

SISTEMA AUTOMATIZADO PARA GERENCIAMENTO DE ESTOQUE

Renata Crescini Ramos¹, Luiz Carlos Calderan²..... Valdir Gil Pillatⁿ

Univap/FEAU, Av. Shishima Hifumi, 2911, renata.ramos@gm.com
Univap/FEAU, Av. Shishima Hifumi, 2911, luizcalderan@yahoo.com.br

Resumo- O ritmo imposto pela evolução tecnológica impõe a criação de sistemas de gerenciamento e controle de fácil instalação, manutenção e baixo custo. Este trabalho apresenta um modelo simples de controle de estoque envolvendo sistema de rádio frequência (coletor de dados), programação em alto nível e banco de dados. Tem como objetivo projetar e desenvolver uma ferramenta que automatize o processo e o gerenciamento das atividades executadas em uma área de trabalho específica dentro de uma grande empresa, tendo como principal critério à redução dos excessos assim como eliminar a incidência de falta de embalagem no estoque.

Palavras-chave: Inventário, Gerenciamento de Estoque, Material de Processo

Área do Conhecimento: Java, VBA, SAP, Banco de Dados

Introdução

Um dos integrantes da equipe trabalha em uma Multinacional no setor automobilístico cujos produtos são exportados á diversos países. Esta empresa já opera com sistema automatizado em quase todos os seus processos, porém no setor de Estoque de material não produtivo os sistemas operam de forma independente e são eles: sistema SAP, Sistema WES e Sistema MGO.

Todos os controles são executados manualmente, pedidos de compras são realizados com base em médias de consumo baseado em contagem física diárias e mantidas em planilhas Excel, em decorrência disso não são levados em considerações variações de volume de produção e como conseqüência ocorre falta ou sobra de embalagens no estoque.

As peças exportadas pela empresa são embaladas e enviadas ao cliente em embalagens como caixas de madeira, caixas metálicas, caixas de papelão, plástico bolha, plástico VCI, polionda, etc.

O estoque de embalagem tem um consumo médio de R\$ 2.200.000,00 mensais e um inventário físico de segurança de R\$ 900.000,00 que se faz necessário devido à falta de um controle eficiente e um planejamento eficaz.

Esse projeto tem o objetivo de desenvolver e implantar um sistema de baixo custo e fácil manutenção e que atue especificamente na comunicação entre os sistemas já existentes para gerar uma programação de compra precisa considerando uma pequena porcentagem de perdas e variações com o intuito de reduzir o inventário físico de segurança.

O projeto já foi iniciado no Estoque de Caixas de Madeira onde obtivemos uma média de R\$ 200.000,00 nos últimos três meses.

Metodologia

Devido à diversidade de sistemas operando na empresa e as diversas interfaces entre eles há muita lentidão nos processos integrados, pois em qualquer sistema quanto menos interfaces melhor. Atualmente não é viável para a empresa o início de um novo projeto para a redução das interfaces.

Cada sistema é responsável por uma atividade, a programação da produção é feita através do Sistema WES, a programação de compra de peças e controle de estoque é feito pelo Sistema MGO, as peças embaladas que já foram retiradas do estoque não tem baixa simultânea, as baixas acontecem na virada do dia, e as embalagens utilizadas para embalar as peças tem estoque controlado manualmente, a entrada da mercadoria é feita através do sistema SAP, porém não há uma inteligência entre os Sistemas MGO e o sistema SAP que realize a baixa das embalagens consumidas para embalar as peças.

Há problemas de unidades de medida, exemplo plástico VCI comprado em Kg e deverá ser baixado no estoque em m², caixa de papelão comprado em Kg deverá ser baixado em unidade etc...;

Uma mesma peça poderá consumir mais de um tipo de embalagem, como por exemplo: plástico bolha + espuma + caixa de papelão (conforme trabalho padronizado), etc..., e um mesmo material de processo poderá ser usado por diferentes peças e em diferentes quantidades de acordo com o pedido que está sendo atendido;

A coleta de peças já embaladas é feita via coletores de dados, banco de dados e uma aplicação desenvolvida em linguagem de alto nível.

