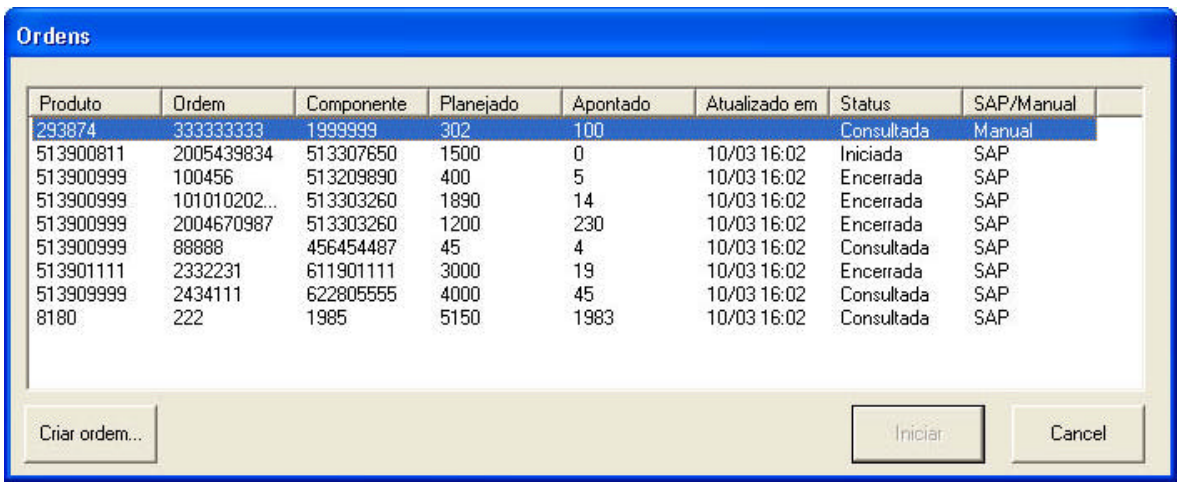

Figura 26: Janela “Ordens”

Aqui, os dados mais relevantes extraídos da consulta ao SAP podem ser visualizados. Se for o caso da conexão com o SAP não ser efetuada com sucesso, o mesmo vai ser indicado nesta janela e os dados serão carregados do banco local.

Caso a conexão com o SAP não esteja ativa ou se deseje iniciar a montagem de uma ordem ainda não disponível, existe a possibilidade de criação de uma nova ordem manualmente, disponibilizada pelo botão "Criar ordem...", que carrega a janela responsável pela criação de nova ordem manual:

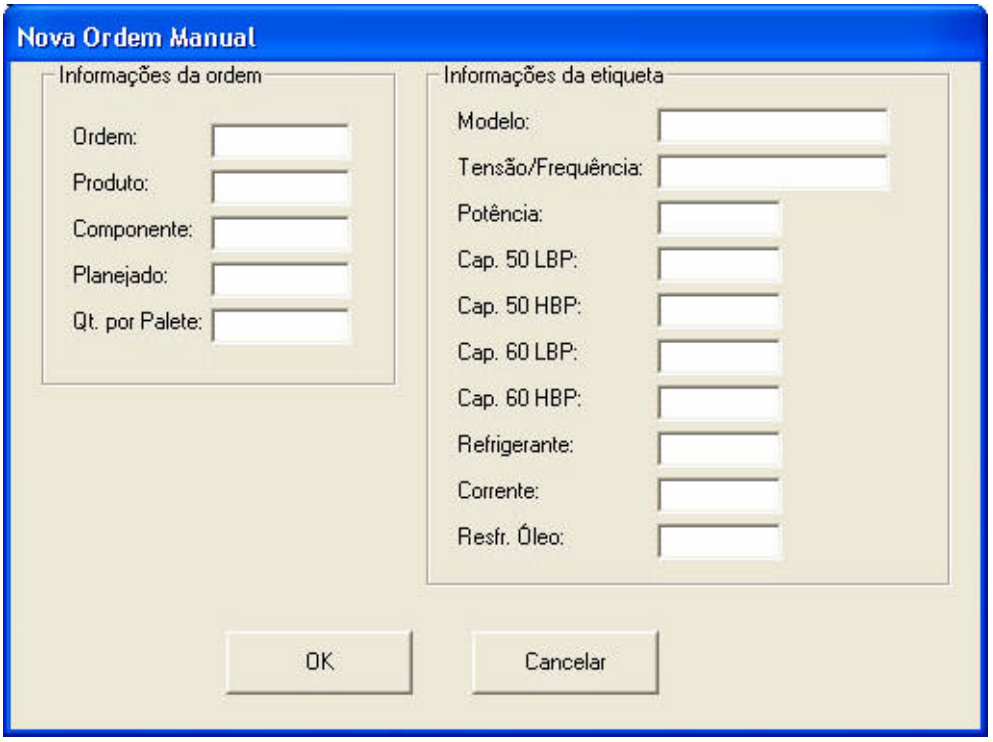

Figura 27: Janela "Nova Ordem Manual"

Aqui, como pode ser visto, estão disponíveis caixas de texto para o preenchimento das diversas informações necessárias para a ordem. Após o clique em OK, o sistema retorna para a janela de ordens, já incluindo a ordem recém-criada na listagem.

Além das telas há ainda, como interface de entrada com o usuário, o leitor de código de barras. Para a entrada dos comandos, foi impresso um cartaz contendo os códigos e seus respectivos significados, que pode ser visto de forma reduzida na figura a seguir:

Figura 28: Códigos de barra dos comandos

5.3.2 Genealogia

A interface do módulo de genealogia é bastante similar à do módulo de paletização. Sua principal tela pode ser visualizada na figura 25:

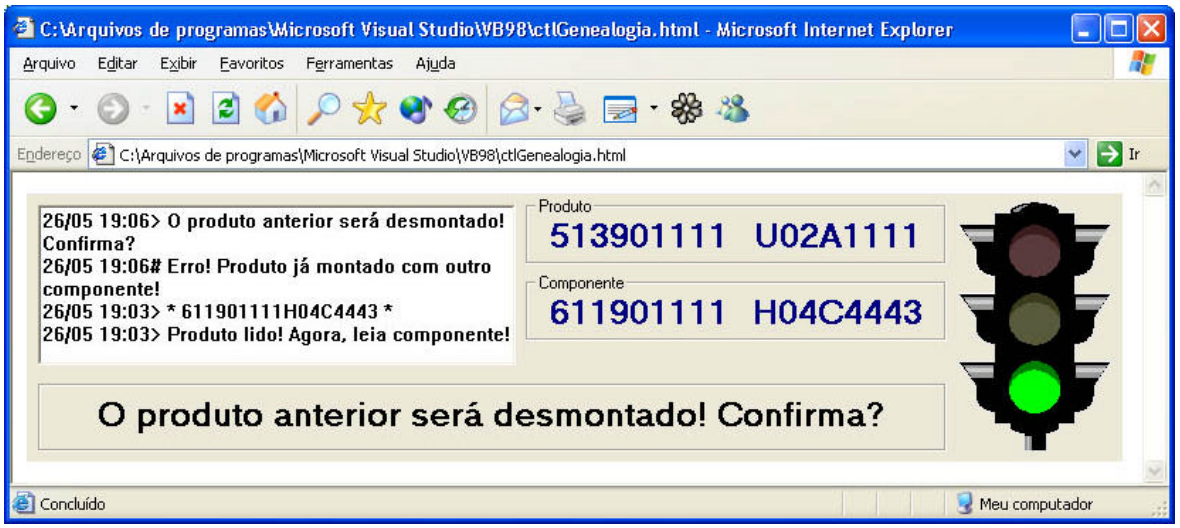

Figura 29: Tela principal do Módulo Genealogia

Da mesma forma que no módulo anterior, uma janela de log indica as últimas operações realizadas, o semáforo indica se o sistema está OK ou se ocorreu algum erro, e o campo maior mostra mensagens para o usuário. Há também um campo para mostrar o produto lido e outro para mostrar o componente lido.

5.3.3 Impressão

O controle da impressora, por se tratar de um driver, não apresenta interface com o usuário. Para isto foi necessário o controle de impressão, que mostra continuamente o status da impressora "traduzido" para o usuário:

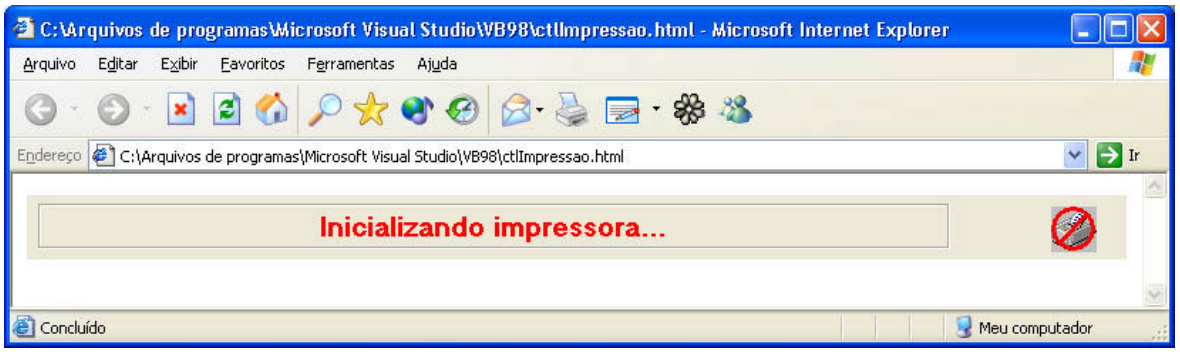

Figura 30: Tela do controle de impressão

Além do campo indicando o status, existe também um ícone no lado direito que retrata se a impressora está operante.

6. CONCLUSÕES E PERSPECTIVAS

Este trabalho demonstrou todo o processo de desenvolvimento de um sistema industrial de coleta de dados de genealogia e rastreabilidade para uma indústria de compressores, de forma integrada ao sistema de gestão já instalado no local.

Inicialmente, foi descrita a etapa de análise. O problema foi definido e foram levantados e elencados todos os requisitos – funcionais, de performance, relativos à disponibilidade, etc. - do sistema. A solução proposta também foi apresentada.

Em seguida, foi detalhada a etapa de projeto. Aqui, a arquitetura física e lógica do sistema foi projetada; o modelo estático – diagrama de modelo de dados e diagramas de classes – seguindo o padrão de modelagem UML foi apresentado e detalhado; o modelo dinâmico, com os diagramas de estados das máquinas de estados para cada um dos módulos, foi elucidado.

Finalmente, os trâmites da etapa de implementação foram comentados, assim como as dificuldades encontradas para se fazerem os testes e as soluções que foram desenvolvidas para que se conseguisse efetivá-los.

Para o desenvolvimento deste sistema, diversas tecnologias tiveram de ser estudadas e aprofundadas. As linguagens XML e XSL, que foram utilizadas para transferências de dados, impressão de etiquetas e como interface para configuração de opções do sistema, por exemplo. SQL e bancos de dados, utilizados na manipulação e nos armazenamentos temporários das informações. ActiveX e HTML, para integração com o Web Browser.

Além disso, foi também necessário um estudo acerca dos equipamentos envolvidos no processo, como leitores de código de barras, impressora e displays digitais. Conhecimentos sobre o próprio processo de produção de compressores também foram adquiridos, pela evidente necessidade de compreensão do problema.

Pode-se, ainda, reafirmar a importância de tais softwares para a indústria citando-se o fato de, antes mesmo deste trabalho ser concluído, outro sistema novo, semelhante ao desenvolvido, já ter sido requisitado. Pela estratégia de modelagem tomada e padrões

adotados, o desenvolvimento de novas versões com alterações e adição de recursos será bastante facilitada.

Por fim, pode-se dizer que o sistema desenvolvido funcionou como o esperado, e possibilitará um maior controle de qualidade para a produção de unidades condensadoras e seladas, podendo-se desta forma concluir que alcançou, com sucesso, os objetivos propostos.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] APPLEMAN, Dan. **"Visual Basic 5.0 Programmer's Guide to the Win32 API"**. USA. Ziff-Davis Press, 1997

[2] CURRAN, Thomas. **SAP R/3 Business Blueprint: understanding the business process reference model**. USA. Prentice Hall, 1998

[3] EMBRACO. **Perfil e Estrutura**. Disponível em: <http://www.embraco.com.br>

[4] HAND HELD PRODUCTS. **IMAGETEAM™ 3800/3900 Hand Held Linear Imager – Users's Guide**. USA, 2000. Disponível em <http://www.hhp.com>

[5] HERNANDEZ, Michael J. **"Database Design for Mere Mortals"**. USA. Addison-Wesley, January 2000

[6] LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e projeto orientados a objetos**. Porto Alegre: Bookman, 2000.

[7] MICROSOFT. **MSDN, The Microsoft Developer Network Library:** essential resources for developers. Disponível em DVD-ROM. Versão de julho de 2001

[8] PRESSMAN, Roger. **Engenharia de Software**. São Paulo: Makron Books, 1995.

[9] WILBOR-TECH. **Display torre**. Manual do Usuário. Setembro, 2000. Disponível em: <http://www.wilbor.com.br>

[10] ZEBRA. **ZPL II Programming Guide.** USA, 1998. Disponível em <http://www.zebra.com>

APÊNDICES

APÊNDICE I – CÓDIGO FONTE

Módulo Genealogia

ctlGenealogia.ctl

```
Option Explicit
Public codBarras As New clsCodigoBarras     'controle e leitura de valores de codigos de barras

Private Impressora As Object                    'controle de impressao
Private maquina As New clsMaqGenealogia         'maquina de estados da genealogia
Private inicializacaoSucesso As Boolean         'verifica o sucesso na inicializacao do controle
Private display As New clsDisplay               'controle de visualizacao no display

Public Sub MsgMostrar(Msg As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Mostra mensagem no label e a escreve
'            na janela de log. Mantem o semáforo verde.
'-------------------------------------------------------------
'Parâmetros: Msg: Mensagem a ser mostrada
'-------------------------------------------------------------
'Autor: GLF, 2003-01-29
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

    'mostra mensagem em preto
    lblMensagem.ForeColor = vbBlack
    lblMensagem.Caption = Msg

    'escreve no log e corta o final
    txtLog = Format(Now, "dd/mm hh:nn") & "> " & Msg & vbCrLf & txtLog
    txtLog = Left(txtLog, TAMANHO_MAX_LOG)

    'seta semaforo
    picVerde.Visible = True
    picVermelho.Visible = False

End Sub

Public Sub DisplayMostrar(Msg As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Mostra mensagem no display. Corta para 21
'            caracteres (maximo que o display suporta
'            para uma das linhas)
'-------------------------------------------------------------
'Parâmetros: Msg: Mensagem a ser mostrada
'-------------------------------------------------------------
'Autor: GLF, 2004-04-08
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

    display.MsgMostrar Left(Msg, 21)

End Sub

Public Sub ErroMostrar(msgErro As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Mostra mensagem de erro no label e a escreve
'            na janela de log. Alterna o semáforo pra vermelho
'            e dorme 2 segundos
'-------------------------------------------------------------
'Parâmetros: msgErro: Mensagem a ser mostrada
'-------------------------------------------------------------
'Autor: GLF, 2003-02-03
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

```
    'mostra mensagem em vermelho
    lblMensagem.ForeColor = vbRed
    lblMensagem.Caption = msgErro

    'escreve no log e corta o final
    txtLog = Format(Now, "dd/mm hh:nn") & "# " & msgErro & vbCrLf & txtLog
    txtLog = Left(txtLog, TAMANHO_MAX_LOG)

    'seta semaforo pra vermelho
    picVerde.Visible = False
    picVermelho.Visible = True

    'Atualiza tela
    Refresh

    'dorme 2 segundos
    Sleep (INTERVALO_MENSAGEM_ERRO)

End Sub

Public Function LeituraTimeout() As String
'******************************************************************************
'Finalidade:    espera leitura da serial ate o timeout definido em
'               tempoTimeout
'Argumentos:    -
'Retorna:       leitura em caso de sucesso, vazio em caso de TimeOut
'Criacao:       GLF 2002-01-16
'Alteracoes:
'******************************************************************************

Dim Leitura As String
Dim tempoExecucao As Long

    'espera leitura ou timeout
    While tempoExecucao < tempoTimeout And Leitura = ""
        DoEvents
        Sleep (INTERVALO_LEITURA)
        'se ha leitura na fila, pega
        Leitura = ctlLeitor.Ler
        tempoExecucao = tempoExecucao + INTERVALO_LEITURA
    Wend

    LeituraTimeout = Leitura

End Function


Public Function LeituraSerial() As String
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Fica aguardando a chegada de uma valor completo
'            (inicio, valor, fim) da serial sem interromper
'            outros processos.
'------------------------------------------------------------
'Parâmetros:
'------------------------------------------------------------
'Retorna: O valor lido, assim que este chegar
'------------------------------------------------------------
'Autor: GLF, 2003-02-13
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

Dim Leitura As String

    'Aguarda leitura
    While Leitura = ""
        DoEvents
        Sleep (INTERVALO_LEITURA)
        'Se há leitura na fila, pega o seu valor
        Leitura = ctlLeitor.Ler
    Wend

    LeituraSerial = Leitura

End Function
```

```
Public Sub ProdutoMostrar(CD As String, NS As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Mostra na janela do controle o codigo e o numero
'            de serie do produto.
'-----------------------------------------------------------
'Parametros: CD - Codigo do produto
'            NS - Numero de serie do produto
'-----------------------------------------------------------
'Autor: GLF, 2003-02-07
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    lblProdutoCD.Caption = CD
    lblProdutoNS.Caption = NS

    display.ProdutoMostrar NS

'    'escreve no log e corta o final
'    txtLog = Format(Now, "dd/mm hh:nn") & "* " & CD & NS & vbCrLf & txtLog
'    txtLog = Left(txtLog, TAMANHO_MAX_LOG)

End Sub

Public Sub ComponenteMostrar(CD As String, NS As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Mostra na janela do controle o codigo e o numero
'            de serie do produto.
'-----------------------------------------------------------
'Parametros: CD - Codigo do produto
'            NS - Numero de serie do produto
'-----------------------------------------------------------
'Autor: GLF, 2003-02-07
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    lblComponenteCD.Caption = CD
    lblComponenteNS.Caption = NS

    display.ComponenteMostrar NS

'    'escreve no log e corta o final
'    txtLog = Format(Now, "dd/mm hh:nn") & "* " & CD & NS & vbCrLf & txtLog
'    txtLog = Left(txtLog, TAMANHO_MAX_LOG)

End Sub

Public Sub IndicadoresLimpar()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Limpa os indicadores de codigo e numero de serie
'            de produto e componente
'-----------------------------------------------------------
'Autor: GLF, 2003-02-11
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    lblProdutoCD.Caption = ""
    lblProdutoNS.Caption = ""
    lblComponenteCD.Caption = ""
    lblComponenteNS.Caption = ""

    'Limpa tela do display
    display.TelaLimpar

End Sub



Private Sub UserControl_Initialize()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Inicializacao do controle. Inicializa maquina
'            de estados, configura controle de leitura
'            e inicia o timer
'-----------------------------------------------------------
'Autor: GLF, 2003-01-29
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

```
On Error GoTo ErrorHandler

Dim resultado As Integer
Dim sucessoXML As Boolean

    'Inicializa sem sucesso
    inicializacaoSucesso = False

    'Mostra msg de inicializacao
    MsgMostrar ("Inicializando")

    'Configuracao do controle, pelo arquivo XML
    resultado = InitXML

    'Inicializa configuracoes pelo XML de configuracao
    Select Case resultado

        Case RESULT_INICIALIZACAO_XML_OK
            'Tudo certo, nao precisa fazer nada
        Case RESULT_ERRO_INICIALIZACAO_LEITOR
            'Indica o erro, indica que nao conseguiu iniciar com
            'sucesso e sai do procedimento
            ErroMostrar MSG_PALETIZACAO_ERRO_INICIALIZACAO_LEITOR
            inicializacaoSucesso = False
            Exit Sub
        Case RESULT_ERRO_INICIALIZACAO_DISPLAY
            ErroMostrar MSG_PALETIZACAO_ERRO_INICIALIZACAO_DISPLAY
            inicializacaoSucesso = False
            Exit Sub
        Case RESULT_ERRO_INICIALIZACAO_IMPRESSORA
            ErroMostrar MSG_PALETIZACAO_ERRO_INICIALIZACAO_IMPRESSORA
            inicializacaoSucesso = False
            Exit Sub
        Case RESULT_ERRO_CARGA_XML
            ErroMostrar MSG_ERRO_CARREGANDO_XML
            inicializacaoSucesso = False
            Exit Sub
    End Select

    'Se chegou aqui, eh porque conseguiu inicializar as configuracoes

    'Conecta com banco de dados
    resultado = BDConectar(CAMINHO_LOCAL & ARQUIVO_MDB)

    Select Case resultado
        Case CONEXAO_PATH_INVALIDO
            ErroMostrar MSG_ERRO_BANCO_PATH_INVALIDO
            inicializacaoSucesso = False
        Case CONEXAO_ERRO_GERAL
            ErroMostrar MSG_ERRO_BANCO_GERAL
            inicializacaoSucesso = False
        Case CONEXAO_OK
            inicializacaoSucesso = True
    End Select

    'Tudo certo
Exit Sub

ErrorHandler:
    ErroMostrar MSG_ERRO_INTERNO & "[" & Err.Number & "]" & Err.Description
    inicializacaoSucesso = False

End Sub

Private Function InitXML() As Integer
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Configura leitor, display e controle de impressao
'            a partir do XML de configuracao
'-----------------------------------------------------------
'Retorna:   RESULT_INICIALIZACAO_XML_OK
'           RESULT_ERRO_CARGA_XML
'           RESULT_ERRO_INICIALIZACAO_DISPLAY
'           RESULT_ERRO_INICIALIZACAO_LEITOR
'           RESULT_ERRO_INICIALIZACAO_IMPRESSORA
```

```
'------------------------------------------------------------
'Autor: GLF, 2003-01-29
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo XMLErrorHandler

Dim xmlConfig As New DOMDocument26

Dim xmlRoot As Object
Dim xmlGenealogia As Object
Dim xmlLeitor As Object
Dim xmlCodigoBarras As Object
Dim xmlDisplay As Object
Dim xmlImpressora As IXMLDOMNode
Dim xslImpressora As New DOMDocument26
Dim sucesso As Boolean

    'inicializa com status de erro
    InitXML = False

    'seta para carregar arquivo sincronizado
    xmlConfig.async = "false"

    'carrega os dados de XML e separa apenas da genealogia
    xmlConfig.Load (CAMINHO_LOCAL & ARQUIVO_CONFIGURACAO)

    Set xmlRoot = xmlConfig.selectSingleNode("GERAL")

    Set xmlGenealogia = xmlRoot.selectSingleNode("GENEALOGIA")

    'Inicializa Leitor
    Set xmlLeitor = xmlGenealogia.selectSingleNode("LEITOR")

    sucesso = ctlLeitor.Init(xmlLeitor)

    If Not sucesso Then
        InitXML = RESULT_ERRO_INICIALIZACAO_LEITOR
        Exit Function
    End If

    'Inicializa Display
    Set xmlDisplay = xmlGenealogia.selectSingleNode("DISPLAY")

    sucesso = ctlDisplay.Init(xmlDisplay)  'Inicializa controle de serial para display
    display.Init ctlDisplay    'Inicializa controle de display

    If Not sucesso Then
        InitXML = RESULT_ERRO_INICIALIZACAO_DISPLAY
        Exit Function
    End If

    'Le Timeout
    tempoTimeout = xmlGenealogia.selectSingleNode("TEMPO_TIMEOUT").Text

    'carrega dados de codigo barras
    Set xmlCodigoBarras = xmlRoot.selectSingleNode("CODIGOBARRAS")
    codBarras.Init xmlCodigoBarras

    'Muda o tratamento de erros, para indicar problemas
    'especificos com a impressora

    On Error GoTo ImpressoraErrHandler

    'Instancia Impressora do user control Impressao
    Set Impressora = CreateObject("Impressao.clsInterfImpressora")

    'carrega dados de configuracao da impressora
    Set xmlImpressora = xmlRoot.selectSingleNode("IMPRESSORA")
    sucesso = Impressora.Inicializar(xmlImpressora)
    If Not sucesso Then
        InitXML = RESULT_ERRO_INICIALIZACAO_IMPRESSORA
        Exit Function
    End If

    'carrega XSL de leiaute de impressao
```

```
    xslImpressora.Load (CAMINHO_LOCAL & ARQUIVO_FORMATO_ETIQUETA)
    Impressora.setXSL xslImpressora.xml

    'Se chegou ateh aqui, eh porque teoricamente deu tudo certo ;)
    InitXML = RESULT_INICIALIZACAO_XML_OK

Exit Function

'Caso tenha ocorrido algum erro
XMLErrorHandler:
    InitXML = RESULT_ERRO_CARGA_XML
    Exit Function
'Caso tenha ocorrido algum erro na parte de inicializacao da impressora
ImpressoraErrHandler:
    InitXML = RESULT_ERRO_INICIALIZACAO_IMPRESSORA
    Exit Function
End Function

Public Sub EtiquetaImprimir(XMLEtiqueta As IXMLDOMNode)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Envia para impressora comando para imprimir
'            etiqueta
'--------------------------------------------------------------
'Parâmetros: XMLEtiqueta - XML contendo informacoes da
'                          etiqueta a ser impressa
'--------------------------------------------------------------
'Retorna:
'--------------------------------------------------------------
'Autor: GLF, 2003-04-09
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
    Impressora.Imprimir XMLEtiqueta

End Sub
Private Sub UserControl_Show()

    'Inicializa maquina de estados, caso tenha ocorrido tudo certo
    'na inicializacao do controle

    If inicializacaoSucesso Then
        maquina.Init

        'Coloca a maquina para rodar
        While True
            maquina.executar Me
            'Caso tenha ocorrido algum erro,
            'dorme e reinicializa a maquina
            Sleep (2000)
            DoEvents
        Wend

    End If
End Sub

Public Sub SerialDisplayEnviar()


End Sub
```

