



ENGINE
LOGÍSTICA



IME-USP

MAC 499 - TRABALHO DE FORMATURA SUPERVISIONADO

Implantação do WMS - Warehouse Management System

Aluno: Paulo Roberto de Rodrigues e Reigadas

Nº USP: 3298082

Supervisora: Professora Ana Cristina Vieira de Melo

Tipo de Trabalho: Estágio

Empresa: Engine Logística Ltda.

AGRADECIMENTOS

Sou muito grato a todos os funcionários da Engine Logística Ltda por toda ajuda e colaboração que tive e por me propiciarem um ambiente muito agradável de trabalho. Queria agradecer também a professora Ana Cristina que me supervisionou neste trabalho, sempre com muita atenção e bom humor.

ÍNDICE

1 – INTRODUÇÃO	01
2 - NATUREZA DA ORGANIZAÇÃO	01
3 - PARTE TÉCNICA	
3.1 - O PROBLEMA ENCONTRADO	02
3.2 - A SOLUÇÃO DESENVOLVIDA	02
3.2.1 – ARQUITETURA DO SISTEMA DESENVOLVIDO	03
3.3 - CRONOGRAMA	04
3.4 – EQUIPE DE TRABALHO	05
3.5 - DESENVOLVIMENTO DO SISTEMA	05
3.5.1 – PARADIGMAS DA ENGENHARIA DE SOFTWARE	05
3.5.1.1 – MODELO CASCATA	06
3.5.1.2 – PROTOTIPAÇÃO	07
3.5.1.3 – MODELO ESPIRAL	09
3.5.1.4 – TÉCNICAS DE QUARTA GERAÇÃO.	10
3.5.2 –MODELO UTILIZADO PARA O DESENVOLVIMENTO	12
3.5.3 –FERRAMENTAS UTILIZADAS	13
3.6 - PROBLEMAS NO PROJETO.	13
3.6.1 – ALTERAÇÃO NO ESCOPO DO PROJETO	13
3.6.2 – FALTA DE PLANEJAMENTO DE MANUTENÇÃO	14
3.6.3 – FALTA DE GARANTIA E DE CONTRATO DE MANUTENÇÃO	14
3.7 - O PROGRAMA.	15
3.7.1 - TELA DE LOGIN	15
3.7.2 - CONFIGURAÇÃO DO SISTEMA.	16
3.7.2.1 - CADASTRO DE COLETORES	16
3.7.2.2 - CADASTRO DE TIMERS	17
3.7.2.3 - CADASTRO DE TOLERÂNCIA	18
3.7.2.4 - CADASTRO DO PATH PARA A IMPORTAÇÃO	19
3.7.3 - CADASTRO DE PERMISSÕES	20
3.7.4 – INVENTÁRIO	21
3.7.5 - ROMANEIO	23
3.10.5.1 - GERENCIAMENTO DOS ROMANEIOS	25
3.7.6 - PROCESSO / CARTA PROGRAMA	26
3.7.7 - GERENCIAMENTO DE USUÁRIOS	27
3.7.8 - RELATÓRIOS GERENCIAIS	28
3.7.8.1 - LOG DE USUÁRIO	28
3.7.8.2 - RELATÓRIO DE DESEMPENHO	29
3.7.8.3 - GRÁFICOS COMPARATIVOS.	30
3.8 – RESULTADOS OBTIDOS	31
3.9 - MINHA PARTICIPAÇÃO NO PROJETO.	31
3.9.1 – TREINAMENTO	31

4 - PARTE SUBJETIVA	32
4.1 - EXPERIÊNCIA PESSOAL	32
4.2 - DESAFIOS E FRUSTRAÇÕES ENCONTRADOS	32
4.3 - DISCIPLINAS CURSADAS NO BCC MAIS RELEVANTES	33
4.4 – MEMBROS DA EQUIPE.	34
4.5 – CONSIDERAÇÕES FINAIS	34
5 – BIBLIOGRAFIA	35

1 INTRODUÇÃO

Esta monografia é parte da disciplina MAC 499 – TRABALHO DE FORMATURA SUPERVISIONADO e tem como objetivo descrever o projeto de implantação de uma solução WMS - *Warehouse Management System*, com ênfase na metodologia de desenvolvimento utilizada.

2 NATUREZA DA ORGANIZAÇÃO

Engine Logística Ltda.

A Engine atua desde 1992 no desenvolvimento e implantação de soluções corporativas em tecnologia de informação.

Entre 1992 e 1999 foi, no Brasil, a representante oficial da Borland Latin America, líder na elaboração de ferramentas para desenvolvedores (Delphi, Kylix, JBuilder, C++ Builder, IntraBuilder). Em 1999 foi parcialmente incorporada pela Borland, criando sua subsidiária no Brasil.

Em 1996, a Engine iniciou suas atividades no mercado logístico, com o objetivo de fornecer soluções de ponta em tecnologia da informação para processos logísticos, tendo crescido exponencialmente desde então e colecionado uma excelente carteira de clientes e parceiros, viabilizando a criação da Engine Logística Ltda.

A Engine Logística é uma empresa dinâmica, sempre atenta as necessidades do mercado, prestando serviços de avaliação, modelagem de processos e sistemas, implantação, treinamento e manutenção de soluções, buscando sempre a plena satisfação das necessidades e urgências de seus clientes.

Possui uma equipe técnica formada por consultores, analistas e programadores em constante treinamento no exterior, objetivando oferecer aos seus clientes o que há de mais avançado em tecnologias de processos e sistemas.

3 PARTE TÉCNICA

3.1 O PROBLEMA ENCONTRADO

A Millenium é uma empresa que atua no segmento têxtil e trabalha com compra/venda de diversas bobinas de tecidos nacionais e importados. Quando uma venda era realizada, o sistema de gestão da empresa, da Microsiga, gerava uma NFS – Nota Fiscal de Saída, que, posteriormente, era enviada para o armazém da empresa onde os produtos solicitados eram separados.

Para cada produto separado, os funcionários anotavam o código de barras em um formulário, que posteriormente era enviado para o setor de vendas e manualmente eram feitos os acertos dos valores e das quantidades da NFS.

A expedição dos produtos levava muito tempo para ser feita e de uma forma totalmente manual, gerando uma taxa alta de erros para uma empresa. A Engine Logística foi contratada por essa empresa para automatizar a sua operação de expedição.

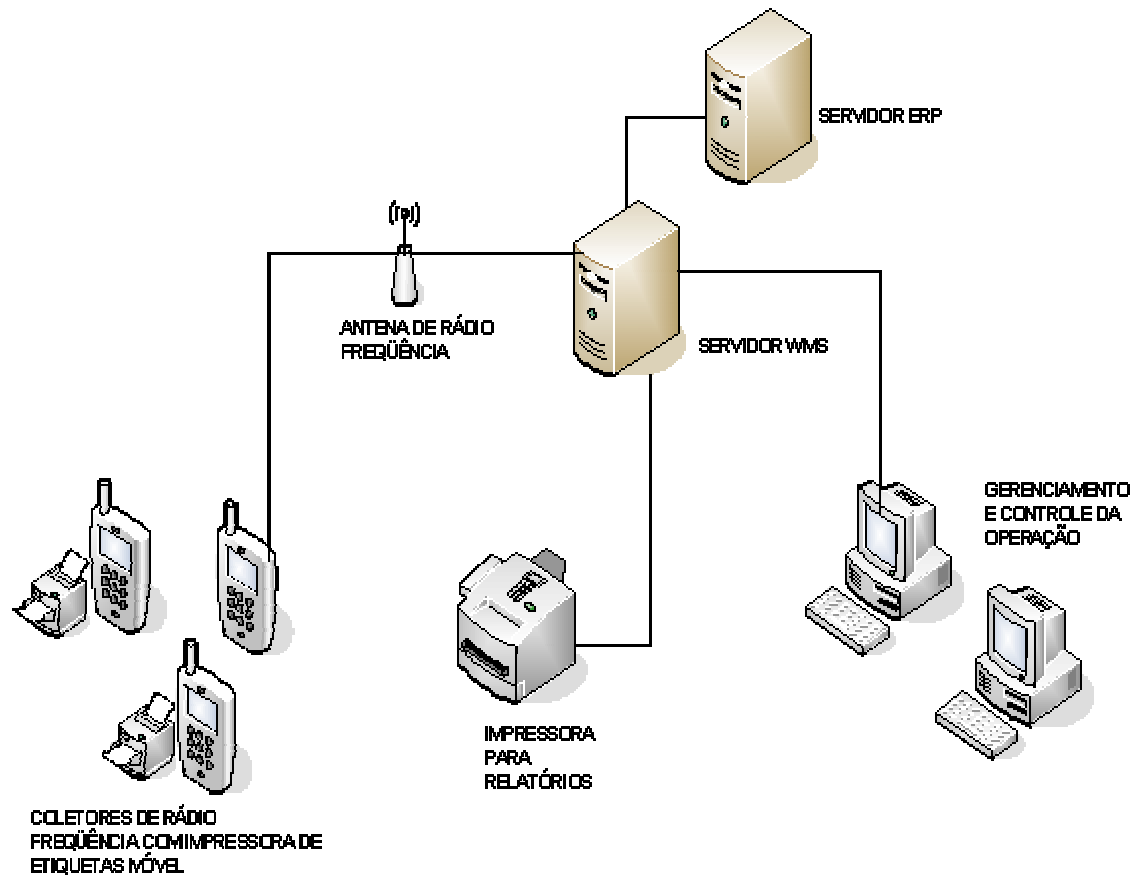
3.2 A SOLUÇÃO DESENVOLVIDA

Para solucionar o problema descrito acima, a Engine Logística desenvolveu um sistema de controle centralizado, integrado com o sistema de gestão da empresa e utilizando coletores de dados com rádio frequência para a leitura dos produtos.

As principais características do sistema desenvolvido são:

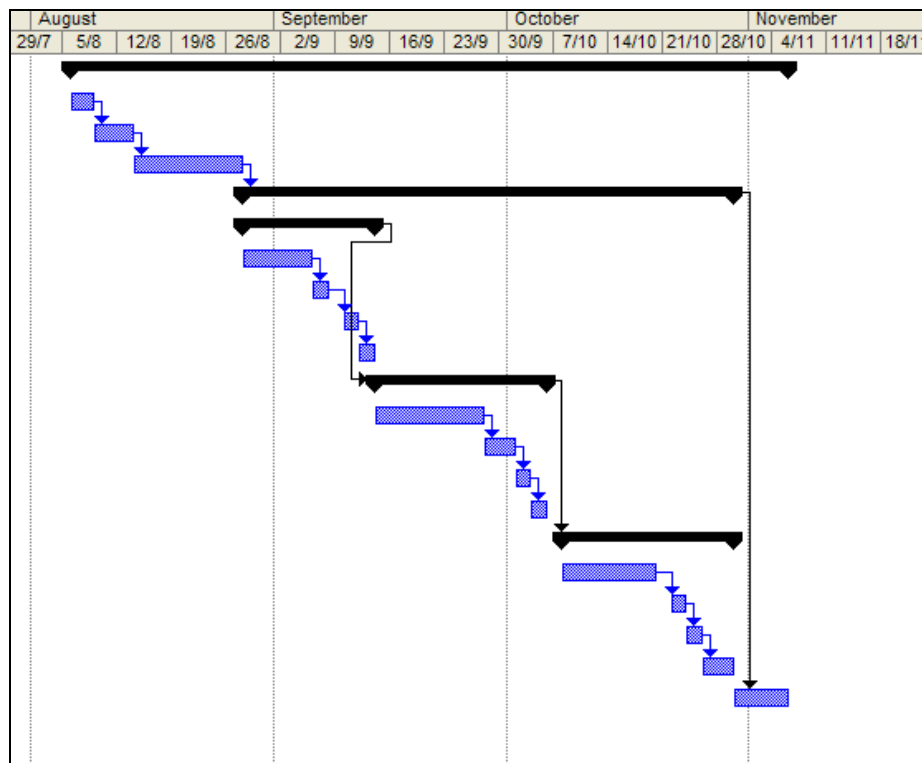
- ↪ configurações gerais de usuários e de parâmetros para o sistema;
- ↪ integração com o software ERP (Enterprise Resource Planning) da empresa;
- ↪ aplicação do coletor (rotinas e interface);
- ↪ controle/consulta das listas de separação;
- ↪ inventário dos produtos;
- ↪ monitorização on-line dos coletores de dados;
- ↪ relatórios gerenciais.

