

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE CIÊNCIAS DA COMPUTAÇÃO**  
(Bacharelado)

**PROTÓTIPO DE SISTEMA DE CONTROLE DA UTILIZAÇÃO  
DE RECURSOS PARA PRODUÇÃO DAS INDÚSTRIAS DE  
MANUFATURA**

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À UNIVERSIDADE  
REGIONAL DE BLUMENAU PARA A OBTENÇÃO DOS CRÉDITOS NA  
DISCIPLINA COM NOME EQUIVALENTE NO CURSO DE CIÊNCIAS DA  
COMPUTAÇÃO — BACHARELADO

**MARCELO LOFFI**

BLUMENAU, JUNHO/1999

1999/1-38

# **PROTÓTIPO DE SISTEMA DE CONTROLE DA UTILIZAÇÃO DE RECURSOS PARA PRODUÇÃO DAS INDÚSTRIAS DE MANUFATURA**

**MARCELO LOFFI**

ESTE TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO  
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE  
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

**BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO**

---

Prof. Oscar Dalfovo — Orientador na FURB

---

Prof. José Roque Voltolini da Silva — Coordenador do TCC

**BANCA EXAMINADORA**

---

Prof. Oscar Dalfovo

---

Prof. Wilson Pedro Carli

---

Prof.

## AGRADECIMENTOS

*À minha noiva Ana,*

*à minha família,*

*ao meu Orientador Oscar Dalfovo,*

*aos colaboradores da Fonte Sistemas de Informática,*

*a todos que colaboraram direta ou indiretamente com este trabalho e*

*em especial, a Deus..*

# SUMÁRIO

SUMÁRIO.....	iv
LISTA DE FIGURAS.....	vii
LISTA DE QUADROS.....	ix
RESUMO .....	x
ABSTRACT .....	xi
1 INTRODUÇÃO .....	1
1.1 OBJETIVOS.....	3
1.2 SINOPSE.....	3
2 REDES PERT-CPM PARA ANÁLISE DE CAMINHO CRÍTICO .....	5
2.1 CONCEITO DE PROJETO E CAMINHO CRÍTICO .....	5
2.2 UTILIZAÇÃO DA PERT-CPM PARA PROGRAMAÇÃO DA PRODUÇÃO .....	6
2.3 O GRÁFICO DE GANTT.....	12
2.4 BENEFÍCIOS DO PERT-CPM.....	13
3 SISTEMAS DE INFORMAÇÃO .....	14
3.1 ORGANIZAÇÕES .....	14
3.2 SISTEMAS DE ADMINISTRAÇÃO DA PRODUÇÃO .....	15
3.2.1 IMPORTÂNCIA ESTRATÉGICA DO SISTEMA DE ADMINISTRAÇÃO DA PRODUÇÃO.....	15
3.2.2 SISTEMAS DE PRODUÇÃO .....	16
3.2.3 PLANEJAMENTO E CONTROLE DA PRODUÇÃO .....	17
4 METODOLOGIA DE ORIENTAÇÃO A OBJETO .....	19
4.1 CLASSE .....	19
4.2 OBJETO .....	19

4.3 MÉTODOS.....	20
4.4 MENSAGENS.....	20
4.5 HERANÇA.....	21
4.6 ENCAPSULAMENTO.....	22
4.7 RELACIONAMENTOS.....	23
4.8 TÉCNICA DE MODELAGEM DE OBJETOS (OMT).....	23
4.8.1 ANÁLISE.....	24
4.8.2 PROJETO DO SISTEMA.....	27
4.8.3 PROJETO DE OBJETOS.....	28
5 TECNOLOGIAS UTILIZADAS.....	31
5.1 MICROGRAFX FLOWCHARTER 7.....	31
5.2 FOXPRO 2.6.....	32
5.3 AMBIENTE VISUAL – DELPHI 4.....	33
6 DESENVOLVIMENTO DO PROTÓTIPO DE SISTEMA DE CONTROLE DA UTILIZAÇÃO DE RECURSOS PARA PRODUÇÃO DAS INDÚSTRIAS DE MANUFATURA.....	36
6.1 ESPECIFICAÇÃO DO SISTEMA.....	36
6.1.1 DESCRIÇÃO.....	36
6.1.2 MODELO DE OBJETOS DA ANÁLISE.....	38
6.1.3 DIAGRAMA DE ESTADOS DA ANÁLISE - SUBSISTEMA PROGRAMAÇÃO DA PRODUÇÃO.....	38
6.1.4 DIAGRAMA DE ESTADOS DA ANÁLISE - APONTAMENTOS DE PRODUÇÃO.....	39
6.1.5 DIAGRAMA DE ESTADOS DA ANÁLISE - SUBSISTEMA RELATÓRIOS.....	40
6.1.6 DIAGRAMA DE EVENTOS DA ANÁLISE.....	41
6.1.7 DIAGRAMA DE FLUXO DE EVENTOS DA ANÁLISE.....	46
6.1.8 VALORES DE ENTRADA E SAÍDA DA ANÁLISE.....	47

6.1.9 DIAGRAMA DE FLUXO DE DADOS DA ANÁLISE .....	48
6.1.10 ARQUITETURA DO SISTEMA - PROJETO DO SISTEMA.....	49
6.1.11 MODELO DE OBJETOS - PROJETO DE OBJETOS.....	49
6.1.12 IMPLEMENTAÇÃO DAS CLASSES DE OBJETOS .....	50
6.2 APRESENTAÇÃO DAS TELAS E OPERACIONALIDADE .....	59
7 CONCLUSÃO .....	68
REFERÊNCIAS BIBLIOGRÁFICAS.....	69

## LISTA DE FIGURAS

Figura 1: Rede com as Etapas do Projeto resultante do Quadro 1. ....	7
Figura 2: Rede PERT-CPM da confecção de um bolo. ....	10
Figura 3: Rede PERT-CPM da confecção do bolo apresentando caminho crítico.....	12
Figura 4: Gráfico de Gantt da confecção do bolo.....	13
Figura 5: Visão geral das atividades do PCP. ....	18
Figura 6: Anatomia de um objeto. ....	20
Figura 7: Objetos se comunicando com solicitações.....	21
Figura 8: Exemplo de Herança - Relógio.....	22
Figura 9: Modelo de Objeto. ....	25
Figura 10: Diagrama de eventos. ....	25
Figura 11: Diagrama de fluxo de eventos. ....	26
Figura 12: Diagrama de estados. ....	26
Figura 13: Valores de entrada e saída. ....	26
Figura 14 - Diagrama de fluxo de dados.....	27
Figura 15: Arquitetura de um sistema.....	28
Figura 16: Modelo de Objetos do Projeto. ....	30
Figura 17: Micrografix FlowCharter sendo executado sob o Windows 95. ....	32
Figura 18: FoxPro 2.6 executando sob o Windows 95. ....	33
Figura 19:O Delphi sendo usado sob o Windows 95.....	35
Figura 20: Modelo de Objeto do Protótipo ....	38
Figura 21: Diagrama de Estados - subsistema programação.....	39
Figura 22: Diagrama de estados - subsistema apontamentos.....	39
Figura 23: Diagrama de estados - subsistema relatórios.....	40

Figura 24: Diagrama de eventos - subsistema programação.....	41
Figura 25: Diagrama de eventos - subsistema programação (continuação).....	42
Figura 26: Diagrama de eventos - subsistema programação (continuação).....	43
Figura 27: Diagrama de eventos - subsistema programação (continuação).....	44
Figura 28: Diagrama de eventos - subsistema apontamentos .....	44
Figura 29: Diagrama de eventos - subsistema apontamentos (continuação) .....	45
Figura 30: Diagrama de eventos - subsistema gerador .....	45
Figura 31: Diagrama de eventos - subsistema gerador (continuação) .....	45
Figura 32: Diagrama de fluxo de eventos do protótipo .....	46
Figura 33: Valores de entrada e saída do protótipo .....	47
Figura 34: Diagrama de fluxo de dados do protótipo. ....	48
Figura 35: Arquitetura do sistema do protótipo, do projeto de sistema.....	49
Figura 36: Modelo de objetos do projeto do protótipo. ....	49



## LISTA DE QUADROS

Quadro 1: Etapas para Projeto.....	7
Quadro 2: Cálculos para achar caminho crítico .....	8
Quadro 3: Etapas do projeto da confecção de um bolo .....	9
Quadro 4: Cálculo do TMT da confecção do bolo .....	10
Quadro 5: Cálculo de TMC da confecção do bolo. ....	11
Quadro 6: Cálculo das folgas da confecção do bolo.....	11
Quadro 7: Relações entre funções do sistema de administração da produção e aspectos competitivos.....	16

## RESUMO

Este trabalho consiste no estudo das técnicas das redes PERT-CPM, do sistema *Enterprise Resource Planning* (ERP) enfatizando o módulo de Planejamento/Programação da Produção e da técnica *Object Modeling Technique* (OMT), de orientação à objetos para o desenvolvimento de um protótipo de Sistemas de Informação para o controle da utilização de recursos da produção. Tem como finalidade auxiliar o executivo de produção na organização da utilização de recursos, mostrando possíveis gargalos, gerando assim maior segurança para garantir as datas de entrega dos pedidos aos clientes. Para isso, no protótipo utilizou-se a técnica de redes PERT-CPM no cálculo do caminho crítico da produção. Para elaborar o protótipo, utilizou-se as seguintes ferramentas e técnicas: na especificação foi utilizada a metodologia de orientação à objeto com a técnica OMT, para a criação dos bancos de dados foi utilizado o FoxPro 2.6, e na implementação, o ambiente visual Delphi 4.

## ABSTRACT

This paper consist on the study of PERT-CPM nets techniques, of *Enterprise Resource Planning* (ERP) system, emphasizing the Production Planning/Programming module, and of *Object Modeling Technique* (OMT) of objects orientation development of Prototype Information System to control the production recourse utilization. The object is helping the production executive on the recourse organization and utilization, showing possible necks, generating a major security to guarantee the delivery dates of client's requests. For this, the prototype utilized the PERT-CPM nets techniques on the critic way production calculus. To elaborate the prototype was utilized the following tools and techniques: in the specification was utilized the Methodology of Objects Orientation through O.M.T.; for the data banks creation was utilized the FoxPro 2.6; and in the visual environment implementation was utilized Delphi 4.

# 1 INTRODUÇÃO

Quando a produção é executada em regime de artesanato com um artesão incumbido de realizar um produto do início ao fim, o empresário pode simplesmente dizer qual produto final que deseja. O artesão (indivíduo que exerce um ofício manual) saberá então determinar como e quando fazer cada operação para resultar no produto final. Diz-se então que a programação da produção é feita pela própria pessoa que trabalha, em bases completamente informais. O controle da produção, também em bases informais, será feito pela supervisão direta do empresário.