clsMaqGenealogia.cls

```
Option Explicit

'Maquina de estados para genealogia

Private Leitura As String   'String que armazenara o valor da leitura da serial
Private sql As String       'armazena instrução sql
Private rs As New Recordset 'armazena os resultados dos queries sql
Private CD As String        'Codigo Generico
Private NS As String        'Numero serial generico
Private prodCD As String    'Codigo do produto
Private prodNS As String    'Numero serial do produto
Private compCD As String    'Codigo do componente
```

```
Private compNS As String    'Numero serial do componente
Private ieOrdem As String   'Nr da ordem do produto sendo montado, de onde
                            'serao tiradas as informacoes para a impressao da
                            'etiqueta

Private estado As enmEstadoGenealogia       'Define o estado da maquina de estados
Private estadoErro As enmEstadoGenealogia   'Armazena o estado anterior a um erro
                                            'do tipo genérico


Public Sub Init()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Inicializa a maquina de estados
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Retorna:
'-------------------------------------------------------------
'Autor: GLF, 2003-02-07
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    estado = GEN_ESTADO_INICIAL

End Sub

Public Sub executar(ctlGen As ctlGenealogia)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Metodo principal da maquina de estados, que
'            é executado ininterruptamente e gerencia
'            as chamadas de leituras, display e
'            transicoes entre estados
'-------------------------------------------------------------
'Argumentos: ctlGen - Uma referencia ao controle
'                     do ctlGenealogia que instanciou a
'                     maquina.
'-------------------------------------------------------------
'Autor: GLF, 2003-02-07
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler

Dim XMLEtiqueta As DOMDocument26    'Armazenará o XML pai pra etiqueta
Dim noRaiz As IXMLDOMNode           'O nó raiz para o XML
Dim novaEtiqueta As IXMLDOMNode     'O nó para a etiqueta
Dim itemEtiqueta As IXMLDOMNode     'Cada um dos itens com informacoes sobre a etiqueta
Dim data As String                  'Data atual
Dim estadoAnterior As enmEstadoGenealogia 'O estado anterior ao estado 'atual', na maquina
                                          'de estados

    While True

    'Verifica em qual estado se encontra o sistema
    Select Case estado

        '-------------------------------------------------------
        Case GEN_ESTADO_INICIAL
        '-------------------------------------------------------

            'Limpa indicadores de NS e CD de prod. e comp.
            ctlGen.IndicadoresLimpar

            'Mostra mensagem inicial solicitando leitura de um componente
            ctlGen.MsgMostrar MSG_GENEALOGIA_LEIA_COMPONENTE
            ctlGen.DisplayMostrar MSG_DISPLAY_LEIA_COMPONENTE

            'Aguarda leitura de algum componente (ou produto)
            Leitura = ctlGen.LeituraSerial

            'Mostra leitura
            ctlGen.MsgMostrar "* " & Leitura & " *"

            Select Case Leitura

                '-----------------------------------------
                'DESMONTAR PRODUTO
```

```
'-----------------------------------------
Case CMD_DESMONTAR_PRODUTO

    'Mostra mensagem para leitura de produto
    ctlGen.MsgMostrar MSG_LEIA_PRODUTO
    ctlGen.DisplayMostrar MSG_DISPLAY_LEIA_PRODUTO

    'Faz a leitura da serial até o tempo máximo permitido
    Leitura = ctlGen.LeituraTimeout

    If Leitura <> "" Then

        'Mostra leitura
        ctlGen.MsgMostrar "* " & Leitura & " *"

        'Extrai o numero de série e codigo do código lido
        If ctlGen.codBarras.ValoresRetornar(Leitura, CD, NS) Then

            'Verifica a existencia do produto no banco local
            sql = "SELECT count(*) as contagem FROM produtos WHERE ns_produto = '" & _
                  NS & "'"
            rs.Open sql, Cnn

            'Se o produto existe
            If rs!contagem > 0 Then

                'Remove o produto especificado do banco local
                sql = "DELETE FROM produtos WHERE ns_produto = '" & NS & "'"
                Cnn.Execute sql

                'Mostra msg indicando que o produto foi desmontado com sucesso
                ctlGen.MsgMostrar MSG_GENEALOGIA_PRODUTO_DESMONTADO
                ctlGen.DisplayMostrar MSG_DISPLAY_PRODUTO_DESMONTADO

                'Retorna ao estado inicial
                estado = GEN_ESTADO_INICIAL

            Else
            'O produto nao existe.
                ctlGen.DisplayMostrar MSG_DISPLAY_PRODUTO_INEXISTENTE
                ctlGen.ErroMostrar MSG_PRODUTO_INEXISTENTE
                estado = GEN_ESTADO_INICIAL
            End If

            rs.Close

        Else 'Erro de codigo invalido/configuracoes invalidas
            ctlGen.DisplayMostrar MSG_DISPLAY_LEITURA_INVALIDA
            ctlGen.ErroMostrar MSG_LEITURA_INVALIDA
            estado = GEN_ESTADO_INICIAL
        End If

    Else 'Erro de timeout na leitura
            ctlGen.DisplayMostrar MSG_DISPLAY_TIMEOUT
            ctlGen.ErroMostrar MSG_TIMEOUT
            estado = GEN_ESTADO_INICIAL
    End If

'-----------------------------------------
'IMPRIMIR ETIQUETA DE REPOSICAO
'-----------------------------------------
Case CMD_IMPRIMIR_ETIQUETA_REPOSICAO

    'Mostra mensagem para leitura de produto
    ctlGen.MsgMostrar MSG_LEIA_PRODUTO
    ctlGen.DisplayMostrar MSG_DISPLAY_LEIA_PRODUTO

    'Faz a leitura da serial até o tempo máximo permitido
    Leitura = ctlGen.LeituraTimeout

    If Leitura <> "" Then

        'Mostra leitura
        ctlGen.MsgMostrar "* " & Leitura & " *"
```

```
            'Extrai o numero de série e codigo do código lido
            If ctlGen.codBarras.ValoresRetornar(Leitura, prodCD, prodNS) Then

                'Verifica a existencia de ordem no banco local com
                'informacoes sobre um produto do mesmo codigo

                sql = "SELECT ie_ordem FROM ordens WHERE cd_produto = '" & prodCD & "'"
                rs.Open sql, Cnn

                If Not rs.EOF Then

                    'Armazena a ordem atual, da qual serao extraídas as informacoes
                    ieOrdem = rs!ie_ordem
                    'Armazena o estado que levou a maquina à GEN_IMPRIMINDO_ETIQUETA
                    estadoAnterior = GEN_ESTADO_INICIAL
                    'Seta maquina para mudança de estado
                    estado = GEN_IMPRIMINDO_ETIQUETA

                Else 'Nao encontrou ordem com produto deste codigo no banco local
                    ctlGen.DisplayMostrar MSG_DISPLAY_PRODUTO_INEXISTENTE
                    ctlGen.ErroMostrar MSG_PRODUTO_INEXISTENTE
                    estado = GEN_ESTADO_INICIAL
                End If

                rs.Close

            Else 'Erro de codigo invalido/configuracoes invalidas
                ctlGen.DisplayMostrar MSG_DISPLAY_LEITURA_INVALIDA
                ctlGen.ErroMostrar MSG_LEITURA_INVALIDA
                estado = GEN_ESTADO_INICIAL
            End If

        Else 'Erro de timeout na leitura
                ctlGen.DisplayMostrar MSG_DISPLAY_TIMEOUT
                ctlGen.ErroMostrar MSG_TIMEOUT
                estado = GEN_ESTADO_INICIAL
        End If


    '-----------------------------------------
    'NÚMERO QUALQUER, COMPONENTE OU PRODUTO
    '-----------------------------------------
    'Foi lido algum codigo, como o de um produto ou de um componente
    Case Else

        'Extrai o numero de série e codigo do código lido
        If ctlGen.codBarras.ValoresRetornar(Leitura, CD, NS) Then

            'Verifica se o codigo lido nao é, por acaso, o de um produto
            sql = "select count(*) as contagem from ordens where cd_produto = '" & CD & "'"
            rs.Open sql, Cnn

            'Caso nao tenha achado nenhum produto com este codigo,
            'eh sinal de que o codigo de barras refere-se realmente
            'a um componente (ou deve ser, ao menos)
            If rs!contagem = 0 Then
            '----------------------------------------
            'CODIGO DE COMPONENTE (teoricamente)
            '----------------------------------------

                'Salva os valores lidos de codigo e numero de série
                compCD = CD
                compNS = NS

                'Mostra nos indicadores
                ctlGen.ComponenteMostrar compCD, compNS

                'Mostra mensagem indicando para ler o produto
                ctlGen.MsgMostrar MSG_LEIA_PRODUTO
                ctlGen.DisplayMostrar MSG_DISPLAY_LEIA_PRODUTO

                'Hora de ler o produto!
                estado = GEN_AGUARDANDO_LEITURA_PRODUTO
```

```
                '----------------------------------------
                'CODIGO DE BARRAS DE PRODUTO
                '----------------------------------------
                Else

                    'Salva os valores lidos de codigo e numero de série
                    prodCD = CD
                    prodNS = NS

                    'Mostra nos indicadores
                    ctlGen.ProdutoMostrar prodCD, prodNS

                    'Mostra mensagem indicando que foi lido um produto,
                    'e pedindo entao para ler o componente

                    ctlGen.MsgMostrar MSG_GENEALOGIA_AGORA_LEIA_COMPONENTE
                    ctlGen.DisplayMostrar MSG_DISPLAY_AGORA_LEIA_COMPONENTE

                    'Hora de ler o produto!
                    estado = GEN_AGUARDANDO_LEITURA_COMPONENTE
                End If

                rs.Close

            Else
                'Erro de codigo invalido/configuracoes invalidas
                ctlGen.DisplayMostrar MSG_DISPLAY_LEITURA_INVALIDA
                ctlGen.ErroMostrar MSG_LEITURA_INVALIDA
                estado = GEN_ESTADO_INICIAL
            End If

        End Select 'Leitura

'-------------------------------------------------------
Case GEN_AGUARDANDO_LEITURA_PRODUTO
'-------------------------------------------------------

    'Le o produto
    Leitura = ctlGen.LeituraTimeout

    'Verifica se algo foi lido
    If Leitura <> "" Then

        'Mostra leitura
        ctlGen.MsgMostrar "* " & Leitura & " *"

        'Extrai o numero de série e codigo do código de barras lido
        If ctlGen.codBarras.ValoresRetornar(Leitura, prodCD, prodNS) Then

            'Mostra nos indicadores o codigo lido
            ctlGen.ProdutoMostrar prodCD, prodNS

            'Seta a maquina para verificar a montagem
            estado = GEN_VERIFICANDO_MONTAGEM

        Else 'Erro de codigo invalido/configuracoes invalidas
            ctlGen.DisplayMostrar MSG_DISPLAY_LEITURA_INVALIDA
            ctlGen.ErroMostrar MSG_LEITURA_INVALIDA
            estado = GEN_ESTADO_INICIAL
        End If

    Else 'Ocorreu timeout
        ctlGen.DisplayMostrar MSG_DISPLAY_TIMEOUT
        ctlGen.ErroMostrar MSG_TIMEOUT
        estado = GEN_ESTADO_INICIAL
    End If

'-------------------------------------------------------
Case GEN_AGUARDANDO_LEITURA_COMPONENTE
'-------------------------------------------------------

    'Le o componente
    Leitura = ctlGen.LeituraTimeout
```

```
    'Verifica se algo foi lido
    If Leitura <> "" Then

        'Mostra leitura
        ctlGen.MsgMostrar "* " & Leitura & " *"

        'Extrai o numero de série e codigo do código de barras lido
        If ctlGen.codBarras.ValoresRetornar(Leitura, compCD, compNS) Then

            'Mostra nos indicadores
            ctlGen.ComponenteMostrar compCD, compNS

            'Seta a maquina para verificar a montagem
            estado = GEN_VERIFICANDO_MONTAGEM

        Else 'Erro de codigo invalido/configuracoes invalidas
            ctlGen.DisplayMostrar MSG_DISPLAY_LEITURA_INVALIDA
            ctlGen.ErroMostrar MSG_LEITURA_INVALIDA
            estado = GEN_ESTADO_INICIAL
        End If

    Else 'Ocorreu timeout
        ctlGen.DisplayMostrar MSG_DISPLAY_TIMEOUT
        ctlGen.ErroMostrar MSG_TIMEOUT
        estado = GEN_ESTADO_INICIAL
    End If


'-----------------------------------------------------------
Case GEN_VERIFICANDO_MONTAGEM
'-----------------------------------------------------------

    'Monta o sql para verificar se existe pelo menos alguma ordem com esta
    'associacao de produto x componente, para verificar se a montagem
    'esta correta.

    'Deve ser usado substr(cd_componente,etc) ao invés de simplesmente
    'o codigo lido por haver diferenças entre os códigos do Brasil e da
    'Eslováquia. Os codigos da eslováquia são armazenados
    'no SAP (e consequentemente, no banco local) com alguns caracteres
    'a mais após o código lido. Devem ser comparados apenas os n primeiros
    'caracteres do código lido com o do banco, onde n = tamanho(codigo lido).

    sql = "SELECT ie_ordem FROM ordens WHERE cd_produto = '" & prodCD & "'" & _
          "AND LEFT(cd_componente," & Len(compCD) & ") = '" & compCD & "'"
    rs.Open sql, Cnn

    If Not rs.EOF Then

        'Correto! Achou uma ordem com este tipo de produto e componente!

        'Salva o nr da ordem para, se tudo estiver correto depois,
        'ter as informacoes do produto para imprimir a etiqueta.
        ieOrdem = rs!ie_ordem

        'Verifica se este componente já não está montado em
        'outro produto
        rs.Close
        sql = "SELECT count(*) as contagem FROM produtos WHERE ns_componente = '" & compNS & "'"
        rs.Open sql, Cnn
        If rs!contagem = 0 Then
            'Correto, componente nao esta montado em nenhum outro produto!
            'Verifica se por acaso o produto já está montado,
            'com outro componente

            rs.Close
            sql = "SELECT  count(*) as contagem FROM produtos WHERE ns_produto = '" & prodNS & "'"
            rs.Open sql, Cnn
            If rs!contagem = 0 Then
                'Certo, sem montagem anterior

                'Armazena o estado que levou a maquina à GEN_IMPRIMINDO_ETIQUETA
                 estadoAnterior = GEN_VERIFICANDO_MONTAGEM
```

```
                'Passa pro proximo estado, comandando a impressao
                'da etiqueta
                estado = GEN_IMPRIMINDO_ETIQUETA

            Else
                'O produto ja estava cadastrado!
                estado = GEN_ERRO_PRODUTO_COM_OUTRO_COMPONENTE
            End If ' Se o produto ja esta cadastrado

        Else
            'Achou algum outro produto já usando esse componente
            estado = GEN_ERRO_COMPONENTE_JA_MONTADO
        End If ' Se algum outro produto ja usa esse componente
        rs.Close
    Else

    'Nao há nenhuma ordem contendo este produto e este componente.
    'Das duas uma:
    '- Ou a montagem esta incorreta (o produto existe localmente mas está associado
    '  a outro componente nas ordens)
    '- Ou nao há dados para este componente na base local, indicando que a ordem SAP
    '  nao foi carregada localmente

        'Checando qual destes dois possíveis erros causou esta situação!
        'Verificar se existe tal produto nas ordens
        rs.Close
        sql = "select count(*) as contagem FROM ordens WHERE (cd_produto = '" & prodCD & "'"
        rs.Open sql, Cnn

        'Caso exista
        If rs!contagem > 0 Then
            'O erro eh que a montagem está incorreta!
            estado = GEN_ERRO_MONTAGEM_INCORRETA

        'Caso contrario, a ordem SAP para o produto nao foi
        'carregada localmente
        Else
            'O erro eh que a montagem está incorreta!
            estado = GEN_ERRO_SEM_ORDEM_SAP_LOCAL

        End If
        rs.Close

    End If 'procura por ordem com esse produto e componente

'-------------------------------------------------------
Case GEN_ERRO_PRODUTO_COM_OUTRO_COMPONENTE
'-------------------------------------------------------

    'Indica esse erro
    ctlGen.DisplayMostrar MSG_DISPLAY_PRODUTO_JA_MONTADO
    ctlGen.ErroMostrar MSG_GENEALOGIA_PRODUTO_JA_MONTADO

    'Pergunta se o usuario deseja desmontar o componente anterior
    ctlGen.MsgMostrar MSG_GENEALOGIA_SOLICITACAO_DESMONTAGEM
    ctlGen.DisplayMostrar MSG_DISPLAY_SOLICITACAO_DESMONTAGEM

    'Lê a resposta do usuario até um tempo limite
    Leitura = ctlGen.LeituraTimeout

    'Verifica se algo foi lido
    If Leitura <> "" Then

        'Mostra leitura
        ctlGen.MsgMostrar "* " & Leitura & " *"

        'Verifica se foi lido a resposta SIM
        If Leitura = CMD_SIM Then

            'CONFIRMACAO PARA DESMONTAR PRODUTO ANTERIOR

            'Remove a montagem antiga cadastrada no banco local
            sql = "DELETE FROM produtos WHERE ns_produto = '" & prodNS & "'"
```

```
                    Cnn.Execute sql

                    'Montagem anterior do produto removida!
                    'Comanda a impressao da etiqueta
                    estado = GEN_IMPRIMINDO_ETIQUETA

                ElseIf Leitura = CMD_NAO Then

                    'CONFIRMACAO PARA DESMONTAR NEGADA

                    ctlGen.DisplayMostrar MSG_DISPLAY_OPERACAO_CANCELADA
                    ctlGen.ErroMostrar MSG_OPERACAO_CANCELADA
                    estado = GEN_ESTADO_INICIAL

                Else 'Erro, leitura invalida
                    ctlGen.DisplayMostrar MSG_DISPLAY_LEITURA_INVALIDA
                    ctlGen.ErroMostrar MSG_LEITURA_INVALIDA
                    estado = GEN_ESTADO_INICIAL
                End If

            Else 'Ocorreu timeout
                ctlGen.DisplayMostrar MSG_DISPLAY_TIMEOUT
                ctlGen.ErroMostrar MSG_TIMEOUT
                estado = GEN_ESTADO_INICIAL
            End If

        '-------------------------------------------------------
        Case GEN_IMPRIMINDO_ETIQUETA
        '-------------------------------------------------------
        'Comanda a impressao da etiqueta

            'Pega os valores do banco
            sql = "SELECT de_modelo, de_tensao_frequencia, de_potencia," & _
                          "de_capacidade_50LBP, de_capacidade_50HBP, de_capacidade_60LBP,
de_capacidade_60HBP," & _
                          "de_refrigerante, de_corrente, de_refrigerador_oleo " & _
                          "FROM ordens WHERE ie_ordem = '" & ieOrdem & "'"

            rs.Open sql, Cnn

            'Cria um novo pai para o XML de etiqueta
            Set XMLEtiqueta = New DOMDocument26

            'Cria a raiz ETIQUETAS
            Set noRaiz = XMLEtiqueta.createNode(NODE_ELEMENT, "ETIQUETAS", "")
            XMLEtiqueta.appendChild noRaiz

            'Cria o nó da etiqueta em questão
            Set novaEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "ETIQUETA", "")
            XMLEtiqueta.childNodes(0).appendChild novaEtiqueta

            'Adiciona nó "DE_MODELO"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "DE_MODELO", "")
            If Not IsNull(rs!de_modelo) Then itemEtiqueta.Text = rs!de_modelo
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "CD_PRODUTO"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "CD_PRODUTO", "")
            itemEtiqueta.Text = prodCD
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "TENSAO_FREQUENCIA"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "TENSAO_FREQUENCIA", "")
            If Not IsNull(rs!de_tensao_frequencia) Then itemEtiqueta.Text = rs!de_tensao_frequencia
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "DE_POTENCIA"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "DE_POTENCIA", "")
            If Not IsNull(rs!de_potencia) Then itemEtiqueta.Text = rs!de_potencia
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "de_capacidade_50lbp"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "DE_CAPACIDADE_50LBP", "")
            If Not IsNull(rs!de_capacidade_50lbp) Then itemEtiqueta.Text = rs!de_capacidade_50lbp
```

```
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "de_capacidade_50hbp"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "DE_CAPACIDADE_50HBP", "")
            If Not IsNull(rs!de_capacidade_50hbp) Then itemEtiqueta.Text = rs!de_capacidade_50hbp
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "de_capacidade_60lbp"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "DE_CAPACIDADE_60LBP", "")
            If Not IsNull(rs!de_capacidade_60lbp) Then itemEtiqueta.Text = rs!de_capacidade_60lbp
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "de_capacidade_60hbp"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "DE_CAPACIDADE_60HBP", "")
            If Not IsNull(rs!de_capacidade_60hbp) Then itemEtiqueta.Text = rs!de_capacidade_60hbp
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "DE_REFRIGERANTE"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "DE_REFRIGERANTE", "")
            If Not IsNull(rs!de_refrigerante) Then itemEtiqueta.Text = rs!de_refrigerante
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "DE_CORRENTE"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "DE_CORRENTE", "")
            If Not IsNull(rs!de_corrente) Then itemEtiqueta.Text = rs!de_corrente
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "DE_REFRIGERADOR_OLEO"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "DE_REFRIGERADOR_OLEO", "")
            If Not IsNull(rs!de_refrigerador_oleo) Then itemEtiqueta.Text = rs!de_refrigerador_oleo
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "NS_PRODUTO"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "NS_PRODUTO", "")
            itemEtiqueta.Text = prodNS
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Adiciona nó "DATA_FABRICACAO"
            Set itemEtiqueta = XMLEtiqueta.createNode(NODE_ELEMENT, "DATA_FABRICACAO", "")
            'Pega data atual
            data = Now()
            'Monta data no formato dd/mmm/yy, com mmm em ingles
            itemEtiqueta.Text = Format(data, "dd") & "/" & StringMesInglesTransformar(Format(data, "mm"))
& "/" & Format(data, "yy")
            XMLEtiqueta.childNodes(0).childNodes(0).appendChild itemEtiqueta

            'Manda imprimir a etiqueta
            ctlGen.EtiquetaImprimir XMLEtiqueta

            Set XMLEtiqueta = Nothing
            Set noRaiz = Nothing
            Set novaEtiqueta = Nothing

            'Verifica o estado anterior, para a partir dele definir o proximo estado
            'Se o estado anterior for o inicial (ou seja, foi dado comando para
            'impressao de etiqueta avulsa)
            If estadoAnterior = GEN_ESTADO_INICIAL Then
                'Retorna para o inicial
                estado = GEN_ESTADO_INICIAL
            Else
                'Comanda o registro da montagem
                estado = GEN_REGISTRANDO_MONTAGEM
            End If

        '-------------------------------------------------------
        Case GEN_REGISTRANDO_MONTAGEM
        '-------------------------------------------------------
        'Cria entrada na tabela de produtos, contendo
        'o produto especificado, os codigos de seu componente,
        'etc

        sql = "INSERT INTO produtos (ns_produto, cd_produto, ns_componente, " & _
                                    "cd_componente, dh_montagem) VALUES ('" & _
                                    prodNS & "', '" & _
```

```
                                    prodCD & "', '" & _
                                    compNS & "', '" & _
                                    compCD & "', '" & _
                                    Now & "')"

        Cnn.Execute sql

        'Tudo certo, mostra mensagem de confirmacao e volta ao estado inicial
        ctlGen.MsgMostrar MSG_GENEALOGIA_PRODUTO_MONTADO
        ctlGen.DisplayMostrar MSG_DISPLAY_PRODUTO_MONTADO

        estado = GEN_ESTADO_INICIAL

        '-------------------------------------------------------
        Case GEN_ERRO_MONTAGEM_INCORRETA
        '-------------------------------------------------------

            ctlGen.DisplayMostrar MSG_DISPLAY_MONTAGEM_INCORRETA
            ctlGen.ErroMostrar MSG_GENEALOGIA_MONTAGEM_INCORRETA
            estado = GEN_ESTADO_INICIAL

        '-------------------------------------------------------
        Case GEN_ERRO_SEM_ORDEM_SAP_LOCAL
        '-------------------------------------------------------

            ctlGen.DisplayMostrar MSG_DISPLAY_ORDEM_INEXISTENTE
            ctlGen.ErroMostrar MSG_GENEALOGIA_ORDEM_INEXISTENTE
            estado = GEN_ESTADO_INICIAL

        '-------------------------------------------------------
        Case GEN_ERRO_COMPONENTE_JA_MONTADO
        '-------------------------------------------------------

            ctlGen.DisplayMostrar MSG_DISPLAY_COMPONENTE_JA_MONTADO
            ctlGen.ErroMostrar MSG_GENEALOGIA_COMPONENTE_JA_MONTADO
            estado = GEN_ESTADO_INICIAL

    End Select

Wend
Exit Sub

ErrorHandler:

    'Na eventualidade de ocorrer algum erro nao previsto,

    'Armazena o estado em que o erro ocorreu
    estadoErro = estado

    'Fecha recordset, caso esteja aberto
    If (rs.State) Then rs.Close

    'Mostra o erro, indicando nr, descricao, e estado em que ocorreu
    ctlGen.DisplayMostrar MSG_DISPLAY_ERRO
    ctlGen.ErroMostrar MSG_ERRO_INTERNO & _
                       " [" & Err.Number & "]'" & Err.Description & "' " & _
                       "Estado: " & estadoErro

    'Seta para estado de reinicio de maquina
    estado = GEN_ESTADO_INICIAL

    'A maquina será reiniciada automaticamente pelo ctlGenealogia
End Sub

Private Function LeituraVerificaLimites(tamMin As Integer, tamMax As Integer, str As String) As Boolean
'Verifica se o nr de caracteres contido na string esta entre os limites
'passados para a funcao

'Retorna true caso esteja,
'false caso contrario

    'Verifica se o codigo lido está entre os limites
    If (Len(Leitura) < tamMin) Or (Len(Leitura) > tamMax) Then
        LeituraVerificaLimites = False
```

```
    Else
        LeituraVerificaLimites = True
    End If


End Function

Private Function StringMesInglesTransformar(mm As Integer)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Converte um mês no formato numérico para o
'            formato mmm em inglês.
'-------------------------------------------------------------
'Argumentos: mm - Mês, representado por um número de 01 a 12
'-------------------------------------------------------------
'Retorna: "" Se número for inválido
'         String com representação "mmm" do número
'-------------------------------------------------------------
'Autor: GLF, 2003-04-07
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    Select Case mm
        Case 1: StringMesInglesTransformar = "JAN"
        Case 2: StringMesInglesTransformar = "FEB"
        Case 3: StringMesInglesTransformar = "MAR"
        Case 4: StringMesInglesTransformar = "APR"
        Case 5: StringMesInglesTransformar = "MAY"
        Case 6: StringMesInglesTransformar = "JUN"
        Case 7: StringMesInglesTransformar = "JUL"
        Case 8: StringMesInglesTransformar = "AUG"
        Case 9: StringMesInglesTransformar = "SEP"
        Case 10: StringMesInglesTransformar = "OCT"
        Case 11: StringMesInglesTransformar = "NOV"
        Case 12: StringMesInglesTransformar = "DEC"
    End Select

End Function
```

clsDisplay.cls

```
Private ctlDisplay As ctlSerialXML

Public Sub Init(ByRef ctlSerialDisplay As Object)

    'Pega a referencia para o controle gerenciador
    'de serial para o qual serao enviados os comandos
    Set ctlDisplay = ctlSerialDisplay

End Sub
Public Sub ProdutoMostrar(produtoNS As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Mostra, no display, informacoes sobre o produto
'            lido, na posicao superior esquerda.
'-------------------------------------------------------------
'Argumentos: produtoNS = Numero serial do produto lido,
'                        a ser mostrado no display
'-------------------------------------------------------------
'Autor: GLF, 2003-03-25
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim comando As String

    'Posiciona o cursor na primeira linha e primeira coluna
    comando = Chr(27) & "P" & Chr(1) & Chr(1)

    'Acrescenta ao comando posicionador o texto propriamente dito
    'que deverá ser mostrado. No caso, o serial do produto
    comando = comando & produtoNS

    'Envia o comando
    'ctlDisplay.Enviar comando

End Sub

Public Sub ComponenteMostrar(componenteNS As String)
```

```
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Mostra, no display, o serial do componente
'            lido, na posicao superior direita.
'-------------------------------------------------------------
'Argumentos: componenteNS = Numero serial do componente lido,
'                           a ser mostrado no display
'-------------------------------------------------------------
'Autor: GLF, 2003-03-25
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim comando As String

    'Posiciona o cursor na primeira linha e 13a coluna
    comando = Chr(27) & "P" & Chr(1) & Chr(13)

    'Acrescenta ao comando posicionador o texto propriamente dito
    'que deverá ser mostrado. No caso, o serial do componente
    comando = comando & componenteNS

    'Envia o comando
    'ctlDisplay.Enviar comando

End Sub

Public Sub MsgMostrar(mensagem As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Mostra, no display, uma msg na segunda
'            linha
'-------------------------------------------------------------
'Argumentos: mensagem: Mensagem a ser mostrada
'-------------------------------------------------------------
'Autor: GLF, 2003-03-25
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim comando As String

    'Posiciona o cursor na segunda linha, 1a coluna
    comando = Chr(27) & "P" & Chr(2) & Chr(1)

    'Acrescenta ao comando posicionador o texto propriamente dito
    'que deverá ser mostrado. No caso, a mensagem
    comando = comando & mensagem

    'Envia o comando
    'ctlDisplay.Enviar comando

End Sub

Public Sub TelaLimpar()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Limpa a tela do display
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Autor: GLF, 2003-03-28
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim comando As String

    comando = Chr(27) & "A"
    'ctlDisplay.Enviar comando

End Sub
```