3.2.1 ARQUITETURA DO SISTEMA DESENVOLVIDO



3.3 CRONOGRAMA DO PROJETO

	f Task Name	Duration	Start	Finish	Predecessors
1	Implantação do WMS	66 days	6 Aug '01	5 Nov '01	
2	Entendimento das necessidades	3 days	6 Aug '01	8 Aug '01	
3	Análise da Situação Atual	3 days	9 Aug '01	13 Aug '01	2
4	Levantamento e Especificação da Solução	10 days	14 Aug '01	27 Aug '01	3
5	Implantação em Fases	45 days	28 Aug '01	29 Oct '01	4
6	Fase 1 - Inventário	13 days	28 Aug '01	13 Sep '01	
7	Customização	7 days	28 Aug '01	5 Sep '01	
8	Teste	2 days	6 Sep '01	7 Sep '01	7
9	Instalação Servidor e clientes	2 days	10 Sep '01	11 Sep '01	8
10	Treinamento	2 days	12 Sep '01	13 Sep '01	9
11	Fase 2 - Romaneio	16 days	14 Sep '01	5 Oct '01	6
12	Customização	10 days	14 Sep '01	27 Sep '01	
13	Teste	2 days	28 Sep '01	1 Oct '01	12
14	Instalação Servidor e clientes	2 days	2 Oct '01	3 Oct '01	13
15	Treinamento	2 days	4 Oct '01	5 Oct '01	14
16	Fase 3 - Processo / Carta Programa	16 days	8 Oct '01	29 Oct '01	11
17	Customização	10 days	8 Oct '01	19 Oct '01	
18	Teste	2 days	22 Oct '01	23 Oct '01	17
19	Instalação Servidor e clientes	2 days	24 Oct '01	25 Oct '01	18
20	Treinamento	2 days	26 Oct '01	29 Oct '01	19
21	Acompanhamento	5 days	30 Oct '01	5 Nov '01	5



3.4 EQUIPE DE TRABALHO

A equipe de trabalho da Engine envolvida nesse projeto era formada por 4 pessoas, conforme descrito abaixo:



3.5 DESENVOLVIMENTO DO SISTEMA

O custo relacionado ao software se tornou o principal item no orçamento da computação, sendo assim, o seu método de desenvolvimento tem cada vez mais importância para as empresas continuarem competitivas no mercado. Através do estágio pude acompanhar na prática todas as etapas de desenvolvimento de um sistema e, depois, pude verificar na teoria, através do curso de MAC0332 - Engenharia de Software, as técnicas e modelos de desenvolvimento existentes.

3.5.1 PARADIGMAS DA ENGENHARIA DE SOFTWARE

A engenharia de software é um rebento da engenharia de sistemas e de hardware. Ela abrange um conjunto de três elementos fundamentais – métodos, ferramentas e procedimentos – que possibilita ao gerente o controle do processo de desenvolvimento do software e oferece ao profissional uma base para construção de software de alta qualidade produtivamente.

- ↪ Os métodos de engenharia de software proporcionam os detalhes de “como fazer” para construir o software. Os métodos envolvem um amplo conjunto de tarefas que incluem: planejamento e estimativa de projeto, análise de requisitos de software e de sistemas, projeto da estrutura de dados, arquitetura de programa e algoritmo de processamento, codificação, teste e manutenção. Os métodos da engenharia de software muitas vezes introduzem uma notação gráfica ou orientada à linguagem especial e introduzem um conjunto de critérios para a qualidade do software.
- ↪ As ferramentas de engenharia de software proporcionam apoio automatizado ou semi-automatizado aos métodos. Atualmente, existem ferramentas para sustentar cada um dos métodos anotados anteriormente. Quando as ferramentas são

integradas de forma que a informação criada por uma ferramenta possa ser usada por outra, é estabelecido um sistema de suporte ao desenvolvimento software chamado engenharia de software auxiliada por computador (CASE – Computer-Aided Software Engineering).

- ↪ Os procedimentos da engenharia de software constituem o elo de ligação que mantém juntos os métodos e as ferramentas e possibilita o desenvolvimento racional e oportuno do software de computador. Os procedimentos definem a seqüência em que os métodos serão aplicados, os produtos (deliverables) que se exige que sejam entregues (documentos, relatórios, formulários, etc), os controles que ajudam a assegurar a qualidade e a coordenar as mudanças, e os marcos de referência que possibilitam aos gerentes de software avaliar o progresso.

A engenharia de software compreende um conjunto de etapas que envolvem métodos, ferramentas e os procedimentos discutidos anteriormente. Essas etapas muitas vezes são citadas como paradigmas de engenharia de software. Um paradigma de engenharia de software é escolhido tendo-se como base a natureza do projeto e da aplicação, os métodos e as ferramentas a serem usados, os controles e os produtos que precisam ser entregues. Quatro paradigmas têm sido amplamente discutidos (e debatidos) e são descritos a seguir:

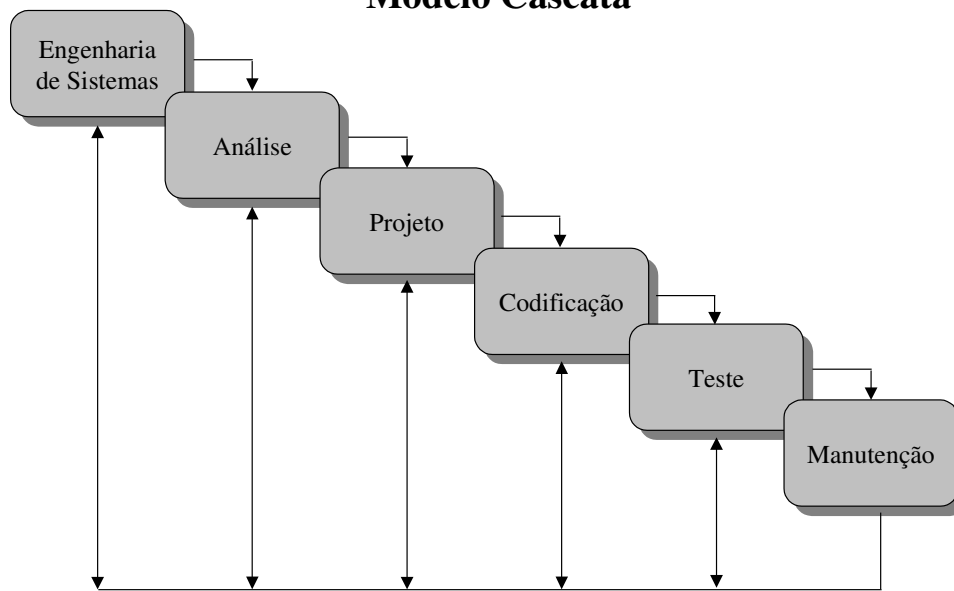
3.5.1.1 MODELO CASCATA

O paradigma do ciclo de vida requer uma abordagem sistemática, seqüencial ao desenvolvimento do software, que se inicia no nível do sistema e avança ao longo da análise, projeto, codificação, teste e manutenção. Modelado em função do ciclo da engenharia convencional, o paradigma do ciclo de vida abrange as seguintes atividades:

- ↪ Análise e engenharia de sistemas: Uma vez que o software sempre faz parte de um sistema mais amplo, o trabalho inicia-se com o estabelecimento dos requisitos para todos os elementos do sistema e prossegue com a atribuição de certo subconjunto desses requisitos ao software. Essa visão do sistema é essencial quando o software deve fazer interface com outros elementos, tais como hardware, pessoas e banco de dados. A análise e engenharia de sistemas envolve a coleta dos requisitos em nível do sistema, com uma pequena quantidade de projeto e análise de alto nível.
- ↪ Análise de requisitos de software: O processo de coleta dos requisitos é intensificado e concentrado especificamente no software. Para entender a natureza do(s) programa(s) a ser(rem) construído(s), o engenheiro (“analista”) de software deve compreender o domínio da informação para o software, bem como a função, desempenho e interface exigidos. Os requisitos, tanto para o sistema como para o software, são documentados e revistos com o cliente.
- ↪ Projeto: O projeto de software é, de fato, um processo de múltiplos passos que se concentra em quatro atributos distintos do programa: estrutura de dados, arquitetura de software, detalhes procedimentais e caracterização de interface. O processo de feitura do projeto traduz as exigências numa representação do software que pode ser avaliada quanto à qualidade antes que a codificação se inicie. Como os requisitos, o projeto é documentado e torna-se parte da configuração do software.

- ↪ Codificação: O projeto deve ser traduzido numa forma legível por máquina. A etapa de codificação executa essa tarefa. Se o projeto for executado detalhadamente, a codificação pode ser executada mecanicamente.
- ↪ Testes: Assim que o código for gerado, iniciar-se-á a realização de testes do programa. O processo de realização de testes concentra-se nos aspectos lógicos internos do software, garantindo que todas as instruções tenham sido testadas, e concentra-se também nos aspectos funcionais externos, ou seja, realizando testes para descobrir erros e garantir que a entrada definida produza resultados reais que concordem com os resultados exigidos.
- ↪ Manutenção: Indubitavelmente, o software sofrerá mudanças depois que for entregue ao cliente. Ocorrerão mudanças porque erros foram encontrados, porque o software deve ser adaptado a fim de acomodar mudanças em seu ambiente externo ou porque o cliente exige acréscimos funcionais ou de desempenho. A manutenção de software re replica cada uma das etapas precedentes do ciclo de vida a um programa existente, e não a um novo.

