



FUNDAÇÃO FACULDADES LUIZ MENEGHEL
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

João Adalberto Pinto

**DESENVOLVIMENTO DE SOFTWARE PARA
CONTROLE DE ESTOQUE PARA AUTO PEÇAS
UTILIZANDO O MODELEO INCREMENTAL**

Bandeirantes - PR
2006

FUNDAÇÃO FACULDADES LUIZ MENEGHEL

Instituição de ensino superior vinculada à Universidade Estadual do Norte do Paraná, em acordo com a Lei Estadual nº 15.300, de 28 de setembro de 2006.

João Adalberto Pinto

DESENVOLVIMENTO DE SOFTWARE PARA CONTROLE DE ESTOQUE PARA AUTO PEÇAS UTILIZANDO O MODELO INCREMENTAL

Trabalho de conclusão de curso submetido à Fundação Faculdades Luiz Meneghel como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. José Reinaldo Merlin

João Adalberto Pinto

**DESENVOLVIMENTO DE SOFTWARE PARA
CONTROLE DE ESTOQUE PARA AUTO PEÇAS
UTILIZANDO O MODELO INCREMENTAL**

COMISSÃO EXAMINADORA

Prof. Orientador José Reinaldo Merlin

Prof.(a) Ederson Marcos Sgarbi

Prof.(a) Carlos Eduardo Ribeiro

Bandeirantes, 10 de Novembro de 2006.

RESUMO

O mundo atual vem exigindo das empresas, novas tecnologias para manter-se competitiva no mercado. A empresa em estudo necessita atualizar-se para obter um melhor atendimento, e ter o controle sobre a entrada e saída de produtos no estoque.

O sistema a ser desenvolvido propõe informatizar todos os setores da empresa, embora em reuniões como o cliente definiu-se o Controle de Estoque como prioritário a ser informatizado. Com isso o software esta sendo desenvolvido utilizando o modelo Incremental e a Análise Essencial, pois a cada etapa do desenvolvimento o cliente tem parte que já pode ser implantado. Facilitando assim esclarecer novos requisitos para este e outros incrementos.

Este sistema será desenvolvido em ambiente *Delphi*, utilizando o banco de dados *FireBird*.

Palavras chaves: Controle de Estoque, Produto, *Delphi*

ABSTRACT

The current world comes demanding of the companies, new technologies to remain itself competitive in the market. The company in study needs to modernize itself to get one better attendance, and to have the control on the entrance and exit of products in the supply.

To be developed system considers to informatizar all the sectors of the company, even so in meetings as the customer defined the Control of Supply as with priority to be implementation. With this software this being developed using Incremental model i the Essential Analysis, therefore to each stage of the development the customer has part that already it can be implanted. Thus facilitating to clarify new requirements for this and other increments.

This system will be developed in *Delphi* environment, using the data base *FireBird*.

Key-Worlds: Control of Supply, Product, *Delphi*

SUMÁRIO

1. Introdução	08
2. Revisão de Literatura	10
2.1 O Processo de Software	10
2.1.1 Análise Especificação	10
2.1.2 Projeto e Implementação do Programas.....	11
2.1.3 Validação	11
2.1.4 Evolução	12
2.2 Modelos de Processo	12
2.2.1 Modelo Seqüencial Linear (Cascata)	12
2.2.2 Modelo de Prototipação	13
2.2.3 O Modelo Incremental.....	13
2.3 Análise Essencial	14
2.3.1 Modelo Ambiental	15
2.3.1.1 Descrição dos Objetos dos Sistemas	15
2.3.1.2 Lista de Eventos	15
2.3.1.3 Diagrama de Contexto	15
2.3.2 Modelo Comportamental.....	16
2.3.2.1 Diagrama de Fluxo de Dados.....	16
2.3.2.2 Diagrama de Entidade Relacionamento.....	16
2.3.2.3 Descrição de Processos	16
2.3.2.4 Dicionário de Dados	17
3. Objetivo	18
3.1 Objetivos Específicos	18
4. Justificativa.....	19
5. Materiais e Métodos.....	20
6. Desenvolvimento.....	21
6.1 Requisitos.....	21
6.2 Modelo Ambiental.....	22
6.2.1 Declaração dos Objetivos do Sistemas	22

6.2.2 Diagrama de Contexto	23
6.2.3 Lista de Eventos	24
6.3 Modelo Comportamental.....	25
6.3.1 Diagrama de Fluxo de Dados	25
6.3.2 Descrição de Processos	27
6.3.3 Dicionários de Dados	30
7. Considerações Finais	32
8. Referências Bibliográficas	33
Anexo A.....	34

1 – Introdução

O mundo atual, cada vez mais competitivo, vem exigindo das empresas novas posturas administrativas. A informatização de empresas é uma necessidade neste cenário.

Com isso a empresa necessita de sistemas de software para se manter competitivo no mercado. Tais sistemas permitem, entre outras coisas, o controle da entrada e saída de produtos, aumentando assim sua eficiência na administração de estoques.

A informatização da empresa, aliada a outros fatores, dá a ela condições de manter-se equiparada aos concorrentes, agilizando o atendimento aos clientes, os pedidos aos fornecedores e a localização física de itens no estoque.

O desenvolvimento de sistemas, no entanto, não é um processo simples, pois muitos problemas podem afetar a qualidade do software desenvolvido. Alguns modelos de processo de software foram desenvolvidos, tais como Cascata, Prototipação, Espiral e Incremental visando diminuir os riscos e incertezas do processo. Também existem metodologias para tratar o processo de software, entre elas, a Análise Essencial, *Rup*, *Uml* entre outras.

Este trabalho propõe o desenvolvimento de um software utilizando o modelo Incremental e a Análise Essencial. Se propõe a informatizar todos os setores da empresa, mas devido as condições e a pouca quantidade de tempo oferecidas pelo cliente, em reuniões com clientes e desenvolvedores tomou-se a decisão de se informatiza-la de maneira incremental, pois a cada etapa do desenvolvimento o cliente já terá em mãos parte do sistema que podem ser implantados.

Com isso chegou-se a conclusão de que o Sistema de Controle de Estoque é um dos principais fatores de urgência da empresa, pois melhorará na rapidez ao atendimento e controle de todos os produtos em estoque. Ao decorrer do desenvolvimento surgirão novos implementos como:

Cadastro de Clientes

Cadastro de Fornecedores

Contas a Pagar

Contas a Receber

Fluxo de Caixa

2 – Revisão Literatura

2.1 – O Processo Software

Processo de software é o nome dado ao conjunto de atividades envolvidas na produção de software. Os desenvolvedores podem contar com o auxílio de ferramentas CASE (*Computer Aided Systems Engineering* ou Engenharia de Software Auxiliada por Computador) para auxiliar no processo, porém estas atividades ainda são de difícil automação.