O banco de dados deverá receber os dados do Sistema WES (programação de produção), Sistema MGO (Baixa material) e Sistema SAP

(Estoque de Embalagem) e outras tabelas de conversão.

O desenvolvimento do projeto seguirá as etapas conforme descrito a baixo:

✓ Especificação

A fase de especificação é constituída pelo Modelo Contextual, Modelo de Objetivos, Modelo de Titulação e Modelo de Requisitos.

✓ Desenvolvimento

O desenvolvimento será constituído por:

- Fase de análise: constituída por Modelo funcional do projeto e Modelo conceitual do projeto
- Fase de Projeto de Sistema: é o modelo Arquitetural do projeto como um todo.

✓ Implementação

A implementação será composta por:

- Fase de codificação – desenvolvimento do modelo físico do projeto
- Fase de Testes – Execução de testes unitários e correção de possíveis erros.

Nesse projeto estamos trabalhando com o Interbase® que é um poderoso banco de dados Cliente/Servidor relacional que é compatível com SQL-ANSI-92, e foi desenvolvido para ser um banco de dados independentes de plataformas e de sistemas operacionais. Este banco de dados dispensa maiores estruturas dentro da empresa, (DBA / Preparação), onde basta instalar o software e usá-lo, sem a interferência freqüente de profissionais, especializados na manutenção do banco de dados de produção.

Acompanhando isso tudo ele ainda dispensa o uso de superservidores, usando pouco espaço em disco para sua instalação e utilizando pouca memória em situações normais de uso. Por isso a plataforma necessária para a sua instalação e utilização pode ser reduzida diminuindo consideravelmente os custos do projeto.

Seu desenvolvimento iniciou em meados de 1985 por uma equipe de engenheiros da DEC (Digital Equipment Corporation). Tendo como nome inicial de Groton, esse produto veio sofrendo varias alterações até finalmente em 1986 receber o nome de Interbase® iniciando na versão 2.0. Nesta época, a idéia era produzir um SGBDR (Sistema Gerenciador de Bancos de Dados Relacionais) que oferecesse benefícios não encontrados em outros da época.

A interface com o usuário será desenvolvida em Java, pois uma interface gráfica com usuários em Java envolve tipicamente a criação de um container, um componente que pode receber outros. Em uma aplicação gráfica autônoma. Após sua criação, os containers da aplicação recebem componentes de interface com usuários, que permitirão apresentar e receber dados dos usuários. Embora seja possível posicionar esses componentes através de coordenadas absolutas em cada container, é recomendável que o projetista utilize sempre um gerenciador de layout

para realizar essa tarefa. Deste modo, garante-se que a aplicação possa ser executada independentemente das características da plataforma onde será executada.

Finalmente, é preciso especificar quais devem ser os efeitos das ações dos usuários — tais como um "clique" do mouse ou uma entrada de texto — quando realizadas sobre cada um desses componentes.

Isto se dá através da especificação de classes manipuladoras de eventos, projetadas para a aplicação. Ao associar objetos dessas classes aos componentes gráficos, o projetista determina o comportamento da aplicação.

A linguagem Java oferece, dentre as funcionalidades incorporadas à sua API padrão, um extenso conjunto de classes e interfaces para o desenvolvimento de aplicações gráficas. Esse conjunto facilita a criação de saídas na forma gráfica e de interfaces gráficas com usuários (GUIs).

As aplicações gráficas são criadas através da utilização de componentes gráficos, que estão agrupados em dois grandes pacotes: java.awt e javax.swing.

AWT é o Abstract Windowing Toolkit, sendo definido através das classes do pacote java.awt e seus subpacotes, tais como java.awt.event. Essas classes agrupam as funcionalidades gráficas que estão presentes desde a primeira versão de Java, que operam tendo por base as funcionalidades de alguma biblioteca gráfica do sistema onde a aplicação é executada.