Modelo Cascata



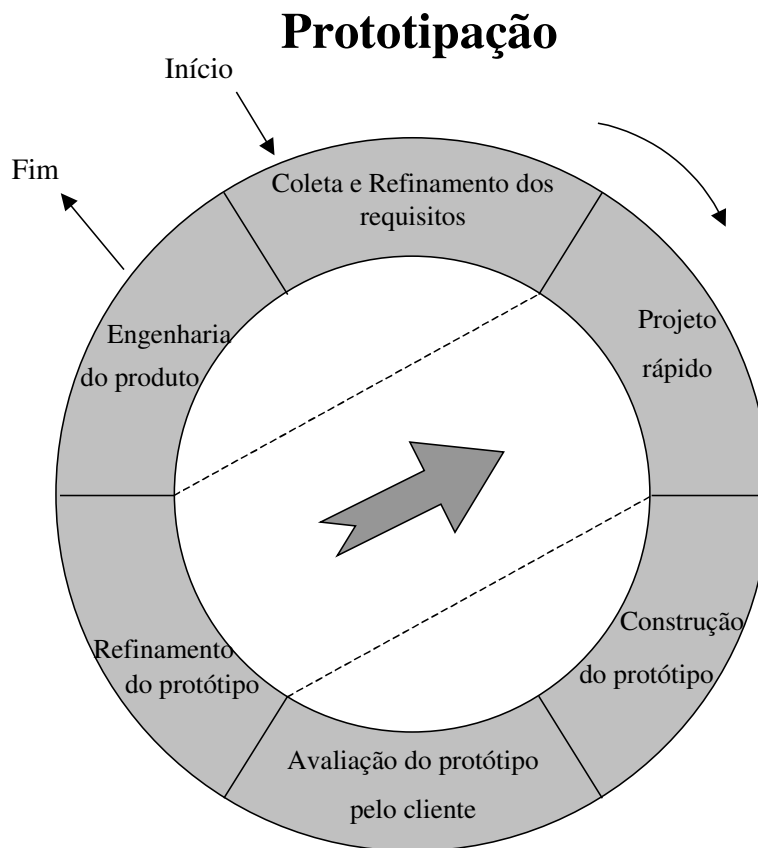
3.5.1.2 PROTOTIPAÇÃO

Muitas vezes o cliente definiu um conjunto de objetivos gerais para o software, mas não identificou requisitos de entrada, processamento e saída detalhados. Em outros casos, o desenvolvedor pode não ter a certeza da eficiência de um algoritmo, da adaptabilidade de um sistema operacional ou da forma que a interação homem-máquina deve assumir. Nessa, em muitas outras situações, uma abordagem de prototipação à engenharia de software pode representar a melhor abordagem.

A prototipação é um processo que capacita o desenvolvedor a criar um modelo de software que será implementado. O modelo pode assumir três formas: (1) um protótipo em papel ou modelo baseado em PC que retrata a interação homem-máquina de uma forma que capacita o usuário a entender quanta ocorrerá; (2) um protótipo de trabalho

que implementa algum subconjunto da função exigida do software desejado; ou (3) um programa existente que executa parte ou toda a função desejada, mas que tem outras características que serão melhoradas em um novo esforço de desenvolvimento.

A seqüência de eventos para o paradigma de prototipação é ilustrada na figura a seguir. Como todas as abordagens ao desenvolvimento de software, a prototipação inicia-se com a coleta de requisitos. O desenvolvedor e o cliente reúnem-se e definem os objetivos globais para o software, identificam as exigências conhecidas e esboçam as áreas em que uma definição adicional é obrigatória. Ocorre então a elaboração de um “projeto rápido”. O projeto rápido concentra-se na representação daqueles aspectos de software que serão visíveis ao usuário. O projeto rápido leva à construção de um protótipo que é avaliado pelo cliente/usuário e é usado para refinar os requisitos para o software a ser desenvolvido. Um processo de iteração ocorre quando é feita uma “sintonia fina” do protótipo para satisfazer as necessidades do cliente, capacitando, ao mesmo tempo, o desenvolvedor a compreender melhor aquilo que precisa ser feito.



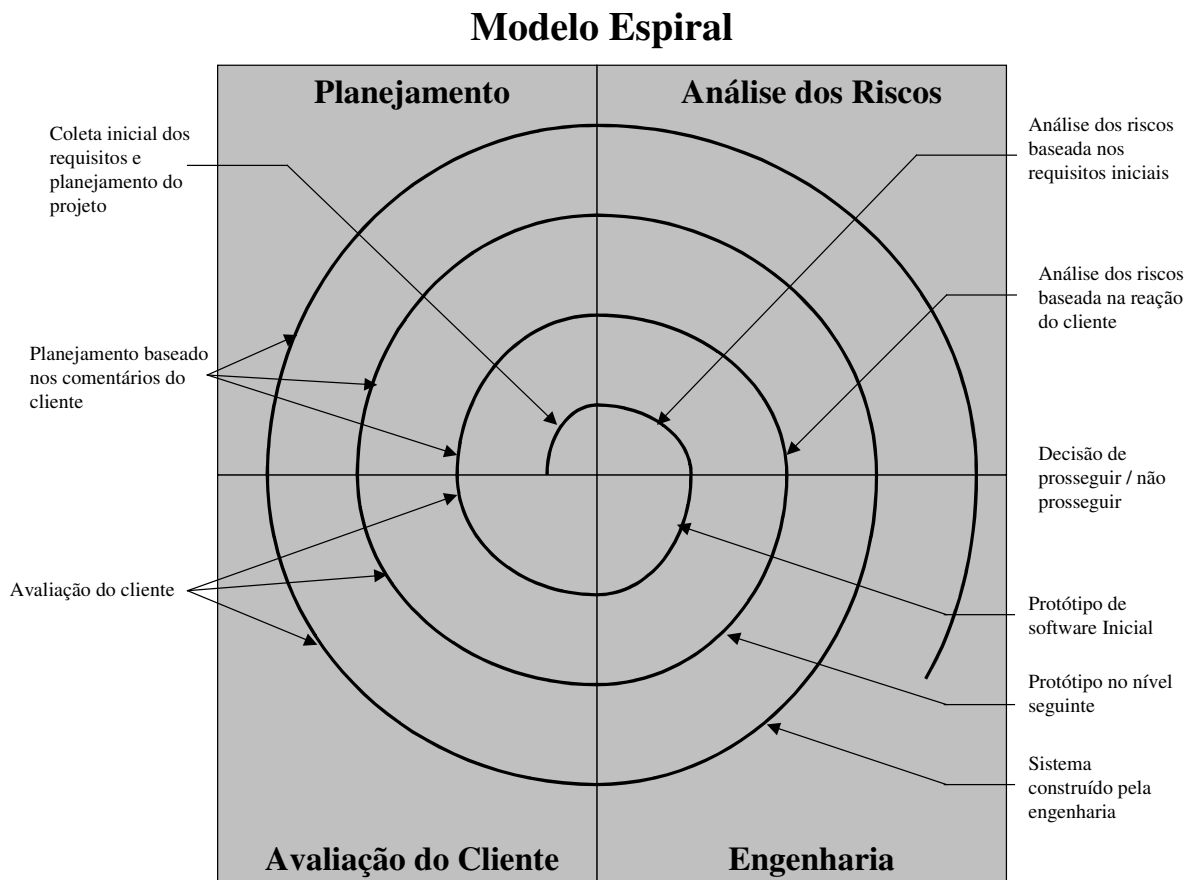
Idealmente, o protótipo serve como um mecanismo para identificar os requisitos de software. Se um protótipo de trabalho for construído, o desenvolvedor tentará usar fragmentos de programas existentes ou aplicará ferramentas que possibilitem que programas de trabalho sejam gerados rapidamente.

3.5.1.3 MODELO ESPIRAL

O modelo espiral foi desenvolvido para abranger as melhores características tanto do modelo em cascata como da prototipação, acrescentando, ao mesmo tempo, um novo elemento – a análise dos riscos – que falta a esses paradigmas. O modelo define quatro importantes atividades:

- ↳ Planejamento: determinação dos objetivos, alternativas e restrições;
- ↳ Análise dos riscos: análise de alternativas e identificação / resolução dos riscos;
- ↳ Engenharia: desenvolvimento do produto no “nível seguinte”;
- ↳ Avaliação feita pelo cliente: avaliação dos resultados da engenharia.

Um aspecto intrigante do modelo espiral torna-se claro quando consideramos a dimensão radial. Com cada iteração ao redor da espiral (iniciando-se ao centro e avançando para fora), versões progressivamente mais completas do software são construídas. Durante o primeiro giro da espiral, os objetivos, alternativas e restrições são definidos e os riscos são identificados e analisados. Se a análise dos riscos indicar que há incertezas nos requisitos, a prototipação pode ser usada no quadrante da engenharia para ajudar tanto o desenvolvedor quanto o cliente. Simulações e outros modelos podem ser usados para definir ainda mais o problema e refinar os requisitos.



O cliente avalia o trabalho de engenharia e apresenta sugestões para modificações. Baseada na entrada do cliente, ocorre a fase seguinte de planejamento e análise dos riscos. Em cada arco da espiral, a conclusão da análise dos riscos resulta numa decisão

de “prosseguir/ não prosseguir”. Se os riscos forem muito grandes, o projeto pode ser encerrado.

Na maioria dos casos, entretanto, o fluxo ao redor de uma trajetória espiral continua, com cada trajetória movimentando os desenvolvedores para fora na direção de um modelo mais completo do sistema e, em última análise, ao próprio sistema operacional. Cada giro ao redor espiral requer um trabalho de engenharia que possa ser levado a efeito usando-se ou a abordagem de ciclo de vida clássico ou prototipação. Deve-se notar que o número de atividades de desenvolvimento que ocorre no quadrante inferior direito eleva-se à medida que as atividades se afastam do centro da espiral.

O paradigma de modelo espiral para a engenharia de software atualmente é a abordagem mais realística para o desenvolvimento de sistemas e de software em grande escala. Ele usa uma abordagem “evolucionária” à engenharia de software, capacitando o desenvolvedor e o cliente a entender e reagir aos riscos em cada etapa evolutiva. O modelo espiral usa a prototipação como um mecanismo de redução dos riscos, mas, o que é mais importante, possibilita que o desenvolvedor aplique a abordagem de prototipação em qualquer etapa da evolução do produto. Ele mantém a abordagem de passos sistemáticos sugerida pelo ciclo de vida clássico, mas incorpora-a numa estrutura iterativa que reflete mais realisticamente o mundo real. O modelo espiral exige uma consideração direta dos riscos técnicos em todas as etapas do projeto e, se adequadamente aplicada, deve reduzir os riscos antes que eles se tornem problemáticos.

3.5.1.4 TÉCNICAS DE QUARTA GERAÇÃO

O termo “técnicas de quarta geração” (4GT) abrange um amplo conjunto de ferramentas de software que têm uma coisa em comum: cada uma delas possibilita que o desenvolvedor de software especifique alguma característica do software num nível elevado. A ferramenta gera então, automaticamente, o código-fonte, tendo como base a especificação do desenvolvedor. Não há muito o que contestar a esse respeito; quanto mais alto o nível em que o software pode ser especificado a uma máquina, mais rapidamente o programa pode ser construído. O paradigma 4GT concentra-se na capacidade de se especificar software a uma máquina em um nível que esteja próximo à linguagem natural ou de se usar uma notação que comunique uma função significativa.

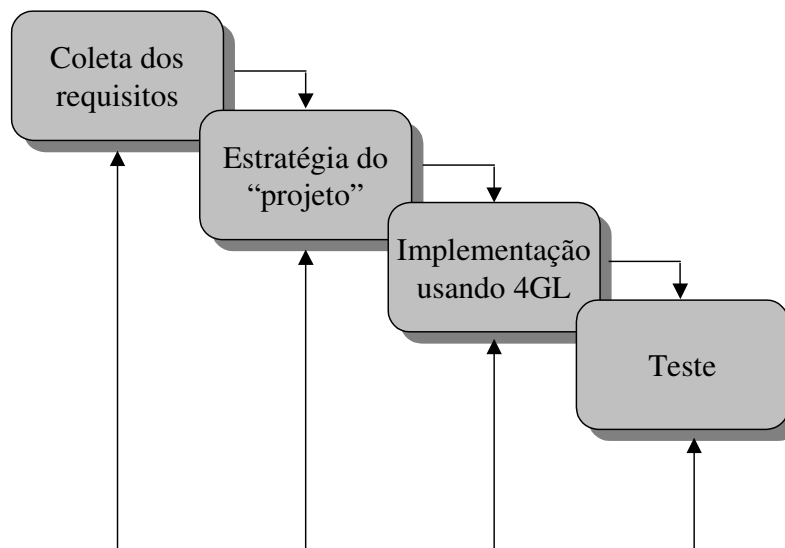
Atualmente, o ambiente de desenvolvimento de software que sustenta o paradigma 4GT inclui algumas, ou todas, das seguintes ferramentas: linguagens não-procedimentais para consulta de bancos de dados, geração de relatórios, manipulação de dados, interação e definição de telas, geração de códigos, capacidade gráfica de alto nível e capacidade de planilhas eletrônicas. Cada uma dessas ferramentas existe, mas somente para domínios de aplicação muito específicos.