Não existe um processo certo para se produzir software, porém existem alguns modelos que foram desenvolvidos ao longo do tempo e que são úteis para atividade (Sommerville, 2003).

Qualquer que seja o modelo de processo utilizado, algumas atividades são fundamentais para o desenvolvimento de software:

- 1 Especificação do software
- 2 Projeto e implementação
- 3 Validação do software
- 4 Evolução do software

2.1.1 – Análise Especificação

A Especificação de Software, também chamada de análise de requisitos ou engenharia de requisitos, é um mecanismo para entender o que o cliente deseja analisar as necessidades, avaliar a exeqüibilidade, negociar uma solução razoável, especificar a solução de maneira não-ambígua. Compreender a natureza dos problemas e estabelecer uma exatidão o que o sistema deve fazer pode ser muito difícil (Sommerville, 2003).

Há vários problemas que dificultam os levantamentos de requisitos, entre eles, problemas de escopo, entendimento e volatilidade.

Os requisitos são classificados como funcionais e não funcionais (Sommerville, 2003). Os funcionais são as declarações do que o sistema deve fazer como deve reagir a entradas específicas e como se comportar em determinadas situações. Os não-funcionais são restrições sobre serviços ou funções oferecidas pelo sistema tais como tempo, consumo médio de memória, entre outros.

2.1.2 – Projeto e implementação de programas

Projeto de software é uma descrição de estrutura de software a ser implementada, dos dados que são parte do sistema, das interfaces entre os componentes e algumas vezes dos algoritmos utilizados (Sommerville, 2003).

Um projeto de qualidade deve ter características como: facilidade de atendimento, facilidade de implantação, de realização de teste e facilidade de modificações das especificações dos requisitos (Pfleeger, 2004).

O estágio de implementação do software é o processo de conversão de uma especificação em um sistema executável.

2.1.3 – Validação

A validação de software destina-se a mostrar que o sistema está de acordo com suas especificações e atende as expectativas do cliente. Esta atividade também é chamada genericamente de teste.

Os grandes sistemas não devem ser testados como uma unidade isolada, são construído a partir de subsistemas, que são construídos a partir de módulos, que são compostos de procedimentos e funções. Devem ser feitos, assim, teste de unidade isoladamente, teste de integração e teste do sistema para que se confirme se o software combina adequadamente com os outros elementos do sistema.

2.1.4 – Evolução

O processo de desenvolvimento de software era visto como encerrado no momento em que era posto em operação. No entanto, uma das características de software é a sua natureza evolucionária.

Evolução ou manutenção é o processo de modificar o sistema depois que ele entrou em operação. Modificações são necessárias para correções (erros são descobertos), adição de funcionalidades (surgem novos requisitos ou os requisitos se modificam) e adaptações a um ambiente operacional diferente (mudança do hardware, sistema operacional, arquitetura).

A manutenção é uma atividade cara, geralmente supera os custos de desenvolvimento. Por isso, procedimentos adotados durante o desenvolvimento, tais como uma boa documentação, pode torná-la mais fácil e barata.

2.2 – Modelos de processo

No desenvolvimento de um sistema é importante percorrer uma série de passos previsíveis, um roteiro que ajuda a criar a tempo um resultado de alta qualidade. Com isso o pessoal que solicitou o software desempenha um papel em relação a esse processo, pois fornece estabilidade, controle e organização.

Diferentes modelos de processos foram desenvolvidos, cada um representa uma tentativa de trazer ordem par uma atividade inerentemente caótica.

2.2.1 – Modelo Seqüencial Linear (cascata)

Sugere uma abordagem seqüencial que começa com a especificação dos requisitos e progride através do projeto, codificação e teste. Sua principal característica

é que cada fase só começa quando a anterior estiver totalmente concluída, tornando difícil para o cliente estabelecerem todos os requisitos do sistema, pois ao decorrer do projeto muitas incertezas surgirão, o mesmo não terá nenhuma versão disponível do programa até que esteja totalmente terminado. Um erro grosseiro pode ser desastroso futuramente até que o programa executado seja revisto.

2.2.2 – Modelo de Prototipação

Tem como base a idéia de desenvolver uma implementação inicial e submetê-la à avaliação do cliente. O cliente define um conjunto de objetivos gerais para o software, mas não identifica detalhadamente requisitos de entrada, processamento e saída.

Um protótipo é desenvolvido para se aprender mais sobre o problema e se ter uma base de uma parte ou de todo software a ser oferecido. Com isso existe a vantagem de um rápido fornecimento do sistema, utilizando a participação do usuário para que o sistema tenha maior probabilidade de atender a seus requisitos.

Porém a prototipagem possui desvantagens, como problemas de gerenciamento, é difícil planejar o custo e prazo de desenvolvimento; problemas de eficiência, nenhum cuidado com a seleção da linguagem de programação é tomada e algoritmos ineficientes podem ser implementados, e problemas de manutenção. Mudanças contínuas fazem com que o software fique com uma estrutura corrompida dificultando a manutenção posterior, especialmente se o responsável pelo software não pertenceu ao grupo de desenvolvimento.

2.2.3 – O Modelo Incremental

O modelo incremental aplica seqüências lineares de uma forma racional a medida que o tempo passa. Cada seqüência produz um incremento factível do software, ou seja, uma versão executável com um grupo de funcionalidades.

Em um processo de desenvolvimento incremental os clientes identificam, em um esboço, as funções que serem fornecidas pelo sistema. Em seguida é definida uma série de estágios de entregas, fornecendo um subconjunto das funcionalidades do sistema. A locação de funções ao estágio depende da prioridade da função. As questões prioritárias são entregues primeiramente ao cliente (Sommerville, 2003).

Uma vez que incremento é desenvolvido e entregue, o cliente pode colocá-lo em operação. Isto significa que recebe com antecedência parte da funcionalidade do sistema o que lhes facilita esclarecer requisitos para os incrementos subseqüentes ao estágio atual.

A vantagem desse desenvolvimento está em que o cliente não precisa esperar até que todo sistema esteja pronto para começar a utilizá-lo, podem utilizar o primeiro incremento para avaliar suas próprias necessidades para estágios posteriores. Existe, desta forma, um risco menor de fracasso completo do sistema.

A dificuldade que pode existir no modelo incremental é saber definir quais são as funcionalidades essenciais e prioritárias do sistema, de modo que as demais possam se adaptar a elas sem problemas.

2.3 – Análise Essencial

A atividade básica do desenvolvimento de um sistema é a análise de requisitos. É preciso analisar e entender o que deve ser feito para, em seguida, projetar a solução adequada. Nos últimos anos, diversas metodologias de análise e projeto foram propostas, entre elas, a Análise Essencial, que será adotada no desenvolvimento deste trabalho. O processo de modelagem do sistema leva a uma representação abstrata do software a ser desenvolvido.