É desnecessária a existência de um sistema formal de programação e controle da produção em um regime de produção artesanal (relativo a artesanato). Mas para reduzir custos de produção e permitir o crescimento das empresas, o regime artesanal tem sido abandonado. Surge então, a especialização do trabalho e a departamentalização. Os operários, na maior parte das indústrias modernas, só sabem fazer um número pequeno de operações. A fábrica em geral, está dividida em seções ou departamentos, cada um deles incumbido de realizar apenas um tipo de operação [ZAC76].

Torna-se necessário comunicar aos departamentos produtivos quais operações devem ser executadas em cada dia para resultar nos produtos finais desejados. Portanto deve-se determinar quais operações são necessárias e quando convém que elas sejam executadas, quais os materiais que deverão ser comprados e em que dia estes materiais deverão estar disponíveis na fábrica.

A determinação de quais operações são necessárias só pode ser feita de forma racional se existir disponibilidade de um certo conjunto de informações como saldo de estoque, vendas previstas, componentes dos produtos finais, processo produtivo de cada componente, capacidade produtiva disponível, tempos de fabricação, etc. Para que estas informações estejam disponíveis com a rapidez necessária, torna-se fundamental a utilização de um sistema de informação.

[FUR94] ressalta que, com sistemas de informação, o executivo pode visualizar em segundos o que antes levava dias para ver pelos métodos tradicionais. Tudo isso com um menor número de pessoas envolvidas no processo, o que diminui a possibilidade de erros e

minimiza custos administrativos. Para [BIN94], a utilização de sistemas de informação pode vir a facilitar o executivo no processo decisório com a obtenção de dados estrategicamente escolhidos e de conteúdos relevantes para qualquer nível e tamanho de empresa.

Segundo [DAL98], a não utilização das informações como recursos estratégicos, leva o executivo, muitas vezes a administrar por impulso ou baseado em modismos. Acredita-se que os Sistemas de Informação resolvem uma deficiência crônica nos processos decisórios da maioria das empresas: a falta de integração das informações.

Para as indústrias de manufatura (Estabelecimento industrial onde são fabricados produtos em série), um dos sistemas de informação mais importantes é o de Planejamento e Controle da Produção (PCP). Um dos conceitos mais atuais que tem o PCP como função centro, é o *Enterprise Resource Planning* - Planejamento de Recursos da Corporação (ERP). Segundo [COR97], um sistema dito ERP, tem a pretensão de suportar todas as necessidades de informações para a tomada de decisão estratégica de um empreendimento como um todo. Entretanto neste trabalho será enfatizado o módulo do *Master Production Schedule* - Plano Mestre da Produção (MPS).

Segundo [COR97] a principal função do MPS é coordenar a demanda do mercado com os recursos internos da empresa, de forma a programar taxas adequadas de produção de produtos finais. Tem como finalidade aumentar a eficiência e a eficácia do processo produtivo da empresa. É portanto, uma dupla finalidade: atuar sobre os meios de produção para aumentar a eficiência e cuidar para que os objetivos de produção sejam plenamente alcançados para aumentar a eficácia.

Uma das técnicas usadas para otimizar o planejamento e controle da produção é a de Caminho Crítico PERT-CPM. O Caminho Crítico é a relação de atividades, e suas atividades pré-requisitos, onde não pode haver atraso e conseqüentemente, haver atraso no produto final. A técnica PERT tem como base os eventos e a técnica CPM tem como base as atividades. Para achar o Caminho Crítico há a necessidade de utilizar as duas técnicas, pois é preciso basear-se tanto nas atividades quanto nos eventos utilizados em determinado produto. Usando as técnicas de Caminho Crítico, as informações serão mais consistentes [BON71].

O protótipo proposto é um sistema de informação que tem como finalidade auxiliar na resolução dos problemas de informações relacionadas com o comando da fabricação determinando “o que”, “quando”, “quanto” e “onde” produzir. Com base nessas informações será feito o Planejamento e Controle da Produção (PCP).

## 1.1 OBJETIVOS

Desenvolver para a pequena e média indústria um protótipo de Sistema de Informações que dê apoio a tomada de decisões táticas e operacionais, aplicando os conceitos de *Enterprise Resource Planning (ERP)* e as técnicas de PERT/CPM.

O protótipo tem como objetivos específicos:

- a) programar as atividades de produção;
- b) prever e evitar gargalos de produção;
- c) apresentar graficamente os resultados encontrados.

## 1.2 SINOPSE

O trabalho foi dividido em seis capítulos descritos a seguir:

O primeiro capítulo define os objetivos do trabalho, apresentando a justificativa para sua elaboração.

O segundo capítulo apresenta definições sobre a técnica PERT-CPM descrevendo seu funcionamento e sua utilidade.

O terceiro capítulo dá uma visão geral sobre os sistemas de informação, que é o tipo de sistema que o protótipo abrange, além de apresentar conceitos sobre sistemas de administração da produção com ênfase no planejamento e programação da produção.

O quarto capítulo apresenta a metodologia de orientação a objetos, usada na elaboração da especificação do protótipo.

O quinto capítulo descreve as ferramentas utilizadas na confecção do trabalho e do protótipo.

O sexto capítulo apresenta o protótipo, sua especificação, características, telas e operacionalização.

O sétimo capítulo completa o trabalho apresentando as conclusões, dificuldades encontradas e sugestões para próximos trabalhos.

## 2 REDES PERT-CPM PARA ANÁLISE DE CAMINHO CRÍTICO

Segundo [LOE99], as etapas de um projeto são o planejamento, a programação e o controle. Durante o planejamento responde-se à pergunta "*o que fazer?*"; assim são detalhadas as atividades e estabelecida a relação de interdependência entre as atividades. Durante a programação responde-se à pergunta "*quando fazer?*" ; assim é estabelecido um cronograma para execução dessas atividades determinando data para início e término das atividades. Durante o controle responde-se à pergunta "*como fazer?*"; deve ocorrer durante a execução do projeto e estabelecer medidas para que a programação estabelecida seja cumprida sem atrasos.

Nas técnicas de Caminho Crítico a execução de um projeto é representada através de um grafo, tornando explícita a relação de dependência entre as atividades e seus tempos.

Para [BON71], as técnicas para análise de Caminho Crítico podem ser classificadas em dois grupos. O primeiro grupo é o Método de Caminho Crítico (*Critical Path Method (CPM)*) – usada onde as tarefas ou atividades têm duração perfeitamente determinadas. O segundo é a Técnica de Avaliação e Revisão de Programas (*Project Evaluation Review Technique (PERT)*) – usada onde as tarefas são planejadas na incerteza, ou seja, as tarefas têm duração aleatória (probabilísticas). Nesta técnica são considerados três tempos para a duração de cada tarefa: tempo mais provável, tempo otimista e tempo pessimista.

### 2.1 CONCEITO DE PROJETO E CAMINHO CRÍTICO

Projeto é um conjunto de tarefas ou atividades elementares bem diferenciáveis que são executadas segundo uma ordem determinada e que exigem tempo para serem executadas. O início ou fim de uma atividade é um instante definido de tempo que corresponde a uma etapa ou nó do projeto. Uma etapa não pode ser iniciada sem que a etapa precedente tenha sido terminada, ou seja, a etapa subsequente de uma tarefa não pode ser realizada até que a tarefa esteja completa. A primeira coisa a fazer é organizar um lista de tudo que tem de ser feito, definindo as diversas tarefas que irão constituir o projeto.

Em qualquer projeto, algumas atividades são flexíveis, em relação ao começo e término, outras não são flexíveis, sendo que qualquer atraso em qualquer delas, atrasará o projeto. As atividades inflexíveis são denominadas críticas e a cadeia que elas formam



denomina-se caminho crítico. O caminho crítico é a duração mais longa através do projeto. Há sempre pelo menos um caminho crítico em um projeto, podendo haver vários. Então o caminho crítico é definido como: “aquele no qual as atividades não tem folga de tempo para começar nem para terminar” [BON71].

## **2.2 UTILIZAÇÃO DA PERT-CPM PARA PROGRAMAÇÃO DA PRODUÇÃO**

Segundo [STO91], a PERT-CPM é uma espécie de análise de rede, que divide um projeto em etapas e determina a seqüência em que devem ser realizadas. Envolve quatro etapas:

- a) determinar as etapas necessárias para se completar o projeto;
- b) determinar as relações de precedência e identificar as tarefas que podem ser realizadas simultaneamente;
- c) avaliar o tempo necessário para cada atividade;
- d) desenhar uma rede de PERT-CPM criando caminhos ou seqüências de eventos e atividades dentro da rede.

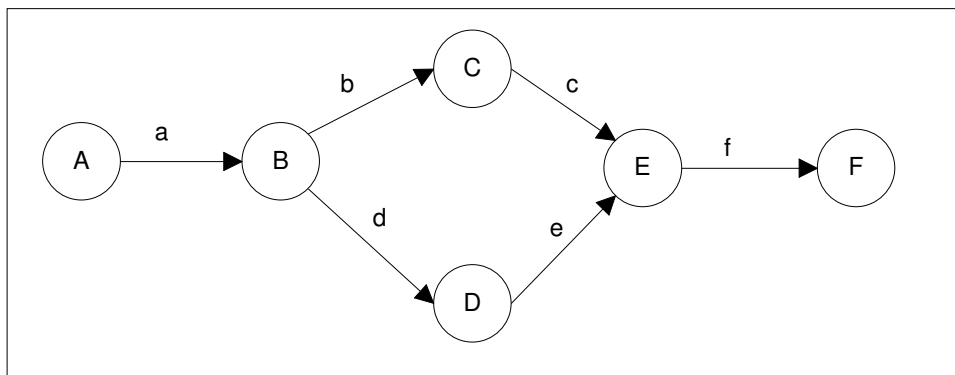
A rede PERT-CPM é então analisada para evitar gargalos potenciais e exigências da programação, ambos importantes para planejar e controlar o projeto.

O quadro 1 apresenta as etapas do projeto, com as características necessárias para a criação da rede. A figura 1 ilustra a rede com etapas do projeto resultantes do quadro 1.

Atividade	Descrição	Predecessores Imediatos	Estimativa de Tempo Otimista ( $ta$ )	Estimativa de Tempo Mais Provável ( $tm$ )	Estimativa de Tempo Pessimista ( $tb$ )	Tempo Estimado ( $te$ )
(a, b, c ...)						