Como os demais paradigmas, o 4GT inicia-se com uma etapa de coleta de requisitos. Idealmente, o cliente descreveria os requisitos, e estes seriam diretamente traduzidos num protótipo operacional. Mas isso não é idealmente possível. O cliente pode não ter certeza daquilo que é exigido, pode ser ambíguo ao especificar fatos que são conhecidos e pode ser inviável especificar as informações de maneira que uma ferramenta 4GT possa receber. Além disso, as atuais ferramentas 4GT não são sofisticadas o bastante para acomodar verdadeiramente “linguagem natural”, e não o serão por algum tempo. No momento, o diálogo cliente-desenvolvedor descrito para outros paradigmas continua sendo uma parte essencial da abordagem 4GT.

A implementação usando uma 4GL possibilita que o desenvolvedor de software represente os resultados desejados de uma forma que resulte em geração automática de código para produzir esses resultados. Obviamente, uma estrutura de dados com informações relevantes deve existir e ser prontamente acessível pela 4GT.

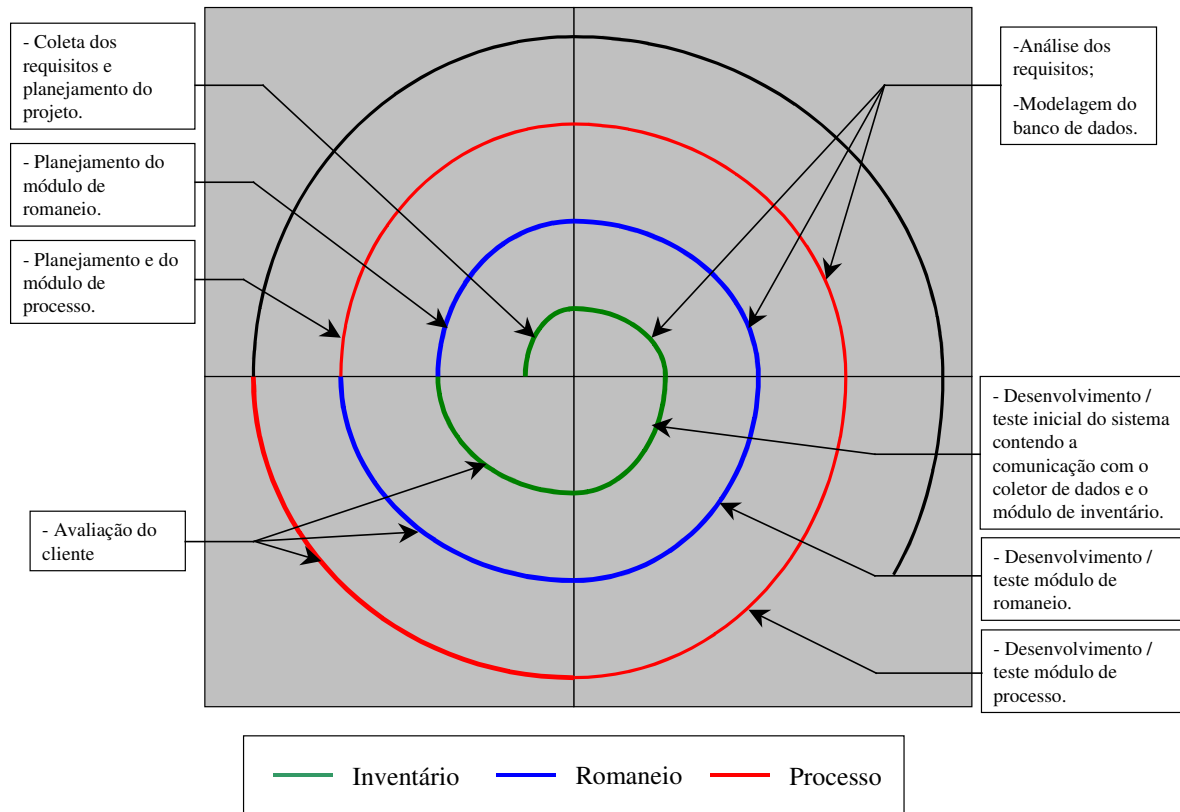
Para transformar a implementação de uma 4Gt num produto, o desenvolvedor deve realizar testes cuidadosos, desenvolver uma documentação significativa e executar todas as demais atividades de “transição” que também são exigidas em outros paradigmas de engenharia de software. Além disso, o software desenvolvido por 4GT deve ser construído de um modo que possibilite rápida manutenção.

Técnicas de Quarta Geração



3.5.2 MODELO UTILIZADO PARA O DESENVOLVIMENTO

Para o desenvolvimento do sistema, a Engine utilizou o modelo espiral, ilustrado na figura a seguir. O sistema foi dividido em módulos e cada módulo desenvolvido era entregue ao cliente para a sua utilização. O sistema foi dividido em três módulos – inventário, romaneio e processo.



Para cada módulo, além das interfaces do sistema, tivemos que desenvolver todo o processamento e interface do coletor de dados.

↳ Módulo de inventário:

- Sistema: Visualiza o inventário realizado pelos coletores de dados;
- Coletor de Dados: Inicia um novo inventário e faz a leitura dos produtos;

↳ Módulo do romaneio:

- Sistema: Importa arquivos texto contendo a descrição da nota fiscal de compra de uma determinada empresa, gerados pelo software Microsiga que a empresa utiliza e atribui aos coletores os romaneios importados.
- Coletor de Dados: Escolhe o romaneio a ser realizado, faz a separação dos produtos através da leitura do código de barras dos mesmos e exporta o romaneio separado contendo os detalhes de cada produto;

- ↪ Módulo do processo:
 - Sistema: Possui relatórios gerenciais para contabilizar as etiquetas geradas e, conseqüentemente, os produtos que entram no depósito;
 - Coletor de Dados: Alguns coletores possuem uma impressora acoplada e geram etiquetas para os produtos através da digitação da metragem dos mesmos.

Na primeira etapa do projeto, tivemos diversas reuniões com o *SPONSOR* da empresa contratante, para entender o processo existente, as necessidades que eles tinham, como a empresa funcionava e também para conhecer as pessoas envolvidas no projeto. Nessa mesma etapa, foram solicitados diversos documentos para estudo (workflow dos documentos). As únicas documentações feitas na especificação dos requisitos foi o fluxo de dados dos módulos e a da modelagem do banco de dados.

3.5.3 FERRAMENTAS UTILIZADAS:

As ferramentas utilizadas no projeto, fabricadas pela Borland, foram:

- ↪ Delphi 5 (para o desenvolvimento);
- ↪ Interbase (banco de dados).

Os coletores de dados adquiridos pela empresa contratante foram indicados pela Engine e emulam um terminal TELNET, assim o sistema desenvolvido teve que cuidar de toda sua interface e processamento.

3.6 PROBLEMAS NO PROJETO

3.6.1 ALTERAÇÃO NO ESCOPO DO PROJETO

Como mencionado no item 3.5.2, a documentação do sistema não foi uma preocupação da Engine, já que se tratava do desenvolvimento de um sistema de pequeno porte. A falta dessa documentação não trouxe problemas para o desenvolvimento do sistema, porém uma mudança ocorrida no final do projeto atrasou a entrega do último módulo do sistema (Processo / Carta Programa). No início do projeto, não estava previsto a utilização de impressora para imprimir etiquetas, pois as bobinas de tecidos já viriam etiquetadas, porém não foi o que aconteceu. A falta de conhecimento técnico com a impressora gerou muito tempo de estudo e de testes, tanto para fazer o design das etiquetas como para fazer a sua comunicação com o coletor de dados. Mas isso não foi um erro da Engine, pelo contrário, foi uma mudança no final do desenvolvimento do sistema e como a Millennium não conseguiria trabalhar sem as etiquetas, a Engine resolveu por atrasar a entrega final do projeto para concluir essas alterações.

3.6.2 FALTA DE PLANEJAMENTO DE MANUTENÇÃO

Por volta de 3 meses de utilização do sistema, a Engine recebeu algumas reclamações de que o sistema estava muito lento e foi chamada para fazer uma limpeza no banco de dados da Millenium. Verificamos que algumas tabelas estavam muito grandes e mesmo apagando alguns registros antigos, o problema voltaria a acontecer, então decidimos alterar o banco de dados e criamos algumas tabelas de histórico diminuindo o tempo de consulta de algumas rotinas do sistema.

3.6.3 FALTA DE GARANTIA E DE CONTRATO DE MANUTENÇÃO

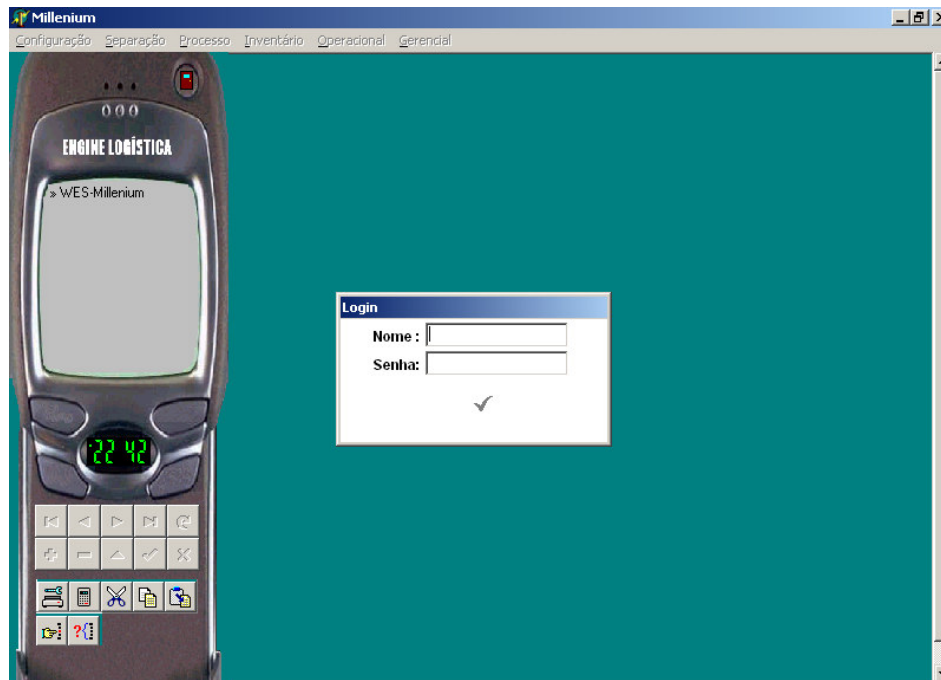
Após a implantação do sistema com todos os módulos, a Engine acompanhou a utilização do sistema por duas semanas, incluindo o treinamento feito para os usuários. Após esse período, não havia nenhum contrato de manutenção entre a empresa contratante e a Engine e todas os atendimentos necessários eram feitos por telefone ou eram agendados em possíveis datas que nem sempre eram de imediato. Isso causou algum desconforto por parte da empresa contratante, mas também gerou alguns prejuízos para a Engine, já que alguns erros eram operacionais e perdíamos tempo para descobrir.