A modelagem de análise usa uma combinação de formas textuais e diagramáticas para mostrar as funções, os dados e o comportamento do sistema, de modo que seja relativamente fácil de entender e fácil de revisar (Pressman, 2000).

Segundo McMenamin e Palmer (1991), cabe ao analista desenvolver uma especificação do sistema que contenha todos os registros verdadeiros e nada mais que isso. Requisitos verdadeiros são características do sistema a ser implementado, e um requisito pode ser falso se o sistema pode realizar todas as funções sem ele.

A análise essencial utiliza dois tipos de modelos: ambiental e comportamental.

2.3.1 Modelo Ambiental

No modelo ambiental é mostrada a interação do sistema com o ambiente que o cerca. A modelagem ambiental utiliza-se basicamente de três ferramentas:

2.3.1.1 Descrição dos objetivos do sistema

É uma declaração textual do sistema que mostra o que ele faz. Deve ser elaboradas em poucas frases, simples e precisas, em linguagem distribuída de jargões técnico, de modo que possa ser entendidas por usuários, clientes e outros.

2.3.1.2 Lista de eventos

Um evento é um acontecimento do ambiente que obriga o sistema a reagir. Para um dado estímulo, o sistema produz, uma resposta. A análise essencial propõe o particionamento por evento.

2.3.1.3 Diagrama de Contexto

O diagrama de contexto é a representação diagramática do sistema e seu relacionamento com as entidades externas que interagem com ele. O sistema é representado como uma caixa preta, ou seja, não se sabe o que acontece dentro dele, apenas se conhecem as entradas e saídas e as entidades externas que interagem com o sistema. As entidades devem ser nomeadas de acordo com o papel que elas representam.

2.3.2. Modelo Comportamental

O modelo comportamental é definido do ponto de vista interno. Descreve a maneira como o sistema reage, internamente, aos estímulos do mundo exterior. As ferramentas do Modelo Comportamental são descritas a seguir.

2.3.2.1 Diagrama de fluxo de dados

É uma representação gráfica do comportamento interno do sistema. Para construção dos DFD, o sistema é particionado em eventos e cada evento é representado no diagrama com as entidades externas, fluxo de dados processos e memórias associados ao evento.

2.3.2.2 Diagrama de entidade Relacionamento

São os produtos da modelagem de dados do sistema. Representam as entidades presentes no sistema e o relacionamento entre elas.

2.3.2.3 Descrição de processos

A especificação de processos é a descrição de como os fluxos de dados de entrada são transformados em fluxos de dados de saída. A linguagem natural não é a melhor forma de descrever os processos, podem dar margem a interpretação diversas da pretendida. Há técnicas por especificação, como português estruturado, pseudocódigo, tabela de decisão, árvore de decisão, entre outros.

2.3.2.4 Dicionário de Dados

É um repositório de informações sobre os comportamentos do sistema. Embora todas estas informações estejam registradas no diagrama, nada se sabe sobre os dados que compõem os fluxos ou então armazenados no depósito, estas informações estão registradas no dicionário de dados.

3. Objetivo

O objetivo deste trabalho é informatizar uma loja de autopeças utilizando o modelo incremental. Serão analisadas as necessidades da empresa. Neste presente trabalho a funcionalidade apresentada será o controle de estoque, com finalidade de mostrar a localização exata da peça e sua quantidade em estoque.

3.1 Objetivos Específicos

- Controlar a entrada e saída de peças.
- Agilizar a busca de peças.
- Melhorar o atendimento ao cliente.

4. Justificativa

O presente trabalho justifica-se pela necessidade atual de informatização de qualquer empresa, em especial da empresa escolhida, que atualmente não utiliza nenhum processo informatizado.

Dentre as vantagens da utilização de um sistema informatizado, especialmente para controle de estoque, está a agilidade no controle da quantidade de mercadorias em estoque, bem como a facilidade de localização de um item no acervo.

O modelo incremental está sendo escolhido devido à dimensão do sistema a ser desenvolvido e o tempo disponível. Esta abordagem permite a entrega de uma primeira versão do sistema em um tempo relativamente curto e posteriores incrementos.

5. Materiais e Métodos

O software proposto será desenvolvido usando-se análise essencial. Será utilizado o ambiente *Delphi* com banco de dados *FireBird*

Vantagens de usar o *Delphi*.

- 1) Gera executáveis compilados, sem a necessidade de distribuir bibliotecas.
- 2) Multi-Plataforma usando a biblioteca CLX (nome dado pela *Borland* para suas bibliotecas de desenvolvimento *Cross Platform*, basicamente as bibliotecas da *Qt Lib's* disponíveis no *Delphi* e no *Kylix*), podendo ser recompilado no *Linux*.
- 3) Ampla ajuda oferecida na Internet, com vários usuários trocando informações.
- 4) Vários livros publicados sobre a linguagem.
- 5) Conectividade extremamente fácil com banco de dados.

Vantagens de usar *FireBird*

- 1) Open-Source, nenhuma empresa controlando o produto, qualquer um pode modificá-lo.
- 2) Multi-Plataforma, roda nativamente em mais de 15 plataformas, entre elas *Windows*, *Linux* e *MacOS*.
- 3) Integração com *Delphi*.
- 4) Oferece recursos exclusivos, como trigger's e procedures e campos calculados.
- 5) Possível geração de *UDF's (User Define Functions)*, funções criadas pelo usuário para aumentar a funcionalidade do software.

6. Desenvolvimento

Para o desenvolvimento do trabalho, definiu-se quais as funcionalidades deveriam ser implementadas inicialmente, para o sistema de controle de estoques. A seguir, os requisitos deste sistema foram levantados e são apresentados a seguir.

6.1 Requisitos

1. **Controle de produtos e estoque:** para gerenciamento de produtos vendidos e controle de seu estoque, devem ser implementadas as funcionalidades a seguir.
 - 1.1. **Incluir produto:** cadastramento de produto (mercadoria) vendido pelo estabelecimento.
 - 1.2. **Alterar produto:** os dados de um produto podem ser alterados, como quando ocorre mudança no preço de venda.
 - 1.3. **Excluir produto:** um produto pode ser excluído quando, por exemplo, ele deixar de ser comercializado pelo estabelecimento.
 - 1.4. **Consultar produto:** os dados de um produto devem poder ser consultados a qualquer momento. Deve ser possível buscar um produto pelo código ou pelo nome.
 - 1.5. **Registrar entrada no estoque:** toda vez que a empresa receba remessas de mercadorias de fornecedor, a quantidade de produtos deve ser adicionada ao estoque.
 - 1.6. **Registrar saída do estoque:** toda vez que ocorrer uma saída de mercadoria (venda, devolução), a quantidade respectiva deve ser subtraída do estoque.
 - 1.7. **Emitir relatório de produtos:** o sistema deve possibilitar a emissão de relatórios de produtos, tais como: produtos classificados por código, por nome, por fornecedor.