Fonte: adaptado de [LOE99]

Quadro 1: Etapas para Projeto



Fonte: adaptado de [LOE99]

Figura 1: Rede com as Etapas do Projeto resultante do Quadro 1.

A partir das informações do quadro pode-se construir a rede mostrada na figura 1. Essa rede mostra quais atividades (a,b,c,...) podem ser realizadas simultaneamente e quais têm que esperar o término da precedente para que seja executada.

O cálculo do tempo estimado é definido por:  $te = \frac{ta + 4 \times tm + tb}{6}$

Para achar o caminho crítico, é preciso achar o Tempo Mais Tarde (TMT) e o Tempo Mais Cedo (TMC) , que vão definir o tempo que deve levar o projeto, e as folgas de cada evento. O modo de calcular cada um deles está representado no quadro 2.

TMT <sub>i</sub>	<p>Cada TMT<sub>i</sub> é o tempo que as atividades precedentes (máximas) ao evento levaram para ser concluídas. O TMT<sub>i</sub> é encontrado através da escolha de qual tempo da sub-rede de atividades tem o maior tempo até o evento mais o tempo da atividade que está entre o último evento desta sub-rede e o evento calculado.</p> $TMT_i = \max\{TMT_{i_{ant1}}, TMT_{i_{ant2}}, \dots, TMT_{i_{antn}}\}$
TMC <sub>i</sub>	<p>O TMC<sub>i</sub> é calculado ao contrário do TMT<sub>i</sub>. É iniciado no evento que representa o término do projeto e faz o caminho inverso. É calculado através da escolha do tempo da sub-rede de atividades de menor tempo até o evento menos o tempo da atividade entre o evento e o seu posterior.</p> $TMC_i = \min\{TMC_{i_{post1}}, TMC_{i_{post2}}, \dots, TMC_{i_{postn}}\}$
F <sub>i</sub>	<p>A folga de cada evento é calculada subtraindo o TMT do TMC deste evento. As folgas são usadas para encontrar o caminho crítico – os eventos onde a folga for igual a zero (ou seja, não há folga) fazem parte do caminho crítico.</p> $F_i = TMC_i - TMT_i$

Quadro 2: Cálculos para achar caminho crítico

Para que seja melhor ilustrada a forma de calcular o caminho crítico e os passos de uma rede PERT-CPM será usado um exemplo prático fácil – a receita de um bolo. Os tempos são usados em minutos. A figura 2 ilustra a rede PERT-CPM do projeto bolo, o quadro 3 mostra as etapas do projeto, o quadro 4 apresenta o cálculo do TMT do projeto, o quadro 5 apresenta o cálculo do TMC do projeto e o quadro 6 apresenta o cálculo das folgas do projeto.

Atividade	Descrição	Predecessores Imediatos	Estimativa de Tempo Otimista ( $t_a$ )	Estimativa de Tempo Mais Provável ( $t_m$ )	Estimativa de Tempo Pessimista ( $t_b$ )	Tempo Estimado ( $t_e$ )
a	misturar manteiga e açúcar	-----	4	5	7	5.17
b	juntar ovos	A	4	5	7	5.17
c	misturar trigo e fermento	-----	3	3	5	3.33
d	juntar o trigo com fermento e o leite	b, c	5	7	9	7
e	untar forma	-----	4	5	7	5.17
f	colocar massa na forma e assar	d, e	50	50	55	50.83
g	misturar chocolate em pó e creme de leite para recheio	-----	5	7	10	7.17
h	bater claras em neve p/ cobertura	-----	4	5	8	5.33
i	cortar massa assada ao meio	F	3	5	8	5.17
j	Colocar recheio	g, i	6	8	10	8
k	Colocar cobertura	h, j	8	10	15	10.5

Quadro 3: Etapas do projeto da confecção de um bolo

A partir de agora para definir o tempo de execução de cada tarefa será considerado o tempo estimado.

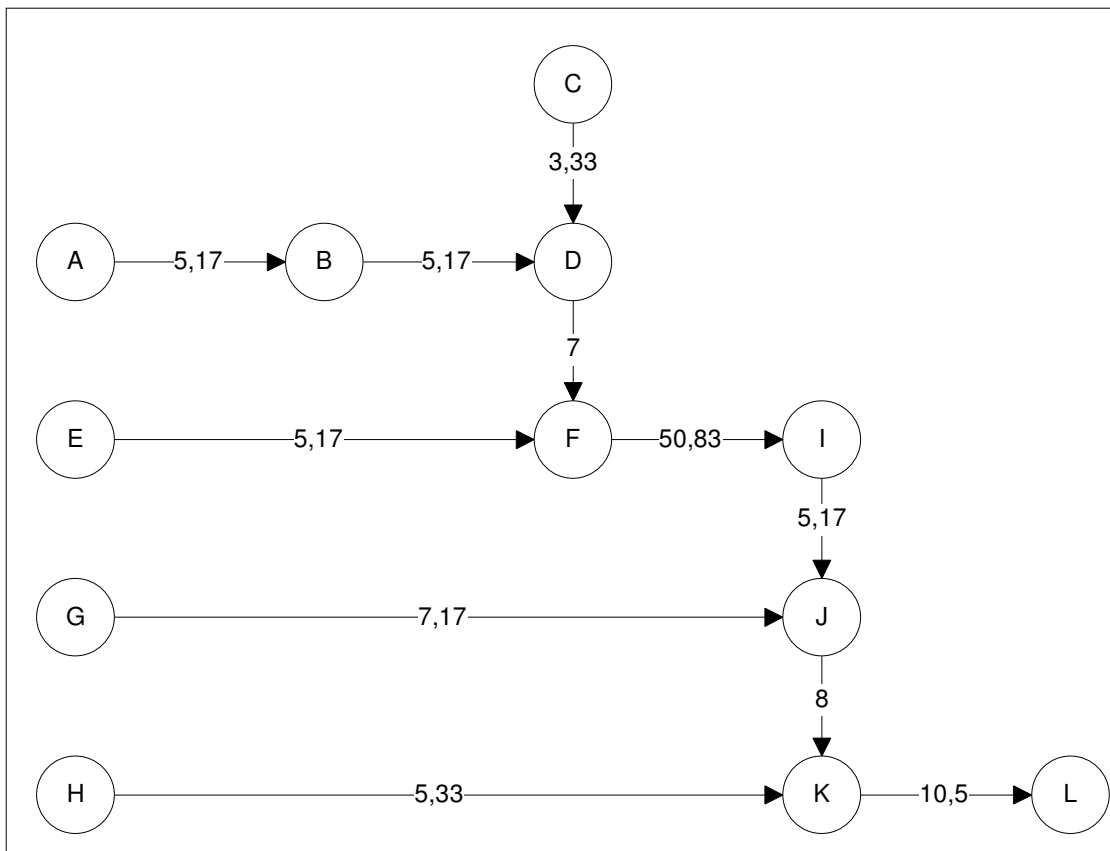


Figura 2: Rede PERT-CPM da confecção de um bolo.

$TMT_A$	0	0
$TMT_B$	$\max\{0 + 5.17\}$	5.17
$TMT_C$	0	0
$TMT_D$	$\max\{5.17+5.17, 3.33\}$	10.34
$TMT_E$	0	0
$TMT_F$	$\max\{5.17, 10.34+7\}$	17.37
$TMT_G$	0	0
$TMT_H$	0	0
$TMT_I$	$\max\{17.34+50.83\}$	68.17
$TMT_J$	$\max\{7.17, 68.17+5.17\}$	73.34
$TMT_K$	$\max\{5.33, 73.34+8\}$	81.34
$TMT_L$	$\max\{81.34+10.5\}$	91.84

Quadro 4: Cálculo do TMT da confecção do bolo

TMC <sub>L</sub>	91.84	
TMC <sub>K</sub>	min {91.84-10.5}	81.34
TMC <sub>J</sub>	min{81.34-8}	73.34
TMC <sub>I</sub>	min{73.34-5.17}	68.17
TMC <sub>H</sub>	min{81.34-5.33}	76.01
TMC <sub>G</sub>	min{73.34-7.17}	66.17
TMC <sub>F</sub>	min{68.17-50.83}	17.34
TMC <sub>E</sub>	min{17.34-5.17}	12.17
TMC <sub>D</sub>	min{17.34-7}	10.34
TMC <sub>C</sub>	min{10.34-3.33}	7.01
TMC <sub>B</sub>	min{10.34-5.17}	5.17
TMC <sub>A</sub>	min{5.17-5.17}	0

Quadro 5: Cálculo de TMC da confecção do bolo.

F <sub>A</sub>	0-0	0
F <sub>B</sub>	5.17-5.17	0
F <sub>C</sub>	7.01-0	7.01
F <sub>D</sub>	10.34-10.34	0
F <sub>E</sub>	12.17-0	12.17
F <sub>F</sub>	17.34-17.34	0
F <sub>G</sub>	66.17-0	66.17
F <sub>H</sub>	76.01-0	76.01
F <sub>I</sub>	68.17-68.17	0
F <sub>J</sub>	73.34-73.34	0
F <sub>K</sub>	81.34-81.34	0
F <sub>L</sub>	91.84-91.84	0

Quadro 6: Cálculo das folgas da confecção do bolo.

Conforme a Quadro 6 das folgas, os eventos A, B, D, F, I, J, K e L não têm folgas e são estes eventos que formam o caminho crítico do projeto Bolo. A figura 3 mostra o caminho crítico do bolo na rede PERT-CPM.