3.7 O PROGRAMA

3.7.1 TELA DE LOGIN

O usuário digita o nome de usuário e a sua senha para acessar o sistema.

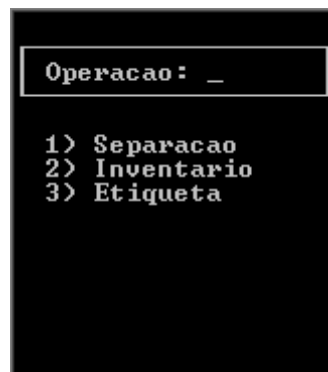
↩ SISTEMA:



↩ COLETOR DE DADOS:



Tela de Login

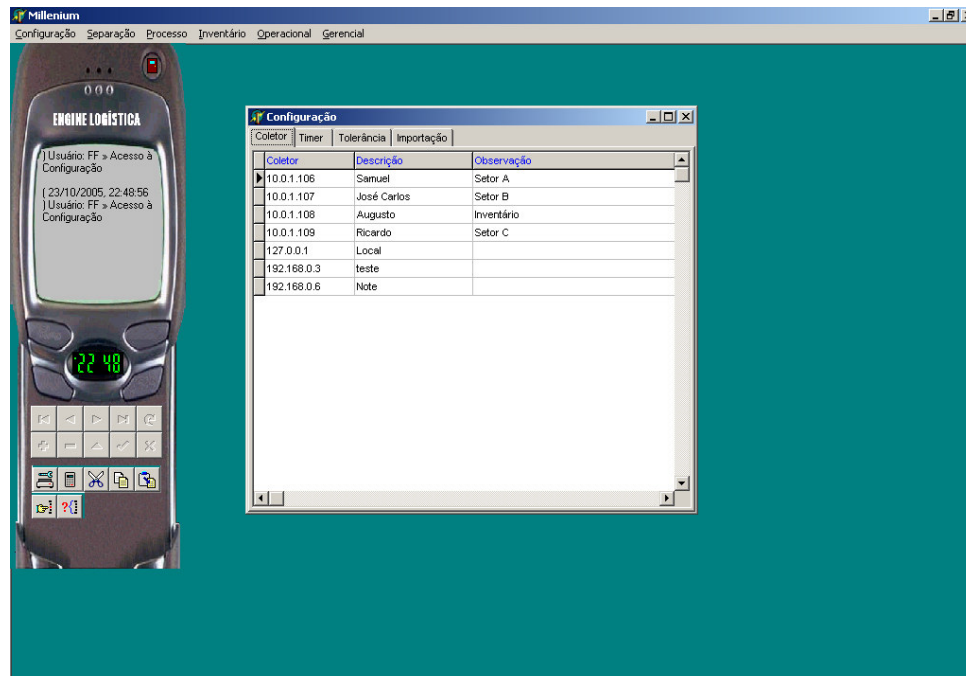


Menu Principal

3.7.2 CONFIGURAÇÃO DO SISTEMA

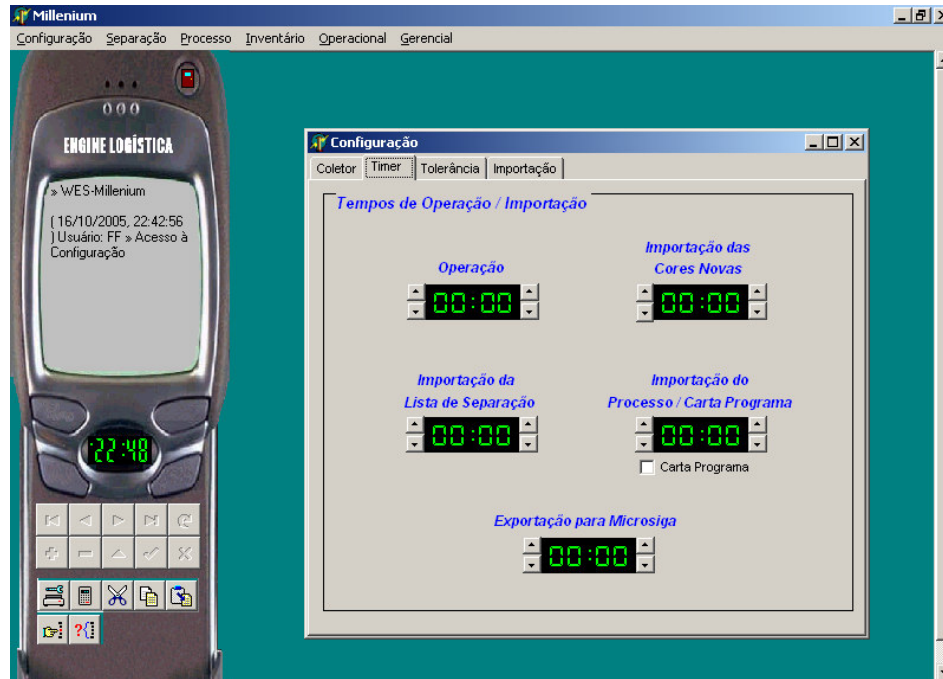
3.7.2.1 CADASTRO DOS COLETORES

Todos os coletores de dados são cadastrados aqui através do seu IP fixo e da sua identificação.



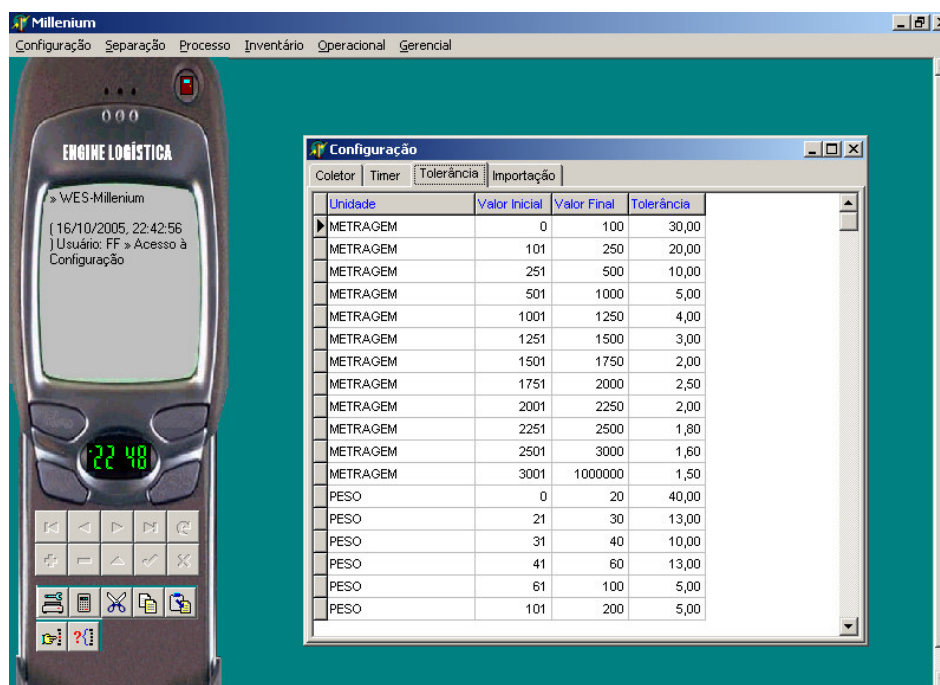
3.7.2.2 CADASTRO DE TIMERS

Aqui são cadastrados os intervalos de tempo para a atualização de dados na tela ou para importação de novas listas de separação.



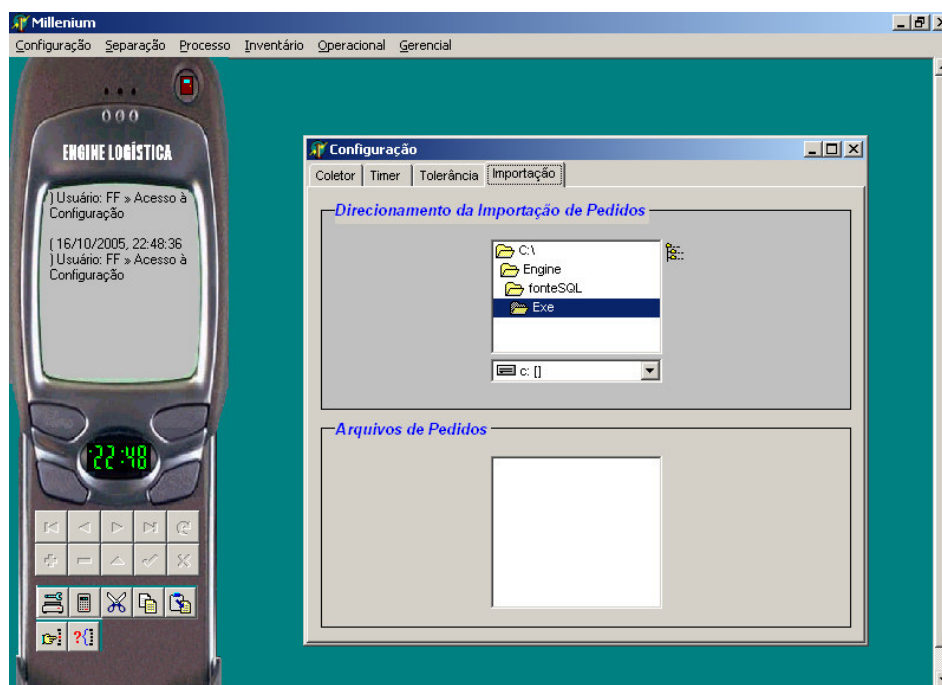
3.7.2.3 CADASTRO DE TOLERÂNCIA

Algumas bobinas são vendidas pela sua metragem, enquanto outras são vendidas pelo seu peso, ambos não padronizados. Para resolver esse problema, foi necessário criar no sistema um cadastro de tolerância para o encerramento e ajuste nas notas fiscais de saída. Assim, quando são solicitados 10 metros de um determinado tecido, por exemplo, o sistema verifica a tolerância da metragem solicitada e, neste caso, pode encerrar a separação com 30% de tolerância (de 7 a 13 metros do tecido).



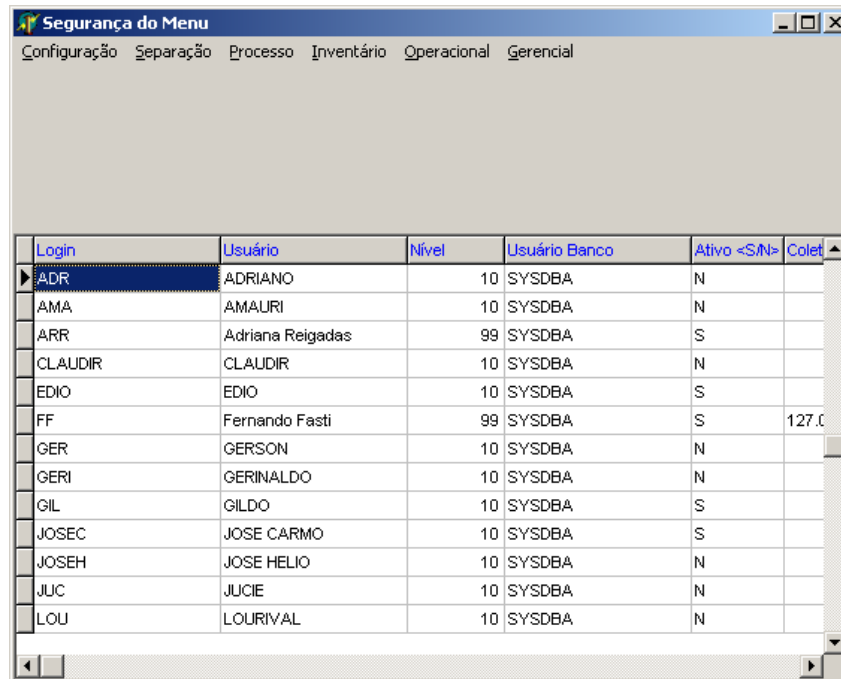
3.7.2.4 CADASTRO DO PATH PARA A IMPORTAÇÃO

Um das interfaces que o sistema WMS possui com o software de gestão da empresa é importação dos romaneios. Quando uma solicitação de compra é aprovada pela área responsável da Millenium, esse software gera um arquivo texto com o detalhe da venda. O sistema WMS verifica periodicamente, através do cadastro do timer (mencionado no item X.X), se existem novos arquivos para serem importados e armazenados na tabela apropriada.