Módulo Paletização

ctlPaletizacao.ctl

```
Option Explicit

Private estado As enmEstadoPaletizacao
Private centroSAP As String             'centro para consultar ordens SAP
Private inicializacaoSucesso As Boolean 'Verificacao de inicializacao do controle

Public linhaCodigo As String            'codigo da linha onde o controle esta rodando
Public codBarras As New clsCodigoBarras


Private Sub cmdOrdens_Click()
    frmOrdens.comportamento = PAL_FORM_ORDENS_CONSULTAR
    frmOrdens.Show vbModal
    Unload frmOrdens

End Sub

Private Sub UserControl_Initialize()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Inicializa o controle, suas configuracoes XML,
'            sua conexao com o banco e seta seu estado inicial.
'-------------------------------------------------------------
'Parâmetros:
'-------------------------------------------------------------
'Autor: GLF, 2003-03-11
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
On Error GoTo ErrorHandler
Dim reply As Integer

    'Inicializa sem sucesso
    inicializacaoSucesso = False

    MsgMostrar ("Inicializando")

    'Inicializa estado
    estado = PAL_ESTADO_INICIAL

    'Inicializacao baseada no XML
    If InitXML Then

        'Conecta com banco de dados
        reply = BDConectar(CAMINHO_LOCAL & ARQUIVO_MDB)

        Select Case reply
            Case CONEXAO_PATH_INVALIDO
                ErroMostrar MSG_ERRO_BANCO_PATH_INVALIDO
                inicializacaoSucesso = False
            Case CONEXAO_ERRO_GERAL
                ErroMostrar MSG_ERRO_BANCO_GERAL
                inicializacaoSucesso = False
            Case CONEXAO_OK
                inicializacaoSucesso = True
                MsgMostrar ("Iniciado com sucesso")
        End Select
    Else
    'Caso nao tenha conseguido inicializar as configuracoes
    'a partir do XML

        ErroMostrar MSG_ERRO_CARREGANDO_XML
        inicializacaoSucesso = False

    End If

Exit Sub

ErrorHandler:
    ErroMostrar MSG_ERRO_INTERNO & "[" & Err.Number & "]" & Err.Description
    inicializacaoSucesso = False
```

```
End Sub

Public Sub MsgMostrar(msg As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Mostra mensagem no label e a escreve
'            na janela de log. Mantem o semáforo verde.
'-------------------------------------------------------------
'Parâmetros: Msg: Mensagem a ser mostrada
'-------------------------------------------------------------
'Autor: GLF, 2003-02-13
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

    'mostra mensagem em preto
    lblMensagem.ForeColor = vbBlack
    lblMensagem.Caption = msg

    'escreve no log e corta o final
    txtLog = Format(Now, "dd/mm hh:nn") & "> " & msg & vbCrLf & txtLog
    txtLog = Left(txtLog, TAMANHO_MAX_LOG)

    'seta semaforo
    picVerde.Visible = True
    picVermelho.Visible = False


End Sub

Public Sub ErroMostrar(msgErro As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Mostra mensagem de erro no label e a escreve
'            na janela de log. Alterna o semáforo pra vermelho
'            e dorme 2 segundos
'-------------------------------------------------------------
'Parâmetros: msgErro: Mensagem a ser mostrada
'-------------------------------------------------------------
'Autor: GLF, 2003-02-13
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

    'mostra mensagem em vermelho
    lblMensagem.ForeColor = vbRed
    lblMensagem.Caption = msgErro

    'escreve no log e corta o final
    txtLog = Format(Now, "dd/mm hh:nn") & "# " & msgErro & vbCrLf & txtLog
    txtLog = Left(txtLog, TAMANHO_MAX_LOG)

    'seta semaforo pra vermelho
    picVerde.Visible = False
    picVermelho.Visible = True

    'Atualiza a tela
    Refresh

    'dorme 2 segundos
    Sleep (2000)

    'seta semaforo pra verde
    picVerde.Visible = True
    picVermelho.Visible = False

End Sub

Public Sub IndicadoresLimpar()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Limpa todos os indicadores da interface
'            principal (ordem, componente, palete, etc)
'-------------------------------------------------------------
'Autor: GLF, 2003-02-13
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
    lblOrdem.Caption = ""
    lblProduto.Caption = ""
    lblPlan.Caption = ""
    lblComponente.Caption = ""
```

```
    lblReal.Caption = ""
    lblPalete.Caption = ""
    lblQtPalete.Caption = ""
End Sub

Public Sub IndicadoresPaleteLimpar()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Limpa todos os indicadores da interface
'            relativos ao palete (NS do Palete, quantidade)
'-----------------------------------------------------------------
'Autor: GLF, 2003-02-26
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
    lblPalete.Caption = ""
    lblQtPalete.Caption = ""

End Sub


Public Function LeituraTimeout() As String
'*****************************************************************************
'Finalidade:    espera leitura da serial ate TIMEOUT
'Argumentos:    -
'Retorna:       leitura em caso de sucesso, vazio em caso de TimeOut
'Criacao:       GLF 2002-01-16
'Alteracoes:
'*****************************************************************************

Dim Leitura As String
Dim tempoExecucao As Long

    'espera leitura ou timeout
    While tempoExecucao < tempoTimeout And Leitura = ""
        DoEvents
        Sleep (INTERVALO_LEITURA)
        'se ha leitura na fila, pega
        Leitura = ctlLeitor.Ler
        tempoExecucao = tempoExecucao + INTERVALO_LEITURA
    Wend

    LeituraTimeout = Leitura

End Function

Public Function LeituraSerial() As String
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Fica aguardando a chegada de uma valor completo
'            (inicio, valor, fim) da serial sem interromper
'            outros processos.
'-----------------------------------------------------------------
'Parâmetros:
'-----------------------------------------------------------------
'Retorna: O valor lido, assim que este chegar
'-----------------------------------------------------------------
'Autor: GLF, 2003-02-13
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

Dim Leitura As String

    'Aguarda leitura
    While Leitura = ""
        DoEvents
        Sleep (INTERVALO_LEITURA)
        'Se há leitura na fila, pega o seu valor
        Leitura = ctlLeitor.Ler
    Wend

    LeituraSerial = Leitura

End Function

Private Function InitXML() As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Configura controle de leitura da porta serial e
'            configuracoes do codigo de barras do arquivo
```

```
'            XML principal de configuracao
'-------------------------------------------------------------
'Retorna:    TRUE, caso tudo se inicialize com sucesso
'            FALSE, se ocorrer erro em alguma coisa
'-------------------------------------------------------------
'Autor: GLF, 2003-01-29
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler

Dim xmlConfig As New DOMDocument26

Dim xmlRoot As Object
Dim xmlPaletizacao As Object
Dim xmlLinha As Object
Dim xmlLeitor As Object
Dim xmlCodigoBarras As Object

    'inicializa com status de erro
    InitXML = False

    'seta para carregar arquivo sincronizado
    xmlConfig.async = "false"

    MsgBox "antes de load"

    'carrega os dados de XML e separa apenas da genealogia
    xmlConfig.Load (CAMINHO_LOCAL & ARQUIVO_CONFIGURACAO)

    MsgBox "depois de load"

    Set xmlRoot = xmlConfig.selectSingleNode("GERAL")

    Set xmlLinha = xmlRoot.selectSingleNode("LINHA")

    'Le o codigo da linha onde o controle esta rodando
    linhaCodigo = xmlLinha.selectSingleNode("CODIGO").Text
    'Le o Centro SAP
    centroSAP = xmlLinha.selectSingleNode("CENTRO_SAP").Text

    Set xmlPaletizacao = xmlRoot.selectSingleNode("PALETIZACAO")

    'Inicializa Leitor
    Set xmlLeitor = xmlPaletizacao.selectSingleNode("LEITOR")
    ctlLeitor.init xmlLeitor

    'Le Timeout
    tempoTimeout = xmlPaletizacao.selectSingleNode("TEMPO_TIMEOUT").Text

    'carrega dados de codigo barras
    Set xmlCodigoBarras = xmlRoot.selectSingleNode("CODIGOBARRAS")
    codBarras.init xmlCodigoBarras

    'Se chegou ateh aqui, eh porque teoricamente deu tudo certo ;)
    InitXML = True

Exit Function

ErrorHandler:
    InitXML = False

End Function


'Private Sub UserControl_KeyDown(KeyCode As Integer, Shift As Integer)
'
'If KeyCode = vbKeyF12 Then
'    frmOrdens.Show vbModal
'End If
'
'End Sub

Private Sub UserControl_Show()
    'Inicializa maquina de estados, caso tenha ocorrido tudo certo
    'na inicializacao do controle
```

```
    If inicializacaoSucesso Then
        Dim maquina As New clsMaqPaletizacao
        maquina.init Me
        maquina.executar
    End If


End Sub

Public Sub BotaoOrdensClicar()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Funcao que dispara o evento cmdOrdens_Click do
'            controle. Necessária pois a captura da tecla
'            de atalho eh feita pelo HTML.
'-------------------------------------------------------------
'Retorna:
'-------------------------------------------------------------
'Autor: GLF, 2003-01-29
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    'Chama o evento
    cmdOrdens_Click

End Sub
Public Sub MostraOrdem(ieOrdem As String)
    lblOrdem.Caption = ieOrdem
End Sub

Public Sub MostraProduto(cdProduto As String)
    lblProduto.Caption = cdProduto
End Sub

Public Sub MostraOrdemPlanejado(qtPlanejada As String)
    lblPlan = qtPlanejada
End Sub

Public Sub MostraComponente(cdComponente As String)
    lblComponente = cdComponente
End Sub

Public Sub MostraOrdemRealizado(qtRealizada As String)

    lblReal = qtRealizada

    'Caso a quantidade realizada seja maior que a produzida, mostrar em vermelho
    If CInt(qtRealizada) > CInt(lblPlan.Caption) Then
        lblReal.ForeColor = vbRed
    Else
        lblReal.ForeColor = &H800000
    End If

End Sub

Public Sub MostraPalete(cdPalete As String)
    lblPalete = cdPalete
End Sub

Public Sub MostraPaleteQuantidade(qtPalete As String, qtMaxPorPalete As String)
    lblQtPalete = qtPalete & "/" & qtMaxPorPalete
End Sub
```

clsMaqPaletizacao

```
Option Explicit

'Maquina de estados para paletizacao

Private Leitura As String   'String que armazenara o valor da leitura da serial
Private sql As String       'armazena instrução sql
Private rs As New Recordset 'armazena os resultados dos queries sql
Private estado As enmEstadoPaletizacao
Private ctlPal As ctlPaletizacao 'Referencia para o controle de paletizacao que
```

```
                                    'instanciou a maquina
Private resultado
Private ordem As New clsOrdem

Public Sub init(ptrCtlPal As ctlPaletizacao)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Inicializa a maquina de estados
'-----------------------------------------------------------------
'Argumentos:
'-----------------------------------------------------------------
'Retorna:
'-----------------------------------------------------------------
'Autor: GLF, 2003-02-13
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
    Set ctlPal = ptrCtlPal
    estado = GEN_ESTADO_INICIAL

End Sub

Public Sub executar()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Metodo principal da maquina de estados, que
'            é executado ininterruptamente e gerencia
'            as chamadas de leituras, display e
'            transicoes entre estados
'-----------------------------------------------------------------
'Autor: GLF, 2003-02-13
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim ordemPalete As String

On Error GoTo ErrorHandler

    While True

    'Verifica em qual estado se encontra o sistema
    Select Case estado

        '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
        Case PAL_CHECAGEM_INICIAL
        '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
        'Checa o banco local para verificar se algum palete ficou
        'aberto, ou se alguma ordem estava iniciada.

            resultado = MaquinaEstadosAjustar

            'Dependendo do resultado, ajusta a maquina
            'de acordo
            Select Case resultado
                Case RESULT_ESTADO_PALETE_ABERTO
                    estado = PAL_PALETE_ABERTO
                Case RESULT_ESTADO_ORDEM_INICIADA
                    estado = PAL_ORDEM_INICIADA
                Case RESULT_ESTADO_INICIAL
                    estado = PAL_ESTADO_INICIAL
                Case RESULT_ERRO_PALETE_ABRIR, RESULT_ERRO_ORDEM_INICIAR
                    ctlPal.ErroMostrar MSG_PALETIZACAO_ERRO_INICIANDO_MAQUINA_ESTADOS
                    Exit Sub
                Case Else
                    ctlPal.ErroMostrar MSG_ERRO_GENERICO
                    Exit Sub

            End Select

        '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
        Case PAL_ESTADO_INICIAL
        '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

            'Limpa indicadores
            ctlPal.IndicadoresLimpar

            'Mostra mensagem inicial
            ctlPal.MsgMostrar MSG_PALETIZACAO_SEM_ORDEM_INICIADA

            'Aguarda leitura de algum comando
```

```
    Leitura = ctlPal.LeituraSerial

    'Mostra comando lido
    ctlPal.MsgMostrar "* " & Leitura & " *"

    Select Case Leitura
        '-------------------------------------------
        'INICIA ORDEM
        '-------------------------------------------
        Case CMD_INICIAR_ORDEM

            'Manda iniciar ordem
            resultado = OrdemIniciar

            Select Case resultado

                Case RESULT_ORDEM_INICIADA
                    estado = PAL_ORDEM_INICIADA
                Case RESULT_ERRO_ORDEM_INICIAR
                    ctlPal.ErroMostrar MSG_ORDEM_ERRO_INICIAR
                    estado = PAL_ESTADO_INICIAL
                Case RESULT_ERRO_NENHUMA_ORDEM_SELECIONADA
                    estado = PAL_ESTADO_INICIAL
                Case Else
                    ctlPal.ErroMostrar MSG_ORDEM_ERRO_INICIAR
                    estado = PAL_ESTADO_INICIAL

            End Select

        '-------------------------------------------
        'QUALQUER OUTRA COISA
        '-------------------------------------------
        Case Else

            ctlPal.ErroMostrar MSG_LEITURA_INVALIDA
            estado = PAL_ESTADO_INICIAL

    End Select 'leitura

'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Case PAL_ORDEM_INICIADA
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

    'Mostra msg indicando estado
    ctlPal.MsgMostrar MSG_ORDEM_INICIADA

    'Aguarda leitura de algum comando
    Leitura = ctlPal.LeituraSerial

    'Mostra comando lido
    ctlPal.MsgMostrar "* " & Leitura & " *"

    Select Case Leitura
        '-------------------------------------------
        'ABRIR PALETE
        '-------------------------------------------
        Case CMD_ABRIR_PALETE

            'Manda abrir o palete. Caso ocorra erro de associacao
            'palete/ordem, a ordem à qual o palete está associado
            'será armazenada em ordemPalete
            resultado = PaleteAbrir(ordemPalete)

            Select Case resultado
                Case RESULT_PALETE_ABERTO
                    estado = PAL_PALETE_ABERTO

                Case RESULT_ERRO_PALETE_ORDEM_INCORRETA
                    ctlPal.ErroMostrar MSG_PALETIZACAO_PALETE_EM_ORDEM_INCORRETA & ordemPalete
                    estado = PAL_ORDEM_INICIADA

                Case RESULT_ERRO_PALETE_TRANSFERENCIA
                    ctlPal.ErroMostrar MSG_PALETIZACAO_PALETE_JA_TRANSFERIDO
                    estado = PAL_ORDEM_INICIADA
```

```
                Case RESULT_ERRO_PALETE_ABRIR_BANCO
                    ctlPal.ErroMostrar MSG_PALETE_ERRO_ABRIR_BANCO
                    estado = PAL_ORDEM_INICIADA

                Case RESULT_ERRO_TIMEOUT
                    ctlPal.ErroMostrar MSG_TIMEOUT
                    estado = PAL_ORDEM_INICIADA

                Case RESULT_ERRO_PALETE_ABRIR_GENERICO
                    ctlPal.ErroMostrar MSG_PALETE_ERRO_ABRIR
                    estado = PAL_ORDEM_INICIADA

                Case RESULT_ERRO_OPERACAO_CANCELADA
                    ctlPal.ErroMostrar MSG_OPERACAO_CANCELADA
                    estado = PAL_ORDEM_INICIADA

                Case RESULT_ERRO_LEITURA_INVALIDA
                    ctlPal.ErroMostrar MSG_LEITURA_INVALIDA
                    estado = PAL_ORDEM_INICIADA

                Case Else
                    ctlPal.ErroMostrar MSG_PALETE_ERRO_ABRIR
                    estado = PAL_ORDEM_INICIADA
            End Select

        '--------------------------------------------
        'ENCERRAR ORDEM
        '--------------------------------------------
        Case CMD_ENCERRAR_ORDEM

            resultado = OrdemEncerrar
            Select Case resultado

                Case RESULT_ORDEM_ENCERRADA
                    ctlPal.MsgMostrar MSG_ORDEM_ENCERRADA
                    estado = PAL_ESTADO_INICIAL

                Case RESULT_ERRO_LEITURA_INVALIDA
                    ctlPal.ErroMostrar MSG_LEITURA_INVALIDA
                    estado = PAL_ORDEM_INICIADA

                Case RESULT_ERRO_OPERACAO_CANCELADA
                    ctlPal.ErroMostrar MSG_OPERACAO_CANCELADA
                    estado = PAL_ORDEM_INICIADA

                Case RESULT_ERRO_TIMEOUT
                    ctlPal.ErroMostrar MSG_TIMEOUT
                    estado = PAL_ORDEM_INICIADA

                Case RESULT_ERRO_ORDEM_ENCERRAR_GENERICO
                    ctlPal.ErroMostrar MSG_ORDEM_ERRO_ENCERRAR
                    estado = PAL_ORDEM_INICIADA

                Case Else
                    ctlPal.ErroMostrar MSG_ORDEM_ERRO_ENCERRAR
                    estado = PAL_ORDEM_INICIADA

            End Select

        '--------------------------------------------
        'QUALQUER OUTRA COISA
        '--------------------------------------------
        Case Else

            ctlPal.ErroMostrar MSG_LEITURA_INVALIDA
            estado = PAL_ORDEM_INICIADA

    End Select


'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Case PAL_PALETE_ABERTO
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
```

```
'Mostra mensagem indicando estado
ctlPal.MsgMostrar MSG_PALETE_ABERTO

'Aguarda leitura de algum comando
Leitura = ctlPal.LeituraSerial

'Mostra comando lido
ctlPal.MsgMostrar "* " & Leitura & " *"

Select Case Leitura

    '-----------------------------------------
    'FECHAR PALETE
    '-----------------------------------------

    Case CMD_FECHAR_PALETE
        resultado = PaleteFechar

        Select Case resultado
            Case RESULT_PALETE_FECHADO
                ctlPal.MsgMostrar MSG_PALETE_FECHADO
                estado = PAL_ORDEM_INICIADA

            Case RESULT_ERRO_CONFIRMACAO_PALETE
                ctlPal.ErroMostrar MSG_PALETIZACAO_ERRO_CONFIRMACAO_PALETE
                estado = PAL_PALETE_ABERTO

            Case RESULT_ERRO_PALETE_FECHAR_BANCO
                ctlPal.ErroMostrar MSG_PALETE_ERRO_FECHAR_BANCO
                estado = PAL_PALETE_ABERTO

            Case RESULT_ERRO_TIMEOUT
                ctlPal.ErroMostrar MSG_TIMEOUT
                estado = PAL_PALETE_ABERTO

            Case RESULT_ERRO_PALETE_FECHAR_GENERICO
                ctlPal.ErroMostrar MSG_PALETE_ERRO_FECHAR
                estado = PAL_PALETE_ABERTO

            Case Else
                ctlPal.ErroMostrar MSG_PALETE_ERRO_FECHAR
                estado = PAL_PALETE_ABERTO

        End Select

    '-----------------------------------------
    'DESMONTAR PALETE
    '-----------------------------------------
    Case CMD_DESMONTAR_PALETE

        resultado = PaleteDesmontar

        Select Case resultado

            Case RESULT_PALETE_DESMONTADO
                ctlPal.MsgMostrar MSG_PALETE_DESMONTADO
                estado = PAL_ORDEM_INICIADA

            Case RESULT_ERRO_PALETE_DESMONTAR_GENERICO
                ctlPal.ErroMostrar MSG_PALETE_ERRO_DESMONTAR
                estado = PAL_PALETE_ABERTO

            Case RESULT_ERRO_PALETE_DESMONTAR_BANCO
                ctlPal.ErroMostrar MSG_PALETE_ERRO_DESMONTAR_BANCO
                estado = PAL_PALETE_ABERTO

            Case RESULT_ERRO_OPERACAO_CANCELADA
                ctlPal.ErroMostrar MSG_OPERACAO_CANCELADA
                estado = PAL_PALETE_ABERTO

            Case RESULT_ERRO_LEITURA_INVALIDA
                ctlPal.ErroMostrar MSG_LEITURA_INVALIDA
                estado = PAL_PALETE_ABERTO
```

```
            Case RESULT_ERRO_TIMEOUT
                ctlPal.ErroMostrar MSG_TIMEOUT
                estado = PAL_PALETE_ABERTO

            Case Else
                ctlPal.ErroMostrar MSG_PALETE_ERRO_DESMONTAR
                estado = PAL_PALETE_ABERTO

        End Select

    '----------------------------------------
    'RETIRAR PRODUTO
    '----------------------------------------
    Case CMD_REMOVER_PRODUTO

        resultado = ProdutoRemover

        Select Case resultado

            Case RESULT_PRODUTO_REMOVIDO
                ctlPal.MsgMostrar MSG_PRODUTO_REMOVIDO
                estado = PAL_PALETE_ABERTO

            Case RESULT_ERRO_PRODUTO_EM_OUTRO_PALETE
                ctlPal.ErroMostrar MSG_PALETIZACAO_PRODUTO_EM_OUTRO_PALETE
                estado = PAL_PALETE_ABERTO

            Case RESULT_ERRO_PRODUTO_REMOVER_BANCO
                ctlPal.ErroMostrar MSG_PALETIZACAO_ERRO_PRODUTO_REMOVER_BANCO
                estado = PAL_PALETE_ABERTO

            Case RESULT_ERRO_TIMEOUT
                ctlPal.ErroMostrar MSG_TIMEOUT
                estado = PAL_PALETE_ABERTO

            Case Else
                ctlPal.ErroMostrar MSG_PALETIZACAO_ERRO_PRODUTO_REMOVER_GENERICO
                estado = PAL_PALETE_ABERTO

        End Select

    '----------------------------------------
    'LEITURA DE ALGO (Produto, provavelmente)
    '----------------------------------------
    Case Else

        resultado = ProdutoInserir

        Select Case resultado

            Case RESULT_PRODUTO_INSERIDO
                ctlPal.MsgMostrar MSG_PALETIZACAO_PRODUTO_INSERIDO
                estado = PAL_PALETE_ABERTO

            Case RESULT_PRODUTO_INSERIDO_COMPLETOU_PALETE
                'Indica que o palete sera fechado automaticamente
                'pois este produto encerrou a capacidade do palete
                ctlPal.MsgMostrar MSG_PALETIZACAO_PRODUTO_INSERIDO
                ctlPal.MsgMostrar MSG_PALETIZACAO_FECHAMENTO_AUTOMATICO

                'Manda fechar o palete automaticamente
                resultado = PaleteFecharAutomaticamente
                '------------------------------------------------------------------------
                Select Case resultado
                    Case RESULT_PALETE_FECHADO
                        ctlPal.MsgMostrar MSG_PALETE_FECHADO
                        estado = PAL_ORDEM_INICIADA

                    Case RESULT_ERRO_PALETE_FECHAR_GENERICO
                        ctlPal.ErroMostrar MSG_PALETE_ERRO_FECHAR_AUTOMATICAMENTE
                        estado = PAL_PALETE_ABERTO

                    Case Else
```

```
                        ctlPal.ErroMostrar MSG_PALETE_ERRO_FECHAR_AUTOMATICAMENTE
                        estado = PAL_PALETE_ABERTO
                    End Select
                    '-------------------------------------------------------------------

                Case RESULT_ERRO_PRODUTO_INEXISTENTE
                    ctlPal.ErroMostrar MSG_PALETIZACAO_PRODUTO_NAO_MONTADO
                    estado = PAL_PALETE_ABERTO

                Case RESULT_ERRO_PRODUTO_SEM_COMPONENTE
                    ctlPal.ErroMostrar MSG_PALETIZACAO_PRODUTO_SEM_COMPONENTE
                    estado = PAL_PALETE_ABERTO

                Case RESULT_ERRO_MONTAGEM_INCORRETA
                    ctlPal.ErroMostrar MSG_PALETIZACAO_PRODUTO_MONTADO_INCORRETAMENTE
                    estado = PAL_PALETE_ABERTO

                Case RESULT_ERRO_PRODUTO_JA_INSERIDO
                    ctlPal.ErroMostrar MSG_PALETIZACAO_PRODUTO_JA_INSERIDO
                    estado = PAL_PALETE_ABERTO

                Case RESULT_ERRO_PRODUTO_JA_INSERIDO_OUTRO_PALETE
                    ctlPal.ErroMostrar MSG_PALETIZACAO_PRODUTO_JA_INSERIDO_OUTRO_PALETE
                    estado = PAL_PALETE_ABERTO

                Case RESULT_ERRO_PRODUTO_INCORRETO
                    ctlPal.ErroMostrar MSG_PALETIZACAO_PRODUTO_INCORRETO
                    estado = PAL_PALETE_ABERTO

                'Esse erro ocorrerá quando o usuário abrir um palete
                'que já está cheio e tentar inserir um produto no mesmo.
                'O Fechamento automático também será disparado.

                Case RESULT_ERRO_PALETE_CHEIO
                    'Mostra o erro
                    ctlPal.ErroMostrar MSG_PALETIZACAO_ERRO_PALETE_CHEIO

                    'Manda fechar o palete automaticamente
                    resultado = PaleteFecharAutomaticamente
                    '-------------------------------------------------------------------
                    Select Case resultado
                        Case RESULT_PALETE_FECHADO
                            ctlPal.MsgMostrar MSG_PALETE_FECHADO
                            estado = PAL_ORDEM_INICIADA

                        Case RESULT_ERRO_PALETE_FECHAR_GENERICO
                            ctlPal.ErroMostrar MSG_PALETE_ERRO_FECHAR_AUTOMATICAMENTE
                            estado = PAL_PALETE_ABERTO

                        Case Else
                            ctlPal.ErroMostrar MSG_PALETE_ERRO_FECHAR_AUTOMATICAMENTE
                            estado = PAL_PALETE_ABERTO
                    End Select
                    '-------------------------------------------------------------------

                Case RESULT_ERRO_PRODUTO_INSERIR_BANCO
                    ctlPal.ErroMostrar MSG_PALETIZACAO_ERRO_PRODUTO_INSERIR_BANCO
                    estado = PAL_PALETE_ABERTO

                Case RESULT_ERRO_PRODUTO_INSERIR_GENERICO
                    ctlPal.ErroMostrar MSG_PALETIZACAO_ERRO_PRODUTO_INSERIR_ERRO
                    estado = PAL_PALETE_ABERTO

                Case RESULT_ERRO_LEITURA_INVALIDA
                    ctlPal.ErroMostrar MSG_LEITURA_INVALIDA
                    estado = PAL_PALETE_ABERTO

                Case Else
                    ctlPal.ErroMostrar MSG_LEITURA_INVALIDA
                    estado = PAL_PALETE_ABERTO
            End Select

    End Select
```

```
    End Select

    Wend

Exit Sub

ErrorHandler:
    MsgBox Err.Description, Err.Number
End Sub

Private Function OrdemIniciar() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Inicia uma ordem
'-----------------------------------------------------------
'Argumentos:
'-----------------------------------------------------------
'Retorna: RESULT_ORDEM_INICIADA
'         RESULT_ERRO_NENHUMA_ORDEM_SELECIONADA
'         RESULT_ERRO_ORDEM_INICIAR
'-----------------------------------------------------------
'Autor: GLF, 2003-02-18
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim ieOrdemAIniciar As String   'Armazena o nr da ordem escolhida pelo usuario,
                                'a ser iniciada

    ctlPal.MsgMostrar MSG_CONSULTANDO_ORDENS

    'Mostra Janela com opcoes
    frmOrdens.comportamento = PAL_FORM_ORDEM_INICIAR
    frmOrdens.Show vbModal

    'Salva a possivel ordem escolhida
    ieOrdemAIniciar = frmOrdens.ieOrdemAIniciar
    'Fecha a form
    Unload frmOrdens

    'Se dados estao OK
        If ieOrdemAIniciar <> "" Then

            If ordem.Iniciar(ieOrdemAIniciar) Then

                'Se a funcao executou e retornou true, eh porque a ordem
                'foi iniciada com sucesso
                OrdemIniciar = RESULT_ORDEM_INICIADA

                'Atualiza os mostradores
                ctlPal.MostraOrdem ieOrdemAIniciar
                ctlPal.MostraProduto ordem.myProdutoCD
                ctlPal.MostraComponente ordem.myComponenteCD
                ctlPal.MostraOrdemPlanejado ordem.myQtPlan
                ctlPal.MostraOrdemRealizado ordem.myQtApont + ordem.myQtApontSAP

                Exit Function
            Else
                OrdemIniciar = RESULT_ERRO_ORDEM_INICIAR
                Exit Function
            End If

        Else
            OrdemIniciar = RESULT_ERRO_NENHUMA_ORDEM_SELECIONADA
            Exit Function
        End If


End Function

Private Function OrdemEncerrar() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Encerra uma ordem
'-----------------------------------------------------------
'Argumentos:
'-----------------------------------------------------------
'Retorna: - RESULT_ORDEM_ENCERRADA
'           RESULT_ERRO_ORDEM_ENCERRAR_GENERICO
```

```
'           RESULT_ERRO_OPERACAO_CANCELADA
'           RESULT_ERRO_LEITURA_INVALIDA
'           RESULT_ERRO_TIMEOUT
'-------------------------------------------------------------
'Autor: GLF, 2003-02-18
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

    'Pede confirmacao do encerramento
    ctlPal.MsgMostrar MSG_ORDEM_CONFIRMA_ENCERRAMENTO

    'Faz a leitura da serial até o tempo máximo permitido
    Leitura = ctlPal.LeituraTimeout

    'Se leu algo
    If Leitura <> "" Then

        'Mostra leitura
        ctlPal.MsgMostrar "* " & Leitura & " *"

        'Confirmou o encerramento
        If Leitura = CMD_SIM Then

            'Encerra a ordem
            If ordem.Encerrar Then
                OrdemEncerrar = RESULT_ORDEM_ENCERRADA
                Exit Function
            Else
                OrdemEncerrar = RESULT_ERRO_ORDEM_ENCERRAR_GENERICO
            End If

        'Negou a confirmacao de encerramento
        ElseIf Leitura = CMD_NAO Then
            OrdemEncerrar = RESULT_ERRO_OPERACAO_CANCELADA
            Exit Function

        'Leu o comando de cancelamento
        ElseIf Leitura = CMD_CANCELAR Then 'Operacao foi cancelada
            OrdemEncerrar = RESULT_ERRO_OPERACAO_CANCELADA
            Exit Function

        'Leu qualquer outra coisa que nao o que deveria ter lido
        Else
            OrdemEncerrar = RESULT_ERRO_LEITURA_INVALIDA
            Exit Function
        End If

    Else 'Dormiu e nao leu nada
        OrdemEncerrar = RESULT_ERRO_TIMEOUT
        Exit Function
    End If

Exit Function

ErrorHandler:
    OrdemEncerrar = RESULT_ERRO_ORDEM_ENCERRAR_GENERICO


End Function


Private Function PaleteAbrir(Optional ByRef ordemAssocAnteriormente As String) As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Recebe leitura de palete, valida e manda o objeto
'            ordem abrir o palete
'-------------------------------------------------------------
'Argumentos: ordemAssocAnteriormente
'                - Ordem à qual está associado o palete
'                  no banco de dados. Necessário para
'                  que o usuário saiba a qual ordem
'                  aquele palete está associado.
'-------------------------------------------------------------
'Retorna: RESULT_PALETE_ABERTO
'         RESULT_ERRO_PALETE_ORDEM_INCORRETA
'         RESULT_ERRO_PALETE_TRANSFERENCIA
```

```
'         RESULT_ERRO_PALETE_ABRIR_BANCO
'         RESULT_ERRO_PALETE_ABRIR_GENERICO
'         RESULT_ERRO_LEITURA_INVALIDA
'         RESULT_ERRO_TIMEOUT
'-------------------------------------------------------------
'Autor: GLF, 2003-02-18
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim Leitura As String
Dim resultado As Long
Dim palCD As String        'Codigo do palete. Nao existe e nao será usado. Para manter o padrao
                           'na chamada da funcao ValoresRetornar do codigo de barras.
Dim palNS As String        'Numero serial do palete

On Error GoTo ErrorHandler

    ctlPal.MsgMostrar MSG_PALETIZACAO_LEIA_PALETE

    'Faz a leitura da serial até o tempo máximo permitido
    Leitura = ctlPal.LeituraTimeout

    If Leitura <> "" Then

        'Mostra leitura
        ctlPal.MsgMostrar "* " & Leitura & " *"

        'Verifica se o serial do palete esta correto, de acordo com as conf. XML
        If ctlPal.codBarras.ValoresRetornar(Leitura, palCD, palNS) Then

            resultado = ordem.PaleteAbrir(palNS, ctlPal.linhaCodigo, ordemAssocAnteriormente)

            'Se abriu com sucesso
            If resultado > 0 Then

                'Atualiza mostrador
                ctlPal.MostraPalete Leitura
                ctlPal.MostraOrdemRealizado ordem.myQtApont + ordem.myQtApontSAP +
ordem.GetPaleteQuantidadeApontada
                ctlPal.MostraPaleteQuantidade ordem.GetPaleteQuantidadeApontada, ordem.myQtPalete

            End If

            PaleteAbrir = resultado
            Exit Function

        Else 'Erro de codigo invalido/configuracoes invalidas
            PaleteAbrir = RESULT_ERRO_LEITURA_INVALIDA
            Exit Function
        End If

    Else 'Erro de timeout na leitura
        PaleteAbrir = RESULT_ERRO_TIMEOUT
        Exit Function
    End If

Exit Function

ErrorHandler:
    PaleteAbrir = RESULT_ERRO_PALETE_ABRIR_GENERICO

End Function

Private Function PaleteFechar() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Fecha um palete aberto, solicitando confirmacao
'            da leitura do codigo de barras do palete
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Retorna: RESULT_PALETE_FECHADO
'         RESULT_ERRO_CONFIRMACAO_PALETE
'         RESULT_ERRO_PALETE_FECHAR_BANCO
'         RESULT_ERRO_TIMEOUT
'         RESULT_ERRO_PALETE_FECHAR_GENERICO
'-------------------------------------------------------------
```

```
'Autor: GLF, 2003-02-26
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler
Dim sql As String
Dim resultado As Long

    ctlPal.MsgMostrar MSG_PALETIZACAO_LEIA_PALETE

    'Faz a leitura da serial até o tempo máximo permitido
    Leitura = ctlPal.LeituraTimeout

    If Leitura <> "" Then

        'Mostra leitura
        ctlPal.MsgMostrar "* " & Leitura & " *"

        'Verifica se leitura eh igual ao palete com que se esta
        'trabalhando
        If Leitura = ordem.GetPaleteNS Then
            'Confirmado o fechamento do palete!
            resultado = ordem.PaleteFechar

            'Se fechou com sucesso
            If resultado > 0 Then
                ctlPal.IndicadoresPaleteLimpar
            End If

            'Retorna resultado
            PaleteFechar = resultado
            Exit Function

        Else
            PaleteFechar = RESULT_ERRO_CONFIRMACAO_PALETE
            Exit Function
        End If

    Else 'Erro de timeout na leitura
        PaleteFechar = RESULT_ERRO_TIMEOUT
        Exit Function
    End If


Exit Function

ErrorHandler:
    PaleteFechar = RESULT_ERRO_PALETE_FECHAR_GENERICO

End Function

Public Function PaleteDesmontar() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Desmonta um palete aberto. Solicita confirmacao
'            para tanto.
'-----------------------------------------------------------
'Argumentos:
'-----------------------------------------------------------
'Retorna: RESULT_PALETE_DESMONTADO
'         RESULT_ERRO_PALETE_DESMONTAR_GENERICO
'         RESULT_ERRO_PALETE_DESMONTAR_BANCO
'         RESULT_ERRO_OPERACAO_CANCELADA
'         RESULT_ERRO_LEITURA_INVALIDA
'         RESULT_ERRO_TIMEOUT
'-----------------------------------------------------------
'Autor: GLF, 2003-02-26
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    'Pede confirmacao
    ctlPal.MsgMostrar MSG_PALETIZACAO_CONFIRMA_DESMONTAGEM

    'Faz a leitura da serial até o tempo máximo permitido
    Leitura = ctlPal.LeituraTimeout

    'Se leu algo
    If Leitura <> "" Then
```

```
        'Mostra leitura
        ctlPal.MsgMostrar "* " & Leitura & " *"

        'Confirmou a desmontagem
        If Leitura = CMD_SIM Then

            'Manda desmontar o palete
            resultado = ordem.PaleteDesmontar

            'Se conseguiu desmontar com sucesso
            If resultado > 0 Then
                'Limpa os indicadores do palete desmontado
                ctlPal.IndicadoresPaleteLimpar
                'Atualiza indicador da ordem, para retirar do total de
                'itens realizados da ordem o nr de produtos que estava sendo
                'contado para este palete
                ctlPal.MostraOrdemRealizado ordem.myQtApont + ordem.myQtApontSAP
            End If

            PaleteDesmontar = resultado

        'Negou a confirmacao da desmontagem
        ElseIf Leitura = CMD_NAO Then
            PaleteDesmontar = RESULT_ERRO_OPERACAO_CANCELADA

        'Leu o comando de cancelamento
        ElseIf Leitura = CMD_CANCELAR Then 'Operacao foi cancelada
            PaleteDesmontar = RESULT_ERRO_OPERACAO_CANCELADA

        'Leu qualquer outra coisa que nao o que deveria ter lido
        Else
            PaleteDesmontar = RESULT_ERRO_LEITURA_INVALIDA
        End If

    Else 'Dormiu e nao leu nada
        PaleteDesmontar = RESULT_ERRO_TIMEOUT
    End If

Exit Function
ErrorHandler:
    PaleteDesmontar = RESULT_ERRO_PALETE_DESMONTAR_GENERICO

End Function


Private Function PaleteFecharAutomaticamente() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Fecha um palete aberto, sem solicitar confirmacoes
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Retorna: RESULT_PALETE_FECHADO
'         RESULT_ERRO_PALETE_FECHAR_BANCO
'         RESULT_ERRO_PALETE_FECHAR_GENERICO
'-------------------------------------------------------------
'Autor: GLF, 2003-02-26
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler
Dim sql As String
Dim resultado As Long

            'Confirmado o fechamento do palete!
            resultado = ordem.PaleteFechar

            'Se fechou com sucesso
            If resultado > 0 Then
                ctlPal.IndicadoresPaleteLimpar
            End If

            'Retorna resultado
            PaleteFecharAutomaticamente = resultado
            Exit Function
```

```
Exit Function

ErrorHandler:
    PaleteFecharAutomaticamente = RESULT_ERRO_PALETE_FECHAR_GENERICO

End Function

Private Function ProdutoInserir() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Insere um produto no palete aberto
'---------------------------------------------------------------
'Argumentos:
'---------------------------------------------------------------
'Retorna: RESULT_PRODUTO_INSERIDO
'         RESULT_ERRO_PRODUTO_INEXISTENTE
'         RESULT_ERRO_PRODUTO_SEM_COMPONENTE
'         RESULT_ERRO_MONTAGEM_INCORRETA
'         RESULT_ERRO_PRODUTO_JA_INSERIDO
'         RESULT_ERRO_PRODUTO_JA_INSERIDO_OUTRO_PALETE
'         RESULT_ERRO_PRODUTO_INCORRETO
'         RESULT_ERRO_PALETE_CHEIO
'         RESULT_ERRO_PRODUTO_INSERIR_BANCO
'         RESULT_ERRO_PRODUTO_INSERIR_GENERICO
'         RESULT_ERRO_LEITURA_INVALIDA
'---------------------------------------------------------------
'Autor: GLF, 2003-02-19
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim resultado As Long
Dim prodCD As String        'Codigo do produto
Dim prodNS As String        'Numero serial do produto

On Error GoTo ErrorHandler

    'Extrai o numero de série e codigo do código do produto lido
    If ctlPal.codBarras.ValoresRetornar(Leitura, prodCD, prodNS) Then

        resultado = ordem.ProdutoInserir(prodCD, prodNS)

        'Se inseriu com sucesso
        If resultado > 0 Then

            'Atualiza mostrador
            ctlPal.MostraOrdemRealizado ordem.myQtApont + ordem.myQtApontSAP +
ordem.GetPaleteQuantidadeApontada
            ctlPal.MostraPaleteQuantidade ordem.GetPaleteQuantidadeApontada, ordem.myQtPalete

        End If

        ProdutoInserir = resultado
        Exit Function

    Else 'Erro de codigo invalido/configuracoes invalidas
        ProdutoInserir = RESULT_ERRO_LEITURA_INVALIDA
        Exit Function
    End If

Exit Function

ErrorHandler:
    ProdutoInserir = RESULT_ERRO_PRODUTO_INSERIR_GENERICO

End Function

Private Function ProdutoRemover() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Remove um produto no palete aberto
'---------------------------------------------------------------
'Argumentos:
'---------------------------------------------------------------
'Retorna: RESULT_PRODUTO_REMOVIDO
'         RESULT_ERRO_PRODUTO_EM_OUTRO_PALETE
'         RESULT_ERRO_PRODUTO_REMOVER_BANCO
'         RESULT_ERRO_PRODUTO_REMOVER_GENERICO
'         RESULT_ERRO_TIMEOUT
```

```
'-------------------------------------------------------------
'Autor: GLF, 2003-02-27
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim resultado As Long
Dim prodCD As String        'Codigo do produto
Dim prodNS As String        'Numero serial do produto

On Error GoTo ErrorHandler

    'Pede confirmacao
    ctlPal.MsgMostrar MSG_LEIA_PRODUTO

    'Faz a leitura da serial até o tempo máximo permitido
    Leitura = ctlPal.LeituraTimeout

    'Se leu algo
    If Leitura <> "" Then

        'Mostra leitura
        ctlPal.MsgMostrar "* " & Leitura & " *"

        'Extrai o numero de série e codigo do código do produto lido
        If ctlPal.codBarras.ValoresRetornar(Leitura, prodCD, prodNS) Then

            resultado = ordem.ProdutoRemover(prodCD, prodNS)

            'Se removeu com sucesso
            If resultado > 0 Then

                'Atualiza mostrador
                ctlPal.MostraOrdemRealizado ordem.myQtApont + ordem.myQtApontSAP +
ordem.GetPaleteQuantidadeApontada
                ctlPal.MostraPaleteQuantidade ordem.GetPaleteQuantidadeApontada, ordem.myQtPalete

            End If

            ProdutoRemover = resultado
            Exit Function

        Else 'Erro de codigo invalido/configuracoes invalidas
            ctlPal.ErroMostrar MSG_LEITURA_INVALIDA
            ProdutoRemover = RESULT_ERRO_LEITURA_INVALIDA
            Exit Function
        End If

    Else 'Dormiu e nao leu nada
        ProdutoRemover = RESULT_ERRO_TIMEOUT
    End If

Exit Function

ErrorHandler:
    ProdutoRemover = RESULT_ERRO_PRODUTO_REMOVER_GENERICO

End Function

Private Function MaquinaEstadosAjustar() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: - Verifica em que estado foi deixado o sistema,
'              quando desligado/resetado/etc.
'            - Ajusta a maquina de estados, preenchendo os
'              objetos Ordem e Palete e mostrando os valores
'              de produtos lidos, etc, no controle.
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Retorna: RESULT_ESTADO_PALETE_ABERTO
'         RESULT_ESTADO_ORDEM_INICIADA
'         RESULT_ESTADO_INICIAL
'         RESULT_ERRO_ORDEM_INICIAR
'         RESULT_ERRO_PALETE_ABRIR
'-------------------------------------------------------------
'Autor: GLF, 2003-02-21
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
```

```
Dim sql As String
Dim rs As New Recordset
Dim sucesso As Boolean

    'Verifica se existe algum palete aberto
    sql = "SELECT * FROM paletes WHERE hc_status = '" & PALETE_HC_STATUS_ABERTO & "'"
    rs.Open sql, Cnn

    If rs.EOF Then
        'Nao tem nenhum palete aberto! Vamos ver se existe alguma ordem iniciada!
        rs.Close
        sql = "SELECT * FROM ordens WHERE hc_status = '" & ORDEM_HC_STATUS_INICIADA & "'"
        rs.Open sql, Cnn

        If rs.EOF Then
            '-----------------------------------
            'SEM ORDEM INICIADA NEM PALETE ABERTO
            '-----------------------------------
            'Nao faz nada, retorna o resultado dizendo que esta no estado inicial mesmo
            rs.Close
            MaquinaEstadosAjustar = RESULT_ESTADO_INICIAL

        Else
        '-----------------------------------
        'ORDEM INICIADA (SEM PALETE ABERTO)
        '-----------------------------------
            'Manda ordem se iniciar passando as informacoes
            sucesso = ordem.IniciarAjustando(rs!ie_ordem, rs!qt_planejada, _
                                             rs!qt_apontada, rs!qt_apontada_sap, _
                                             rs!cd_produto, rs!cd_componente, _
                                             rs!qt_palete)

            If sucesso Then

                'Fecha RecordSet
                rs.Close

                'Mostra valores da ordem no controle
                ctlPal.MostraOrdem ordem.myIe
                ctlPal.MostraOrdemPlanejado ordem.myQtPlan
                ctlPal.MostraOrdemRealizado ordem.myQtApont + ordem.myQtApontSAP +
ordem.GetPaleteQuantidadeApontada
                ctlPal.MostraProduto ordem.myProdutoCD
                ctlPal.MostraComponente ordem.myComponenteCD

                'Retorna o estado
                MaquinaEstadosAjustar = RESULT_ESTADO_ORDEM_INICIADA

            Else 'Sem sucesso na inicializacao do objeto ordem
                MaquinaEstadosAjustar = RESULT_ERRO_ORDEM_INICIAR
            End If

        End If 'Checagem de ordem iniciada

    Else 'Se existe algum palete aberto,
    '-------------
    'PALETE ABERTO
    '-------------

        'Manda ordem se inicializar já abrindo o palete em questao
        sucesso = ordem.IniciarComPalete(rs!ns_palete, rs!cd_produto, rs!cd_linha, rs!ie_ordem)

        If sucesso Then

            'Fecha RecordSet
            rs.Close

            'Mostra valores da ordem no controle
            ctlPal.MostraOrdem ordem.myIe
            ctlPal.MostraOrdemPlanejado ordem.myQtPlan
            ctlPal.MostraOrdemRealizado ordem.myQtApont + ordem.myQtApontSAP +
ordem.GetPaleteQuantidadeApontada
            ctlPal.MostraProduto ordem.myProdutoCD
            ctlPal.MostraComponente ordem.myComponenteCD
```

```
            'Mostra valores do palete no controle
            ctlPal.MostraPalete ordem.GetPaleteNS
            ctlPal.MostraPaleteQuantidade ordem.GetPaleteQuantidadeApontada, ordem.myQtPalete

            'Retorna o estado
            MaquinaEstadosAjustar = RESULT_ESTADO_PALETE_ABERTO

        Else 'Sem sucesso na inicializacao com abertura de palete
            MaquinaEstadosAjustar = RESULT_ERRO_PALETE_ABRIR
        End If

    End If 'Se nao existe palete aberto

End Function
```