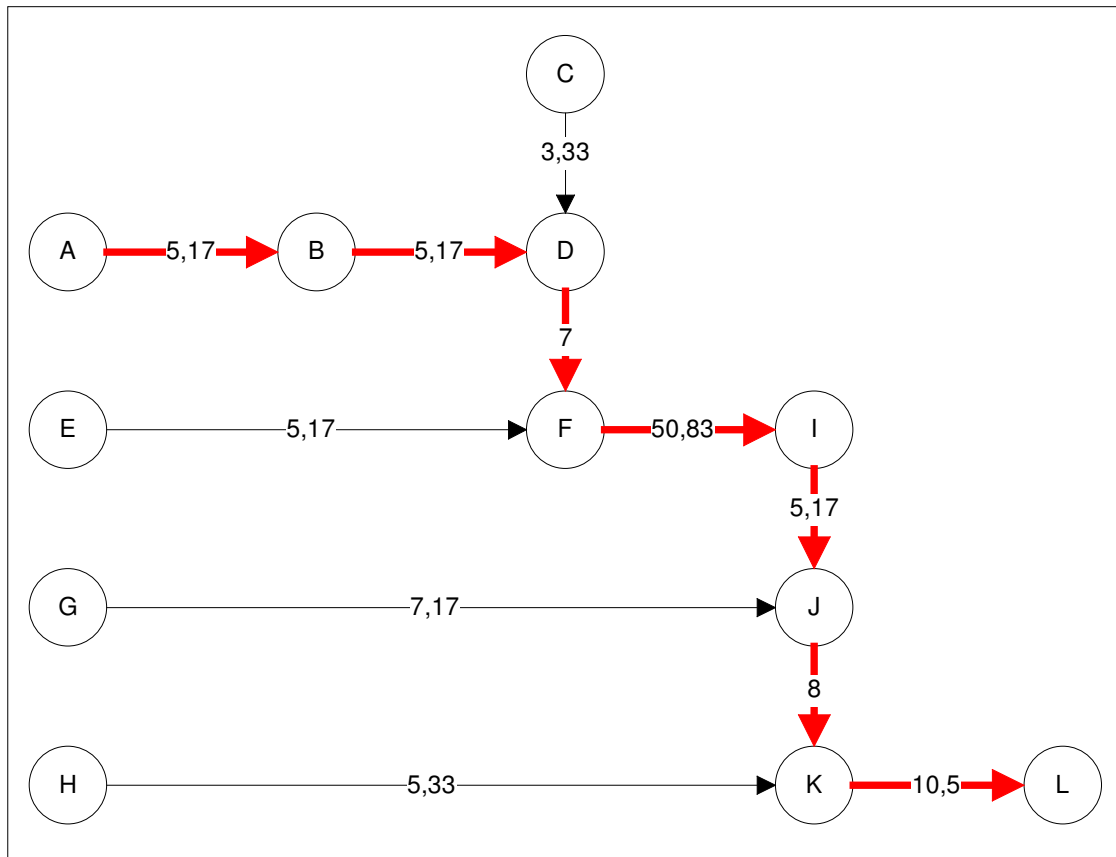


Figura 3: Rede PERT-CPM da confecção do bolo apresentando caminho crítico

## 2.3 O GRÁFICO DE GANTT

Conforme [STO91], o Gráfico de Gantt é o método gráfico de planejamento e controle que permite a um administrador ver as datas de início e término de várias tarefas. O projeto é dividido em tarefas separadas e são feitas estimativas de quanto tempo cada uma requer, bem como do tempo total para completar o projeto.

Os Gráficos de Gantt são úteis para acompanhar projetos compostos por tarefas que sejam realizadas numa seqüência. No protótipo o Gráfico de Gantt será usado para a visualização da Programação de um determinado lote.

Na figura 4 visualiza-se o exemplo da confecção do bolo em um Gráfico de Gantt.

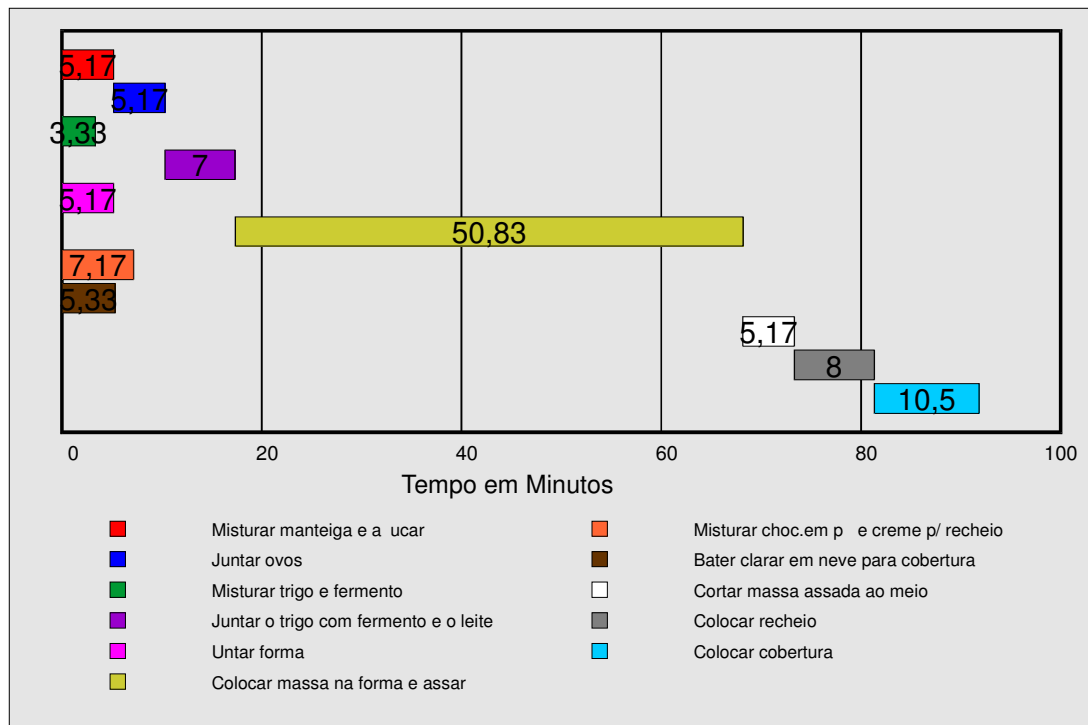


Figura 4: Gráfico de Gantt da confecção do bolo.

## 2.4 BENEFÍCIOS DO PERT-CPM

O PERT-CPM oferece vários benefícios dos quais pode-se destacar:

- é útil para planejar tanto no nível estratégico como no operacional, pois indica onde os recursos sejam alocados para que os objetivos sejam alcançados com eficiência;
- mostra aos administradores como economizar tempo, conseqüentemente economizar dinheiro já que tempo e custo estão comumente relacionados;
- ajuda os administradores a alcançar seus objetivos dentro do prazo, indicando quais são as atividades críticas e prioritárias dentro do projeto.

Segundo [STO91] o PERT-CPM costuma ser anunciado como uma *técnica de administração de projetos*, em vez de apenas uma ferramenta de programação.



## 3 SISTEMAS DE INFORMAÇÃO

De acordo com [DAL98], os Sistemas de Informações resolvem uma deficiência crônica no processo decisório da maioria das empresas, isto é, a falta de integração das informações.

Apesar das palavras dados e informações serem usadas de modo semelhante, há uma diferença entre os dois conceitos. Dados são os números e fatos brutos, não analisados, como o número de camas produzidas por semana em uma fábrica de móveis. A informação resulta quando os dados são organizados ou analisados de algum modo significativo, como comparar a produção de camas de uma semana com a semana anterior.

As empresas irão avaliar as informações que recebem a partir de quatro fatores:

- a) qualidade da informação: quanto mais precisa for a informação com mais segurança os administradores poderão contar com ela no momento de tomar decisões;
- b) a oportunidade da informação: a informação deve estar disponível à pessoa certa no momento certo para que a ação corretiva seja aplicada antes que ocorra um desvio muito grande do plano ou do padrão;
- c) quantidade de informações: dificilmente os administradores podem tomar decisões precisas e oportunas sem informações suficientes. Contudo, os administradores são freqüentemente inundados por informações irrelevantes ou inúteis. Se recebem mais informações do que podem utilizar produtivamente, eles podem não perceber informações sobre problemas sérios;
- d) relevância da informação: as informações que os administradores recebem devem ter relevância as suas responsabilidades e tarefas.

### 3.1 ORGANIZAÇÕES

Segundo [STO91], a definição de organização é "*duas ou mais pessoas trabalhando juntas e de modo estruturado para alcançar um objetivo específico ou um conjunto de objetivos*". O mais óbvio desses elementos talvez seja um objetivo ou finalidade, e as organizações devem ter programas ou métodos para alcançar seus objetivos. Sem nenhum plano é provável que nenhuma organização seja muito eficaz.

Uma organização de produção preocupa-se principalmente em produzir bens físicos, que podem ser produzidos em massa e armazenados para consumo posterior. Porém a produção em massa gera custos como armazenagem e compra de matéria prima excessiva. Para evitar que a produção seja excessiva, há a necessidade de um sistema que administre a produção, para que cada produção seja feita no tempo certo de atender às necessidades.

## **3.2 SISTEMAS DE ADMINISTRAÇÃO DA PRODUÇÃO**

Segundo [COR97], chama-se genericamente Sistemas de Administração da Produção os Sistemas de Informação para apoio à tomada de decisões estratégicas da organização. Existem diversas alternativas de técnicas e lógicas que podem ser utilizadas com este objetivo. As três principais, que têm sido mais extensivamente usadas ao longo dos últimos 15 anos são: os sistemas MRP/ERP, que se baseiam fundamentalmente na lógica do cálculo das necessidades futuras de produtos, os sistemas *Just in Time*, de inspiração japonesa, e os sistemas de produção com capacidade finita, que se utilizam fundamentalmente das técnicas de simulação em computador.

### **3.2.1 IMPORTÂNCIA ESTRATÉGICA DO SISTEMA DE ADMINISTRAÇÃO DA PRODUÇÃO**

O Sistema de Administração da Produção deve planejar a produção e controlar o seu desempenho. Antecipadamente deve estabelecer o que a empresa irá produzir e conseqüentemente o que deverá dispor de matérias-primas e materiais, de pessoas, de máquinas e equipamentos, bem como de estoques de produtos acabados para suprir as vendas.

Ao desenvolver suas funções, o Sistema de Administração da Produção procura utilizar racionalmente os recursos empresariais, sejam eles materiais, humanos, financeiros, etc e desta forma mantém uma rede de relações com as demais áreas da empresa [COR97].

O quadro 7 traz um resumo dos relacionamentos entre as sete principais funções a cargo dos sistemas de administração da produção e os aspectos de desempenho competitivo que estão dentro do escopo dos sistemas de operações produtivas nas organizações. Embora tendo algum grau de influência sobre todos os aspectos competitivos dos sistemas produtivos, o maior potencial dos sistemas de produção concentra-se nos aspectos referentes a custos e tempos, tanto em termos de velocidade como de confiabilidade de entrega.

Relação entre funções do sistema de administração da produção e aspectos competitivos						
	Custo	Velocidade	Confiabilidade	Flexibilidade	Qualidade	Serviço
1	4	4	4			
2	4					
3	4	4	4	4		
4	4	4	4			
5			4		4	4
6	4		4			
7		4		4		

Legenda:

1 - Planejar as necessidades futuras de capacidade de produção  
2 - Planejar os materiais comprados.  
3 - Planejar níveis adequados de estoques de matérias primas, semi-acabados e produtos finais, nos pontos certos.  
4 - Programar atividades de produção para garantir que os recursos produtivos envolvidos estejam sendo utilizados, em cada momento, nas atividades certas e prioritárias.  
5 - Ser capaz de saber informar corretamente a respeito da situação corrente dos recursos (pessoas, equipamentos, instalações, materiais) e das ordens (de compra e produção).  
6 - Ser capaz de prometer aos clientes os menores prazos possíveis e depois fazer cumprilos.  
7 - Ser capaz de reagir eficazmente.