1.8. **Efetuar acerto:** Os dados dos produtos em estoque, podem ser acertados devido a divergências existentes e quantidades.

6.2 Modelo Ambiental

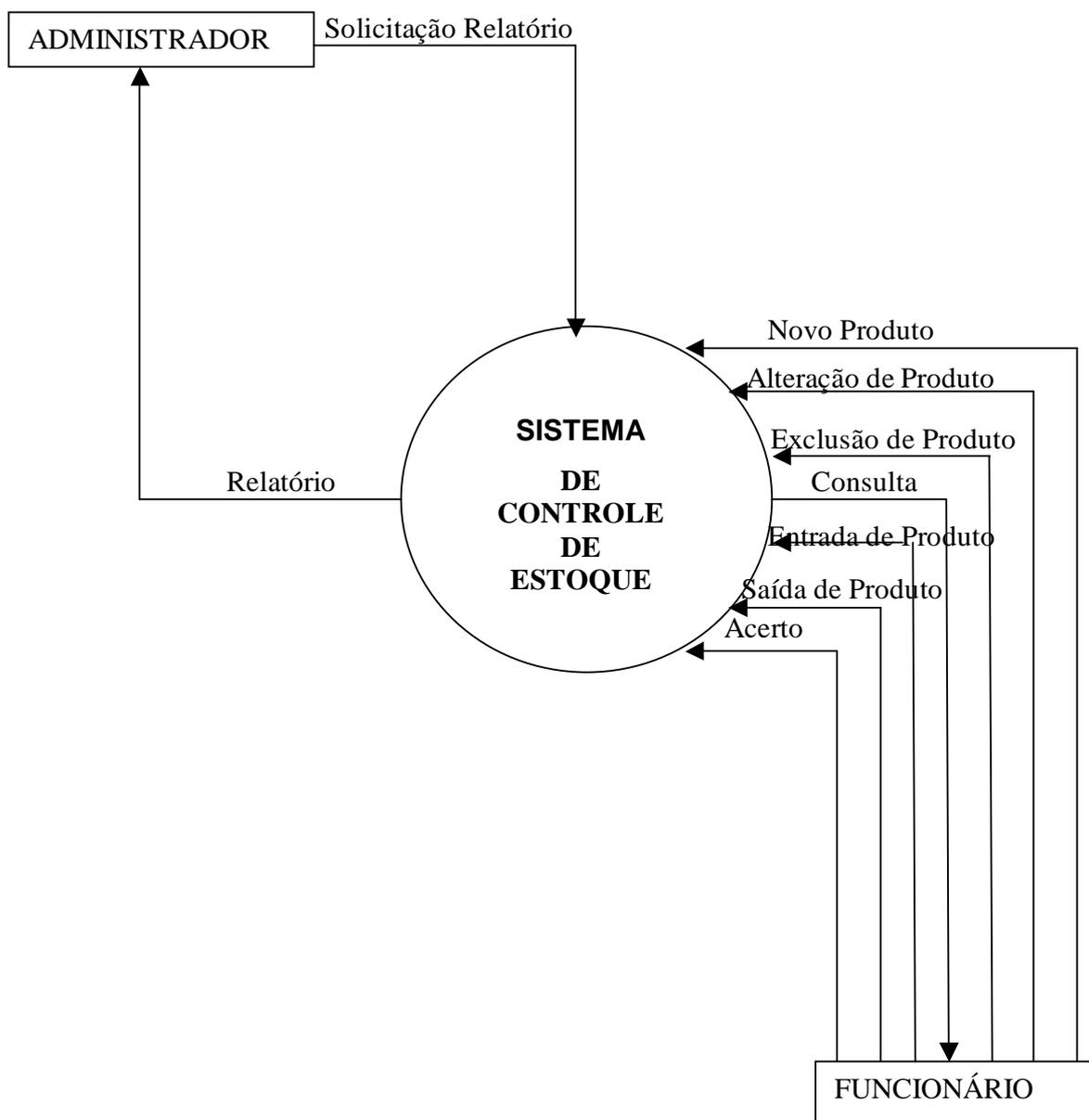
6.2.1 Declaração dos Objetivos do Sistema

O software para controle de estoque para autopeças tem como objetivo, satisfazer as necessidades do cliente, e obter um melhor rendimento dos funcionários.

Objetivos específicos

1. Registrar a entrada e saída de produtos da empresa
2. Tornar mais rápido o atendimento aos clientes
3. Ter controle sobre todas as negociações da empresa
4. Permitir a rápida localização de um produto no estoque, mesmo para novos funcionários.

6.2.2 Diagrama de Contexto



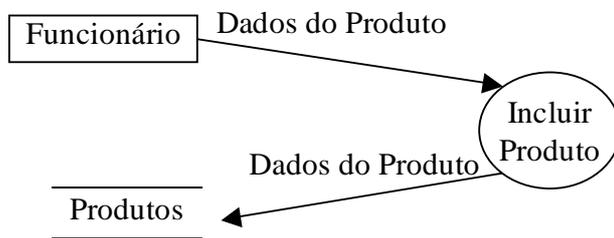
6.2.3 Lista de Eventos

Nº	Evento	Descrição	Estimulo	Ação	Resposta
1	Funcionário Inclui Produto	Funcionário Cadastra os produtos vendidos no estabelecimento	Novo Produto	Incluir Produto	
2	Funcionário Altera Produto	Funcionário Altera dados do produto	Alteração de Produto	Alterar Produto	
3	Funcionário Exclui Produto	Exclui produtos não existentes no estoque	Exclusão Produto	Excluir Produto	
4	Funcionário Consulta Produtos	Consultar Produtos existentes no estoque	Consulta	Consultar Produto	
5	Funcionário Solicita Registrar Entrada Dados Estoques	Entrada Produtos do Estoque	Entrada Produto	Registrar Entrada Produto	
6	Funcionário Registra Saída Dados Estoques	Saída Produtos do Estoque	Saída de Produto	Registra Saída do Produto	
7	Administrador Solicita Relatório de Produtos	Mostra na Tela para possível Impressão	Solicitação do Relatório	Emitir Relatório de Produtos	Relatório de Produtos
8	Funcionário Efetua Acerto de Estoque	Quando há constatação de erro na quantidade, funcionário realiza acerto	Acerto	Acertar estoque	