clsOrdem.cls

```
Option Explicit

Public myIe As String
Public myQtPlan As Long
Public myQtApont As Long
Public myQtApontSAP As Long
Public myProdutoCD As String
Public myComponenteCD As String
Public myQtPalete As Integer

Private myPalete As New clsPalete

Public Function PaleteAbrir(paleteNS As String, linhaCodigo As String, Optional ByRef
ordemAssocAnteriormente As String) As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Manda o objeto palete abrir-se
'-----------------------------------------------------------------
'Argumentos: nsPalete - Numero serial do palete a ser aberto
'            linhaCodigo - Codigo da linha onde o palete
'                          sera preenchido
'            ordemAssocAnteriormente
'                        - Ordem à qual está associado o palete
'                          no banco de dados. Necessário para
'                          que o usuário saiba a qual ordem
'                          aquele palete está associado, caso
'                          nao seja à ordem que está atualmente
'                          aberta.
'-----------------------------------------------------------------
'Retorna: O resultado da funcao abrir() do palete
'-----------------------------------------------------------------
'Autor: GLF, 2003-02-18
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler
Dim resultado As Long
Dim sql As String
Dim nrItensAlterados As Integer

    'Inicia Transacao
    Cnn.BeginTrans

    'Manda o palete se abrir
    resultado = myPalete.Abrir(paleteNS, myProdutoCD, myIe, linhaCodigo, ordemAssocAnteriormente)

    'Se o palete foi aberto com sucesso
    If resultado = RESULT_PALETE_ABERTO Then

        'Este resultado ocorrerá se o palete estava fechado
        'e foi, entao, reaberto.

        'Deixa, na tabela de ordens, a quantidade armazenada como apontada
        'para o que estava armazenado antes menos o que ja esta no palete
        'aberto

        sql = "UPDATE ordens SET qt_apontada = '" & _
```

```
                myQtApont - myPalete.myQtApont & _
               "' WHERE ie_ordem = '" & myIe & "'"

        Cnn.Execute sql, nrItensAlterados

        If nrItensAlterados > 0 Then

            'Fecha a transacao
            Cnn.CommitTrans

            'Atualiza a qtdade do objeto ordem tambem
            myQtApont = myQtApont - myPalete.myQtApont

            'Retorna o resultado
            PaleteAbrir = RESULT_PALETE_ABERTO
        Else

            'Volta a transacao
            Cnn.RollbackTrans
            'Retorna o erro
            PaleteAbrir = RESULT_ERRO_PALETE_ABRIR_BANCO
        End If

    ElseIf resultado = RESULT_PALETE_ABERTO_CRIADO_NOVO Then

        'Efetiva a transacao
        Cnn.CommitTrans
        PaleteAbrir = RESULT_PALETE_ABERTO

    Else

        'Retorna o banco ao estado que estava
        Cnn.RollbackTrans
        'Devolve o erro que o objeto palete retornou
        PaleteAbrir = resultado
    End If

Exit Function

ErrorHandler:
    Cnn.RollbackTrans
    PaleteAbrir = RESULT_ERRO_PALETE_ABRIR_GENERICO
End Function

Public Function PaleteFechar() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Manda o objeto palete fechar-se
'-----------------------------------------------------------------
'Argumentos:
'-----------------------------------------------------------------
'Retorna: O resultado da funcao fechar() do palete
'-----------------------------------------------------------------
'Autor: GLF, 2003-02-26
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler
Dim resultado As Long
Dim nrProdApontPalete As Long
Dim sql As String
Dim nrItensAlterados As Integer

    'Armazena o nr de produtos apontados no palete sendo fechado
    nrProdApontPalete = myPalete.myQtApont

    'Inicia transacao
    Cnn.BeginTrans

    'Manda o palete se fechar
    resultado = myPalete.Fechar

    'Se teve sucesso fechando o palete
    If resultado > 0 Then

        '- Acrescenta ao nr de itens apontados na ordem, a quantidade
        '  relativa ao palete que foi fechado.
```

```
        '- Seta a data de fechamento do palete.

        sql = "UPDATE ordens SET qt_apontada = '" & _
        myQtApont + nrProdApontPalete & "'," & _
        "dh_utilizacao = '" & Format(Now, "dd/mm hh:nn") & "'" & _
        "WHERE ie_ordem = '" & myIe & "'"

        Cnn.Execute sql, nrItensAlterados

        'Se conseguiu atualizar o banco
        If nrItensAlterados > 0 Then
             'Efetiva a transacao
             Cnn.CommitTrans

            'Faz o mesmo com o objeto atual
             myQtApont = myQtApont + nrProdApontPalete

             PaleteFechar = resultado
        Else
            Cnn.RollbackTrans
            PaleteFechar = RESULT_ERRO_PALETE_FECHAR_BANCO
        End If

    Else
        Cnn.RollbackTrans
        PaleteFechar = resultado
    End If

Exit Function

ErrorHandler:
    Cnn.RollbackTrans
    PaleteFechar = RESULT_ERRO_PALETE_FECHAR_GENERICO

End Function

Public Function ProdutoInserir(produtoCD As String, produtoNS As String) As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Manda o objeto palete inserir um produto
'-----------------------------------------------------------------
'Argumentos: produtoCD - Codigo do produto a ser inserido
'            produtoNS - Numero serial do produto a ser inserido
'-----------------------------------------------------------------
'Retorna: O resultado da funcao produtoInserir() do palete
'-----------------------------------------------------------------
'Autor: GLF, 2003-02-19
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim resultado As Long

    'Manda o palete inserir o produto
    resultado = myPalete.ProdutoInserir(produtoCD, produtoNS, Me)

    ProdutoInserir = resultado

End Function

Public Function ProdutoRemover(produtoCD As String, produtoNS As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Manda o objeto palete remover um produto
'-----------------------------------------------------------------
'Argumentos: produtoCD - Codigo do produto a ser removido
'            produtoNS - Numero serial do produto a ser
'                        removido
'-----------------------------------------------------------------
'Retorna: O resultado da funcao ProdutoRemover() do palete
'-----------------------------------------------------------------
'Autor: GLF, 2003-02-27
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim resultado As Long

    'Manda o palete remover o produto
    resultado = myPalete.ProdutoRemover(produtoCD, produtoNS)

    ProdutoRemover = resultado
```

```
End Function

Public Function Iniciar(ieOrdem As String) As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Marca uma ordem como iniciada no banco local
'-------------------------------------------------------------
'Argumentos: ieOrdem =     O nr da ordem a ser iniciada
'            linhaCodigo = O código da linha onde está sendo
'                          iniciada a ordem
'-------------------------------------------------------------
'Retorna: TRUE se deu tudo certo
'         FALSE caso contrario
'-------------------------------------------------------------
'Autor: GLF, 2003-02-13
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler

'Marca uma ordem como iniciada no banco local
Dim sql As String
Dim rs As New Recordset
Dim nrItensModificados As Integer

    'Marca ordem como iniciada
    sql = "UPDATE ordens SET hc_status = '" & ORDEM_HC_STATUS_INICIADA & "' " & _
                          "WHERE ie_ordem = '" & ieOrdem & "'"
    Cnn.Execute sql, nrItensModificados

    'Checa se foi iniciada corretamente, e coleta novos valores
    If nrItensModificados > 0 Then

        'Requisita os valores
        sql = "SELECT cd_produto, cd_componente, qt_planejada, qt_apontada, qt_apontada_sap, " & _
                      "qt_palete FROM ordens WHERE ie_ordem = '" & ieOrdem & "'"

        rs.Open sql, Cnn

        'Preenche os atributos
        myIe = ieOrdem
        myProdutoCD = rs!cd_produto
        myComponenteCD = rs!cd_componente
        myQtPlan = rs!qt_planejada
        myQtApont = rs!qt_apontada
        myQtApontSAP = rs!qt_apontada_sap
        myQtPalete = rs!qt_palete
        Iniciar = True

        rs.Close

    Else 'Deu algum problema, a ordem nao pôde ser alterada no banco

        'Retorna erro
        Iniciar = False
    End If

Exit Function

ErrorHandler:

    Call VariaveisLimpar

    'Retorna o erro
    Iniciar = False

End Function

Public Function IniciarAjustando(ieOrdem As String, qt_planejada As String, qt_apontada As String,
qt_apontada_sap As String, cd_produto As String, cd_componente As String, qt_palete As String) As Boolean

'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Inicia o objeto Ordem com os valores passados
'            como parâmetro
'            Essa funçao deve ser usada quando uma ordem
'            foi deixada como 'iniciada' no banco, sem ter
```

```
'            sido encerrada.
'-----------------------------------------------------------
'Argumentos: ie_ordem        nr da ordem
'            qt_planejada    quantidade planejada
'            qt_apontada     quantidade apontada
'            qt_apontada_sap quantidade apontada pelo sap
'            cd_produto      codigo do produto
'            cd_componente   codigo do componente
'            qt_palete       quantidade max de itens no palete
'-----------------------------------------------------------
'Retorna: TRUE se deu tudo certo
'         FALSE caso contrario
'-----------------------------------------------------------
'Autor: GLF, 2003-02-26
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler

    myIe = ieOrdem
    myQtPlan = qt_planejada
    myQtApont = qt_apontada
    myQtApontSAP = qt_apontada_sap
    myProdutoCD = cd_produto
    myComponenteCD = cd_componente
    myQtPalete = qt_palete

    IniciarAjustando = True

Exit Function

'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Controle de erros
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
ErrorHandler:

    Call VariaveisLimpar

    IniciarAjustando = False

End Function

Public Function IniciarComPalete(ns_palete As String, cd_produto As String, cd_linha As String, ie_ordem
As String) As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: - Preenche os atributos do objeto Ordem retirando
'              do banco os valores de uma ordem previamente
'              iniciada
'            - Preenche os atributos do objeto Palete
'              de acordo com os valores fornecidos
'-----------------------------------------------------------
'Argumentos: ns_palete   Numero serial do palete
'            cd_produto  Codigo do produto
'            cd_linha    Codigo da linha
'            ie_ordem    Nr da ordem
'-----------------------------------------------------------
'Retorna: TRUE se deu tudo certo
'         FALSE caso contrario
'-----------------------------------------------------------
'Autor: GLF, 2003-02-26
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler
Dim sql As String
Dim rs As New Recordset
Dim sucesso As Boolean

    'Pega as informacoes relativas à ordem do palete aberto
    sql = "SELECT * FROM ordens WHERE ie_ordem = '" & ie_ordem & "'"
    rs.Open sql, Cnn

    'Inicializa as variáveis
    sucesso = Me.IniciarAjustando(rs!ie_ordem, rs!qt_planejada, rs!qt_apontada, _
                                  rs!qt_apontada_sap, rs!cd_produto, rs!cd_componente, _
                                  rs!qt_palete)

    'Fecha o recordset
```

```
    rs.Close

    'Checa o sucesso na inicializacao
    If Not sucesso Then
        IniciarComPalete = False
        Exit Function
    End If

    'Manda o palete se abrir
    sucesso = myPalete.AbrirAjustando(ns_palete, cd_produto, cd_linha, ie_ordem)

    If Not sucesso Then
       IniciarComPalete = False
       Exit Function
    End If

    IniciarComPalete = True

Exit Function
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Controle de erros
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
ErrorHandler:

    Call VariaveisLimpar

    IniciarComPalete = False

End Function

Public Function Encerrar() As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Marca uma ordem como encerrada no banco local
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Retorna:
'-------------------------------------------------------------
'Autor: GLF, 2003-02-18
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim sql As String
Dim nrItensModificados As Integer

    'Marca ordem como encerrada
    sql = "UPDATE ordens SET hc_status = '" & ORDEM_HC_STATUS_ENCERRADA & "' " & _
                        "WHERE ie_ordem = '" & myIe & "'"
    Cnn.Execute sql, nrItensModificados

    'Checa se foi encerrada corretamente
    If nrItensModificados > 0 Then

        'Encerrado com sucesso. Limpa os valores do objeto ordem
        myIe = ""
        myProdutoCD = ""
        myQtPlan = 0
        myQtApont = 0
        myQtApontSAP = 0

        Encerrar = True
    Else
        Encerrar = False
    End If

Exit Function

ErrorHandler:
    Encerrar = False

End Function

Public Function PaleteDesmontar() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Manda o objeto palete desmontar-se
'-------------------------------------------------------------
```

```
'Argumentos:
'-------------------------------------------------------------
'Retorna: O resultado da funcao desmontar() do palete
'-------------------------------------------------------------
'Autor: GLF, 2003-02-26
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim resultado As Long

    'Manda o palete se desmontar
    resultado = myPalete.Desmontar

    PaleteDesmontar = resultado

End Function

Public Function GetPaleteQuantidadeApontada() As Integer
    GetPaleteQuantidadeApontada = myPalete.myQtApont
End Function

Public Function GetPaleteNS() As String
    GetPaleteNS = myPalete.myNS
End Function

Public Sub VariaveisLimpar()

    myIe = ""
    myQtPlan = 0
    myQtApont = 0
    myProdutoCD = ""
    myComponenteCD = ""
    myQtPalete = 0
    myPalete.VariaveisLimpar



End Sub
```