Fonte: [COR97]

Quadro 7: Relações entre funções do sistema de administração da produção e aspectos competitivos.

### 3.2.2 SISTEMAS DE PRODUÇÃO

Conforme [COR97], o Sistema de Administração está intimamente relacionado com o sistema de produção adotado pela empresa e dele depende para planejar e controlar a produção. Existem três tipos de Sistemas de Produção e cada um desses exige um Sistema de Administração da Produção específico:

- a) produção sob encomendas – a empresa apenas produz após ter efetuado um contrato

ou pedido de venda de seus produtos. É a encomenda que vai definir como a produção deverá ser planejada e controlada;

- b) produção em lotes – a empresa produz uma quantidade limitada de um tipo de produto de cada vez. A produção é planejada antecipadamente e a empresa pode aproveitar melhor os seus recursos;
- c) produção contínua ou em série – a empresa produz um produto durante um longo período de tempo e sem qualquer modificação. O planejamento é feito antecipado e pode cobrir maior extensão de tempo.

### **3.2.3 PLANEJAMENTO E CONTROLE DA PRODUÇÃO**

Planejar significa pensar antecipadamente em seus objetivos e ações, e agir com base em algum método, plano ou lógica, e não em palpites [STO91].

Segundo [CHI90] o planejamento da produção é vital para o sucesso da empresa: fundamenta-se na previsão de vendas como base do que a empresa pretende colocar no mercado e na capacidade de produção da empresa como base do que a empresa tem condições de produzir.

Para [TUB97], em um sistema produtivo, ao serem definidas suas metas e estratégias, faz-se necessário formular planos e controles (PCP) para atingi-las. O mesmo autor descreve as atividades do PCP como sendo o Planejamento Estratégico da Produção, O Planejamento-mestre da Produção, a Programação da Produção e o Acompanhamento e Controle da Produção.

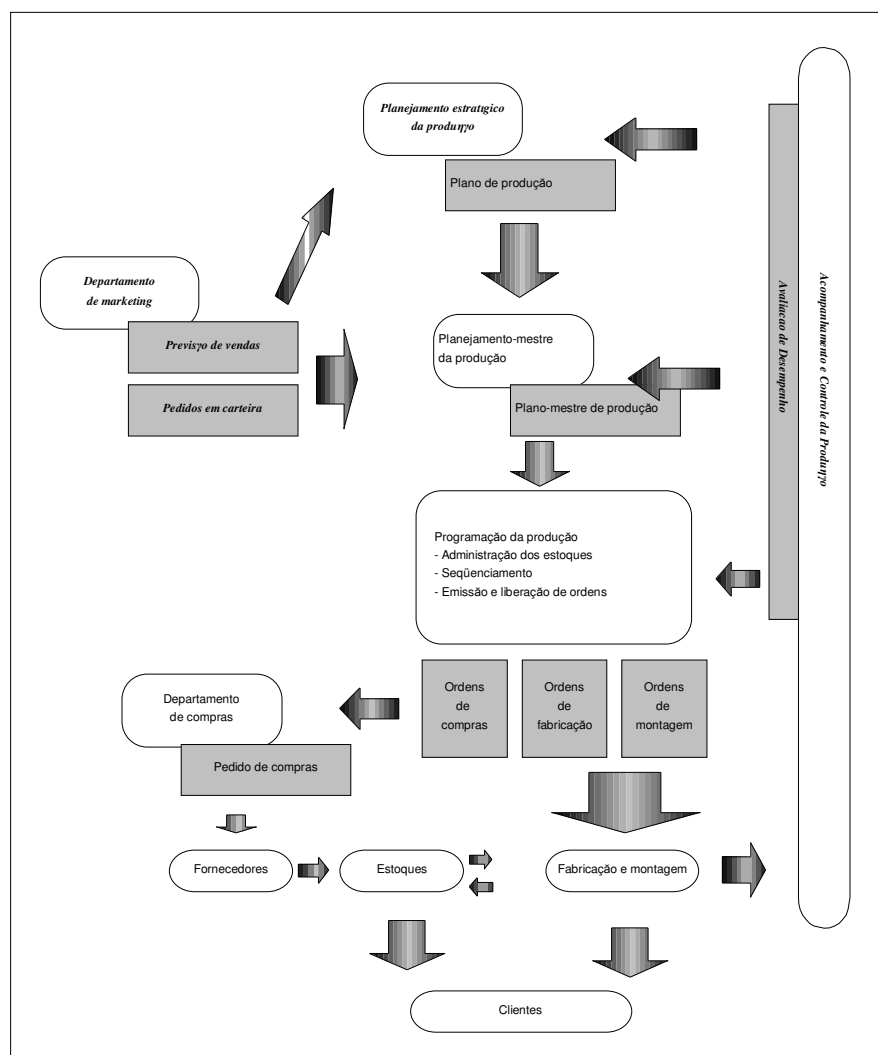
O Planejamento Estratégico da Produção consiste em estabelecer um Plano de Produção para um determinado período (longo prazo) segundo as estimativas de vendas e a disponibilidade financeira e produtiva.

O Planejamento-mestre da Produção consiste em estabelecer um Plano-mestre de Produção (PMP) de produtos finais, detalhado a médio prazo, período a período, a partir do Plano de Produção, com base nas previsões de vendas de médio prazo ou nos pedidos em carteira já confirmados.

A Programação da Produção estabelece a curto prazo (com base no PMP e nos registros de controle de estoques) quando e quanto comprar, fabricar ou montar de cada item necessário à composição dos produtos finais.

O Acompanhamento e Controle da Produção, por meio da coleta e análise de dados, busca garantir que o programa de produção emitido seja executado a contento.

A figura 5 apresenta uma visão geral das atividades do PCP.



Fonte: [TUB97]

Figura 5: Visão geral das atividades do PCP.

O protótipo irá enfatizar a atividade de Programação da Produção.

## 4 METODOLOGIA DE ORIENTAÇÃO A OBJETO

Para entender a Orientação a Objetos é necessário antes entender alguns conceitos básicos dessa metodologia. São eles:

- a) classe;
- b) objeto;
- c) método;
- d) mensagens;
- e) herança;
- f) encapsulamento;

### 4.1 CLASSE

O termo classe refere-se à implementação de software de um tipo de objeto. O tipo de objeto é uma noção conceitual, especifica uma família de objetos sem estipular como o tipo e o objeto são implementados. Os tipos de objetos são especificados durante a análise orientada a objeto, quando se implementam tipos de objetos são usados outros termos.

Uma classe é então, uma implementação de um tipo de objeto. Ela especifica uma estrutura de dados e os métodos operacionais permissíveis que se aplicam a cada um dos seus objetos [MAR95].

### 4.2 OBJETO

[MAR95] conceitua um objeto como qualquer coisa, real ou abstrata, a respeito da qual armazena-se dados e os métodos que os manipulam. Um objeto pode ser composto de outros objetos, que por sua vez podem ser compostos de outros objetos. É como uma máquina que é composta por componentes e esses componentes por outros componentes.

Um tipo objeto é uma categoria de objeto. Um objeto é uma instância do tipo objeto. O objeto preocupa-se tanto com seus dados quanto com os métodos com os quais os dados são manipulados.

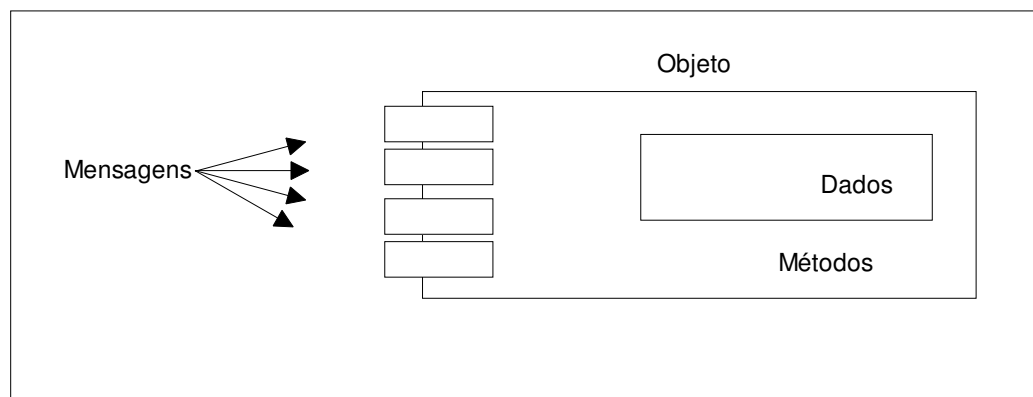
## 4.3 MÉTODOS

Os métodos especificam a maneira pela qual os dados de um objeto são manipulados. Os métodos de um tipo de objeto referenciam somente as estruturas de dados desse tipo objeto, não devendo acessar diretamente as estruturas de dados de outro tipo objeto. Para o acesso a estrutura de outro tipo objeto devem ser usadas as mensagens. Dessa forma, um objeto é qualquer coisa com suas propriedades representadas pelos tipos de dados e seu comportamento representado por métodos [MAR95].