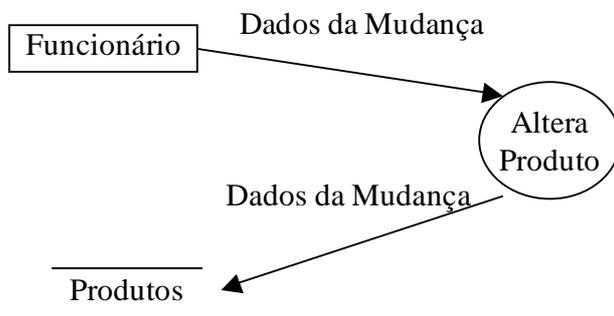
6.3 Modelo Comportamental

6.3.1 Diagrama de Fluxo de Dados

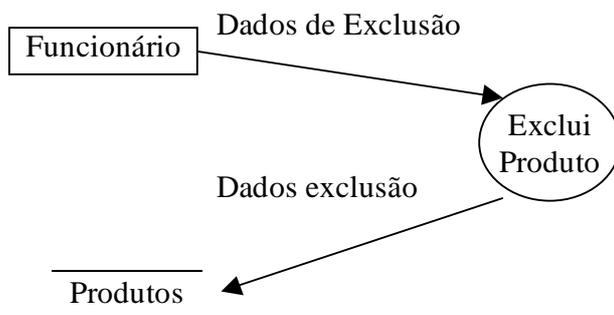
1. INCLUIR PRODUTOS



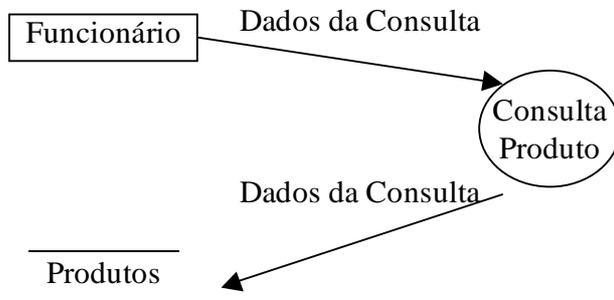
2. ALTERAR PRODUTOS



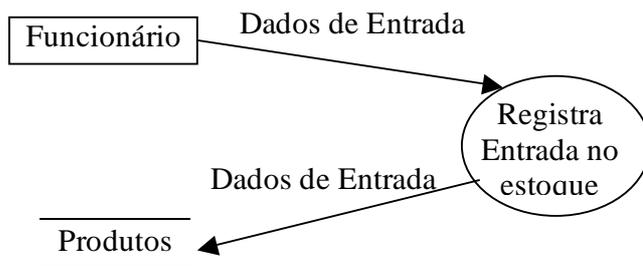
3. EXCLUIR PRODUTOS



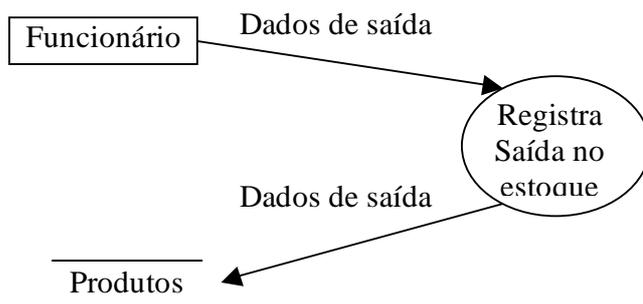
4. CONSULTAR PRODUTOS



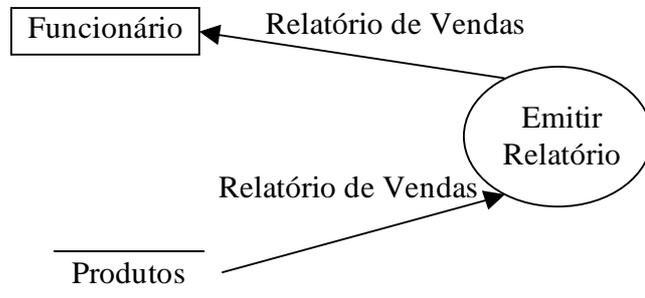
5. REGISTRAR ENTRADA NO ESTOQUE



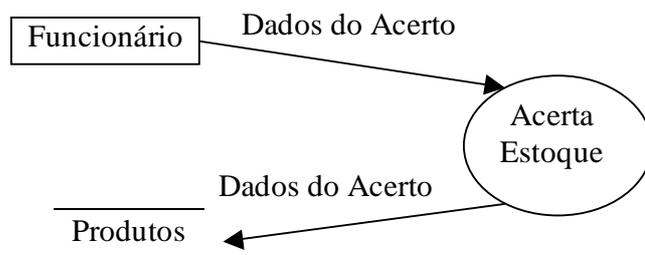
6. REGISTRAR SAÍDA NO ESTOQUE



7. EMITIR RELATÓRIOS



8. ACERTAR ESTOQUE



6.3.2 Descrição de Processos

Descrição do Processo:

Processo nº 1 – Cadastra Produto

Função: Incluir no Sistema um novo produto

Fluxo de entrada: Dados do Produto

Resumo Lógico:

Início

Ler Dados do Produto

Buscar se existe produto cadastrado com o código informado em Produtos

Se não existir

Armazenar Dados do produto em produtos

Senão

Recusar o cadastro

Fim.

Processo nº 2- Alterar Produto

Função: Alterar Dados de Produtos já existentes

Fluxo de entrada: Dados da Mudança

Resumo Lógico:**Início**

Ler Dados da Mudança

Buscar o Produto Cadastrado com o código informado em produtos

Se código existe então

Início

Inserir mudanças ao produto

Fim**Processo nº 3- Excluir Produtos**

Função: Excluir Produtos Inválidos no Sistema

Fluxo de entrada: Dados do Produto

Resumo Lógico:**Início**

Ler Dados do Produto

Buscar se existe produto cadastrado com o código informado em Produtos

Se código existir

Início

Excluir Produto

Senão

Exibe "Código não válido"

Fim**Fim****Processo nº 4 – Consultar Produtos**

Função: Consulta Produtos existentes no Sistema

Fluxo de entrada: Dados da Consulta

Resumo Lógico:**Início**

Ler Dados do Produto

Consulta Lista de Dados Existentes na Tabela Produtos

Se dados existentes então

Lista Dados do Estoque

Fim**Processo nº 5 – Registrar Entrada no Estoque**

Função: Listar os dados de Entrada no Estoque

Fluxo de entrada: Dados da Entrada

Resumo Lógico:

Início

Ler Dados da Entrada
 Lista os dados de Entrada na tabela produto
Se dados válidos então
Início
 Registra entrada de Dados
 Somar a Quantidade
 Listar nova quantidade

Fim

Processo nº 6 – Registrar Saída no Estoque

Função: Listar os dados de Entrada no Estoque

Fluxo de entrada: Dados de Saída

Resumo Lógico:

Início

Ler Dados de Saída
 Lista os dados de Saída na tabela produto
Se dados válidos então
Início
 Registra Dados de Saída
 Diminuir a Quantidade
 Listar a nova Quantidade

Fim

Processo nº 7 – Emitir Relatórios

Função:

Fluxo de entrada:

Resumo Lógico:

Início

Ler Relatório
 Emite Relatórios
Se dados válidos então
 Emite Relatório

Fim

Processo nº 8 – Acertar Estoque

Função: Autorizar a quantidade de estoque

Fluxo de entrada: Dados Acerto

Resumo Lógico:

Início

Ler Dados Acerto
 Lista os dados Acerto estoque

Se dados válidos então
Atualiza os dados do estoque

Fim

6.3.3 Dicionário de Dados

1. DADOS DO PRODUTO

CÓDIGO PRODUTO	NUM (4)
CÓDIGO PEÇA	NUM (5)
DESCRIÇÃO	CHAR (35)
QUANTIDADE	NUM (3)
FORNECEDOR	CHAR (20)
NÚMERO DA NOTA FISCAL	NUM (6)
PREÇO DE CUSTO	DECIMAL (5,2)
PREÇO DE VENDA	DECIMAL (5,2)
CORREDOR	NUM (2)
PLATELEIRA	NUM (2)

2. DADOS DE MUDANÇA

CÓDIGO DO PRODUTO	NUM (4)
CÓDIGO DA PEÇA	NUM (5)
DESCRIÇÃO	CHAR (35)
QUANTIDADE	NUM (2)
PREÇO CUSTO	DECIMAL (5,2)
PREÇO VENDA	DECIMAL (5,2)

3. EXCLUIR

CÓDIGO PRODUTO	NUM (4)
CÓDIGO PEÇA	NUM (5)

4. DADOS DE CONSULTA

PREÇO CUSTO	DECIMAL (5,2)
PREÇO VENDA	DECIMAL (5,2)
QUANTIDADE	NUM (3)
FORNECEDOR	CHAR (20)

5. DADOS ENTRADA

CÓDIGO PRODUTO	NUM (4)
QUANTIDADE	NUM (3)

6. DADOS SAÍDA

CÓDIGO PRODUTO	NUM (4)
QUANTIDADE	NUM (3)

7. RELATÓRIO

DATA	DATE
DESCRIÇÃO	CHAR (35)
PREÇO VENDA	DECIMAL (5,2)
PREÇO CUSTO	DECIMAL (5,2)

8. ACERTO

CÓDIGO PRODUTO	NUM (5)
QUANTIDADE	NUM (3)
DATA	DATE

7. Considerações Finais

A implantação de um sistema de controle de estoque tem como objetivo controlar a entrada e saída de produtos vendidos, manter a quantidade destes produtos atualizada no estoque e permitir conhecer a localização física de cada produto nas prateleiras. Com isto é possível melhorar o fluxo de trabalho e melhor atender os clientes da empresa, pois os controles são precisos e o atendimento mais rápido. Destaca-se a rapidez no aprendizado para os novos funcionários, com a utilização do novo sistema.

O desenvolvimento de software, no entanto, é uma atividade complexa e demorada. O modelo incremental mostrou-se adequado, pois o controle de estoque já pode ser implantado e utilizado enquanto as outras partes do sistema são desenvolvidas.

8. Referências Bibliográficas

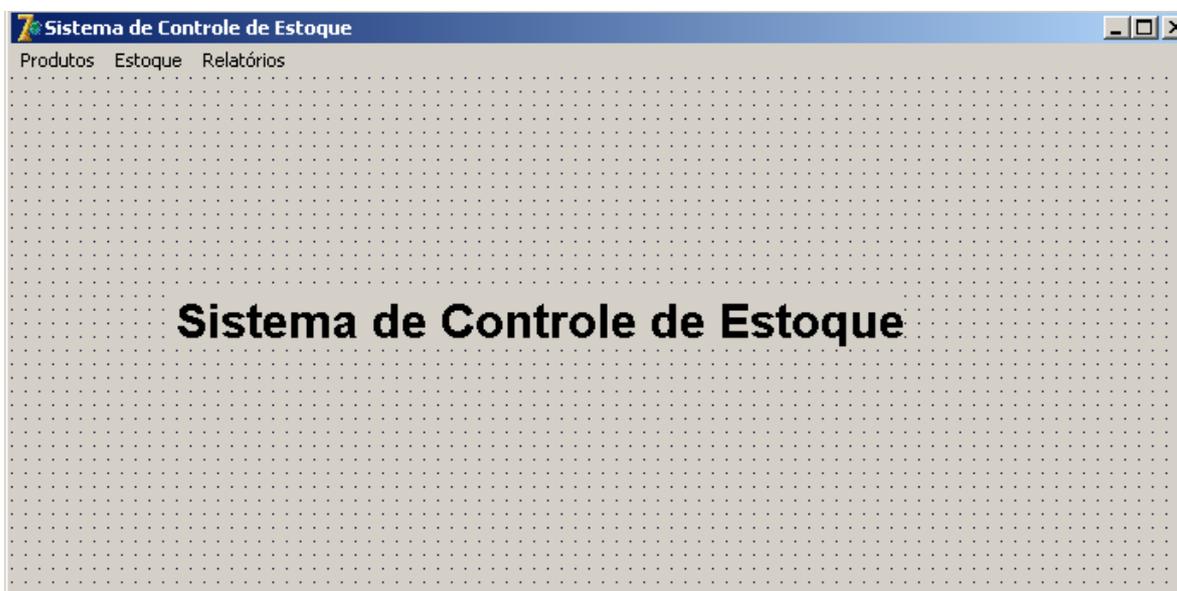
SOMMERVILLE, I. *Engenharia de Software*. Addison Wesley, 2003.

PRESSMAN, R. S. *Software Engineering – A Practitioner’s Approach*. McGraw-Hill, 5 edition, 2000.