Já o pacote javax.swing define uma extensão padronizada a partir de AWT que congrega componentes gráficos que utilizam exclusivamente Java (lightweight components), com funcionalidades e aparência independentes do sistema onde a aplicação é executada.

A API JDBC é um conjunto de classes e interfaces que prevê um padrão para tornar possível a construção de softwares que acesse banco de dados e utilize toda a API Java. Por meio dessa API, torna-se possível o acesso, genérico e uniforme a qualquer SGBDR, sendo que os dados são efetivamente acessados por meio do SQL. Desenvolveremos uma interface com os elementos da API JDBC para conectar-se ao banco de dados, com o auxílio do SQL, deverão se desenvolvidas as operações de consulta, de inclusão, de remoção ou de alteração de dados.

Resultados

Foram realizados testes e análise para adotarmos uma linguagem de programação na qual será desenvolvidos a interface com o usuário. Nos testes e análises realizados podemos dizer que VB é utilizado principalmente para aplicações

desktop, o VB é rápido, aparentemente até mais que o Java, cujas interfaces gráficas, independentes de plataforma é um pouco lenta. Podemos dizer também que se pode trabalhar com VB no servidor, trabalhando, por exemplo, com componentes (dlls) COM+. Nesse ponto, o VB já não é tão rápido e pouco confiável (Talvez pela própria plataforma COM, que prende objetos na memória). No caso do Java no servidor, seja com Servlet, EJB, CORBA, RMI e outros mais, podemos afirmar que têm umas plataformas bastante robustas, confiáveis e muito mais rápido que o VB.

O desenvolvimento em VB é extremamente rápido, enquanto que no Java o processo pode ser um pouco mais lento (razoavelmente mais trabalhoso escrever um código Java do que em Basic.), porém, como o java é 100% orientado a objetos, a reutilização é um ponto chave. Depois que se desenvolveu sua primeira aplicação, caso esta tenha sido bem desenhada, podemos reaproveitar muito do que foi feito em projetos futuros. Assim, depois de algum tempo programando em java, provavelmente não terá mais que programar tanto, pois reutilizará muito do código. Isso já não é possível com VB. Uns podem dizer que pode fazer reutilização copiando os códigos do que já foi feito e colando nas aplicações novas, ou reutilizando Módulos. Até pode ser, porém qualquer alteração no código deverá ser feita na mão em todos os lugares onde foi utilizada uma cópia. Além disso, com o conceito de herança em Java, podemos ter um código mais genérico e ir modificando-o, com base na herança, visando ajustar o código a algo mais específico, sem alterar seu código Pai e assim não afetando nenhum dos filhos que utilizam esse código. Além disso, qualquer alteração no código pai reflete em todo mundo que o utiliza. Esse conceito o VB não implementa.

No geral, Java (desde que não se utilize APIs Proprietárias dependentes de plataforma) é Multiplataforma (Write once run anywhere) e o VB não tem esta funcionalidade. Java é gratuito, e pode-se montar toda uma plataforma para aplicação, como por exemplo, Linux + Apache + Tomcat + JBOSS + MySQL +Velocity + Struts, utilizando softwares também gratuitos .

VB precisa de Windows, que pode precisar de MTS, que pode precisar de SQL Server, e isso gera um custo beneficio não viável para nosso projeto. Nesse ponto pode-se dizer que o suporte dos "Gratuitos" é caríssimo, não valendo a pena. Pode até ser uma verdade que o suporte seja caro, porém o material que se encontra na internet sobre qualquer um destes é suficiente para não se precisar do suporte.

- CAVALCANTI, J. C. C. et all. Visual Basic 2005: Teoria e Aplicações Práticas. São Paulo: Érica. 1ª edição, 2007.
- MACHADO, F. N. R. Banco de Dados – Projeto e Implementação. São Paulo: Érica. 1ª edição, 2004.
- BERTAGLIA, P. R. Logística e Gerenciamento. São Paulo: Saraiva. 1ª edição, 2003.
- LIMA, A. S. Visual Basic 6.0. São Paulo: Érica. 1ª edição, 1999.
- www.sap.com acessado em 25/03/2008.

Referências