clsPalete.cls

```
Option Explicit

Public myNS As String       'Numero serial do palete
Public myQtApont As Long    'Quantidade de produtos ja inserida no palete
Public myProdutoCD As String 'Codigo dos produtos que devem ser inseridos neste palete
Public myLinhaCD As String  'Codigo da linha onde o palete esta sendo preenchido
Public myOrdemIE As String  'Nr da ordem para a qual o palete esta sendo preenchido


Public Function Abrir(paleteNS As String, produtoCD As String, ordemIE As String, linhaCodigo As String,
Optional ByRef ordemAssocAnteriormente As String) As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Faz as verificacoes no banco e abre ou
'            reabre o palete, se tudo estiver certo
'-------------------------------------------------------------
'Argumentos: paleteNS  - Numero serial do palete a ser aberto
'            produtoCD - Codigo do produto que deve ser colocado
'                        no palete
'            ordemIE   - A ordem para a qual o palete esta
'                        sendo preenchido
'            linhaCodigo - Codigo da linha onde
'                        o palete sera preenchido
'            ordemAssocAnteriormente
'                        - Ordem à qual está associado o palete
'                          no banco de dados. Necessário para
'                          que o usuário saiba a qual ordem
'                          aquele palete está associado, caso
'                          nao seja à ordem que está atualmente
'                          aberta.
'-------------------------------------------------------------
'Retorna: RESULT_PALETE_ABERTO
'         RESULT_PALETE_ABERTO_CRIADO_NOVO
'         RESULT_ERRO_PALETE_ORDEM_INCORRETA
'         RESULT_ERRO_PALETE_TRANSFERENCIA
```

```
'         RESULT_ERRO_PALETE_ABRIR_GENERICO
'         RESULT_ERRO_PALETE_ABRIR_BANCO
'         (> 0 se deu tudo certo, < 0 em caso de erro)
'-------------------------------------------------------------
'Autor: GLF, 2003-02-18
'=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler

Dim sql As String                   'Query SQL
Dim rs As New Recordset             'Recordset para consultas
Dim nsPaleteFechado As String       'NS de palete fechado, a ser reaberto
Dim nrItensAlterados As Integer

    'Verifica se o palete existe
    sql = "SELECT count(*) as contagem from paletes WHERE ns_palete = '" & paleteNS & "'"
    rs.Open sql, Cnn

    If rs!contagem = 0 Then
    '-------------------
    'PALETE NAO EXISTE
    '-------------------

        'O Palete nao existe. Cria um palete, vazio
        sql = "INSERT INTO paletes (ns_palete, cd_produto, hc_status," & _
                                  " cd_linha, ie_ordem) VALUES ('" & _
                                    paleteNS & "', '" & _
                                    produtoCD & "', '" & _
                                    PALETE_HC_STATUS_ABERTO & "', '" & _
                                    linhaCodigo & "', '" & _
                                    ordemIE & "')"

        Cnn.Execute sql, nrItensAlterados

        If nrItensAlterados > 0 Then

            'Preenche os atributos do objeto
            myNS = paleteNS
            myProdutoCD = produtoCD
            myQtApont = 0 'Esta vazio, logo nao tem produtos
            myLinhaCD = linhaCodigo
            myOrdemIE = ordemIE

            Abrir = RESULT_PALETE_ABERTO_CRIADO_NOVO
            Exit Function

        Else 'Deu problema na insercao no banco

            Abrir = RESULT_ERRO_PALETE_ABRIR_BANCO
            Exit Function
        End If
    Else
    '---------------------------
    'PALETE EXISTE, VERIFICACOES
    '---------------------------

        'Verifica se esta fechado
        rs.Close
        sql = "SELECT * FROM paletes WHERE ns_palete = '" & paleteNS & "'"
        rs.Open sql, Cnn

        If rs!hc_status = PALETE_HC_STATUS_FECHADO Then

            'Verifica se o palete fechado pertence à ordem que esta aberta
            If rs!ie_ordem = ordemIE Then

                rs.Close

                '"Reabre" o palete
                sql = "UPDATE paletes SET hc_status = '" & PALETE_HC_STATUS_ABERTO & "' " & _
                      "WHERE ns_palete = '" & paleteNS & "'"

                Cnn.Execute sql, nrItensAlterados

                If nrItensAlterados > 0 Then
```

```
                    'Preenche atributos principais
                    myNS = paleteNS
                    myProdutoCD = produtoCD
                    myOrdemIE = ordemIE
                    myLinhaCD = linhaCodigo

                    'Descobre quantos produtos foram paletizados nesse palete
                    'e atualiza a variavel contadora de produtos apontados
                    sql = "SELECT count(*) as contagem FROM produtos WHERE ns_palete = '" & myNS & "'"
                    rs.Open sql, Cnn

                    myQtApont = rs!contagem

                    'Finaliza com resultado de "tudo certo"
                    rs.Close
                    Abrir = RESULT_PALETE_ABERTO

                Else 'Problemas setando o status do palete para "aberto" no banco

                    Abrir = RESULT_ERRO_PALETE_ABRIR_BANCO
                    rs.Close
                    Exit Function

                End If

            Else 'Palete nao pertence a ordem que esta aberta

                'Armazena a que ordem pertence esse palete
                ordemAssocAnteriormente = rs!ie_ordem
                rs.Close
                Abrir = RESULT_ERRO_PALETE_ORDEM_INCORRETA
                Exit Function
            End If

        Else 'Palete nao esta com status de fechado. Deve estar
             'como "transferindo" ou "transferido"
            rs.Close
            Abrir = RESULT_ERRO_PALETE_TRANSFERENCIA
            Exit Function
        End If
    End If

Exit Function

ErrorHandler:
    Abrir = RESULT_ERRO_PALETE_ABRIR_GENERICO

End Function

Public Function ProdutoInserir(produtoCD As String, produtoNS As String, ordem As clsOrdem) As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Insere produto no palete
'-------------------------------------------------------------
'Argumentos: produtoNS  - Numero serial do produto a ser inserido
'            produtoCD  - Codigo do produto a ser inserido
'            ordem      - Ordem para o produto a ser inserido
'-------------------------------------------------------------
'Retorna: RESULT_PRODUTO_INSERIDO
'         RESULT_ERRO_PRODUTO_INEXISTENTE
'         RESULT_ERRO_PRODUTO_SEM_COMPONENTE
'         RESULT_ERRO_MONTAGEM_INCORRETA
'         RESULT_ERRO_PRODUTO_JA_INSERIDO
'         RESULT_ERRO_PRODUTO_JA_INSERIDO_OUTRO_PALETE
'         RESULT_ERRO_PRODUTO_INCORRETO
'         RESULT_ERRO_PALETE_CHEIO
'         RESULT_ERRO_PRODUTO_INSERIR_GENERICO
'         RESULT_ERRO_PRODUTO_INSERIR_BANCO
'         (> 0 se deu tudo certo, < 0 em caso de erro)
'-------------------------------------------------------------
'Autor: GLF, 2003-02-19
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler
```

```
Dim sql As String                   'Query SQL
Dim rs As New Recordset             'Recordset para consultas
Dim nrItensAlterados As Integer     'Nr de itens modificados no banco

    'Verifica se o produto em questao existe e ja pega o codigo
    'do componente agregado

    sql = "SELECT cd_componente, ns_palete FROM produtos WHERE ns_produto = '" & produtoNS & "'"
    rs.Open sql, Cnn

    If Not rs.EOF Then

        'Produto existe!
        'Verifica se tem componente agregado

        If Not IsNull(rs!cd_componente) Then

            'Produto tem componente.
            'Verifica se o codigo do produto eh igual ao espec. na ordem

            If produtoCD = ordem.myProdutoCD Then

                'Codigo do produto correto
                'Verifica se o componente montado é igual ao especificado na ordem
                'Faz com Left pois o codigo do componente, na tabela de ordens, pode estar no formato
                'grande que atualmente o SAP retorna.

                If rs!cd_componente = Left(ordem.myComponenteCD, Len(CStr(rs!cd_componente))) Then

                    'Produto está com componente certo.
                    'Verifica se ele ja nao esta no palete

                    If IsNull(rs!ns_palete) Then

                        'Produto nao esta inserido em nenhum palete.

                        'Verifica, por fim(graças!), se o palete nao esta cheio
                        If myQtApont < ordem.myQtPalete Then

                            'Tudo certo!
                            'Insere o produto no palete!

                            sql = "UPDATE produtos SET ns_palete = '" & myNS & "' " & _
                                  "WHERE ns_produto = '" & produtoNS & "'"

                            Cnn.Execute sql, nrItensAlterados

                            If nrItensAlterados > 0 Then

                                'Incrementa a variavel interna indicando quantos produtos existem no
palete atual
                                myQtApont = myQtApont + 1

                                'Inseriu com sucesso. Verifica se o palete nao ficou cheio
                                If myQtApont = ordem.myQtPalete Then
                                    'Palete ficou cheio com a inserção.
                                    'Retorna o sucesso da operação, com a ressalva
                                    ProdutoInserir = RESULT_PRODUTO_INSERIDO_COMPLETOU_PALETE
                                    Exit Function
                                Else
                                    'Finaliza com sucesso
                                    ProdutoInserir = RESULT_PRODUTO_INSERIDO
                                    Exit Function
                                End If

                            Else 'Problemas ao associar o produto com o palete no banco

                                ProdutoInserir = RESULT_ERRO_PRODUTO_INSERIR_BANCO
                                Exit Function
                            End If

                        Else 'Palete cheio!
                            ProdutoInserir = RESULT_ERRO_PALETE_CHEIO
                            Exit Function
```

```
                        End If

                    'Produto já associado a outro (ou a este) palete
                    ElseIf rs!ns_palete = myNS Then
                        ProdutoInserir = RESULT_ERRO_PRODUTO_JA_INSERIDO
                        Exit Function
                    Else
                        ProdutoInserir = RESULT_ERRO_PRODUTO_JA_INSERIDO_OUTRO_PALETE
                        Exit Function
                    End If

                Else 'Componente montado no produto diferente do especificado na ordem
                    ProdutoInserir = RESULT_ERRO_MONTAGEM_INCORRETA
                    Exit Function
                End If

            Else 'Codigo do produto diferente do especificado na ordem
                ProdutoInserir = RESULT_ERRO_PRODUTO_INCORRETO
                Exit Function
            End If

        Else 'Produto sem componente agregado
            ProdutoInserir = RESULT_ERRO_PRODUTO_SEM_COMPONENTE
            Exit Function
        End If

    Else 'Produto com esse NS nao existe!
         ProdutoInserir = RESULT_ERRO_PRODUTO_INEXISTENTE
        Exit Function
    End If

Exit Function

ErrorHandler:
    ProdutoInserir = RESULT_ERRO_PRODUTO_INSERIR_GENERICO

End Function

Public Function ProdutoRemover(produtoCD As String, produtoNS As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Insere produto no palete
'-------------------------------------------------------------
'Argumentos: produtoNS  - Numero serial do produto a ser inserido
'            produtoCD  - Codigo do produto a ser inserido
'-------------------------------------------------------------
'Retorna: RESULT_PRODUTO_REMOVIDO
'         RESULT_ERRO_PRODUTO_EM_OUTRO_PALETE
'         RESULT_ERRO_PRODUTO_REMOVER_BANCO
'         RESULT_ERRO_PRODUTO_REMOVER_GENERICO
'-------------------------------------------------------------
'Autor: GLF, 2003-02-27
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler

Dim sql As String                    'Query SQL
Dim rs As New Recordset              'Recordset para consultas
Dim nrItensAlterados As Integer      'Nr de itens modificados no banco

    'Verifica se produto esta nesse palete
    sql = "SELECT ns_palete FROM produtos WHERE ns_produto = '" & produtoNS & "'"
    rs.Open sql, Cnn

    If rs!ns_palete = myNS Then
        'Correto, produto esta neste palete

        'Desassocia produto com palete no banco local
        sql = "UPDATE produtos SET ns_palete = null WHERE ns_produto = '" & produtoNS & "'"
        Cnn.Execute sql, nrItensAlterados

        'Se deu tudo certo
        If nrItensAlterados > 0 Then

            'Decrementa contador interno de produtos no palete
            myQtApont = myQtApont - 1
```

```
            'Tudo certo!
            ProdutoRemover = RESULT_PRODUTO_REMOVIDO

        Else 'Erro removendo associacao no produto no banco

            ProdutoRemover = RESULT_ERRO_PRODUTO_REMOVER_BANCO
            Exit Function

        End If

    Else 'Produto nao estava no palete
        ProdutoRemover = RESULT_ERRO_PRODUTO_EM_OUTRO_PALETE
    End If

Exit Function
ErrorHandler:
    ProdutoRemover = RESULT_ERRO_PRODUTO_REMOVER_GENERICO

End Function
'Public Function AbrirAjustando(rsPalete) As Boolean
Public Function AbrirAjustando(ns_palete As String, cd_produto As String, cd_linha As String, ie_ordem As String)

'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: -  Inicia o objeto Palete com os valores retirados
'               no banco, ja fornecidos pelos parametros
'            -  Calcula a quantidade apontada para o palete,
'               verificando o nr de produtos que estao associados
'               com ele
'
'            Essa funçao deve ser usada quando um palete
'            foi deixada como 'aberto' no banco, sem ter
'            sido fechado.
'-------------------------------------------------------------
'Argumentos: rsOrdem =  RecordSet com as informacoes relativas
'                       ao palete em questão
'------------------------------------------------------------
'Retorna: TRUE se deu tudo certo
'         FALSE caso contrario
'------------------------------------------------------------
'Autor: GLF, 2003-02-26
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler
Dim rs As New Recordset
Dim sql As String

    'Preenche atributos do objeto com os valores do banco
    myNS = ns_palete
    myProdutoCD = cd_produto
    myLinhaCD = cd_linha
    myOrdemIE = ie_ordem

    'Descobre quantos produtos foram paletizados nesse palete
    'e atualiza a variavel contadora de produtos apontados
    sql = "SELECT count(*) as contagem FROM produtos WHERE ns_palete = '" & myNS & "'"
    rs.Open sql, Cnn

    myQtApont = rs!contagem

    rs.Close

    AbrirAjustando = True

Exit Function
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Controle de erros
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
ErrorHandler:

    myNS = ""
    myProdutoCD = ""
    myLinhaCD = ""
    myOrdemIE = ""
```

```
    myQtApont = 0

    AbrirAjustando = False

End Function

Public Function Fechar() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Fecha o palete
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Retorna: RESULT_PALETE_FECHADO
'         RESULT_ERRO_PALETE_FECHAR_GENERICO
'         RESULT_ERRO_PALETE_FECHAR_BANCO
'         (> 0 se deu tudo certo, < 0 em caso de erro)
'-------------------------------------------------------------
'Autor: GLF, 2003-02-26
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler
Dim sql As String
Dim nrItensAlterados As Integer     'Nr de itens modificados no banco

    'Altera o status do palete no banco para refletir o status de fechado,
    'e salva a data e hora atual em dh_utilizacao
    sql = "UPDATE paletes SET hc_status = '" & PALETE_HC_STATUS_FECHADO & "'" & _
          "WHERE ns_palete = '" & myNS & "'"

    Cnn.Execute sql, nrItensAlterados

    'Se deu tudo certo
    If nrItensAlterados > 0 Then

        'Zera os atributos do objeto
        myLinhaCD = ""
        myNS = ""
        myOrdemIE = ""
        myProdutoCD = ""
        myQtApont = 0

        Fechar = RESULT_PALETE_FECHADO

    Else 'Erro setando status de palete para fechado

        Fechar = RESULT_ERRO_PALETE_FECHAR_BANCO

    End If

Exit Function

'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Controle de erros
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
ErrorHandler:
    Fechar = RESULT_ERRO_PALETE_FECHAR_GENERICO
End Function


Public Function Desmontar() As Long
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Desmonta o palete. Exclui do banco local seus
'            dados e desassocia os produtos a ele
'            previamente associados.
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Retorna: RESULT_PALETE_DESMONTADO
'         RESULT_ERRO_PALETE_DESMONTAR_GENERICO
'         RESULT_ERRO_PALETE_DESMONTAR_BANCO
'         (> 0 se deu tudo certo, < 0 em caso de erro)
'-------------------------------------------------------------
'Autor: GLF, 2003-02-26
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler
```

```
Dim sql As String
Dim rs As New Recordset
Dim nrItensAlterados As Integer

            'Inicia Transacao
            Cnn.BeginTrans

            'Desassocia todos os produtos previamente associados a este palete
            sql = "UPDATE produtos SET ns_palete = null WHERE ns_palete = '" & myNS & "'"
            Cnn.Execute sql

            'Remove o palete do banco local
            sql = "DELETE FROM paletes WHERE ns_palete = '" & myNS & "'"
            Cnn.Execute sql, nrItensAlterados

            If nrItensAlterados > 0 Then

                'Zera os atributos do objeto
                myLinhaCD = ""
                myNS = ""
                myOrdemIE = ""
                myProdutoCD = ""
                myQtApont = 0

                Desmontar = RESULT_PALETE_DESMONTADO

                'Finaliza transacao
                Cnn.CommitTrans
            Else 'Problemas na hora de excluir o palete da tabela de paletes

                Cnn.RollbackTrans
                Desmontar = RESULT_ERRO_PALETE_DESMONTAR_BANCO

            End If

Exit Function

'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Controle de erros
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
ErrorHandler:

    'Caso algo tenha ocorrido errado, dá um rollback na transacao,
    'desfazendo as desassociacoes e/ou a remocao do palete
    Cnn.RollbackTrans

    'Retorna o erro
    Desmontar = RESULT_ERRO_PALETE_DESMONTAR_GENERICO

End Function

Public Sub VariaveisLimpar()
    myNS = ""
    myQtApont = 0
    myProdutoCD = ""
    myLinhaCD = ""
    myOrdemIE = ""
End Sub
```