## 4.4 MENSAGENS

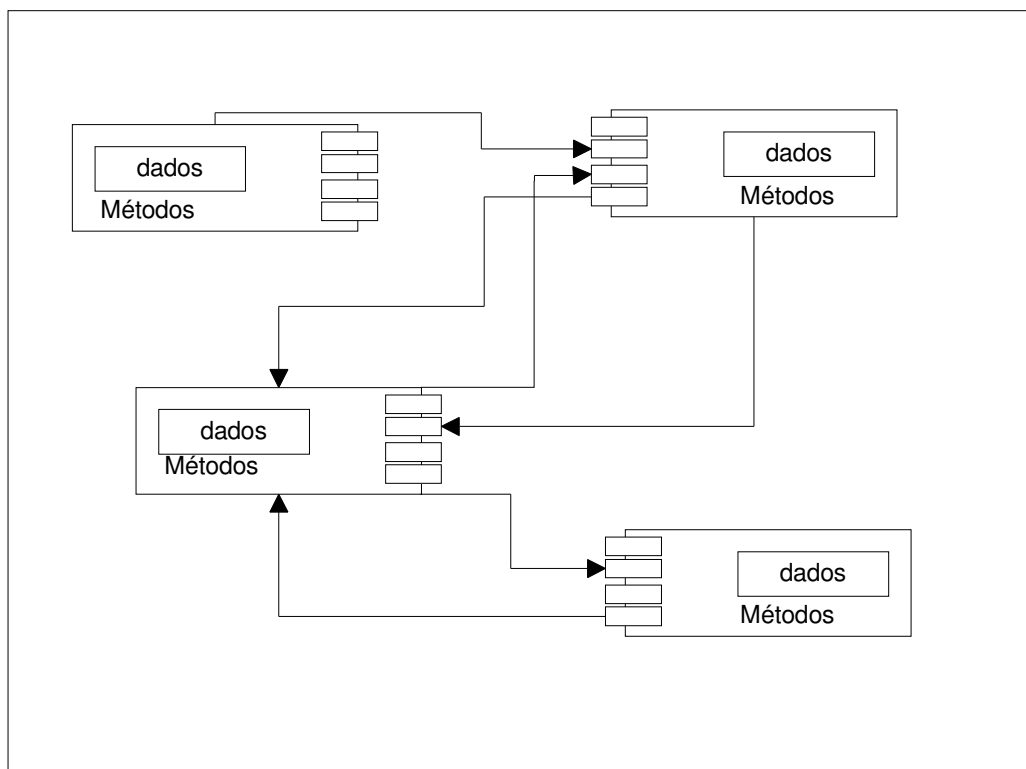
Para que um objeto realize alguma coisa, envia-se uma solicitação, que faz com que uma operação seja invocada. A operação executa o método apropriado e opcionalmente, retorna uma resposta. A mensagem que constitui a solicitação contém o nome do objeto, o nome da operação e, às vezes um grupo de parâmetros.

A figura 6 ilustra a anatomia de um objeto, e a figura 7 ilustra objetos se comunicando através de solicitações, onde cada solicitação é uma mensagem especificando que uma operação indicada seja executada.



Fonte: adaptado de [WIN93]

Figura 6: Anatomia de um objeto.



Fonte: adaptado de [WIN93]

Figura 7: Objetos se comunicando com solicitações.

## 4.5 HERANÇA

Um tipo de objeto de alto nível pode ser especializado em tipos de objetos de níveis mais baixos. Um tipo de objetos pode ter subtipos.

Um exemplo clássico é de um relógio, conforme ilustra a figura 8. O relógio pode ser analógico ou digital, sem perder as características de relógio; um relógio analógico pode ser despertador, sem perder as características de relógio analógico e de relógio.



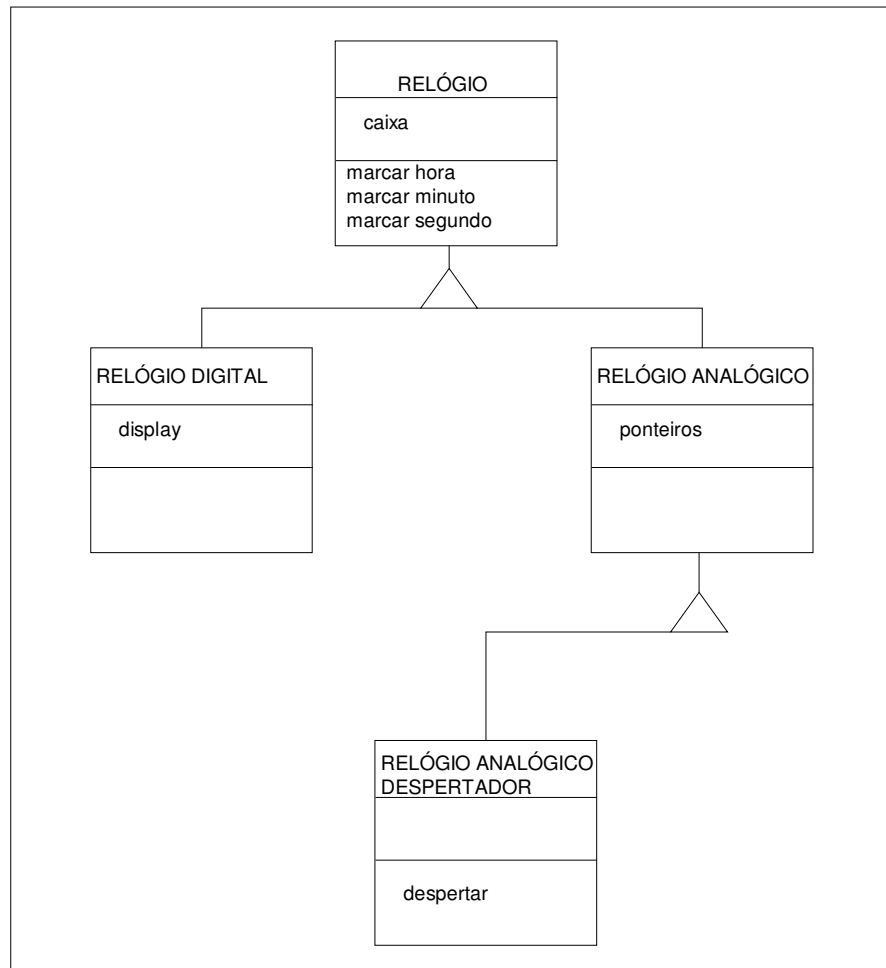


Figura 8: Exemplo de Herança - Relógio

Uma classes implementa o tipo de objeto. Uma subclasse herda as propriedades de sua classe mãe. Uma subclasse pode herdar a estrutura de dados e os métodos ou alguns dos métodos, de sua superclasse. Ela também tem métodos e, às vezes, tipos de dados próprios.

## 4.6 ENCAPSULAMENTO

Para [MAR95], um conceito para encapsulamento pode ser "*o ato de empacotar ao mesmo tempo dados e métodos*". O objeto esconde seus dados de outros objetos e permite que os dados sejam acessados por intermédio de seus próprios métodos. O encapsulamento protege os dados contra adulterações.






O encapsulamento oculta os detalhes de sua implementação interna aos usuários de um objeto. Os usuários entendem quais operações do objeto podem ser solicitadas, mas não

conhecem detalhes de como a operação é executada. Então pode-se dizer que Encapsulamento é o ato de ocultar do usuário os detalhes da implementação de um objeto.

O encapsulamento é importante porque separa a maneira como um objeto se comporta da maneira como ele é implementado.

## 4.7 RELACIONAMENTOS

Os relacionamentos entre os objetos é demonstrado através dos seguintes símbolos:

	Dependência – o objeto relacionado a ele depende dele para existir.
	Herança – os objetos relacionados ao objeto acima, herdam as características dele.
	Cardinalidade – o objeto se relaciona uma ou várias vezes (1, N).
	Cardinalidade – o objeto pode se relacionar várias vezes (0, N).
	Cardinalidade – o objeto se relaciona uma vez (1, 1).

## 4.8 TÉCNICA DE MODELAGEM DE OBJETOS (OMT)

Segundo [RUM94], a metodologia OMT fundamenta-se na utilização de uma notação baseada em objetos para descrever as classes e os relacionamentos durante o ciclo de vida. Ao Modelo de Objetos são acrescentados o Modelo Dinâmico e o Modelo Funcional para descrever todos os aspectos de um sistema. A fase de análise consiste no desenvolvimento de um modelo daquilo que se espera que o sistema faça, independente de como será implementado. A fase de projeto consiste na otimização, refinamento e ampliação de Modelos de Objetos, Dinâmico e Funcional até que eles estejam suficientemente detalhados para implementação, fazendo com que a implementação seja uma simples questão de tradução do projeto para um código.

O modelo de objetos descreve a estrutura de objetos de um sistema: identidade, relacionamentos com outros objetos, atributos e operações. Este modelo deve proporcionar a estrutura necessária na qual podem ser colocados os modelos dinâmico e funcional.

O modelo dinâmico descreve os aspectos de um sistema relacionados ao tempo e à seqüência de operações. Este modelo incorpora o controle (seqüência de operações que ocorrem, independente do que elas fazem, sobre o que elas atuam ou como são implementadas).

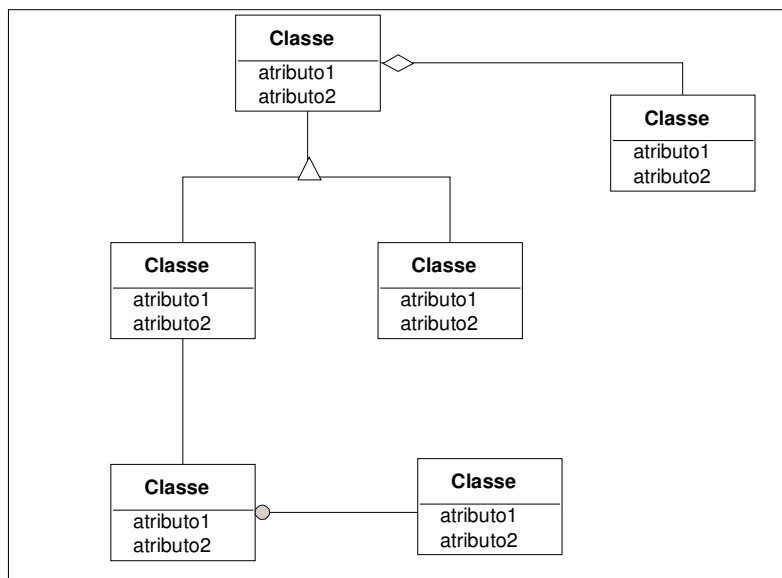
O modelo funcional descreve os aspectos de um sistema relacionados a transformações de valores; funções, mapeamentos, restrições e dependências funcionais. Este modelo abrange o que o sistema faz, independente de como ou quando é feito.

### **4.8.1 ANÁLISE**

O objetivo da análise é desenvolver um modelo do que o sistema irá fazer. Esse modelo é expresso em termos de objeto e relacionamentos, fluxo de controle dinâmico e transformações funcionais. As fases que devem ser seguidas são [RUM94]:

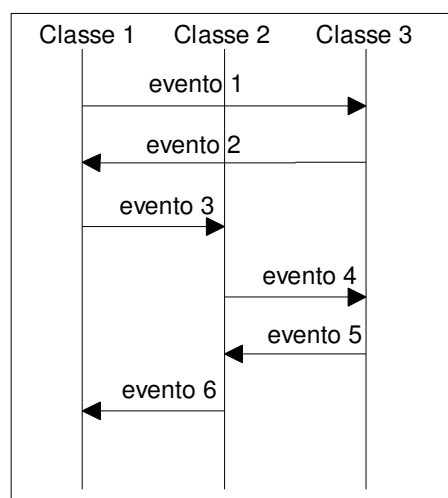
- a) escrever ou obter uma descrição inicial do problema (enunciado);
- b) construir um Modelo de Objetos - identificar as classes, iniciar a geração de um dicionário de dados contendo descrições das classes, atributos e associações, acrescentar as associações entre as classes, acrescentar os atributos para objetos e ligações, organizar e simplificar as classes de objetos utilizando herança, testar os caminhos de acesso utilizando roteiros e repetindo os passos anteriores, se necessário;
- c) desenvolver um modelo dinâmico - preparar roteiros das seqüências típicas de interação, identificar eventos entre objetos e preparar uma seqüência de eventos para cada roteiro, preparar um diagrama de fluxo de eventos para o sistema, desenvolver um diagrama de estados para cada classe que tenha comportamento dinâmico importante, verificar a consistência dos eventos compartilhados pelos diagramas de estados;
- d) construir um modelo funcional - identificar os valores de entrada e saída, utilizar diagramas de fluxo de dados quando necessário para mostrar dependências funcionais, descrever o que cada função faz, identificar as restrições, especificar os critérios de otimização;
- e) verificar, repetir e refinar os três modelos.