3.7.3 CADASTRO DE PERMISSÕES

Cada usuário cadastrado no sistema possui um nível de acesso podendo acessar somente o que é necessário.



Login	Usuário	Nível	Usuário Banco	Ativo <S/N>	Colet
ADR	ADRIANO	10	SYSDBA	N	
AMA	AMAURI	10	SYSDBA	N	
ARR	Adriana Reigadas	99	SYSDBA	S	
CLAUDIR	CLAUDIR	10	SYSDBA	N	
EDIO	EDIO	10	SYSDBA	S	
FF	Fernando Fasti	99	SYSDBA	S	127.0
GER	GERSON	10	SYSDBA	N	
GERI	GERINALDO	10	SYSDBA	N	
GIL	GILDO	10	SYSDBA	S	
JOSEC	JOSE CARMO	10	SYSDBA	S	
JOSEH	JOSE HELIO	10	SYSDBA	N	
JUC	JUCIE	10	SYSDBA	N	
LOU	LOURIVAL	10	SYSDBA	N	

3.7.4 INVENTÁRIO

O inventário é feito pelo coletor de dados e pode ser visualizado e impresso através do sistema.

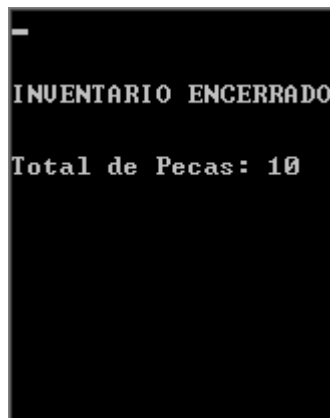
↩ COLETOR DE DADOS



Tela de inventário
(Lista todos os inventários não encerrados)



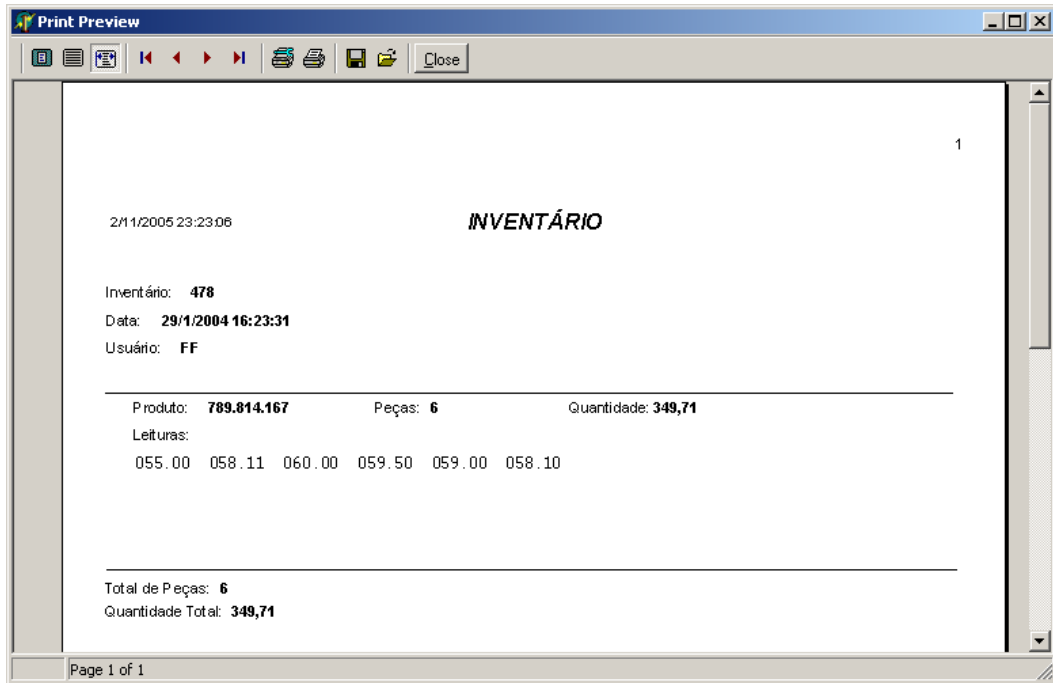
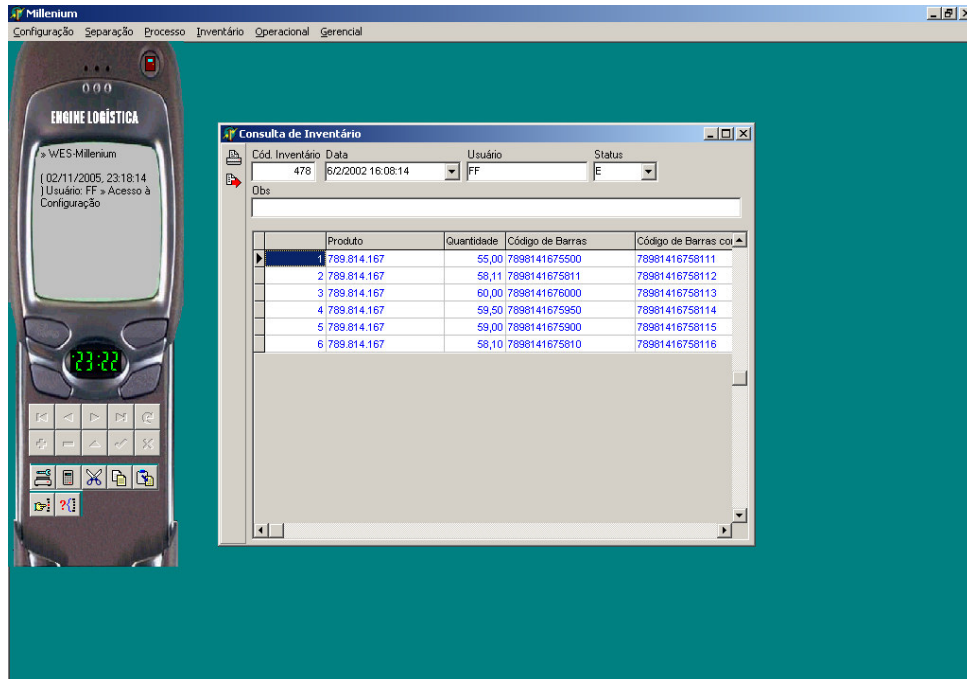
Tela com um novo inventário
(O usuário lê as etiquetas e fica armazenado no inventário)



Tela de encerramento de Inventário
(mostrando o total de peça lidas)

↩ SISTEMA:

O usuário escolhe o número do inventário e verifica todos os produtos lidos, podendo também imprimir um relatório consolidado do inventário.



3.7.5 ROMANEIO

Após a importação dos romaneios, é necessário atribuí-los a um determinado coletor de dados, para que o usuário possa visualizá-lo e iniciar a sua operação.

Cliente	Pedido	Separação	Coletor	UF	Produto	Unidade	Qtde Pedida	Qtde Separada
A. LEGATO	376528	70998		SP	215.447	MT	600,00	182,00
ADAR (B.F. 2002)	190959	71118		SP	022.161	MT	75586,00	0,00
ADAR (B.F. 2002)	370102	69611		SP	385.693	KG	1,00	0,00
ADAR (B.F. 2002)	370103	69612	FF	SP	385.680	KG	6,60	0,00
ADAR (B.F. 2002)	370117	70415	104	SP	225.424	MT	6,00	0,00
ADAR (B.F. 2002)	370121	70416	104	SP	135.338	MT	7,70	0,00
ADAR (B.F. 2002)	370122	70417		SP	385.996	KG	4,90	0,00
ADAR (B.F. 2002)	370124	70472		SP	105.850	MT	68,00	0,00
ADAR (B.F. 2002)	370125	70473		SP	122.139	MT	108,00	0,00

Código do Produto	Descrição	Qtde Pedida	Qtde Separada	Unidade
215.447.X21	HI MULTI CHIFFON MELAO	100,00	0	MT
215.447.VM12	HI MULTI CHIFFON GIRASOL	200,00	182	MT

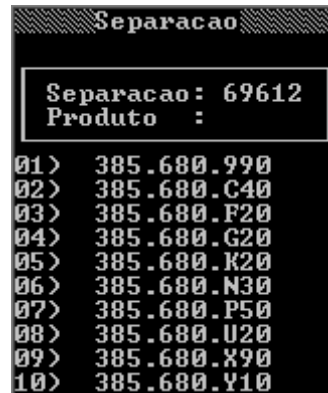
Operação						
Coletor	Usuário	Separações Atribuídas	Metr.Total	Metr. Faltante	Peso Total	Peso Faltante
127.0.0.1	Fernando Fasti	1			6,60	6,60
	Adriana Reigadas	0				
	EDIO	0				
	GILDO	0				
	JOSE CARMO	0				
	MARCOS	0				
	MOISES	0				
	Paulo Reigadas	0				

Metragem Total (m) :	0
Peso Total (Kg) :	0

↩ **NO COLETOR:**



Quando o usuário entra no menu de separação, aparecem todos os romaneios atribuídos a ele.



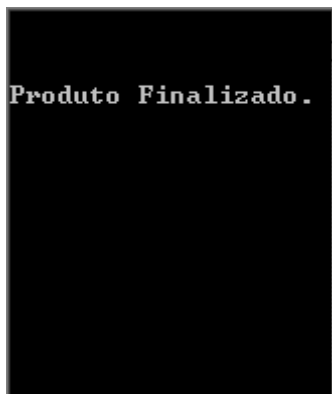
Após a escolher a separação, são listados todos os produtos a serem separados



Após escolher um produto, alguns detalhes são mostrados na tela inclusive a quantidade a ser separada. O sistema vai adicionando os produtos ao romaneio conforme eles vão sendo lidos pelos coletores de dados



Quando um produto é lido, o sistema armazena todos os dados do mesmo e mostra ao separador a quantidade separada. (Qtde Real)

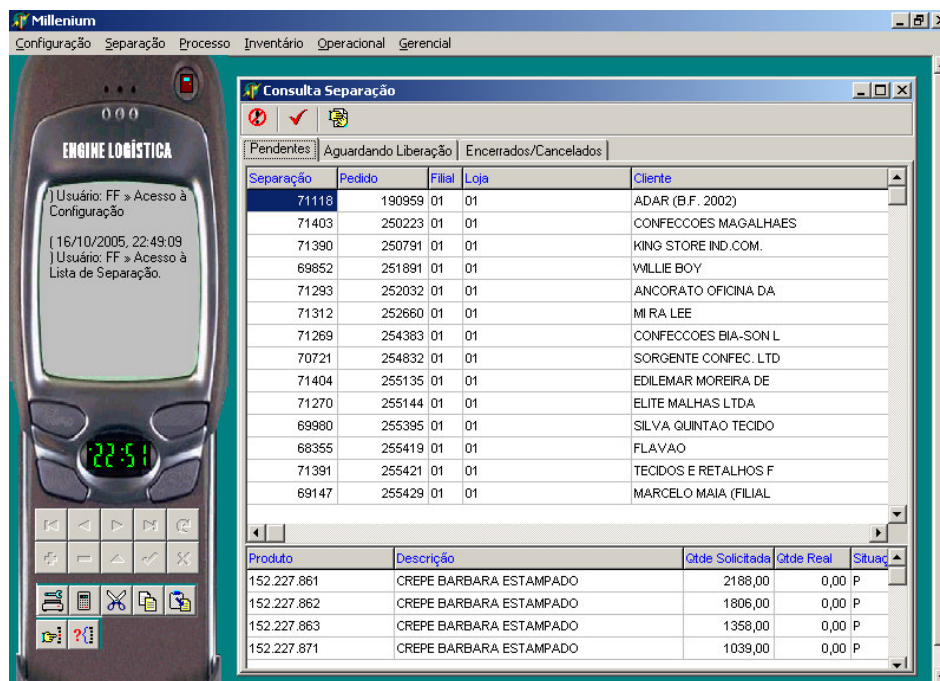


Tela avisando o termino da separação do produto.