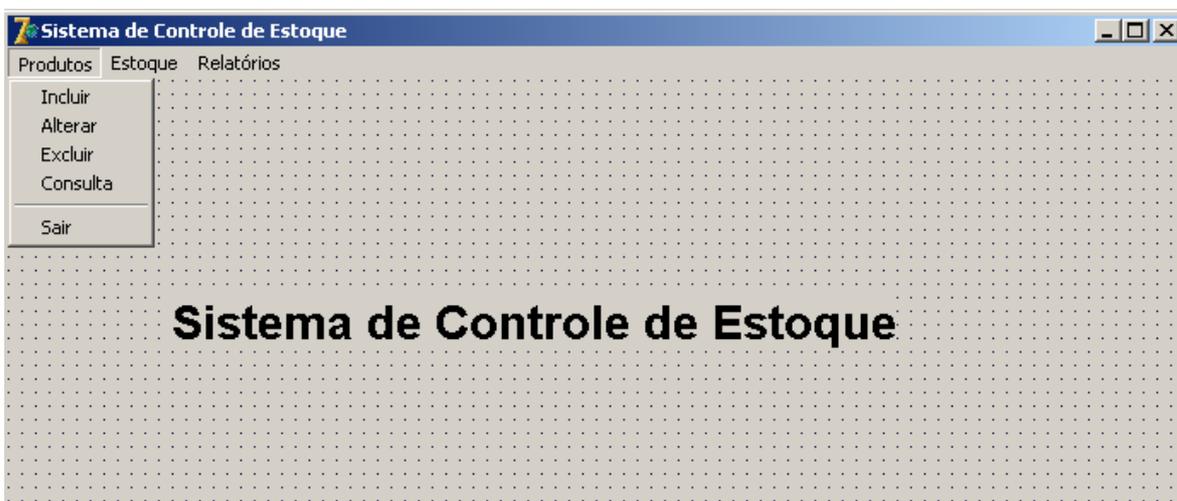
PFLEEGER, S. L. *Engenharia de Software – Teoria e Prática*. Prentice Hall, São Paulo, 2004.

McMENMIN, S. e Palmer, J. F. *Análise Essencial de Sistemas*. McGraw-Hill, 1991.

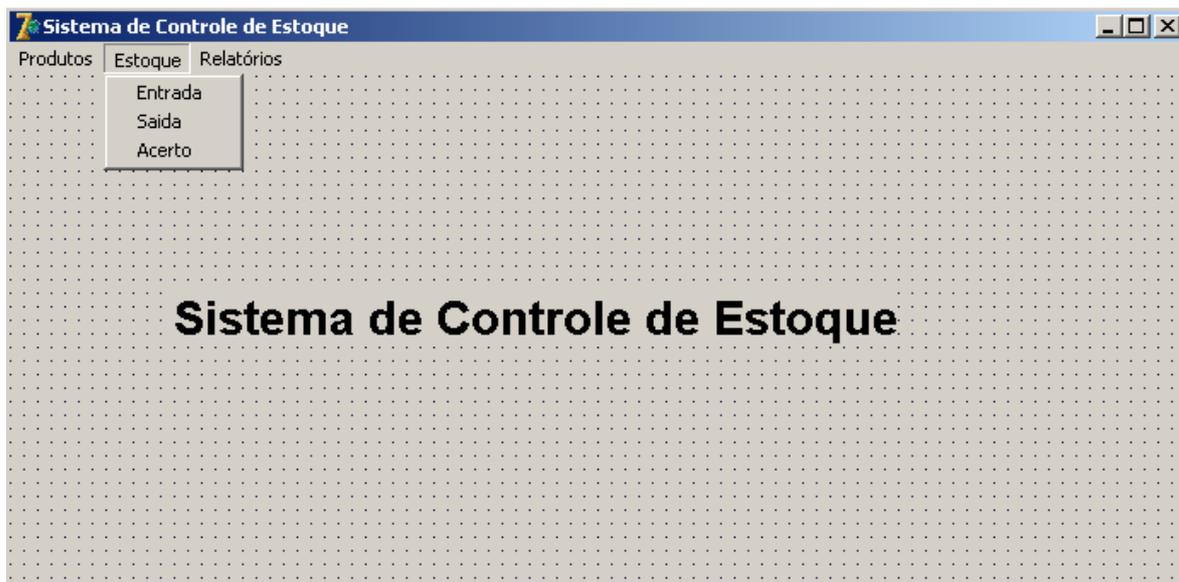
Anexo A - Telas do Sistema



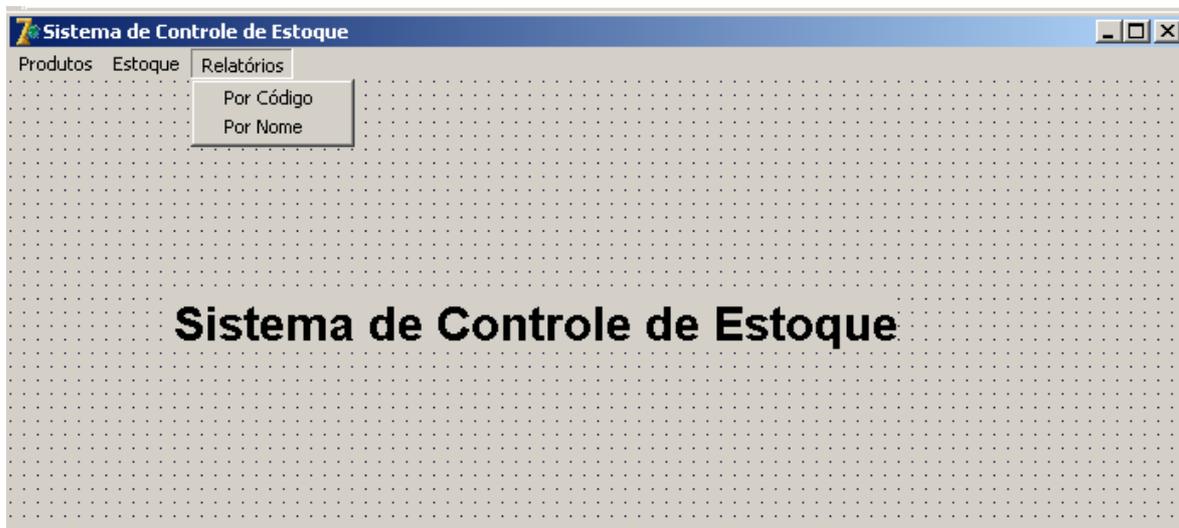
Tela Inicial do Sistema.



Esta tela inicial com o Menu Produto aberto.



Tela Inicial com o Menu Estoque aberto.



Tela Inicial com o Menu Relatórios aberto.