Os documentos resultantes da análise são o enunciado do problema, o modelo de objetos (figura 9), o modelo dinâmico (diagrama de eventos para o cenário - figura 10, diagrama de fluxo de eventos - figura 11 e diagrama de estados - figura 12) e o modelo funcional (valores de entrada e saída - figura 13, diagrama de fluxo de dados - figura 14).



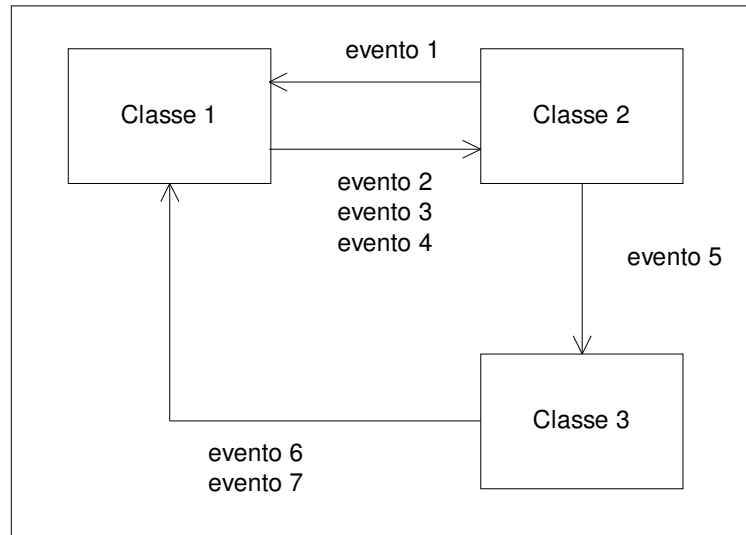
Fonte: adaptado de [RUM94].

Figura 9: Modelo de Objeto.



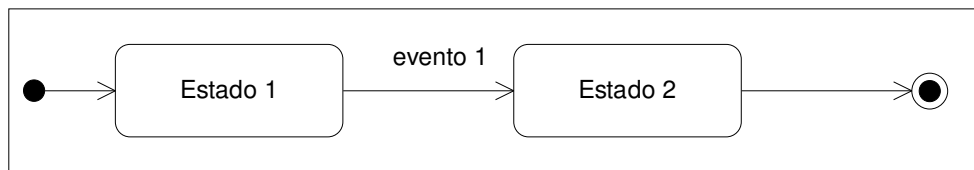
Fonte: adaptado de [RUM94]

Figura 10: Diagrama de eventos.



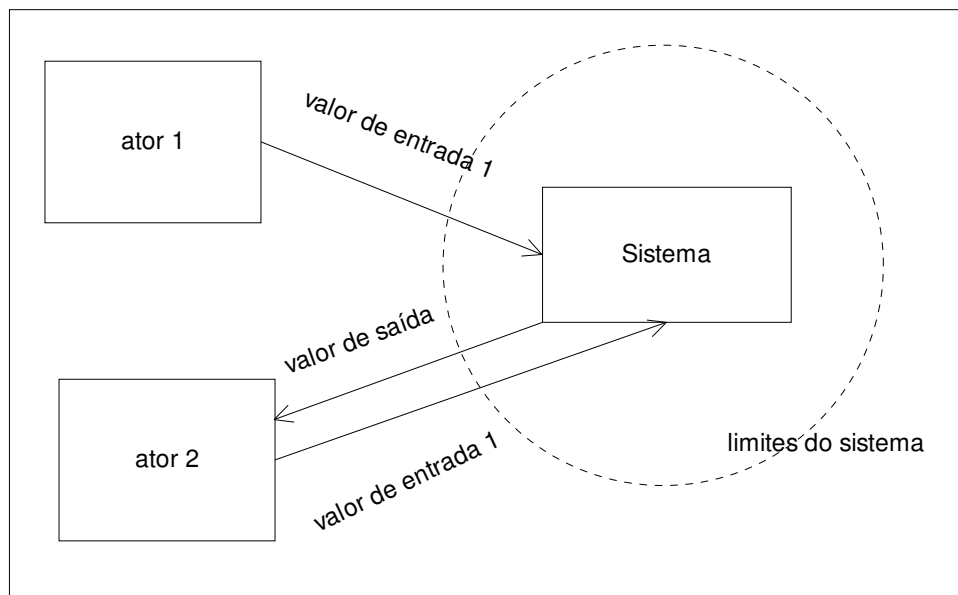
Fonte: adaptado de [RUM94].

Figura 11: Diagrama de fluxo de eventos.



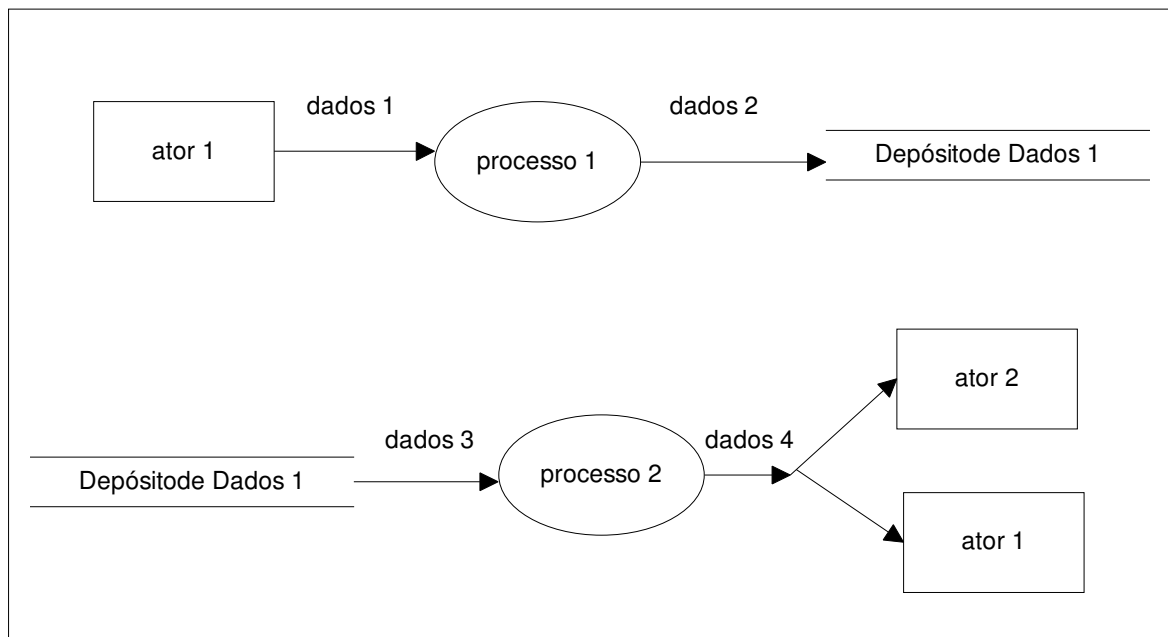
Fonte: adaptado de [RUM94].

Figura 12: Diagrama de estados.



Fonte: adaptado de [RUM94].

Figura 13: Valores de entrada e saída.



Fonte: adaptado de [RUM94] e [NOG99].

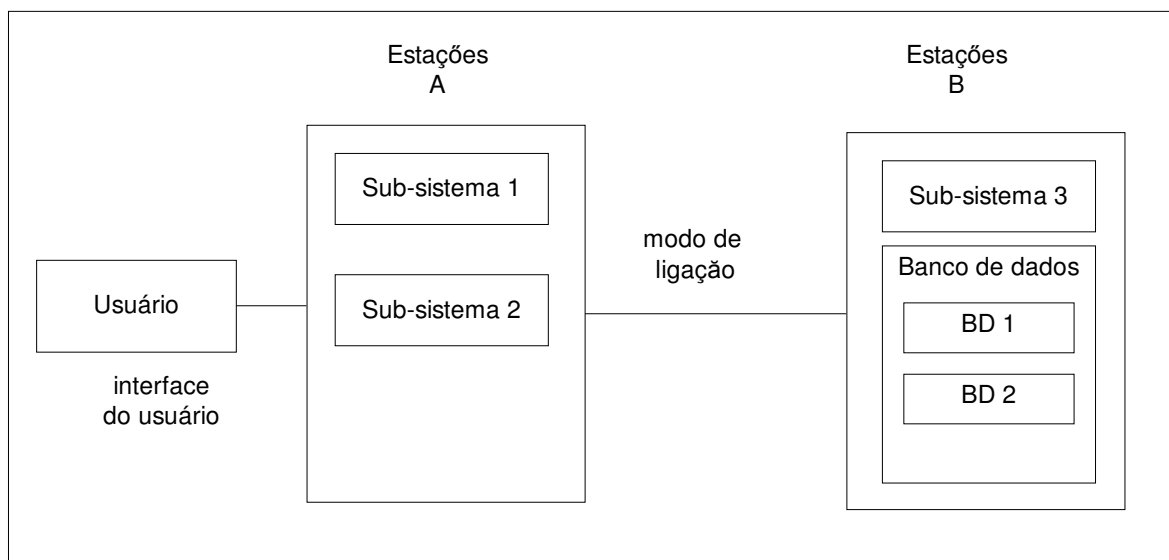
Figura 14 - Diagrama de fluxo de dados.

## 4.8.2 PROJETO DO SISTEMA

Durante o projeto do sistema é determinada a estrutura de alto nível do sistema. As fases que devem ser seguidas são [RUM94]:

- a) organizar o sistema e subsistemas;
- b) identificar ocorrências inerentes ao problema;
- c) alocar os subsistemas a processadores e tarefas;
- d) escolher a estratégia básica para implementação dos depósitos de dados em termos de estrutura de dados, arquivos e bancos de dados;
- e) identificar recursos globais e determinar mecanismos para controlar o acesso a eles;
- f) definir uma abordagem para a implementação do controle de software - utilizar uma área dentro do programa para salvar o estado, ou implementar diretamente uma máquina de estados, ou utilizar tarefas concorrentes;
- g) considerar condições extremas;
- h) estabelecer prioridades.