3.7.5.1 GERENCIAMENTO DOS ROMANEIOS

Além da tela de atribuições dos romaneios, o sistema possui uma outra tela para a sua administração. O status do romaneio pode ser:

- ↩ Pendente: romaneio a ser separado;
- ↩ Aguardando Liberação: romaneio bloqueado, pode ser por falta do produto no armazém, por exemplo;
- ↩ Encerrado / Cancelado: romaneios terminados ou cancelados.



3.7.6 PROCESSO / CARTA PROGRAMA

O Processo/Carta Programa funcionam de maneira parecido com o romaneio, mas aqui nos referimos ao produtos que estão entrando no armazém. O Processo se trata das notas fiscais de entrada de produtos nacionais e a Carta Programa se trata de produtos internacionais.

Processo	Tipo Processo	Coletor	Ordem de Compra	Container	Lote	Unidade	Ctde Pedida
1002	P		809344			MT	0
1003-A	P		809339	TEXU 7248370		MT	2177
1003-B	P	FF	809318	FSCU 3174495		MT	128
1006	P		809319	TTNU 3114371		MT	148
1007	P		809287	TRLU 5748517		MT	99
1020	P		809241	NYKU6124625		MT	2465
1021	P		809160	TRLU 2108497		MT	1209
1025	P		809337	NYKU 6560871		MT	284
			809288	NYKU 6218365		MT	3032

Produto	Composição do Produto	Peso Total	Peso Recebido	Descrição do Produto	Ctde Pedida
045.893.990	100%POLIESTER	4050,00		LINEN DIAMOND PRETO	
045.893.K40	100%POLIESTER	4050,00		LINEN DIAMOND ACUCENA	

Operação					
Coletor	Usuário	Processos Atribuidos	Metr. Total	Peso Total	
127.0.0.1	Fernando Fasti	3	276	688,00	
	Adriana Reigadas		0		
	EDIO		0		
	GILDO		0		
	JOSE CARMO		0		
	MARCOS		0		
	MOISES		0		
	Paulo Reigadas		0		

```

Processo: __

01> 1003-A
02> 1003-B
03> M-0094-A
    
```

Processos atribuídos a usuário

```

Cor: __

01> 355.712.010
    
```

Lista com os produtos

```

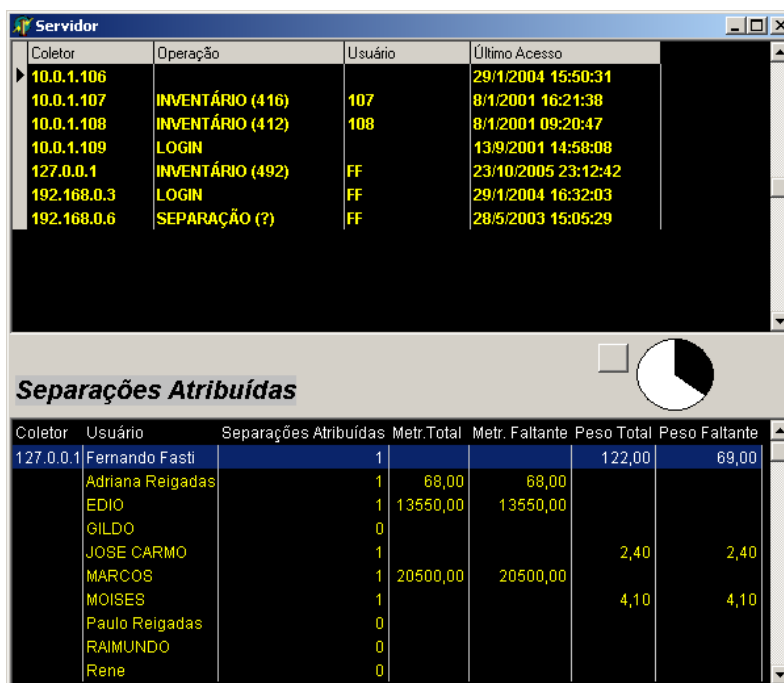
ETIQUETAS

Artigo: TRICOLINE MI
STA BRANCO
Composicao: 65%POLIE
STER 35%ALGODAO
Cor: BRANCO
Peso: __
    
```

O usuário digita o peso/metragem da bobina e a etiqueta é impressa.

3.7.7 GERENCIAMENTO DE USUÁRIOS

O sistema WMS possui uma tela de gerenciamento on-line (atualizada periodicamente através da configuração do timer apropriado), podendo visualizar as tarefas realizadas pelos coletores e a quantidade de separações que ainda possui para separar.



Coletor	Operação	Usuário	Último Acesso
10.0.1.106			29/1/2004 15:50:31
10.0.1.107	INVENTÁRIO (416)	107	8/1/2001 16:21:38
10.0.1.108	INVENTÁRIO (412)	108	8/1/2001 09:20:47
10.0.1.109	LOGIN		13/9/2001 14:58:08
127.0.0.1	INVENTÁRIO (492)	FF	23/10/2005 23:12:42
192.168.0.3	LOGIN	FF	29/1/2004 16:32:03
192.168.0.6	SEPARAÇÃO (?)	FF	28/5/2003 15:05:29

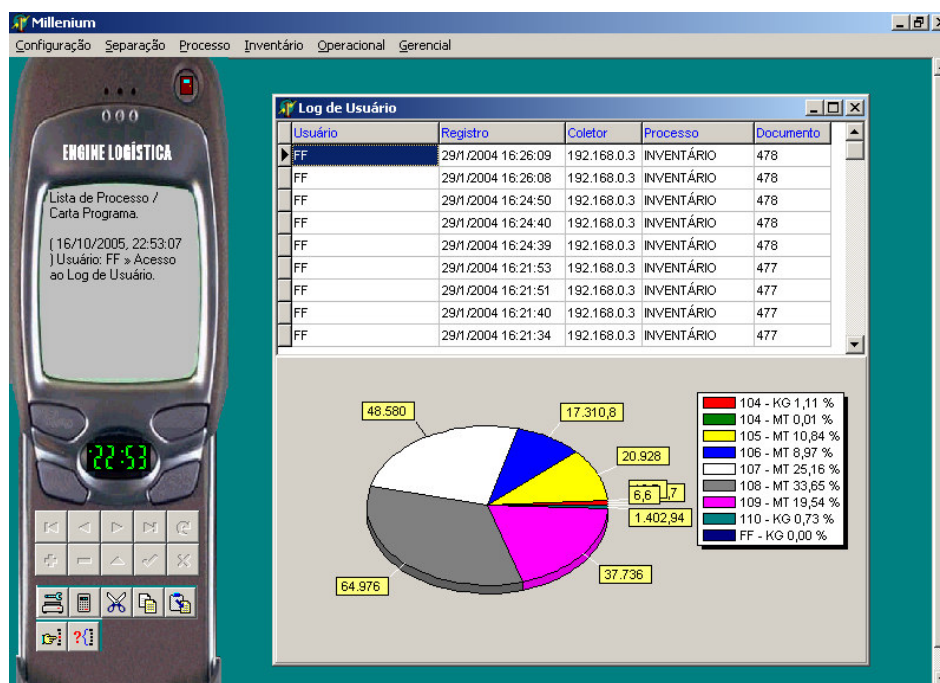
Coletor	Usuário	Separações Atribuídas	Metr.Total	Metr. Faltante	Peso Total	Peso Faltante
127.0.0.1	Fernando Fasti	1			122,00	69,00
	Adriana Reigadas	1	68,00	68,00		
	EDIO	1	13550,00	13550,00		
	GILDO	0				
	JOSE CARMO	1			2,40	2,40
	MARCOS	1	20500,00	20500,00		
	MOISES	1			4,10	4,10
	Paulo Reigadas	0				
	RAIMUNDO	0				
	Rene	0				

3.7.8 RELATÓRIOS GERENCIAIS

O sistema WMS possui três tipos de relatórios gerenciais:

3.7.8.1 LOG DE USUÁRIO

Verifica os acessos ao sistema e os menus acessados.



3.7.8.2 RELATÓRIO DE DESEMPENHO

O sistema mostra estatística sobre um usuário.

Seleção de Relatório

Operador: FF

Hora Inicial: 00:01

Hora Final: 23:59

Data Inicial: 01/01/2004

Data Final: 02/11/2005

OK

Cancela

2/11/2005 23:46:07

RELATÓRIO DE DESEMPENHO

Operador: **102**

Período:

Data: 01/01/2004 - 02/11/2005

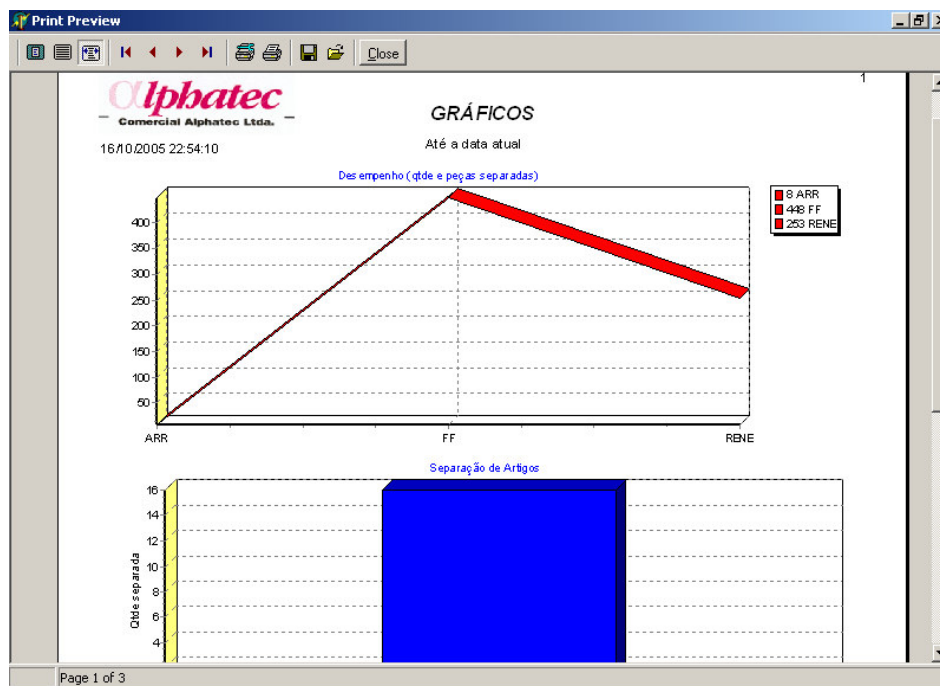
	Quantidade separada	% por unidade de medida	% por unidade de medida sobre total
(Mt)	52030,69	76,75%	100,00%
(Kg)	15765,24	23,25%	100,00%