frmOrdens.frm

```
Option Explicit
Public comportamento As enmComportamentoFormOrdens  'Determina o comportamento da form
Public ieOrdemAIniciar As String                   'Armazena a ordem escolhida, a ser iniciada pela
maquina

Private Sub ListViewPreencher()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Consulta ordens no banco de dados local e
'            apresenta no listview
'-----------------------------------------------------------
'Argumentos:
```

```
'-------------------------------------------------------------
'Autor: GLF, 2003-03-10
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Dim rs As New Recordset, sql As String
Dim registro As Integer

    'Limpa itens anteriores
    lsvOrdens.ListItems.Clear

    registro = 0
    sql = "SELECT * FROM ordens ORDER BY cd_produto, ie_ordem"
    rs.Open sql, Cnn

    While Not rs.EOF
        registro = registro + 1
        lsvOrdens.ListItems.Add , , rs!cd_produto
        lsvOrdens.ListItems(registro).SubItems(1) = rs!ie_ordem
        lsvOrdens.ListItems(registro).SubItems(2) = rs!cd_componente
        lsvOrdens.ListItems(registro).SubItems(3) = rs!qt_planejada
        lsvOrdens.ListItems(registro).SubItems(4) = rs!qt_apontada
        lsvOrdens.ListItems(registro).SubItems(5) = Format(rs!dh_atualizacao, "dd/mm hh:nn")
        lsvOrdens.ListItems(registro).SubItems(6) = rs!hc_status
        lsvOrdens.ListItems(registro).SubItems(7) = rs!hc_sapmanual
        rs.MoveNext
    Wend
    rs.Close

End Sub

Private Sub cmdCancel_Click()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Cancela início de ordem
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Autor: GLF, 2003-03-10
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    ieOrdemAIniciar = ""

    frmOrdens.Visible = False
End Sub

Private Sub cmdIniciar_Click()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Coloca em variável global da form o ie_ordem
'            da ordem a ser iniciada pela máquina de estados
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Retorna:
'-------------------------------------------------------------
'Autor: GLF, 2003-03-11
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
    ieOrdemAIniciar = lsvOrdens.SelectedItem.SubItems(1)
    Me.Visible = False

End Sub


Private Sub cmdNova_Click()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Abre a janela para entrada de dados de nova ordem
'            manual. Verifica se ordem ja nao existe na base
'            local e, nao existindo, a acrescenta à listView
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Autor: GLF, 2003-03-10
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim itemOrdem As ListItem
Dim nrOrdem As String
Dim sql As String
```

```
Dim rs As New Recordset
Dim nrItensAlterados As Integer

    frmNovaOrdem.Show vbModal
    'Se o usuario clicou OK na janela de insercao de nova ordem
    If frmNovaOrdem.clicouOK Then

        'Se já nao existe ordem com esse mesmo numero na lista
        Set itemOrdem = lsvOrdens.FindItem(frmNovaOrdem.txtOrdem.Text, lvwSubItem)
        If itemOrdem Is Nothing Then

            'Adiciona a ordem no banco local
            sql = "INSERT INTO ordens (ie_ordem, cd_produto, qt_planejada," & _
                                "qt_apontada, hc_status, hc_sapmanual," & _
                                "dh_atualizacao, qt_palete, cd_componente," & _
                                "de_modelo, de_tensao_frequencia, de_potencia ," & _
                                "de_capacidade_50LBP, de_capacidade_50HBP, " & _
                                "de_capacidade_60LBP, de_capacidade_60HBP," & _
                                "de_refrigerante, de_corrente, de_refrigerador_oleo) VALUES ('" & _
                          frmNovaOrdem.txtOrdem.Text & "','" & _
                          frmNovaOrdem.txtProduto.Text & "','" & _
                          frmNovaOrdem.txtPlanejado.Text & "','" & _
                          "0" & "','" & _
                          ORDEM_HC_STATUS_CONSULTADA & "','" & _
                          "Manual" & "','" & _
                          Format(Now, "dd/mm hh:nn") & "','" & _
                          frmNovaOrdem.txtQtPalete.Text & "','" & _
                          frmNovaOrdem.txtComponente.Text & "','" & _
                          frmNovaOrdem.txtModelo.Text & "','" & _
                          frmNovaOrdem.txtTensao.Text & "','" & _
                          frmNovaOrdem.txtPotencia.Text & "','" & frmNovaOrdem.txtCap50LBP.Text &
"','" & frmNovaOrdem.txtCap50HBP.Text & "','" & frmNovaOrdem.txtCap60LBP.Text & "','" &
frmNovaOrdem.txtCap60HBP.Text & "','" & _
                          frmNovaOrdem.txtRefrigerante.Text & "','" & _
                          frmNovaOrdem.txtCorrente.Text & "','" & _
                          frmNovaOrdem.txtResfriador.Text & "')"
             Cnn.Execute sql, nrItensAlterados

             'Checa para ver se deu algum problema na insercao
             If nrItensAlterados <= 0 Then
                MsgBox MSG_INSERCAO_ORDEM_ERRO & "[" & Err.Number & "] :" & Err.Description
             End If

        Else
            MsgBox MSG_ORDEM_JA_EXISTENTE
        End If


    End If
End Sub

Private Sub Form_Activate()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Evento que ocorre toda vez que a form se torna
'             a janela ativa.
'             Tenta consultar as ordens do SAP, e depois carrega
'             os dados do banco local, indicando se houve ou
'             nao erro de conexao.
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Autor: GLF, 2003-03-17
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler
Dim sucesso As Boolean
Dim cdProduto As String

    ieOrdemAIniciar = ""
    Me.MousePointer = vbHourglass

    'Verifica o comportamento da form
    Select Case comportamento
        Case PAL_FORM_ORDENS_CONSULTAR
            'Desabilita botao Iniciar
```

```
            cmdIniciar.Enabled = False

        Case PAL_FORM_ORDEM_INICIAR
            'Habilita botao Iniciar
            cmdIniciar.Enabled = True
    End Select

    cdProduto = InputBox("Entre com o código do produto a ser pesquisado:", "Codigo Material")
    If cdProduto <> "" Then

        'Consulta ordens SAP
        sucesso = OrdensSAPConsultar(cdProduto)

        'Verifica se a consulta teve sucesso
        If sucesso Then
            'Mantém o caption normal
            Me.Caption = CAP_ORDENS
        Else
            'Avisa no caption da janela que estamos sem conexao SAP.
            Me.Caption = CAP_ORDENS_SEM_CONEXAO
        End If


    Else

        'Msg de erro por nao ter indicado o material
        MsgBox "Voce deve indicar o codigo do produto que deseja pesquisar!"

    End If

    'Consulta as ordens do banco local (que podem ser as atualizadas, pelos dados do SAP,
    'ou as antigas, quando sem conexao) e preenche o listview para escolha
    ListViewPreencher

    Me.MousePointer = vbDefault

Exit Sub

ErrorHandler:

    Me.MousePointer = vbDefault

End Sub



Private Sub lsvOrdens_Click()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Evento que ocorre no clique de uma ordem na
'            list view, garantindo que não se tente
'            iniciar uma ordem que já esteja iniciada
'-------------------------------------------------------------
'Argumentos:
'-------------------------------------------------------------
'Retorna:
'-------------------------------------------------------------
'Autor: GLF, 2003-03-11
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

On Error Resume Next

'Caso a ordem selecionada ja esteja iniciada,
'ou o form esteja em modo consulta, desabilita o botao
'iniciar

If lsvOrdens.SelectedItem.ListSubItems(6).Text = "Iniciada" Or _
   comportamento = PAL_FORM_ORDENS_CONSULTAR Then
    cmdIniciar.Enabled = False
Else
    cmdIniciar.Enabled = True
End If


End Sub
```

```
Private Function OrdensSAPConsultar(cdProduto As String) As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Carrega os dados do SAP e os repassa para
'            tratamento local
'-------------------------------------------------------------
'Argumentos: cdProduto - Codigo do produto a ser
'                        pesquisado (necessario para a funcao
'                        do SAP)
'-------------------------------------------------------------
'Retorna: TRUE   se der tudo certo
'         FALSE  caso ocorram problemas na tentativa de
'                solicitar o XML ou este venha vazio
'-------------------------------------------------------------
'Autor: GLF, 2003-03-11
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler

Dim xmlDoc As New DOMDocument26
Dim myXml As String
Dim sucesso As Boolean

    'Transferencia http
    InetHTTP.url = Trim(HTTP_ORDENS_CONSULTAR & "?cd_produto=" & cdProduto)

     'Seta para carregar arquivo sincronizado
     xmlDoc.async = "false"

    'Carrega o arquivo XML da pagina
     myXml = InetHTTP.OpenURL()

    'Verifica se veio alguma informacao
    If myXml <> "" Then
        'Passa os dados de XML para o objeto XMLDOM

        '#########################################################################
        'DESENVOLVIMENTO - Linha 1
        'IMPLEMENTACAO - Linha 2
        'sucesso = xmlDoc.Load("C:\rastreabilidade\ordensSAP.xml")  'LINHA 1
         sucesso = xmlDoc.loadXML(myXml)                           'LINHA 2
        '#########################################################################
        If Not sucesso Then
            OrdensSAPConsultar = False
            Exit Function
        End If

        'Estou com o XML na mao. Hora de passá-lo adiante para o devido
        'tratamento no banco local
        sucesso = OrdensSAPCarregar(xmlDoc)

        If sucesso Then
            OrdensSAPConsultar = True
        Else
            OrdensSAPConsultar = False
        End If

    Else 'Veio um arquivo vazio
      OrdensSAPConsultar = False
    End If

Exit Function

ErrorHandler:
    'Seta que ocorreu o erro
    OrdensSAPConsultar = False
End Function

Private Function OrdensSAPCarregar(ordensSAP As DOMDocument26) As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Recebe um XML com os dados provenientes do SAP
'            e atualiza o banco local
'-------------------------------------------------------------
'Argumentos: ordensSAP: XML contendo as informacoes
'                       com as ordens do SAP a serem carre-
```

```
'                       gadas
'-------------------------------------------------------------
'Retorna: TRUE   se der tudo certo
'         FALSE  caso ocorram problemas
'-------------------------------------------------------------
'Autor: GLF, 2003-03-11
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error GoTo ErrorHandler

Dim listaOrdens As IXMLDOMNodeList  'Lista das ordens do XML
Dim ordemSAP As IXMLDOMNode         'Cada uma das ordens do XML
Dim sql As String
Dim rs As New Recordset
Dim colOrdensVistas As New Collection 'Colecao com as ordens que foram vistas
                                      'no processo (atualizadas ou inseridas)
Dim strOrdensVistas As String
Dim nrItensAlterados As Long

Dim ieOrdem As String

    Set listaOrdens = ordensSAP.selectSingleNode(TAG_SAP_ORDENS).childNodes

    'Inicia transacao
    Cnn.BeginTrans

    For Each ordemSAP In listaOrdens

        'Extrai o nr da ordem
        ieOrdem = ordemSAP.firstChild.Text

        'Verifica se tal ordem já existe no banco local
        sql = "SELECT count(*) as contagem FROM ordens WHERE ie_ordem = '" & _
                    ieOrdem & "'"

        rs.Open sql, Cnn

        'Se ela existe no banco local
        If rs!contagem <> 0 Then
            'Atualiza com os dados do SAP. Vai sobrescrever tudo, menos o status
            'e o qt_apontada (que vai pra qt_apontada_sap)

            sql = "UPDATE ordens SET " & _
            "cd_produto = '" & ordemSAP.selectSingleNode(TAG_SAP_ORDENS_CD_PRODUTO).Text & "', " & _
            "qt_planejada = '" & ordemSAP.selectSingleNode(TAG_SAP_ORDENS_QT_PLANEJADA).Text & "', " & _
            "qt_apontada_sap = '" & ordemSAP.selectSingleNode(TAG_SAP_ORDENS_QT_APONTADA).Text & "', "& _
            "hc_sapmanual = 'SAP'," & _
          "dh_atualizacao = '" & ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DH_ATUALIZACAO).Text & "', " &_
         "cd_componente = '" & ordemSAP.selectSingleNode(TAG_SAP_ORDENS_CD_COMPONENTE).Text & "', " & _
          "de_modelo = '" & ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_MODELO).Text & "', " & _
          "de_tensao_frequencia = '"& ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_TENSAO_FREQUENCIA).Text
&"', "&_
         "de_potencia = '" & ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_POTENCIA).Text & "', " & _
          "de_capacidade_50LBP = " &
IIf(ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_50LBP).Text = "", "null", "'" &
ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_50LBP).Text & "'") & ", " & _
          "de_capacidade_50HBP = " &
IIf(ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_50HBP).Text = "", "null", "'" &
ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_50HBP).Text & "'") & ", " & _
          "de_capacidade_60LBP = " &
IIf(ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_60LBP).Text = "", "null", "'" &
ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_60LBP).Text & "'") & ", " & _
          "de_capacidade_60HBP = " &
IIf(ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_60HBP).Text = "", "null", "'" &
ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_60HBP).Text & "'") & ", " & _
        "de_refrigerante = '" & ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_REFRIGERANTE).Text & "', " &_
        "de_corrente = '" & ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CORRENTE).Text & "', " & _
        "de_refrigerador_oleo = '" & ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_REFRIGERADOR_OLEO).Text &
"'" & _
          "WHERE ie_ordem = '" & ieOrdem & "'"

            Cnn.Execute sql, nrItensAlterados

            If nrItensAlterados > 0 Then
```

```
                'Adiciona o nr dessa ordem na collection com a lista das ordens atualizadas/inseridas
                colOrdensVistas.Add ieOrdem

            Else 'Problemas na atualizacao do item!
                'Indica erro
                OrdensSAPCarregar = False

                'Volta as alteracoes feitas
                Cnn.RollbackTrans

                'Sai da funcao
                Exit Function
            End If

        Else

            'Insere no banco local
            sql = "INSERT into ordens (ie_ordem, cd_produto, qt_planejada," & _
                "qt_apontada_sap, hc_status, hc_sapmanual, dh_atualizacao," & _
                "cd_componente, de_modelo, de_tensao_frequencia, de_potencia, " & _
               "de_capacidade_50LBP, de_capacidade_50HBP, de_capacidade_60LBP, de_capacidade_60HBP, " & _
                "de_refrigerante, de_corrente, de_refrigerador_oleo) VALUES ('" & _
                ieOrdem & "', '" & _
                ordemSAP.selectSingleNode(TAG_SAP_ORDENS_CD_PRODUTO).Text & "', '" & _
                ordemSAP.selectSingleNode(TAG_SAP_ORDENS_QT_PLANEJADA).Text & "', '" & _
                ordemSAP.selectSingleNode(TAG_SAP_ORDENS_QT_APONTADA).Text & "', '" & _
                ordemSAP.selectSingleNode(TAG_SAP_ORDENS_HC_STATUS).Text & "', '" & _
                "SAP','" & _
                ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DH_ATUALIZACAO).Text & "', '" & _
                ordemSAP.selectSingleNode(TAG_SAP_ORDENS_CD_COMPONENTE).Text & "', '"
            sql = sql & ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_MODELO).Text & "', '" & _
                ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_TENSAO_FREQUENCIA).Text & "', '" & _
                ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_POTENCIA).Text & "', " & _

IIf(ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_50LBP).Text = "", "null", "'" &
ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_50LBP).Text & "'") & "," & _

IIf(ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_50HBP).Text = "", "null", "'" &
ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_50HBP).Text & "'") & "," & _

IIf(ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_60LBP).Text = "", "null", "'" &
ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_60LBP).Text & "'") & "," & _

IIf(ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_60HBP).Text = "", "null", "'" &
ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CAPACIDADE_60HBP).Text & "'") & ", '" & _
            ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_REFRIGERANTE).Text & "', '" & _
            ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_CORRENTE).Text & "', '" & _
            ordemSAP.selectSingleNode(TAG_SAP_ORDENS_DE_REFRIGERADOR_OLEO).Text & "')"

            Cnn.Execute sql, nrItensAlterados

            If nrItensAlterados > 0 Then

                'Adiciona o nr dessa ordem na collection com a lista das ordens atualizadas/inseridas
                colOrdensVistas.Add ieOrdem

            Else 'Problemas na atualizacao do item!
                'Indica erro
                OrdensSAPCarregar = False
                'Volta as alteracoes feitas
                Cnn.RollbackTrans
                'Sai da funcao
                Exit Function
            End If

        End If

        'Fecha o recordset
        rs.Close

    Next

    'Feitas as atualizações/inserções, temos que excluir as ordens que estejam
    'no banco local, como consultadas, e nao tenham vindo nessa nova consulta do
```

```
    'SAP.

    'Tudo começa coletando os nrs das ordens do banco local

    sql = "SELECT ie_ordem FROM ordens"
    rs.Open sql, Cnn

    'Em seguida, montamos uma string com cada um dos itens separados por
    'aspas, virgulas e parênteses... em suma, ('nrItem1', 'nrItem2', ...)
    strOrdensVistas = "('"
    While Not rs.EOF
        strOrdensVistas = strOrdensVistas & rs!ie_ordem & "','"
        rs.MoveNext
    Wend
    rs.Close

    'Tira o ultimo "','" e fecha
    strOrdensVistas = Left(strOrdensVistas, Len(strOrdensVistas) - 2) & ")"

    'Montamos agora o sql
    sql = "DELETE FROM ordens WHERE hc_status = '" & ORDEM_HC_STATUS_CONSULTADA & "' " & _
          "AND ie_ordem NOT IN " & strOrdensVistas

    'Pimba!
    Cnn.Execute sql

    'Confirma as modificacoes
    Cnn.CommitTrans

    'Retorna sucesso
    OrdensSAPCarregar = True

Exit Function

ErrorHandler:

    'Volta atrás
    Cnn.RollbackTrans
    'Seta o erro
    OrdensSAPCarregar = False

End Function

Public Function ExisteNaCollection(str As String, col As Collection) As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Indica se a string str existe ou nao na
'            collection de strings col
'-----------------------------------------------------------------
'Argumentos: str - string a ser procurada
'            col - collection possivelmente contendo a
'                  string em questao
'-----------------------------------------------------------------
'Retorna: TRUE   se der tudo certo
'         FALSE  caso ocorram problemas
'-----------------------------------------------------------------
'Autor: GLF, 2003-03-17
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim strCol As Variant

    'Começa como se nao houvesse tal item na collection
    ExisteNaCollection = False

    'Para cada string contida na collection
    For Each strCol In col
        'Se a string de procura for igual à string da collection
        If strCol = str Then
            'Eh porque o item existe. Está lá. Eh verdadeiro!
            ExisteNaCollection = True
            Exit Function
        End If
    Next

    'Caso nao tenha achado nenhum, o valor será falso mesmo.
End Function
```

frmNovaOrdem.frm

```
Option Explicit

Public clicouOK As Boolean    'Define se o usuario saiu do FORM clicando em
                        'OK ou cancelando a form (clicando em Cancel ou
                        'fechando a janela)



Private Sub cmdCancelar_Click()
    txtOrdem.Text = ""
    txtProduto.Text = ""
    txtComponente.Text = ""
    txtPlanejado.Text = ""
    clicouOK = False
    Unload Me

End Sub

Private Sub cmdOK_Click()
    'Se algum dos itens nao estiverem preenchidos

    If txtOrdem.Text = "" Or _
        txtProduto.Text = "" Or _
        txtComponente.Text = "" Or _
        txtPlanejado.Text = "" Or _
        txtQtPalete.Text = "" Or _
        txtModelo.Text = "" Or _
        txtTensao.Text = "" Or _
        txtPotencia.Text = "" Or _
        txtRefrigerante.Text = "" Or _
        txtCorrente.Text = "" Or _
        txtResfriador.Text = "" Or _
       (txtCap50LBP.Text = "" And txtCap50HBP.Text = "" And txtCap60LBP.Text = "" And txtCap60HBP.Text =
"") _
        Then
            MsgBox MSG_ERRO_CAMPOS_INCOMPLETOS
    Else
        clicouOK = True
        Me.Visible = False
    End If

End Sub

Private Sub Form_Load()

    clicouOK = False

End Sub

Private Sub txtProduto_LostFocus()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Procura nas ordens locais se existe alguma
'            ordem que descreva as caracteristicas do
'            produto para o codigo digitado em txtProduto.
'            Caso exista, preenche os campos da form com os
'            valores contidos no banco.
'-------------------------------------------------------------
'Autor: GLF, 2003-04-22
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
On Error Resume Next

Dim rs As New Recordset
Dim sql As String

    If txtProduto.Text <> "" Then
        sql = "SELECT * FROM ordens WHERE cd_produto = '" & txtProduto.Text & "'"
        rs.Open sql, Cnn

        If Not rs.EOF Then
```

```
            txtModelo.Text = rs!de_modelo
            txtTensao.Text = rs!de_tensao
            txtPotencia.Text = rs!de_tensao_frequencia
            txtCap50LBP.Text = rs!de_capacidade_50LBP
            txtCap50HBP.Text = rs!de_capacidade_50HBP
            txtCap60LBP.Text = rs!de_capacidade_60LBP
            txtCap60HBP.Text = rs!de_capacidade_60HBP
            txtRefrigerante.Text = rs!de_refrigerante
            txtCorrente.Text = rs!de_corrente
            txtResfriador.Text = rs!de_refrigerador_oleo
        End If
    End If

    rs.Close
    Set rs = Nothing

End Sub
```

Módulo Impressão

ctlImpressao.ctl

```
Private Sub tmrChecaStatus_Timer()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: - Verifica o status da impressora e o "traduz",
'               indicando o seu significado.
'             - Ativa o ícone correto, de acordo com o stats
'-------------------------------------------------------------
'Autor: GLF, 2003-04-16
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    lblMensagem = StatusTraduzir(ctlZebraDriver.Status)

    'Caso esteja com erro
    If ctlZebraDriver.Status < 0 Then
        picErro.Visible = True
        picOK.Visible = False
    Else
        picErro.Visible = False
        picOK.Visible = True
    End If


End Sub

Private Sub UserControl_Initialize()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Seta uma referencia global para que a classe
'             publica de interface possa ter acesso ao controle
'-------------------------------------------------------------
'Autor: GLF, 2003-04-08
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    Set controle = Me


End Sub

Public Function Inicializar(inXMLConfig As IXMLDOMNode) As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Inicializa o driver da impressora
'-----------------------------------------------------------------------------------------
'Argumentos: inXMLConfig - nó XML contendo as configuracoes.
'        Tags do XML:
'        PORTA_SERIAL: obrigatório (1,2,3,...)
'        BAUD_RATE:    obrigatório (1200,2400,...)
'        PARITY:       obrigatório (e,o,n)
'        DATA_BITS:    obrigatório (7,8)
'        STOP_BITS:    obrigatório (1,2)
'---------------------------------------------------------------------------------------
'Retorna: TRUE se tudo deu certo
'          FALSE caso contrário (XML com erro, driver nao conseguindo inicializar)
'---------------------------------------------------------------------------------------
'Autor: GLF, 2003-04-08
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
On Error GoTo ErrorHandler

Dim portaSerial As String
Dim baudRate As String
Dim parity As String
Dim dataBits As String
Dim stopBits As String
Dim carControle As String
Dim carFormatacao As String

  portaSerial = inXMLConfig.selectSingleNode("PORTA_SERIAL").Text
  baudRate = inXMLConfig.selectSingleNode("BAUD_RATE").Text
  parity = inXMLConfig.selectSingleNode("PARITY").Text
  dataBits = inXMLConfig.selectSingleNode("DATA_BITS").Text
  stopBits = inXMLConfig.selectSingleNode("STOP_BITS").Text
```

```
  carControle = inXMLConfig.selectSingleNode("CARACTER_CONTROLE").Text
  carFormatacao = inXMLConfig.selectSingleNode("CARACTER_FORMATACAO").Text

  Inicializar = ctlZebraDriver.Inicializar(portaSerial, baudRate, dataBits, stopBits, parity,
CInt(carControle), CInt(carFormatacao))

  tmrChecaStatus.Enabled = True

Exit Function

ErrorHandler:
    Inicializar = False

End Function

Public Sub MsgMostrar(msg As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Mostra uma mensagem na tela principal do controle,
'            em preto
'-------------------------------------------------------------
'Argumentos: msg - Mensagem a ser mostrada
'-------------------------------------------------------------
'Autor: GLF, 2004-04-08
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    lblMensagem.ForeColor = vbBlack
    lblMensagem.Caption = msg
End Sub

Public Sub ErroMostrar(msg As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Mostra uma mensagem na tela principal do controle,
'            em vermelho
'-------------------------------------------------------------
'Argumentos: msg - Mensagem a ser mostrada
'-------------------------------------------------------------
'Autor: GLF, 2004-04-08
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    lblMensagem.ForeColor = vbRed
    lblMensagem.Caption = msg
End Sub

Public Function Imprimir(XMLEtiqueta As DOMDocument26) As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Envia o comando de impressão para o driver da
'            impressora
'-------------------------------------------------------------
'Argumentos: XMLEtiqueta - Um XML contendo a etiqueta a ser
'            impressa
'-------------------------------------------------------------
'Retorna: TRUE  se conseguiu enviar a impressao para o driver
'         FALSE caso nao tenha conseguido inicializar o
'               driver (porta aberta, por exemplo)
'-------------------------------------------------------------
'Autor: GLF, 2004-04-08
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    ctlZebraDriver.Imprimir XMLEtiqueta

End Function

Public Sub setXSL(leiauteXSL As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Configura no driver a formatação da etiqueta a ser
'            impressa
'-------------------------------------------------------------
'Argumentos: leiauteXSL - XSL com a formatacao da etiqueta
'-------------------------------------------------------------
'Retorna:
'-------------------------------------------------------------
'Autor: GLF, 2004-04-08
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
    ctlZebraDriver.leiauteImpressaoXSL = leiauteXSL
```

```
End Sub

Private Function StatusTraduzir(codStatus As Long) As String
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Traduz um codigo de status para uma frase
'            explicativa
'---------------------------------------------------------------
'Argumentos: codStatus - O codigo de status devolvido pelo
'                        driver da impressora.
'                        O arquivo com as ctes com os valores
'                        dos codigos deve ser inserido no
'                        projeto
'---------------------------------------------------------------
'Retorna: string com o texto descrevendo o status
'---------------------------------------------------------------
'Autor: GLF, 2004-04-08
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    Select Case codStatus
        Case STATUS_DESLIGADA, STATUS_OFFLINE
            StatusTraduzir = "Impressora desligada/offline!"
        Case STATUS_SEM_PAPEL
            StatusTraduzir = "Impressora sem papel!"
        Case STATUS_SEM_RIBBON
            StatusTraduzir = "Impressora sem ribbon!"
        Case STATUS_CABECOTE_ABERTO
            StatusTraduzir = "Cabeçote aberto!"
        Case STATUS_INICIALIZANDO
            StatusTraduzir = "Inicializando impressora..."
        Case STATUS_IMPRIMINDO
            StatusTraduzir = ""
        Case STATUS_PRONTA
            StatusTraduzir = ""
        Case STATUS_BUFFER_CHEIO
            StatusTraduzir = "Buffer cheio!"
        Case STATUS_FORMATO_ETIQUETA_INCOMPLETO
            StatusTraduzir = "Formato de etiqueta incompleto!"
        Case STATUS_RAM_CORROMPIDA
            StatusTraduzir = "Memória RAM corrompida!"
        Case STATUS_TEMPERATURA_ALTA
            StatusTraduzir = "Temperatura muito alta!"
        Case STATUS_TEMPERATURA_BAIXA
            StatusTraduzir = "Temperatura muito baixa!"
        Case STATUS_ERRO_GERAL
            StatusTraduzir = "Erro geral!"
        Case Else
            StatusTraduzir = "Erro geral!"
    End Select

End Function
```

clsImpressora.cls

```
Option Explicit

Public Sub setXSL(leiauteXSL As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Determina a formatação da etiqueta a ser impressa
'---------------------------------------------------------------
'Argumentos: leiauteXSL - XSL com a formatacao da etiqueta
'---------------------------------------------------------------
'Retorna:
'---------------------------------------------------------------
'Autor: GLF, 2004-04-08
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
    controle.setXSL leiauteXSL

End Sub
Public Function Imprimir(XMLEtiqueta As DOMDocument26) As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Envia o comando de impressao para o controle
'---------------------------------------------------------------
```

```
'Argumentos: XMLEtiqueta - Um XML contendo a etiqueta a ser
'            impressa
'-------------------------------------------------------------
'Retorna: TRUE  se conseguiu enviar a impressao para o driver
'         FALSE caso nao tenha conseguido inicalizar o
'               driver (porta aberta, por exemplo)
'-------------------------------------------------------------
'Autor: GLF, 2004-04-08
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
       controle.Imprimir XMLEtiqueta

End Function

Public Function Inicializar(inXMLConfig As IXMLDOMNode) As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Inicializa o driver de impressora
'-------------------------------------------------------------
'Argumentos: inXMLConfig - nó XML contendo as configuracoes da
'                          impressora, no formato:
'                          <IMPRESSORA>
'                              <PORTA_SERIAL></PORTA_SERIAL>
'                              <BAUD_RATE></BAUD_RATE>
'                              <DATA_BITS></DATA_BITS>
'                              <STOP_BITS></STOP_BITS>
'                              <PARITY></PARITY>
'                          </IMPRESSORA>
'-------------------------------------------------------------
'Retorna: TRUE  se conseguiu enviar a impressao para o driver
'         FALSE caso nao tenha conseguido inicalizar o
'               driver (porta aberta, por exemplo)
'-------------------------------------------------------------
'Autor: GLF, 2004-04-08
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    Inicializar = controle.Inicializar(inXMLConfig)

End Function
```