Incluir Produtos

Código: Cód. Peça: Descrição:

QTD.: Preço Custo: Preço Venda:

FORNECEDOR: CORREDOR: PRATELEIRA:

NNOTA:

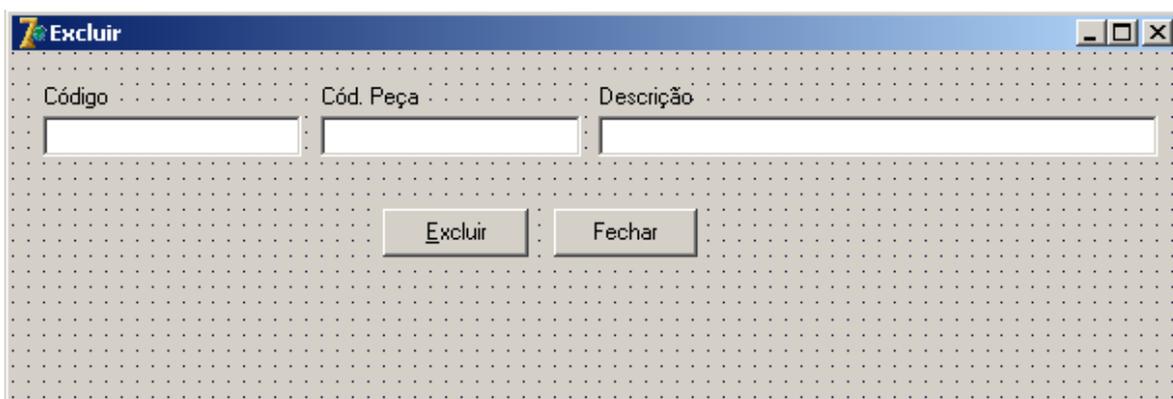
Tela Incluir Produtos, inclui os produtos no estoque.

Alterar

Código: Cód. Peça: Descrição:

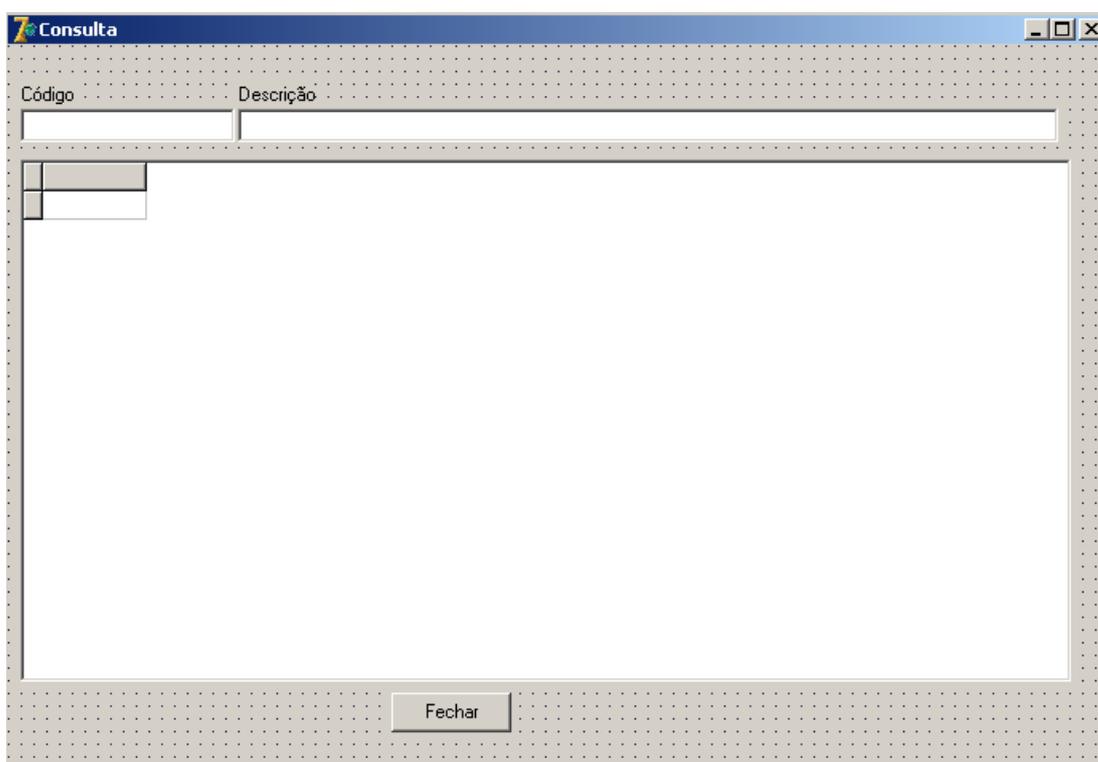
Preço Custo: Preço Venda:

Tela Alterar, altera os dados do produto.



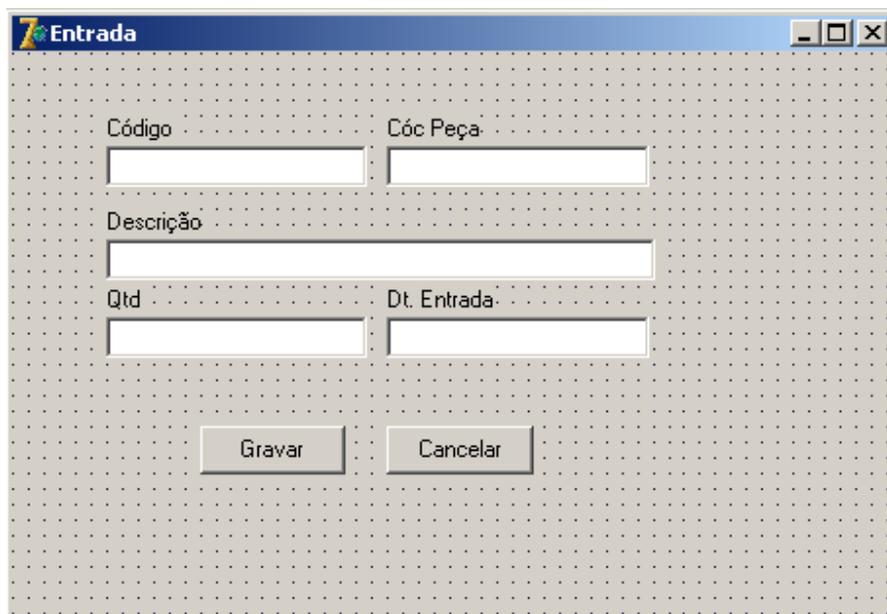
The screenshot shows a window titled "Excluir" with a blue header bar. The window has a dotted background. It contains three input fields: "Código", "Cód. Peça", and "Descrição". Below these fields are two buttons: "Excluir" and "Fechar".

Tela Excluir, exclui um produto cadastrado e seus dados nele contido.



The screenshot shows a window titled "Consulta" with a blue header bar. The window has a dotted background. It contains two input fields: "Código" and "Descrição". Below these fields is a large empty rectangular area, likely a table or list. At the bottom right of the window is a "Fechar" button.

Tela Consulta, Consulta os dados do produto existentes no estoque.



The image shows a software window titled "Entrada" with a blue header bar. The window contains a form with the following fields and buttons:

- Código**: A text input field.
- Cód. Peça**: A text input field.
- Descrição**: A larger text input field.
- Qty**: A text input field.
- Dt. Entrada**: A text input field.
- Gravar**: A button to save the entry.
- Cancelar**: A button to cancel the entry.

Tela Entrada, Entrada de um produto, na reposição do estoque.