	PERÍODO DE LIBERAÇÃO		PRAZO DE ENCERRAMENTO	
Período 1	11:01 - 16:30		11:01 - 17:00	

	Pedidos liberados	Pedidos separados no prazo	% de atendimento	Max peças separadas
Período 1	24	20	83,33%	96,00
Período 2	33	33	100,00%	95,00

0% Page 1 of 1

3.7.8.3 GRÁFICOS COMPARATIVOS



3.8 RESULTADOS OBTIDOS

O sistema desenvolvido pela Engine automatizou as atividades do armazém fazendo a empresa ganhar agilidade, rapidez e conseqüentemente maior produtividade. O sistema trouxe também outros benefícios:

- ↪ Visualização, em tempo real, de todas as operações que estão sendo realizadas no armazém;
- ↪ Verificação do desempenho das equipes;
- ↪ Com o uso dos coletores de dados, os separadores aumentaram suas produtividades e reduziram os erros relacionados à separação das bobinas;

3.9 MINHA PARTICIPAÇÃO NO PROJETO

No início do projeto, acompanhei a equipe da Engine nas reuniões com a empresa contratante, para o entendimento do problema e para conhecer as pessoas envolvidas no projeto. Fizemos o levantamento dos requisitos do sistema e a modelagem do banco de dados utilizado pelo sistema.

Na parte de desenvolvimento do sistema, já comecei a ter mais responsabilidades e maior atuação no projeto, comecei a desenvolver algumas rotinas do sistema e também a testar outras. As interfaces e as rotinas usadas pelo coletor de dados foram as partes do sistema em que tive maior dificuldade em trabalhar, pois como os coletores trabalhavam paralelamente, em apenas uma instância, usamos os conceitos de Threads para a comunicação, que era um conceito novo e um pouco difícil de entender nas atuais circunstâncias.

Nas demais etapas, também tive uma grande participação no projeto:

- ↪ nos testes do sistema realizados na Engine e na empresa contratante, na fase de implantação do sistema;
- ↪ no treinamento dos usuários, tanto do sistema como dos coletores de dados;
- ↪ no suporte por telefone ou em campo, quando necessário, após a implantação;
- ↪ na manutenção do sistema.

3.9.1 TREINAMENTO

Assim que iniciei o estágio, tive cerca de um mês dedicado exclusivamente para treinamento e integração com a equipe. Como nunca havia programado em Delphi, fiquei cerca de três semanas lendo apostilas e fazendo pequenos programas para ir me habituando com esse ambiente de desenvolvimento. Após esse período, fiz um curso de desenvolvimento de aplicações utilizando Delphi, na Borland do Brasil, que me deu uma boa base para começar a trabalhar nos projetos da Engine.

4 PARTE SUBJETIVA

4.1 EXPERIÊNCIA PESSOAL

Um dos grandes méritos do IME - Instituto de Matemática e Estatística, é fazer com que o aluno busque por informações complementares às vistas em sala de aula, por meio de pesquisas e estudo, pois nem sempre o que é visto em aula é suficiente para fazer os projetos e as provas. Isso é um grande diferencial no mercado de trabalho, a pessoa se torna auto-ditada e muito independente. Isso é fundamental para a área de informática onde sempre estão surgindo novas tecnologias e cada vez mais os profissionais desse segmento tem que estar alinhados com essas tecnologias.

No meu estágio, realizado na empresa Engine Logística Ltda., não foi diferente. Eu me deparei com diversas tecnologias novas (na época) e tive que aprendê-las muito rapidamente, assim como alguns conceitos novos, que estudei posteriormente no curso, como banco de dados, concorrência, rede, entre outros.

Fazer o estágio no início do curso, contrariando o que sugere a maioria dos professores do BCC, tem seus pontos positivos e negativos. Entre os aspectos positivos, acho que aprender em paralelo a teoria e a prática torna o ensino muito mais completo. Você acaba tendo um outro ponto de vista da matéria dada em sala de aula, já pensando onde pode ser aplicado, na prática, um conceito novo visto. Um exemplo disso, é o curso de Engenharia de Software, que ficou muito mais fácil o entendimento para mim que já havia tido contato com um projeto realizado no estágio do que para um aluno que não fazia idéia do que se tratava.

Um dos aspectos negativos, é que com o estágio não sobra muito tempo para você se dedicar o suficiente às disciplinas que esta cursando. Além disso, eu tive outro problema, não pude fazer os cursos no período considerado ideal pelos professores por dois motivos, além do estágio, que me impedia de fazer matérias de tarde. Com eu ingressei no curso do BCC através da transferência interna, minha grade de horário ficou uma verdadeira bagunça, em todos os semestres tive que cursar disciplinas de semestres diferentes, e, em alguns casos por exemplo, apesar de uma matéria de um semestre não ser pré-requisito para uma do semestre posterior, ela sempre apresenta alguns conceitos que ajudam na próxima disciplina, prejudicando um pouco no aprendizado.

Como a Engine é uma empresa de pequeno porte e com poucos funcionários, o meu estágio não ficou limitado só na parte de desenvolvimento, fiz um pouco de tudo. Aprendi um pouco de administração de empresas, configurar redes e micros e até um pouco de marketing. Isso, e todo o restante do meu estágio, contribuiu muito para o meu crescimento pessoal e profissional.

4.2 DESAFIOS E FRUSTRAÇÕES ENCONTRADAS

Quando iniciei o meu estágio, estava cursando Bacharelado em Matemática, mas já tinha o intuito de pedir transferência para o curso de Ciência da Computação. Como essa transferência se dava por meio de avaliação do currículo do aluno, eram exigidas

notas boas em diversas matérias, e este foi um dos principais desafios que enfrentei quando comecei a estagiar. Tudo no estágio era novidade e, às vezes, exigia muita dedicação fora do horário de trabalho, mesmo com o horário flexível disponibilizado pela Engine, não tinha muitas horas para me dedicar à faculdade e tive que me dedicar nos finais de semana para os estudos e para fazer os trabalhos (Eps). A administração do meu tempo livre não era mais um privilégio, mas sim uma necessidade para conseguir conciliar o estudo e o trabalho.

No estágio, fiquei as primeiras semanas em treinamento para aprender a programar em Delphi, e logo depois já fui envolvido no projeto descrito nesta monografia. Enfrentei diversos obstáculos, entre eles, o de lidar com clientes e trabalhar sob pressão. No começo não tinha muito “jogo de cintura” para dar suporte aos clientes e até mesmo em visitas realizadas na empresa contratante, acabava falando coisas que não deveria e depois em reuniões com a equipe recebia dicas de como me portar. Acho que isso foi muito importante para meu crescimento profissional e pessoal. Nas primeiras visitas fui acompanhado, mas nas demais, devido à indisponibilidade dos demais funcionários da Engine, fui sozinho e comecei a aprender a lidar com as diversas situações encontradas. Muitas vezes, o responsável da empresa René, me ligava falando que tinha dado alguns problemas no sistema e eu tinha que resolver o problema, ou por telefone ou diretamente na empresa (caso não detectasse o erro via telefone). Como as operações dessa empresa já dependiam do nosso sistema, a pressão era muito grande para corrigir o erro, pois se corria o risco da empresa ter certos prejuízos financeiros. Mas conforme o tempo foi passando, aprendi a lidar melhor com essas situações.

Uma das frustrações que tive foi justamente essa falta de tempo para me dedicar à faculdade e a minha vida pessoal. Em diversas matérias gostaria de ter me aprofundado mais em diversos temas, mas como apareciam outras pendências com maior prioridade, isso acaba ficando de lado.

4.3 DISCIPLINAS CURSADAS NO BCC MAIS RELEVANTES

O projeto no qual fiz parte teve início em 2001 (ano que comecei a fazer algumas matérias do curso de Bacharelado em Ciência da Computação), e durante o seu período de desenvolvimento, eu cursei muitas disciplinas que foram de extrema importância para o meu estágio. Dentre as mais importantes gostaria de destacar:

↪ **MAC0110 Introdução à Computação / MAC0122 Princípios de Desenvolvimento de Algoritmos**

Essas matérias me forneceram uma base imprescindível para a programação, com conceitos e técnicas básicas que foram úteis em todo o curso e no estágio também.

↪ **MAT0138 Álgebra I para Computação / MAC0239 Métodos Formais em Programação**

Essas matérias podem parecer pouco úteis no dia a dia, mas com certeza desenvolvem muito o raciocínio dos alunos.

↳ **MAC0426 Sistemas de Bancos de Dados**

Como o projeto desenvolvido envolvia banco de dados relacional, foi muito importante aprender a teoria e os conceitos de banco de dados e um pouco da prática de modelagem e de linguagem SQL.

↳ **MAC0332 Engenharia de Software**

Engenharia de Software foi extremamente importante na minha formação acadêmica, pois aprendi a importância da documentação dos sistemas e principalmente do levantamento de requisitos de um projeto, conhecer os processos de desenvolvimento e como produzir software de qualidade.

4.4 MEMBROS DA EQUIPE

Os meus mentores no projeto foram a Adriana Reigadas (parte técnica) e Fernando Fasti (parte administrativa). No começo do projeto eu não tinha muito conhecimento técnico para as atividades que tinha que desenvolver, então sempre trabalhava em conjunto com a Adriana e na medida que fui conhecendo mais a linguagem e as ferramentas utilizadas, comecei a desenvolver grande parte do sistema sozinho. O mesmo aconteceu com as implantações e treinamentos do sistema, nas primeiras vezes eu fui acompanhado, mas nas demais eu ia sozinho, ganhando uma responsabilidade muito grande já que estava representando a Engine e as vezes tinha que tomar algumas decisões sozinho.

A Engine tinha mais projetos em paralelo ao citado no trabalho e muitas vezes eu ficava alguns dias sem encontrar com o Fernando ou com a Adriana no escritório, mas sempre que era necessário, nos falávamos por telefone e resolvíamos o problema pendente. Como o Fernando era o gerente do trabalho, ele se preocupava muito com o cronograma do projeto, quando ocorriam eventuais atrasos sempre conversávamos para tentar ganhar tempo em outras coisas voltar a ficar em dia. Isso algumas vezes dava certo, mas em algumas tínhamos que negociar com o cliente outras datas.

4.5 CONSIDERAÇÕES FINAIS

O projeto descrito neste trabalho já foi encerrado e entregue ao cliente, o sistema final esta funcionando na Millenium desde 2002. Os seus resultados são/foram muito positivos para todos os participantes.

BIBLIOGRAFIA

- [1] - Site oficial da Engine Logística Ltda - www.enginelogistica.com.br;
- [2] - Site oficial da Borland – <http://www.borland.com>;
- [3] - PRESSMAN, R. S.. *Software Engineering*. McGraw-Hill, 2001.
- [4] - ANDREWS, G. R. *Foundations of Multithreaded, Parallel and Distributed Programming*, 2000.
- [5] - Alves, William Pereira. *Fundamentos de Banco de Dados*
- [6] - Rumbaugh, James. *Uml Guia do Usuário*.