Driver da Impressora

ctlZebraS400Driver.ctl

```
Option Explicit
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Driver para impressora Zebra Stripe 400, implementado segundo
'os novos padrões para drivers de impressoras.
'-------------------------------------------------------------
'Autor: GLF, 2003-03-28
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

'Propriedade indicando status da impressora
Private myStatus As Long

'Objeto contendo leiaute de etiqueta
Private myXSLEtiqueta As DOMDocument26

'Parametro de impressao
Const fimcomando = vbCrLf

'Estilo de mensagem
Public Enum MensagemEstilo
    eNormal = 1
    eErro
End Enum

'controle status
Private RecebeuResposta As Boolean
Private FormatosBufferImpressora As Integer

'Spool contendo as etiquetas, ja formatadas, a serem impressas
Private spool As New Collection

'numero de vezes que o aplicativo tenta inicializar a impressora
Private vezes As Integer

Private chrCtrl As String   'Caracter de controle
Private chrFrm As String 'Caracter de formatacao


Public Property Let leiauteImpressaoXSL(inXSL As String)
'**********************************************************************
'Finalidade: Seta o atributo com o formato de impressao dos dados e
'            das etiquetas
'Autor:      GLF em 2002-10-30
'**********************************************************************

    'Carrega o leiaute de etiqueta
    myXSLEtiqueta.loadXML (inXSL)

End Property

Public Property Get status()
'**********************************************************************
'Finalidade: Informa o status da impressora
'Autor:      GLF em 2003-03-28
'**********************************************************************
    status = myStatus

End Property

Private Sub ctlSerial_RecebeMensagem(ByVal Msg As String)
    Me.MensagemRecebeu Msg
End Sub


Private Sub tmrImpressao_Timer()
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Checa se há algo para ser impresso no spool.
'            Caso exista, checa status da impressora pra ver
'            se ela está apta a imprimir. Sendo este o caso,
```

```
'            comanda a impressão e retira o item impresso do
'            spool.
'-------------------------------------------------------------
'Autor: GLF, 2003-04-01
'=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim C As Integer

On Error Resume Next

    'Se há algo no spool
    If spool.Count > 0 Then

        'Checa status da impressora
        C = ImpressoraChecar

        'Se a resposta for que nao há erros, e se o nr. de formatos
        'no buffer for até NR_MAX_FORMATOS_NO_BUFFER, envia o item
        'para impressão e o retira do spool.

        'Caso contrário, aguarda até que o erro seja resolvido ou
        'que a impressora esteja com mais memória

        If C >= 0 And C <= NR_MAX_FORMATOS_NO_BUFFER Then

            'DoEvents

            'Envia o primeiro item do spool pra impressora
            ctlSerial.Enviar spool.Item(1) & vbCrLf
            'Remove-o da lista
            spool.Remove 1
            'Caso ainda tenham itens no spool
            If spool.Count > 0 Then
                'Seta status
                myStatus = STATUS_IMPRIMINDO
            'Caso contrario, deixa o status como pronta
            Else
                myStatus = STATUS_PRONTA
            End If

        'Caso o buffer já esteja no limite especificado
        ElseIf C > NR_MAX_FORMATOS_NO_BUFFER Then

            myStatus = STATUS_IMPRIMINDO
            'Aguarda a proxima ativacao do timer para verificar se ja pode
            'enviar mais dados pra impressora

        'Caso o status da impressora não esteja ok
        ElseIf C < 0 Then
            'Seta o status indicando o erro
            myStatus = StatusTraduzir(C)

        End If 'Verificacao de status, erros e limite de buffer

    Else 'Caso nao tenha nada no spool

        'Faz a checagem mesmo assim
        C = ImpressoraChecar

        'Tendo mais de um item no buffer
        If C > 0 Then
            myStatus = STATUS_IMPRIMINDO
        'Estando o buffer zerado
        ElseIf C = 0 Then
            myStatus = STATUS_PRONTA
        'Tendo ela respondido com codigo de erro
        Else
            myStatus = StatusTraduzir(C)
        End If

    End If

End Sub

Private Sub tmrSetagem_Timer()
```

```
'****************************************************************
'Finalidade: testa o status da impressora e inicializa a impressora
'
'Autor:     GLF em 2003-04-01
'****************************************************************
Dim C As Integer
On Error GoTo ErrorHandler

  'Verifica status da impressora
  C = ImpressoraChecar

  'Se estiver ok, inicializa, limpando os buffers, e habilita a impressao
  If C >= 0 Then

    DoEvents
    'Inicializa impressora
    ImpressoraBufferLimpar
    'Seta status como pronta
    myStatus = STATUS_PRONTA
    'Desliga o timer de setagem
    tmrSetagem.Enabled = False
    'Liga o timer de impressão
    tmrImpressao.Enabled = True

  'Do contrario, pula e tenta de novo
  Else

    myStatus = StatusTraduzir(C)

  End If

Exit Sub

ErrorHandler:
'Caso dê erro (no caso, na limpeza do buffer), nao chega a setar status de pronta.
'O timer vai ser ativado e ocorrerá uma nova tentativa, até que
'funcione.

End Sub

Public Function Inicializar(portaSerial As String, baudRate As String, dataBits As String, stopBits As
String, parity As String, carControle As Integer, carFormatacao As Integer) As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Inicializa o controle, abre a porta serial, liga o timer
'            para configurar a impressora, se tudo certo habilita
'            a impressão. Caso a porta serial nao consiga ser aberta,
'            seta o status de erro e sai da função.
'----------------------------------------------------------------------
'Parâmetros: portaSerial - Nr da porta serial da impressora
'            baudRate    - baudRate a ser utilizada. Ex: 9600
'            dataBits    - Ex: 8
'            stopBits    - Ex: 1
'            parity      - Paridade (N, E)
'            carControle - Código ASCII do caracter a ser utilizado
'                          para controle (que é, normalmente, ~)
'            carFormatacao-Codigo ASCII do caracter a ser utilizado
'                          para formatacao (normalmente, ^)
'----------------------------------------------------------------------
'Retorna:    TRUE   se conseguiu abrir a porta. Nao garante que a
'                   impressora esteja on-line, bonitinha, pronta pra
'                   imprimir
'            FALSE  caso nao tenha conseguido abrir a porta.
'----------------------------------------------------------------------
'Autor:      GLF, 2003-04-01
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
On Error GoTo ErrorHandler

    'Comeca com status de erro na inicalizacao
    Inicializar = False

    'Seta status indicando a inicialização da impressora
    myStatus = STATUS_INICIALIZANDO

    'Tenta iniciar a comunicação serial
```

```
    If Not ctlSerial.Init(CInt(portaSerial), baudRate, parity, dataBits, stopBits,
Chr(ASCIICaracterInicio), Chr(ASCIICaracterFim)) Then
        Inicializar = False
        Exit Function
    End If

    'Cria objeto para leiaute de etiqueta
    Set myXSLEtiqueta = New DOMDocument26
    myXSLEtiqueta.async = False

    'numero de tentativas de inicialização da impressora
    vezes = 0

    'Seta o tempo de intervalo dos timers
    tmrImpressao.Interval = TempoTimer
    tmrSetagem.Interval = TempoTimer

    'Define o caracter de controle e de formatacao
    chrCtrl = Chr(carControle)
    chrFrm = Chr(carFormatacao)

    'Manda inicializar a impressora, checando o seu status e limpando os buffers
    tmrSetagem.Enabled = True

    'Indica que a abertura da porta serial deu-se com sucesso
    Inicializar = True

Exit Function

ErrorHandler:
    Inicializar = False
End Function

Private Sub ImpressoraBufferLimpar()
'*********************************************************************
'Finalidade: Limpa o buffer da impressora
'
'Modif.:     GLF em 2003-04-01
'*********************************************************************
Dim Buffer As String

  '*** Limpa buffer da impressora ***
  Buffer = ""
  'Limpa formatos na memória da impressora
  Buffer = chrCtrl & "EF" & fimcomando
  'Limpa gráficos da memória
  Buffer = Buffer & chrCtrl & "EG" & fimcomando
  'Limpa formatos do buffer (só para garantir)
  Buffer = Buffer & chrCtrl & "JA" & fimcomando
  'Limpa última etiqueta impressa da memória
  Buffer = Buffer & chrFrm & "XA" & chrFrm & "MCY" & chrFrm & "XZ" & fimcomando
  'Não reemprimir depois de erro
  'IMPORTANTÍSSIMO!!!!
  Buffer = Buffer & chrFrm & "XA" & chrFrm & "JZN" & chrFrm & "XZ" & fimcomando

  'manda para a impressora
  ctlSerial.Enviar Buffer

End Sub

Friend Sub MensagemRecebeu(Msg As String)
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade:  Procedimento que é chamado quando foi recebido
'             algo da porta serial
'-----------------------------------------------------------
'Argumentos: Msg - Mensagem enviada pela impressora contendo
'            o status da mesma
'-----------------------------------------------------------
'Efeito:
'Seta globais: RecebeuResposta quando recebeu os 3 strings
'              FormatosBufferImpressora, com os valores:
'              - 1: sem papel
'              - 2: buffer lotado
'              - 3: formato incompleto
```

```
'              - 4: memória RAM corrompida
'              - 5: Temperatura baixa
'              - 6: Temperatura alta
'              - 7: cabeçote levantado
'              - 8: sem ribbon
'              - 96: erro ao verificar formatos no buffer
'              - 97: erro na resposta
'-------------------------------------------------------------
'Autor: GLF em 2003-04-03
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=

On Error GoTo ErrorHandler

  Debug.Print "Recebi Mensagem: " & Msg

    'se tiver 34 caracteres eh string com dados - 1a ou 2a
    If Len(Msg) = 34 Then
        '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
        '1a String
        '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
        ' se tiver virgula na 31 eh primeira string
        If Mid(Msg, 31, 1) = "," Then
            'seta qt formatos no buffer
            If IsNumeric(FormatosBufferImpressora) Then
                FormatosBufferImpressora = Val(Mid(Msg, 14, 3))
            Else
                FormatosBufferImpressora = -96
            End If
            'procura por erros
            'sem papel
            If Mid(Msg, 5, 1) <> "0" Then
                FormatosBufferImpressora = -1
            End If
            'buffer full
            If Mid(Msg, 18, 1) <> "0" Then
                FormatosBufferImpressora = -2
            End If
            'formato incompleto
            If Mid(Msg, 22, 1) <> "0" Then
                FormatosBufferImpressora = -3
            End If
            'RAM corrompida
            If Mid(Msg, 28, 1) <> "0" Then
                FormatosBufferImpressora = -4
            End If
            'Temperatura baixa
            If Mid(Msg, 30, 1) <> "0" Then
                FormatosBufferImpressora = -5
            End If
            'Temperatura alta
            If Mid(Msg, 32, 1) <> "0" Then
                FormatosBufferImpressora = -6
            End If
        '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
        '2a String
        '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
        Else 'eh segunda string
            'procura por erros
            'head up
            If Mid(Msg, 7, 1) <> "0" Then
                FormatosBufferImpressora = -7
            End If
            'ribbon out
            If Mid(Msg, 9, 1) <> "0" Then
                FormatosBufferImpressora = -8
            End If
        End If
    '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
    '3a String
    '-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
    ElseIf Len(Msg) = 8 Then 'e terceira string
        RecebeuResposta = True
    Else 'deu erro
        FormatosBufferImpressora = -97
```

```
        RecebeuResposta = True
    End If

Exit Sub

ErrorHandler:
    Debug.Print "Erro na decodificação da mensagem"
End Sub

Private Function StatusTraduzir(codigoErroImpressora As Integer) As enmStatusZebraS400
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Pega os códigos de retorno da impressora
'            e os converte para status no padrão da HarboR
'-------------------------------------------------------------
'Argumentos: codigoErroImpressora - Codigos da impressora
'            especifica do driver. Neste caso, sao os retornos
'            possiveis da funcao ImpressoraChecar
'-------------------------------------------------------------
'Retorna:   Enumerado contendo o status da impressora
'           Nota: Todos os valores de erro sao negativos
'-------------------------------------------------------------
'Autor: GLF, 2003-04-01
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

    Select Case codigoErroImpressora
        Case -1 'Sem papel
            StatusTraduzir = STATUS_SEM_PAPEL
        Case -2 'Buffer lotado
            StatusTraduzir = STATUS_BUFFER_CHEIO
        Case -3 'Formato etiqueta incompleto
            StatusTraduzir = STATUS_FORMATO_ETIQUETA_INCOMPLETO
        Case -4 'Memória RAM corrompida
            StatusTraduzir = STATUS_RAM_CORROMPIDA
        Case -5 'Temperatura baixa
            StatusTraduzir = STATUS_TEMPERATURA_BAIXA
        Case -6 'temperatura alta
            StatusTraduzir = STATUS_TEMPERATURA_ALTA
        Case -7 'head up
            StatusTraduzir = STATUS_CABECOTE_ABERTO
        Case -8 'ribbon out
            StatusTraduzir = STATUS_SEM_RIBBON
        Case -96, -97, -99
                '-96: erro ao verificar formatos no buffer
                '-97: erro ao decodificar a resposta da impressora
                '-99: erro geral ao checar status
            StatusTraduzir = STATUS_ERRO_GERAL
        Case -98 'sem resposta
            StatusTraduzir = STATUS_DESLIGADA
    End Select

End Function

Private Function ImpressoraChecar() As Integer
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Verifica status da impressora, enviando comando
'            '~HS' e aguardando resposta pela serial até
'            TimeoutResposta
'-------------------------------------------------------------
'Retorna:   número de formatos no buffer
'           ou código de erro
'           -1 sem papel
'           -2 buffer lotado
'           -3 formato etiqueta incompleto
'           -4 memória RAM corrompida
'           -5 Temperatura baixa
'           -6 temperatura alta
'           -7 head up
'           -8 ribbon out
'           -96 erro ao verificar formatos no buffer
'           -97 erro ao decodificar a resposta da impressora
'           -98 sem resposta
'           -99 erro geral ao checar status
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
Dim t As Long
```

```
Dim BufferOut As String
Dim i As Integer

On Error GoTo Error

    RecebeuResposta = False

    'vamos tentar status por n vezes
    Do While Not RecebeuResposta And i < 10
      t = 0
      'pede o status da impressora
      BufferOut = chrCtrl & "HS" & fimcomando
      ctlSerial.Enviar BufferOut

      Sleep (10)
      'verifica pela resposta da impressora
      DoEvents
      'enquanto não receber resposta, espera
      Do While Not RecebeuResposta And t < TimeoutResposta
        Sleep (10)
        DoEvents
        t = t + 20
      Loop
      i = i + 1
    Loop

    'Se recebeu resposta, retorna a interpretação da resposta,
    'que pode ser o nr. de formatos no buffer (>= 0) ou o codigo de erro (< 0)
    If RecebeuResposta Then
      ImpressoraChecar = FormatosBufferImpressora
    'Do contrário, reporta um erro
    Else
      ImpressoraChecar = -98
    End If

Exit Function

Error:
   ImpressoraChecar = -99

End Function

Private Sub UserControl_Initialize()

#If Desenvolvimento = 1 Then
        On Error Resume Next
        Inicializar 1, 9600, 8, 1, "N", 126, 94

'        Dim xmlTeste As New DOMDocument26
'        xmlTeste.Load ("C:\rastreabilidade\etiqueta.xml")
'        Dim xslTeste As New DOMDocument26
'        xslTeste.Load ("c:\rastreabilidade\etiqueta.xsl")
'        Me.leiauteImpressaoXSL = xslTeste.xml
'        Me.Imprimir xmlTeste

#End If

End Sub

Private Function XMLTraduzir(inXML) As String
'*********************************************************************
'Finalidade: Traduzir escapes de XML
'Retorna:    string traduzida
'Autor: GLF em 2003-04-03
'*********************************************************************
Const PROC_SIG = "XMLTraduzir"
Dim myAux As String
Dim myResultado As String
Dim i As Integer
Dim myInicio As Integer

On Error GoTo ErrorHandler

    myInicio = 1
```

```
    myResultado = inXML

    'Se existir escape \\a no XML
    If InStr(myInicio, myResultado, "\\a") > 0 Then
        While InStr(myInicio + 1, myResultado, "\\a") > 0
            'Valor da string antes do escape
            myInicio = InStr(myInicio, myResultado, "\\a")
            'Valor definido no scape
            myAux = Mid(myResultado, myInicio + 3, 3)
            If myAux > 155 Then
                'Substitui o escape pelo caracter definido
                myResultado = Mid(myResultado, 1, myInicio - 1) & Chr(myAux) & Mid(myResultado, myInicio +
6)
            End If
        Wend
    Else
        myResultado = inXML
    End If


    XMLTraduzir = myResultado
    Exit Function

ErrorHandler:
    Err.Raise ErroXMLTraduzir, "XMLTraduzir", Err.Description
End Function

Private Function CaracterEspecialEscape(inPalavra As String) As String
'************************************************************
'Finalidade: substituir caracteres que não podem ser impressos
'            na impressora Zebra S400 por caracteres que podem
'
'Parametros: inPalavra - string contendo um caracter
'Retorna:    caracter que pode ser impresso pela impressora
'
'Autor: GLF em 2003-04-03
'************************************************************

    '&   <    >
    inPalavra = Replace(inPalavra, "&gt;", ">")
    inPalavra = Replace(inPalavra, "&lt;", "<")
    inPalavra = Replace(inPalavra, "&amp;", "&")

    'A
    inPalavra = Replace(inPalavra, "Ã", "A")
    inPalavra = Replace(inPalavra, "Â", "A")
    inPalavra = Replace(inPalavra, "Á", "A")
    inPalavra = Replace(inPalavra, "À", "A")
    inPalavra = Replace(inPalavra, "Ä", "A")
    'a
    inPalavra = Replace(inPalavra, "ã", "a")
    inPalavra = Replace(inPalavra, "â", "a")
    inPalavra = Replace(inPalavra, "á", "a")
    inPalavra = Replace(inPalavra, "à", "a")
    inPalavra = Replace(inPalavra, "ä", "a")
    'E
    inPalavra = Replace(inPalavra, "Ê", "E")
    inPalavra = Replace(inPalavra, "É", "E")
    inPalavra = Replace(inPalavra, "È", "E")
    inPalavra = Replace(inPalavra, "Ë", "E")
    'e
    inPalavra = Replace(inPalavra, "ê", "e")
    inPalavra = Replace(inPalavra, "é", "e")
    inPalavra = Replace(inPalavra, "è", "e")
    inPalavra = Replace(inPalavra, "ë", "e")
    'I
    inPalavra = Replace(inPalavra, "Î", "I")
    inPalavra = Replace(inPalavra, "Í", "I")
    inPalavra = Replace(inPalavra, "Ì", "I")
    inPalavra = Replace(inPalavra, "Ï", "I")
    'i
    inPalavra = Replace(inPalavra, "î", "i")
    inPalavra = Replace(inPalavra, "í", "i")
    inPalavra = Replace(inPalavra, "ì", "i")
```

```
    inPalavra = Replace(inPalavra, "ï", "i")
    'O
    inPalavra = Replace(inPalavra, "Õ", "O")
    inPalavra = Replace(inPalavra, "Ô", "O")
    inPalavra = Replace(inPalavra, "Ó", "O")
    inPalavra = Replace(inPalavra, "Ò", "O")
    inPalavra = Replace(inPalavra, "Ö", "O")
    'o
    inPalavra = Replace(inPalavra, "õ", "o")
    inPalavra = Replace(inPalavra, "ô", "o")
    inPalavra = Replace(inPalavra, "ó", "o")
    inPalavra = Replace(inPalavra, "ò", "o")
    inPalavra = Replace(inPalavra, "ö", "o")
    'U
    inPalavra = Replace(inPalavra, "Û", "U")
    inPalavra = Replace(inPalavra, "Ú", "U")
    inPalavra = Replace(inPalavra, "Ù", "U")
    inPalavra = Replace(inPalavra, "Ü", "U")
    'u
    inPalavra = Replace(inPalavra, "û", "u")
    inPalavra = Replace(inPalavra, "ú", "u")
    inPalavra = Replace(inPalavra, "ù", "u")
    inPalavra = Replace(inPalavra, "ü", "u")
    'Ç
    inPalavra = Replace(inPalavra, "Ç", "C")
    'ç
    inPalavra = Replace(inPalavra, "ç", "c")

    'retorna resultado da função
    CaracterEspecialEscape = inPalavra

End Function

Public Function Imprimir(xmlEtiquetas As Object) As Integer
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Finalidade: Recebe um XML com uma ou mais etiquetas, aplica a formatação
'            XSL definida na variavel leiauteXSL e joga no spool
'            para impressão.
'-----------------------------------------------------------------------
'Parâmetros: xmlEtiquetas - XML contendo informações da(s) etiqueta(s)
'                           a ser(em) impressa(s)
'-----------------------------------------------------------------------
'Retorna:    RESULT_ERRO_XSL
'            RESULT_ERRO_XML
'-----------------------------------------------------------------------
'Autor:      GLF, 2003-04-01
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
Dim xmlNodes As Object           'Contém nós da árvore, ou etiquetas
Dim xmlEtiqueta As DOMDocument26 'Contém o nó traduzido
Dim etiqueta As String           'String contendo a versao transformada e pronta da etiqueta
Dim i As Integer                 'Auxiliar

On Error GoTo ErrorHandler

    'Seta para carregar arquivo sincronizado
    xmlEtiquetas.async = False

    If (myXSLEtiqueta.xml = "") Then
        Imprimir = RESULT_ERRO_XSL
        Exit Function
    End If

    'Cria objeto que conterá os dados de uma etiqueta individual
    Set xmlEtiqueta = New DOMDocument26

    On Error GoTo ErrorHandler

    'Se o XML das etiquetas nao for vazio
    If xmlEtiquetas.Text <> "" Then

        'Captura os nós "ETIQUETA", ou seja, captura todas os dados de
        'etiquetas do xml passado a serem impressas
        Set xmlNodes = xmlEtiquetas.getElementsByTagName("ETIQUETA")
```

```
        'Monta todas as etiquetas passadas no xml e adiciona ao
        'spool
        For i = 0 To xmlNodes.length - 1

            'Captura os dados de uma etiqueta
            xmlEtiqueta.loadXML (XMLTraduzir(xmlNodes.Item(i).xml))

            'Faz a transformacao dos dados a partir do formato definido em XslEtiqueta,
            'e converte os caracteres especiais em caracteres que possam ser impressos
            etiqueta = CaracterEspecialEscape(xmlEtiqueta.transformNode(myXSLEtiqueta))

            'Adiciona etiqueta já processada e formatada ao spool de impressao
            spool.Add etiqueta

        Next i

    Else
    'XML com a(s) etiqueta(s) esta vazio
        Imprimir = RESULT_ERRO_XML
    End If

    Set xmlEtiqueta = Nothing

    Imprimir = RESULT_ENVIO_IMPRESSAO_OK

Exit Function

'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Controle de erros
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
ErrorHandler:
    'Indica que esta impressao nao funcionou, nao foi adicionada
    'ao spool corretamente em virtude de algum problema com o XML
    Imprimir = RESULT_ERRO_XML

End Function

Private Sub UserControl_Resize()

    'Ajusta o tamanho do user control para o mesmo tamanho da figura
    UserControl.Height = 32 * 15            '32 pixels
    UserControl.Width = 32 * 15             '32 pixels

End Sub
```

ctesInternasZebraS400.bas

```
'Constantes gerais do driver
Public Const ASCIICaracterInicio = 2
Public Const ASCIICaracterFim = 10
Public Const TamanhoSaida = 512
Public Const TempoTimer = 400
Public Const TimeoutResposta = 1000 'was: 200
Public Const NumMaxVezes = 20

'Constante que indica o número máximo de formatos no buffer que o driver
'permitirá (para evitar problemas de buffer cheio).
Public Const NR_MAX_FORMATOS_NO_BUFFER = 2

'Propriedades txtBox de Log (Máximo suportado pelo controle = 32K)
Public Const MaxLenTxLog = 5000

Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

'arquivo de log para debug
Public Const debugNumArquivo = 99

'arquivos defaults de formato de etiqueta
'Public Const ARQUIVO_XSL_DEFAULT_REDE = "http://laminas.embraco.com.br/xml/etiqueta_formato.xsl"
'Public Const ARQUIVO_XSL_DEFAULT_LOCAL = "c:/temp/etiqueta_formato.xsl"

Public Const ErroXMLTraduzir = vbObjectError + 1000
'Public Const ErroEnviar = vbObjectError + 1001
```

```
'Public Const ErroReceber = vbObjectError + 1002
```

ctesZebraS400.bas

```
'Constantes publicas para uso com controle ZebraS400Driver

Public Enum enmStatusZebraS400
'Status genericos do driver de impressao
    STATUS_DESLIGADA = -1   ' ou offline
    STATUS_OFFLINE = -2
    STATUS_SEM_PAPEL = -3
    STATUS_SEM_RIBBON = -4  'ou tinta
    STATUS_CABECOTE_ABERTO = -5
    STATUS_INICIALIZANDO = -6
    STATUS_IMPRIMINDO = -7
    STATUS_PRONTA = 1

'Status especificos para esta impressora
    STATUS_BUFFER_CHEIO = -10
    STATUS_FORMATO_ETIQUETA_INCOMPLETO = -11
    STATUS_RAM_CORROMPIDA = -12
    STATUS_TEMPERATURA_BAIXA = -13
    STATUS_TEMPERATURA_ALTA = -14
    STATUS_ERRO_GERAL = -15

End Enum

'Resultados de funcoes
Public Const RESULT_ERRO_XSL = -1
Public Const RESULT_ERRO_XML = -2
```

Arquivos de uso comum

clsCodigoBarras.cls

```
Option Explicit

Private listaConfigs As IXMLDOMNodeList  'Lista de nós contendo, cada um, as configuracoes

Public Sub init(inXMLConfig As Object)

    'Pega a lista de configuracoes
    Set listaConfigs = inXMLConfig.childNodes

End Sub

Public Function ValoresRetornar(codigoBarrasRaw As String, CD As String, NS As String) As Boolean
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Lê do arquivo XML as configuracoes possiveis de
'            posicoes de NS e CD no codigo de barras, e
'            descobre, baseado no tamanho do codigo e na
'            presenca ou nao de unidade separadora, de qual
'            configuracao aquele codigo se trata.
'            Preenche NS e CD baseado nessas posicoes.
'-------------------------------------------------------------
'Argumentos: codigoBarrasRaw - O valor bruto, completo, do
'                              codigo de barras
'            CD - É preenchido com o código do
'                 produto/componente
'            NS - É preenchido com o número serial do
'                 produto/componente
'-------------------------------------------------------------
'Retorna: TRUE se achou a configuracao do código e ele estava
'              correto de acordo com essa configuracao
'
'         FALSE caso nao se tenha achado uma configuracao valida
'               no XML de configuracao, ou o codigo nao esteja
'               de acordo com a configuração.
'-------------------------------------------------------------
'Autor: GLF, 2003-01-30
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Dim codRawTamanho As Integer                'Tamanho da string com o codigo bruto
Dim codRawTemUnidadeSeparadora As Boolean   'Verificacao da existencia de carac. sep. na string
Dim configTemUnidadeSeparadora As Boolean   'Verificacao da existencia de carac. sep. em uma config.
Dim unidadeSeparadora As String             'Caracter separador para cada config.

Dim CDpos As Integer                        'Posicao do codigo
Dim CDlen As Integer                        'Comprimento do codigo
Dim NSpos As Integer                        'Posicao do numero de série
Dim NSlen As Integer                        'Comprimento do numero de série

Dim config As IXMLDOMNode                      'Cada uma das configuracoes possiveis

On Error GoTo ErrorHandler

    'Limpa eventuais valores que estejam nas variaveis
    'de retorno
    CD = ""
    NS = ""

    'Determina o tamanho da string bruta do codigo de barras
    codRawTamanho = Len(codigoBarrasRaw)

    'Para cada configuracao possivel
    For Each config In listaConfigs

        'Se o nr de caracteres dessa configuracao for o mesmo do atual
        'codigo de barras
        If config.Attributes.getNamedItem("LEN").Text = codRawTamanho Then

            'Pesquisa qual a unidade separadora, se existente, dessa
```

```
'configuracao
On Error Resume Next
    unidadeSeparadora = config.selectSingleNode("SEPARADOR").Text
On Error GoTo ErrorHandler

'Verifica se a atual configuracao tem ou nao unidade separadora
configTemUnidadeSeparadora = (unidadeSeparadora <> "")

'-------------------------------------------------------------
'SE A CONFIGURACAO SENDO PESQUISADA TEM A UNIDADE SEPARADORA
'-------------------------------------------------------------
If configTemUnidadeSeparadora Then

    'Descobre se a string com o codigo de barras tem tal unidade separadora
    codRawTemUnidadeSeparadora = CBool(InStr(codigoBarrasRaw, unidadeSeparadora))

    'Se a string tambem apresenta a unidade separadora, tudo certo!
    'Trata e pega os valores. Caso nao tenha, eh sinal que essa configuracao
    'nao é valida para essa string. Passa para a proxima e testa novamente.
    If codRawTemUnidadeSeparadora Then

        'Pega a posicao, no codigo de barras, onde está o código
        'Aqui, a posicao refere-se ao bloco (antes ou entre as unidades separadoras)
        'onde se encontra o campo desejado
        'Ex:      1     2   3
        '      1000001#300#7543

        CDpos = config.selectSingleNode("CD_POS").Text

        'Obtém o conteúdo do bloco
        CD = CampoRetornar(codigoBarrasRaw, CDpos, unidadeSeparadora)

        'Pega do config a posicao do numero de série
        NSpos = config.selectSingleNode("NS_POS").Text

        'Obtém o valor do campo correspondente
        NS = CampoRetornar(codigoBarrasRaw, NSpos, unidadeSeparadora)

        'Verifica se obteve os valores com sucesso
        If (CD <> "") And (NS <> "") Then
            ValoresRetornar = True
            Exit Function
        Else
            ValoresRetornar = False
            Exit Function
        End If

    End If

'SE A CONFIGURACAO SENDO PESQUISADA NAO APRESENTA UNIDADE SEPARADORA
Else

    'Extrai do config a posicao do codigo do produto no codigo de barras
    CDpos = config.selectSingleNode("CD_POS").Text

    'Extrai o comprimento do codigo
    CDlen = config.selectSingleNode("CD_LEN").Text

    'Extrai o valor do codigo
    CD = Mid(codigoBarrasRaw, CDpos, CDlen)

    'Extrai do config a posicao do numero serial no codigo de barras
    NSpos = config.selectSingleNode("NS_POS").Text

    'Extrai o comprimento do numero de serie
    NSlen = config.selectSingleNode("NS_LEN").Text

    'Extrai o valor do NS
    NS = Mid(codigoBarrasRaw, NSpos, NSlen)

    'Verifica se obteve os valores com sucesso
    If (CD <> "") And (NS <> "") Then
         ValoresRetornar = True
         Exit Function
```

```
                Else
                     ValoresRetornar = False
                     Exit Function
                End If

            End If 'Fim de se tem unidade separadora
        End If 'Fim de se o tamanho do codigo for o mesmo da atual configuracao

    Next 'Vai para a proxima configuracao

    'Se chegou aqui, é porque nenhuma das configuracoes existentes
    'no XML é valida para o codigo de barras passado para a funcao.
    'Indica isso.
    ValoresRetornar = False

    Exit Function

'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Controle de erros
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
ErrorHandler:

   ValoresRetornar = False

End Function


Private Function CampoRetornar(str As String, nrCampo As Integer, caracterSeparador As String) As String
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-
'Finalidade: Descobre, em uma determinada string contendo
'            campos separados por um caracter separador,
'            o campo de numero especificado e retorna
'            o seu valor.
'------------------------------------------------------------
'Argumentos: str -      A string com os campos e caracteres separadores
'            nrCampo -  Numero do campo do qual deseja-se obter
'                       o valor, partindo de 1.
'
'                       Ex:      1     2   3
'                             1000001#300#7543
'            caracterSeparador -
'                       Define o caracter usado para separar os
'                       campos.
'--------------------------------------------------------------------
'Retorna: -  "" caso o nr do campo desejado nao exista na string
'            fornecida
'         -  O valor do campo, caso este seja encontrado com
'            sucesso
'-------------------------------------------------------------
'Autor: GLF, 2003-01-30
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-

Dim pos As Integer           'Posicao do ponteiro
Dim aux As String
Dim contador As String       'Contador de campos ja pesquisados

On Error GoTo ErrorHandler

    'Inicia o contador de campos e a primeira posicao
    contador = 1
    pos = 1

    'Passa a string total para uma variavel auxiliar, que inicia com a string inteira
    aux = str

    'Descobre a posicao do primeiro caracter separador, delimitando o fim do primeiro campo
    pos = InStr(pos, aux, caracterSeparador)

    'Enquanto nao chegar no campo desejado
    While contador <= nrCampo

        'Se ja esta no campo certo
        If nrCampo = contador Then
```

```
            'Descobre a posicao do proximo caracter separador
            pos = InStr(pos, aux, caracterSeparador)

            'Se a pos for zero, eh sinal de que a string nao contém
            'mais caracteres separadores. Ou seja, o campo requisitado
            'é o ultimo campo da string
            If pos = 0 Then
                'Retorna a string inteira
                CampoRetornar = Left(aux, Len(aux))

                'Sai da funcao
                Exit Function

            Else
                'Retorna o valor do comeco da string até o separador, tirando-o
                CampoRetornar = Left(aux, pos - 1)

                'Sai da funcao
                Exit Function
            End If

        Else
            'Descobre a posicao do proximo caracter separador
            pos = InStr(pos, aux, caracterSeparador)

            'Se a pos for zero, eh sinal de que a string nao contém
            'mais caracteres separadores. Ou seja, terminaram os campos
            'mas nao se chegou no campo desejado.
            If pos = 0 Then

                'Indica que ocorreu este erro
                CampoRetornar = ""

                'Sai da funcao
                Exit Function
            Else

                'Pega a string dessa posicao pra frente
                aux = Mid(aux, pos + 1)
                'Incrementa a posicao de campo
                contador = contador + 1

            End If

        End If

    Wend

    'Se chegou aqui, é porque nao encontrou o campo
    CampoRetornar = ""

    Exit Function

'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
'Controle de erros
'-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
ErrorHandler:

   CampoRetornar = False

End Function
```

Constantes.bas

```
Attribute VB_Name = "Constantes"
Option Explicit

'Constantes utilizadas em todo aplicativo

'Tempos
Public Const INTERVALO_LEITURA = 100 'Tempo entre leituras da serial (milisegundos)
Public tempoTimeout As Long 'tempo máximo para usuário responder ao sistema
```

```
Public Const INTERVALO_MENSAGEM_ERRO = 2000 'Intervalo de sleep para mensagens de erro, em milissegundos

'Constantes de status de paletes no banco de dados
Public Const PALETE_HC_STATUS_ABERTO = "Aberto"
Public Const PALETE_HC_STATUS_FECHADO = "Fechado"
Public Const PALETE_HC_STATUS_TRANSFERIDO = "Transferido"
Public Const PALETE_HC_STATUS_TRANSFERINDO = "Transferindo"
Public Const PALETE_HC_STATUS_ERRO = "Erro"

'Constantes de status de ordens no banco de dados
Public Const ORDEM_HC_STATUS_INICIADA = "Iniciada"
Public Const ORDEM_HC_STATUS_CONSULTADA = "Consultada"
Public Const ORDEM_HC_STATUS_ENCERRADA = "Encerrada"

'Constantes de status do banco
Public Const CONEXAO_OK = 1
Public Const CONEXAO_PATH_INVALIDO = -1
Public Const CONEXAO_ERRO_GERAL = -2

'Retornos de funcoes paletizacao
Public Const RESULT_ORDEM_INICIADA = 1
Public Const RESULT_PALETE_ABERTO = 2
Public Const RESULT_ORDEM_ENCERRADA = 3
Public Const RESULT_PRODUTO_INSERIDO = 4
Public Const RESULT_ESTADO_PALETE_ABERTO = 5
Public Const RESULT_ESTADO_ORDEM_INICIADA = 6
Public Const RESULT_ESTADO_INICIAL = 7
Public Const RESULT_PALETE_ABERTO_CRIADO_NOVO = 8
Public Const RESULT_PALETE_FECHADO = 9
Public Const RESULT_PALETE_DESMONTADO = 10
Public Const RESULT_PRODUTO_REMOVIDO = 11
Public Const RESULT_PRODUTO_INSERIDO_COMPLETOU_PALETE = 12


Public Const RESULT_ERRO_ORDEM_INICIAR = -1
Public Const RESULT_ERRO_OPERACAO_CANCELADA = -2
Public Const RESULT_ERRO_LEITURA_INVALIDA = -3
Public Const RESULT_ERRO_TIMEOUT = -4
Public Const RESULT_ERRO_PALETE_ABRIR = -5
Public Const RESULT_ERRO_PALETE_TRANSFERENCIA = -6
Public Const RESULT_ERRO_PALETE_ORDEM_INCORRETA = -7
Public Const RESULT_ERRO_PALETE_ABRIR_GENERICO = -8
Public Const RESULT_ERRO_GENERICO = -9
Public Const RESULT_ERRO_ORDEM_ENCERRAR_GENERICO = -10
Public Const RESULT_ERRO_PRODUTO_INSERIR_GENERICO = -11
Public Const RESULT_ERRO_PRODUTO_INEXISTENTE = -12
Public Const RESULT_ERRO_PRODUTO_SEM_COMPONENTE = -13
Public Const RESULT_ERRO_MONTAGEM_INCORRETA = -14
Public Const RESULT_ERRO_PRODUTO_JA_INSERIDO = -15
Public Const RESULT_ERRO_PRODUTO_INCORRETO = -16
Public Const RESULT_ERRO_PALETE_CHEIO = -17
Public Const RESULT_ERRO_PRODUTO_JA_INSERIDO_OUTRO_PALETE = -18
Public Const RESULT_ERRO_CONFIRMACAO_PALETE = -19
Public Const RESULT_ERRO_PALETE_FECHAR_GENERICO = -20
Public Const RESULT_ERRO_PALETE_DESMONTAR_GENERICO = -21
Public Const RESULT_ERRO_PRODUTO_EM_OUTRO_PALETE = -22
Public Const RESULT_ERRO_PRODUTO_REMOVER_GENERICO = -23
Public Const RESULT_ERRO_PALETE_DESMONTAR_BANCO = -24
Public Const RESULT_ERRO_PALETE_FECHAR_BANCO = -25
Public Const RESULT_ERRO_PRODUTO_REMOVER_BANCO = -26
Public Const RESULT_ERRO_PRODUTO_INSERIR_BANCO = -27
Public Const RESULT_ERRO_PALETE_ABRIR_BANCO = -28
Public Const RESULT_ERRO_NENHUMA_ORDEM_SELECIONADA = -29

'Retornos pra funcao de inicializacao de XML paletizacao
Public Const RESULT_INICIALIZACAO_XML_OK = 1
Public Const RESULT_ERRO_INICIALIZACAO_DISPLAY = -30
Public Const RESULT_ERRO_INICIALIZACAO_LEITOR = -31
Public Const RESULT_ERRO_INICIALIZACAO_IMPRESSORA = -32
Public Const RESULT_ERRO_CARGA_XML = -33

'Retornos de funcoes (mantidos por compatibilidade.
'                     Checar depois se ainda sao utilizados )
```

```
Public Const ORDEM_INICIADA = 1
Public Const ORDEM_ENCERRADA = 1
Public Const PALETE_ABERTO = 1
Public Const PALETE_FECHADO = 1
Public Const PALETE_DESMONTADO = 1
Public Const PRODUTO_INSERIDO = 1
Public Const PRODUTO_RETIRADO = 1
Public Const PRODUTO_LIDO = 1

Public Const OPERACAO_CANCELADA = -1
Public Const TIMEOUT = -2
Public Const LEITURA_INVALIDA = -3

Public Const ERRO_ORDEM_INICIAR = -12
Public Const ERRO_ORDEM_ENCERRAR = -13
Public Const ERRO_ORDENS_SAP_CONSULTAR = -14

Public Const ERRO_PALETE_ABRIR = -21
Public Const ERRO_PALETE_FECHAR = -22

Public Const ERRO_CONSULTA = -1
Public Const ERRO_CONSULTA_HTTP = -2

Public Const XML_RECEBIDO = 1
Public Const XML_ERRO_SERVIDOR = -1
Public Const XML_ERRO_COMUNICAÇÂO = -2

'Retornos de funcoes genealogia
Public Const ERRO_PRODUTO_LEITURA = -23

'Constantes de comandos
Public Const CMD_INICIAR_ORDEM = "I"
Public Const CMD_ENCERRAR_ORDEM = "E"
Public Const CMD_ABRIR_PALETE = "A"
Public Const CMD_FECHAR_PALETE = "F"
Public Const CMD_DESMONTAR_PALETE = "D"
Public Const CMD_REMOVER_PRODUTO = "R"
Public Const CMD_DESMONTAR_PRODUTO = "D"
Public Const CMD_IMPRIMIR_ETIQUETA_REPOSICAO = "I"
Public Const CMD_SIM = "S"
Public Const CMD_NAO = "N"
Public Const CMD_CANCELAR = "C"

'Constantes de interface
Public Const TAMANHO_MAX_LOG = 4000

'*** Chamadas http
Public Const HTTP_ERROS_CONSULTAR = "http://componentes/rastreabilidade/apontamentos/errosConsultar.asp"
Public Const HTTP_PALETES_RECEBER = "http://componentes/rastreabilidade/apontamentos/paletesReceber.asp"
Public Const HTTP_ORDENS_CONSULTAR = "http://componentes/rastreabilidade/ordensPesquisar.asp"
Public Const XML_POST_CONTENT_TYPE = "Content-Type: application/x-www-form-urlencoded"

'*** Especificos PALETIZACAO

'Estados estaveis possiveis
Enum enmEstadoPaletizacao
   PAL_CHECAGEM_INICIAL = 0
   PAL_ESTADO_INICIAL = 1
   PAL_ORDEM_INICIADA = 2
   PAL_PALETE_ABERTO = 3
End Enum
'Comportamentos form Ordens
Enum enmComportamentoFormOrdens
    PAL_FORM_ORDEM_INICIAR = 0
    PAL_FORM_ORDENS_CONSULTAR = 1
End Enum


'*** Especificos GENEALOGIA
Enum enmEstadoGenealogia
   GEN_ESTADO_INICIAL = 0
   GEN_AGUARDANDO_LEITURA_COMPONENTE = 1
   GEN_AGUARDANDO_LEITURA_PRODUTO = 2
   GEN_VERIFICANDO_MONTAGEM = 3
```

```
   GEN_ERRO_COMPONENTE_JA_MONTADO = 4
   GEN_ERRO_PRODUTO_COM_OUTRO_COMPONENTE = 5
   GEN_ERRO_MONTAGEM_INCORRETA = 6
   GEN_ERRO_SEM_ORDEM_SAP_LOCAL = 7
   GEN_ERRO_GENERICO = 8
   GEN_IMPRIMINDO_ETIQUETA = 9
   GEN_REGISTRANDO_MONTAGEM = 10
End Enum

'declara funcao sleep da API do Windows
Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)

'constantes para o XML de palete
Public Const TAG_PALETE = "palete"
Public Const TAG_PRODUTO = "produto"
Public Const TAG_IE_ORDEM = "ie_ordem"
Public Const TAG_NS_PALETE = "ns_palete"
Public Const TAG_DH_PRODUCAO = "dh_producao"
Public Const TAG_CD_LINHA = "cd_linha"

'constantes do produto no XML de palete
Public Const TAG_CD_PRODUTO = "cd_produto"
Public Const TAG_NS_PRODUTO = "ns_produto"
Public Const TAG_CD_COMPONENTE = "cd_componente"
Public Const TAG_NS_COMPONENTE = "ns_componente"
Public Const TAG_DH_MONTAGEM = "dh_montagem"
Public Const TAG_DH_PALETIZACAO = "dh_paletizacao"

'constantes para o XML de ordens do SAP
Public Const TAG_SAP_ORDENS = "Ordens"
Public Const TAG_SAP_ORDENS_QT_PLANEJADA = "qt_planejada"
Public Const TAG_SAP_ORDENS_QT_APONTADA = "qt_apontada"
Public Const TAG_SAP_ORDENS_CD_PRODUTO = "cd_produto"
Public Const TAG_SAP_ORDENS_HC_STATUS = "hc_status"
Public Const TAG_SAP_ORDENS_DH_ATUALIZACAO = "dh_atualizacao"
Public Const TAG_SAP_ORDENS_QT_PALETE = "qt_palete"
Public Const TAG_SAP_ORDENS_CD_COMPONENTE = "cd_componente"

Public Const TAG_SAP_ORDENS_DE_MODELO = "de_modelo"
Public Const TAG_SAP_ORDENS_DE_TENSAO_FREQUENCIA = "de_tensao_frequencia"
Public Const TAG_SAP_ORDENS_DE_POTENCIA = "de_potencia"
Public Const TAG_SAP_ORDENS_DE_CAPACIDADE_50LBP = "de_capacidade_50LBP"
Public Const TAG_SAP_ORDENS_DE_CAPACIDADE_50HBP = "de_capacidade_50HBP"
Public Const TAG_SAP_ORDENS_DE_CAPACIDADE_60LBP = "de_capacidade_60LBP"
Public Const TAG_SAP_ORDENS_DE_CAPACIDADE_60HBP = "de_capacidade_60HBP"
Public Const TAG_SAP_ORDENS_DE_REFRIGERANTE = "de_refrigerante"
Public Const TAG_SAP_ORDENS_DE_CORRENTE = "de_corrente"
Public Const TAG_SAP_ORDENS_DE_REFRIGERADOR_OLEO = "de_refrigerador_oleo"
Public Const TAG_SAP_ORDENS_DE_FASES = "de_fases"

'respostas do servidor WEB
Public Const HTTP_RESP_OK = "RESP_OK"
Public Const HTTP_RESP_ERROR = "RESP_ERROR"
```

Arquivos.bas

```
Attribute VB_Name = "Arquivos"
Option Explicit

'constantes

Public Const ARQUIVO_CONFIGURACAO = "config.xml"
Public Const ARQUIVO_FORMATO_ETIQUETA = "etiqueta.xsl"
Public Const ARQUIVO_HTML_INIT = "index.html"
Public Const ARQUIVO_HTML_RUN = "run.html"
Public Const ARQUIVO_MDB = "local.mdb"
Public Const TMP_ARQUIVO_MDB = "tmplocal.mdb"
Public Const CAMINHO_LOCAL = "c:\rastreabilidade\"

'constantes de tabelas

Public Const TAB_ORDENS = "ordens"
Public Const TAB_PALETES = "paletes"
```

```
Public Const TAB_PRODUTOS = "produtos"
```

mdlBD.bas

```
Attribute VB_Name = "mdlBD"
Option Explicit

Global Cnn As New ADODB.Connection
Public Function BDConectar(inPathBanco As String) As Long
'*********************************************************************
'Finalidade: Conecta ao banco de dados Access, com a conexao global
'Parametros: Caminho local do banco de dados
'Saida:      Sucesso ou código de erro
'Criacao:    GLF 2003-01-21
'*********************************************************************

Dim arquivoMdb As String

On Error GoTo ErrorHandler

    'verifica se existe arquivo
    arquivoMdb = Dir(inPathBanco)
    If arquivoMdb = "" Then
        BDConectar = CONEXAO_PATH_INVALIDO
    Else
        'conecta
        Cnn.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
            "Data Source=" & inPathBanco & ";"
        BDConectar = CONEXAO_OK
    End If

Exit Function

ErrorHandler:
    BDConectar = CONEXAO_ERRO_GERAL

End Function

Public Function BDCriar(inPathBanco As String) As Long
'*********************************************************************
'Finalidade: Cria banco de dados Access, no caminho especificado, sem tabelas,
'            com a conexao global
'Parametros: Caminho local do banco de dados
'Saida:      Sucesso ou código de erro
'Criacao:    GLF em 2003-02-03
'*********************************************************************
Dim catNewDB As New ADOX.Catalog 'variável com objeto catálogo

On Error GoTo ErrorHandler

    Set catNewDB = New ADOX.Catalog 'catNewDB recebe novo objeto

    'Catálogo cria um novo banco de dados com o caminho especificado
    catNewDB.Create "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & inPathBanco

    'objeto recebe valor nulo
    Set catNewDB = Nothing


Exit Function

ErrorHandler:
    BDCriar = CONEXAO_ERRO_GERAL

End Function
```

Lingua.bas

```
Attribute VB_Name = "Lingua"
```

```
Option Explicit

'Mensagens de erro genéricas
Public Const MSG_LEITURA_INVALIDA = "Leitura Inválida!"
Public Const MSG_LEITURA_INVALIDA_INICIAR_ORDEM = "Leitura inválida! Inicie Ordem."
Public Const MSG_OPERACAO_CANCELADA = "Operacao cancelada!"
Public Const MSG_TIMEOUT = "Tempo esgotado!"
Public Const MSG_PRODUTO_INEXISTENTE = "Produto inexistente!"
Public Const MSG_ERRO_GENERICO = "Ocorreu um erro!"
Public Const MSG_ERRO_CAMPOS_INCOMPLETOS = "Erro! Todos os campos devem ser preenchidos!"
Public Const MSG_ERRO_BANCO_PATH_INVALIDO = "Banco local inválido!"
Public Const MSG_ERRO_BANCO_GERAL = "Erro geral na conexao com o banco!"
Public Const MSG_ERRO_INTERNO = "Erro interno!"
Public Const MSG_ERRO_CARREGANDO_XML = "Erro carregando arquivo XML!"


'Mensagens Ordem
Public Const MSG_CONSULTANDO_ORDENS = "Consultando ordens SAP"
Public Const MSG_ORDEM_INICIADA = "Ordem iniciada"
Public Const MSG_ORDEM_ENCERRADA = "Ordem encerrada"
Public Const MSG_ORDEM_ERRO_INICIAR = "Erro ao iniciar ordem!"
Public Const MSG_ORDEM_ERRO_ENCERRAR = "Erro ao encerrar ordem!"
Public Const MSG_ORDEM_CONFIRMA_ENCERRAMENTO = "Confirma encerramento da ordem?"
Public Const MSG_ORDEM_JA_EXISTENTE = "A nova ordem nao pôde ser inserida. Ela já existe na lista de
ordens!"
Public Const MSG_INSERCAO_ORDEM_ERRO = "Erro inserindo ordem no banco!"

'Mensagens Palete
Public Const MSG_PALETE_ABERTO = "Palete aberto"
Public Const MSG_PALETE_FECHADO = "Palete fechado"
Public Const MSG_PALETE_ERRO_ABRIR = "Erro ao abrir palete!"
Public Const MSG_PALETE_ERRO_FECHAR = "Erro ao fechar palete!"
Public Const MSG_PALETE_ERRO_FECHAR_AUTOMATICAMENTE = "Erro ao fechar palete automaticamente"
Public Const MSG_PALETE_DESMONTADO = "Palete desmontado!"
Public Const MSG_PALETE_ERRO_DESMONTAR = "Erro ao desmontar palete!"
Public Const MSG_PALETE_ERRO_ABRIR_BANCO = "Erro ao abrir dados do palete no banco!"
Public Const MSG_PALETE_ERRO_FECHAR_BANCO = "Erro ao tentar fechar o palete no banco!"
Public Const MSG_PALETE_ERRO_DESMONTAR_BANCO = "Erro ao desmontar palete no banco!"

Public Const MSG_PRODUTO_REMOVIDO = "Produto removido!"
Public Const MSG_LEITURA_INVALIDA_ABRIR_PALETE = "Leitura invalida! Abra palete."
Public Const MSG_PALETE_ERRO_INICIANDO_ORDEM = "Erro iniciando ordem!"

'Msgs paletizacao
Public Const MSG_PALETIZACAO_LEIA_PALETE = "Leia palete"
Public Const MSG_PALETIZACAO_SEM_ORDEM_INICIADA = "Sem ordem iniciada"
Public Const MSG_PALETIZACAO_PALETE_EM_ORDEM_INCORRETA = "Erro! Palete associado à ordem: "
Public Const MSG_PALETIZACAO_PALETE_JA_TRANSFERIDO = "Erro! Palete ja fechado e transferido!"
Public Const MSG_PALETIZACAO_ERRO_CONFIRMACAO_PALETE = "Erro! Palete lido nao confere!"
Public Const MSG_PALETIZACAO_PRODUTO_EM_OUTRO_PALETE = "Erro! Produto não inserido neste palete!"
Public Const MSG_PALETIZACAO_PRODUTO_INSERIDO = "Produto inserido no palete!"
Public Const MSG_PALETIZACAO_PRODUTO_NAO_MONTADO = "Produto não montado!"
Public Const MSG_PALETIZACAO_PRODUTO_SEM_COMPONENTE = "Erro! Produto sem componente!"
Public Const MSG_PALETIZACAO_PRODUTO_MONTADO_INCORRETAMENTE = "Erro! Produto montado incorretamente!"
Public Const MSG_PALETIZACAO_PRODUTO_JA_INSERIDO = "Produto já inserido neste palete!"
Public Const MSG_PALETIZACAO_PRODUTO_JA_INSERIDO_OUTRO_PALETE = "Produto já inserido em outro palete!"
Public Const MSG_PALETIZACAO_PRODUTO_INCORRETO = "Produto incorreto para esta ordem!"
Public Const MSG_PALETIZACAO_ERRO_PALETE_CHEIO = "Erro! Palete está cheio!"
Public Const MSG_PALETIZACAO_ERRO_PRODUTO_INSERIR_ERRO = "Erro ao inserir produto!"
Public Const MSG_PALETIZACAO_CONFIRMA_DESMONTAGEM = "Confirma desmontagem do palete?"
Public Const MSG_PALETIZACAO_ERRO_INICIANDO_MAQUINA_ESTADOS = "Erro fatal ao inicializar valores do
banco!"
Public Const MSG_PALETIZACAO_ERRO_PRODUTO_REMOVER_GENERICO = "Erro ao remover produto!"
Public Const MSG_PALETIZACAO_ERRO_PRODUTO_INSERIR_BANCO = "Erro ao tentar inserir produto no banco!"
Public Const MSG_PALETIZACAO_ERRO_PRODUTO_REMOVER_BANCO = "Erro ao tentar remover o produto do banco!"
Public Const MSG_PALETIZACAO_FECHAMENTO_AUTOMATICO = "Palete completo. Fechando automaticamente..."
Public Const MSG_PALETIZACAO_ERRO_INICIALIZACAO_LEITOR = "Erro ao inicializar leitor de codigo de barras!"
Public Const MSG_PALETIZACAO_ERRO_INICIALIZACAO_DISPLAY = "Erro ao inicializar display!"
Public Const MSG_PALETIZACAO_ERRO_INICIALIZACAO_IMPRESSORA = "Erro ao inicializar impressora!"

'Mensagens Genealogia
Public Const MSG_GENEALOGIA_ERRO = "Erro!"
Public Const MSG_GENEALOGIA_LEIA_COMPONENTE = "Leia componente"
Public Const MSG_GENEALOGIA_MONTAGEM_INCORRETA = "Montagem incorreta!"
```

```
Public Const MSG_GENEALOGIA_ORDEM_INEXISTENTE = "Ordem SAP para este produto não encontrada! " 'Reconsulte
as ordens!"
Public Const MSG_GENEALOGIA_COMPONENTE_JA_MONTADO = "Erro! Componente já montado em outro produto!"
Public Const MSG_GENEALOGIA_PRODUTO_JA_MONTADO = "Erro! Produto já montado com outro componente!"
Public Const MSG_GENEALOGIA_SOLICITACAO_DESMONTAGEM = "O produto anterior será desmontado! Confirma?"
Public Const MSG_GENEALOGIA_PRODUTO_MONTADO = "Produto montado!"
Public Const MSG_GENEALOGIA_PRODUTO_DESMONTADO = "Produto desmontado!"
Public Const MSG_GENEALOGIA_AGORA_LEIA_COMPONENTE = "Produto lido! Agora, leia componente!"

'Mensagens Display
Public Const MSG_DISPLAY_ERRO = "Erro!"
Public Const MSG_DISPLAY_LEIA_COMPONENTE = "Leia componente"
Public Const MSG_DISPLAY_LEIA_PRODUTO = "Leia produto"
Public Const MSG_DISPLAY_MONTAGEM_INCORRETA = "Montagem Incorreta"
Public Const MSG_DISPLAY_ORDEM_INEXISTENTE = "Sem ordem pra prod."
Public Const MSG_DISPLAY_COMPONENTE_JA_MONTADO = "Comp. ja montado"
Public Const MSG_DISPLAY_PRODUTO_JA_MONTADO = "Produto ja montado!"
Public Const MSG_DISPLAY_SOLICITACAO_DESMONTAGEM = "Desmontar anterior?"
Public Const MSG_DISPLAY_PRODUTO_MONTADO = "Produto montado!"
Public Const MSG_DISPLAY_PRODUTO_DESMONTADO = "Produto desmontado!"
Public Const MSG_DISPLAY_AGORA_LEIA_COMPONENTE = "Prod OK. Leia comp."
Public Const MSG_DISPLAY_OPERACAO_CANCELADA = "Operacao cancelada!"
Public Const MSG_DISPLAY_PRODUTO_INEXISTENTE = "Produto inexistente!"
Public Const MSG_DISPLAY_LEITURA_INVALIDA = "Leitura invalida!"
Public Const MSG_DISPLAY_TIMEOUT = "Tempo esgotado!"

'Mensagens Genericas
Public Const MSG_LEIA_PRODUTO = "Leia produto"

'Caption janelas
Public Const CAP_ORDENS_SEM_CONEXAO = "Ordens (Sem conexão SAP)"
Public Const CAP_ORDENS = "Ordens"

'MENSAGENS DE ERRO DO MÓDULO DE TRANSFERÊNCIA

Public Const AVISO_ERRO_REMOTO = " erro(s) encontrados no banco de dados remoto!"
Public Const AVISO_ERRO_LOCAL = " erro(s) encontrados no banco de dados local!"
Public Const FALHA_COMUNICACAO_REMOTA = "Falha na comunicação remota!"
Public Const FALHA_BANCO_LOCAL = "Falha no acesso ao banco local!"
Public Const ERRO_PATH_INVALIDO = "Arquivo não encontrado"
Public Const ERRO_INTERNO_SERVIDOR = "Erro interno no servidor!"

Public Const ERRO_ENVIO_XML = "Falha ao enviar palete ao servidor"
Public Const ERRO_MONTAGEM_XML = "Falha ao preparar dados para envio"
Public Const ERRO_TRANSFERENCIA = "Falha na transferência de dados"
Public Const ERRO_INIT_XML = "Falha na configuração do módulo"

Public Const MSG_OK_BANCO_LOCAL = "Conectado ao banco de dados"


'MENSAGENS DE ERRO DO MÓDULO INIT

Public Const MSG_CRIANDO_BANCO = "Criando banco de dados local..."
Public Const MSG_PREPARANDO_BANCO = "Preparando banco de dados local..."
Public Const MSG_ARQUIVO_COPIANDO = "Copiando arquivos"
Public Const MSG_ARQUIVO_COPIADO = "Arquivos copiados com sucesso!"

Public Const ERRO_BANCO_LOCAL = "Erro geral no banco de dados local!"
Public Const ERRO_ARQUIVO_COPIAR = "Erro ao copiar arquivos localmente"
```

APÊNDICE II – EXEMPLO DE ARQUIVO DE CONFIGURAÇÃO XML

```
<GERAL>
   <LINHA>
      <NOME>LINHA 1</NOME>
      <CODIGO>U</CODIGO>
      <CENTRO_SAP>ITA</CENTRO_SAP>
   </LINHA>
   <IMPRESSORA>
      <PORTA_SERIAL>4</PORTA_SERIAL>
      <BAUD_RATE>9600</BAUD_RATE>
      <DATA_BITS>8</DATA_BITS>
      <STOP_BITS>1</STOP_BITS>
      <PARITY>n</PARITY>
   </IMPRESSORA>
   <PALETIZACAO>
      <TEMPO_TIMEOUT>5000</TEMPO_TIMEOUT>
      <LEITOR>
         <PORTA_SERIAL>1</PORTA_SERIAL>
         <BAUD_RATE>9600</BAUD_RATE>
         <DATA_BITS>8</DATA_BITS>
         <STOP_BITS>1</STOP_BITS>
         <PARITY>n</PARITY>
         <CARACTER_INICIO>2</CARACTER_INICIO>
         <CARACTER_FIM>3</CARACTER_FIM>
         <ALFA_NUMERICO>0</ALFA_NUMERICO>
      </LEITOR>
   </PALETIZACAO>
   <GENEALOGIA>
      <TEMPO_TIMEOUT>5000</TEMPO_TIMEOUT>
      <LEITOR>
         <PORTA_SERIAL>2</PORTA_SERIAL>
         <BAUD_RATE>9600</BAUD_RATE>
         <DATA_BITS>8</DATA_BITS>
         <STOP_BITS>1</STOP_BITS>
         <PARITY>n</PARITY>
         <CARACTER_INICIO>2</CARACTER_INICIO>
         <CARACTER_FIM>3</CARACTER_FIM>
         <ALFA_NUMERICO>0</ALFA_NUMERICO>
      </LEITOR>
      <DISPLAY>
         <PORTA_SERIAL>3</PORTA_SERIAL>
         <BAUD_RATE>9600</BAUD_RATE>
         <DATA_BITS>8</DATA_BITS>
         <STOP_BITS>1</STOP_BITS>
         <PARITY>n</PARITY>
      </DISPLAY>
   </GENEALOGIA>
   <TRANSFERENCIA>
      <TEMPO_ESPERA_TRANSFERENCIA>10</TEMPO_ESPERA_TRANSFERENCIA>
      <PRAZO_PALETE>2</PRAZO_PALETE>
   </TRANSFERENCIA>
   <CODIGOBARRAS>
      <CONFIG LEN="17">
         <SEPARADOR>$</SEPARADOR>
         <CD_POS>1</CD_POS>
         <CD_LEN>1</CD_LEN>
         <NS_POS>2</NS_POS>
         <NS_LEN>1</NS_LEN>
      </CONFIG>
      <CONFIG LEN="17">
         <CD_POS>1</CD_POS>
         <CD_LEN>9</CD_LEN>
         <NS_POS>10</NS_POS>
         <NS_LEN>8</NS_LEN>
      </CONFIG>
   </CODIGOBARRAS>
</GERAL>
```