Os documentos resultantes do Projeto do Sistema são: estrutura da arquitetura básica do sistema (figura 15), decisões estratégicas de alto nível.



Fonte: adaptado de [RUM94].

Figura 15: Arquitetura de um sistema.

### 4.8.3 PROJETO DE OBJETOS

Durante o projeto de objetos é preparado o modelo de análise, sendo elaborada uma base detalhada para a implementação. Tomam-se as decisões necessárias para efetivar o sistema sem entrar nos pequenos detalhes de uma linguagem específica ou de um sistema de banco de dados. O projeto de objetos afasta-se da orientação do mundo real para se aproximar de um modelo orientado para o computador, necessário para a implementação prática. As fases que devem ser seguidas são [RUM94]:

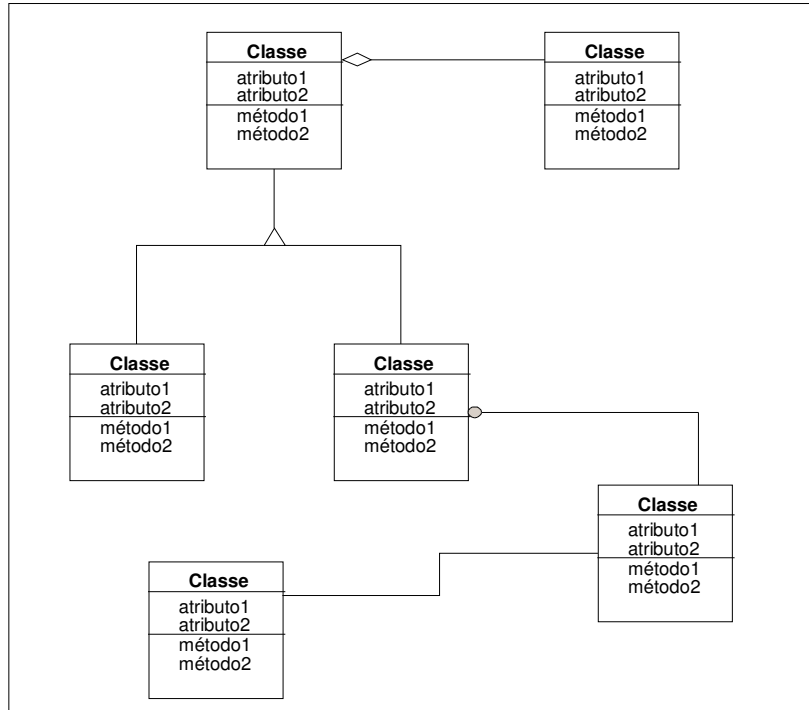
- a) obter as operações para o modelo de objetos a partir de outros modelos - definir uma operação para cada modelo funcional, definir uma operação para cada evento do modelo dinâmico dependendo da implementação de controle;
- b) projetar algoritmos para implementar as operações - escolher algoritmos que minimizem o custo da implementação das operações, selecionar as estruturas de dados adequadas aos algoritmos, definir novas classes internas e operações quando necessário, atribuir responsabilidades para operações que não estejam claramente associadas a uma única classe;
- c) otimizar as vias de acesso aos dados - acrescentar associações redundantes para

diminuir o custo de acesso e maximizar a facilidade, refazer a computação para maior eficiência, salvar valores derivados para evitar que expressões complicadas sejam recalculadas;

- d) implementar controle de software pelo enxugamento da abordagem escolhida durante o projeto do sistema;
- e) ajustar a estrutura de classes para aperfeiçoar a herança - rearranjar e ajustar classes e operações para aperfeiçoar a herança, abstrair os comportamentos comuns dos grupos de classes, utilizar a delegação para compartilhar os comportamentos em que a herança é semanticamente inválida;
- f) projetar a implementação das associações - analisar a travessia das associações, implementar cada associação como um objeto diferente ou acrescentando atributos de valor baseado em objetos a uma ou a ambas as classes da associação;
- g) determinar a representação exata dos atributos de objetos;
- h) arrumar as classes e associações em módulos

Os documentos resultantes do Projeto devem ser uma extensão do documento da Análise. Assim, o documento do projeto incluirá uma descrição revista e muito mais detalhada do Modelo de Objetos (figura 16). O Modelo Funcional também é estendido durante a fase do projeto e mantido atualizado, utilizando a mesma notação da análise, porém com mais detalhes e especificações. Para o Modelo Dinâmico, se foi implementado com utilização de um controle explícito de estados ou tarefas concorrentes, o modelo da análise já será adequado [RUM94].





Fonte: adaptado de [RUM94].

Figura 16: Modelo de Objetos do Projeto.

## 5 TECNOLOGIAS UTILIZADAS

Para este trabalho foram utilizadas as seguintes tecnologias:

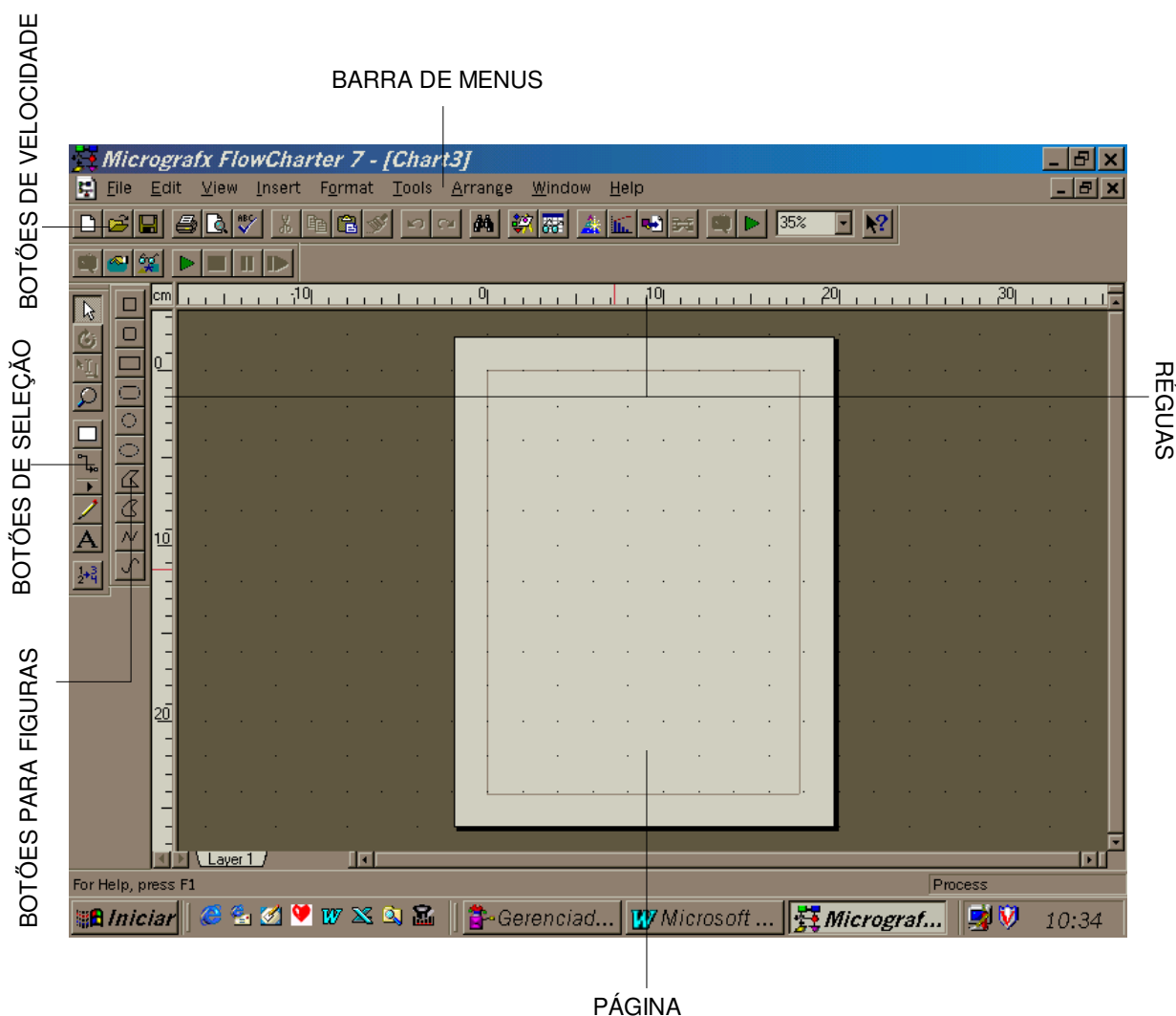
- a) Micrografix FlowCharter 7.0 para plataforma Windows 95 da INSO Corporation, - utilizado na confecção das figuras do trabalho e dos diagramas da análise e projeto;
- b) FoxPro 2.6 for Windows, da Microsoft Corporation utilizado na criação do banco de dados do protótipo;
- c) Ambiente visual Delphi 4 para plataforma Windows 95, da empresa Borland, utilizado na implementação do protótipo.

### 5.1 MICROGRAFX FLOWCHARTER 7

O Micrografix FlowCharter versão 7 para plataforma Windows 95, da INSO Corporation, é uma ferramenta gráfica que permite criar uma grande variedade de diagramas. Mesmo não oferecendo nenhum suporte a Análise Orientada a Objetos foi usado no trabalho para a confecção das figuras e dos diagramas da análise e projeto do protótipo. Seus elementos principais são [INS96]:

- a) barra de menus - menu padrão Windows;
- b) botões de velocidade - botões tipo apontar e clicar em comando de menu selecionado, por exemplo, para salvar um arquivo existe o botão "save", que aparece com a figura de um disquete;
- c) botões de seleção - usados para selecionar o que será feito, por exemplo, pressionando no botão com "A" poderá ser digitado um texto, no botão com uma setinha, uma seta de ligação entre duas figuras;
- d) botões para figuras - a figura que estiver no botão selecionado aparecerá na página. Clicando em uma das figuras, automaticamente será selecionado o botão de figura (representado por um retângulo) dos botões de seleção;
- e) régua - auxiliar no alinhamento das figuras;
- f) página - é a página onde será desenhada a figura.

A figura 17 apresenta o Flow sendo executado sob o Windows 95, apontando cada um dos elementos principais acima citados.



Fonte: [INS96].

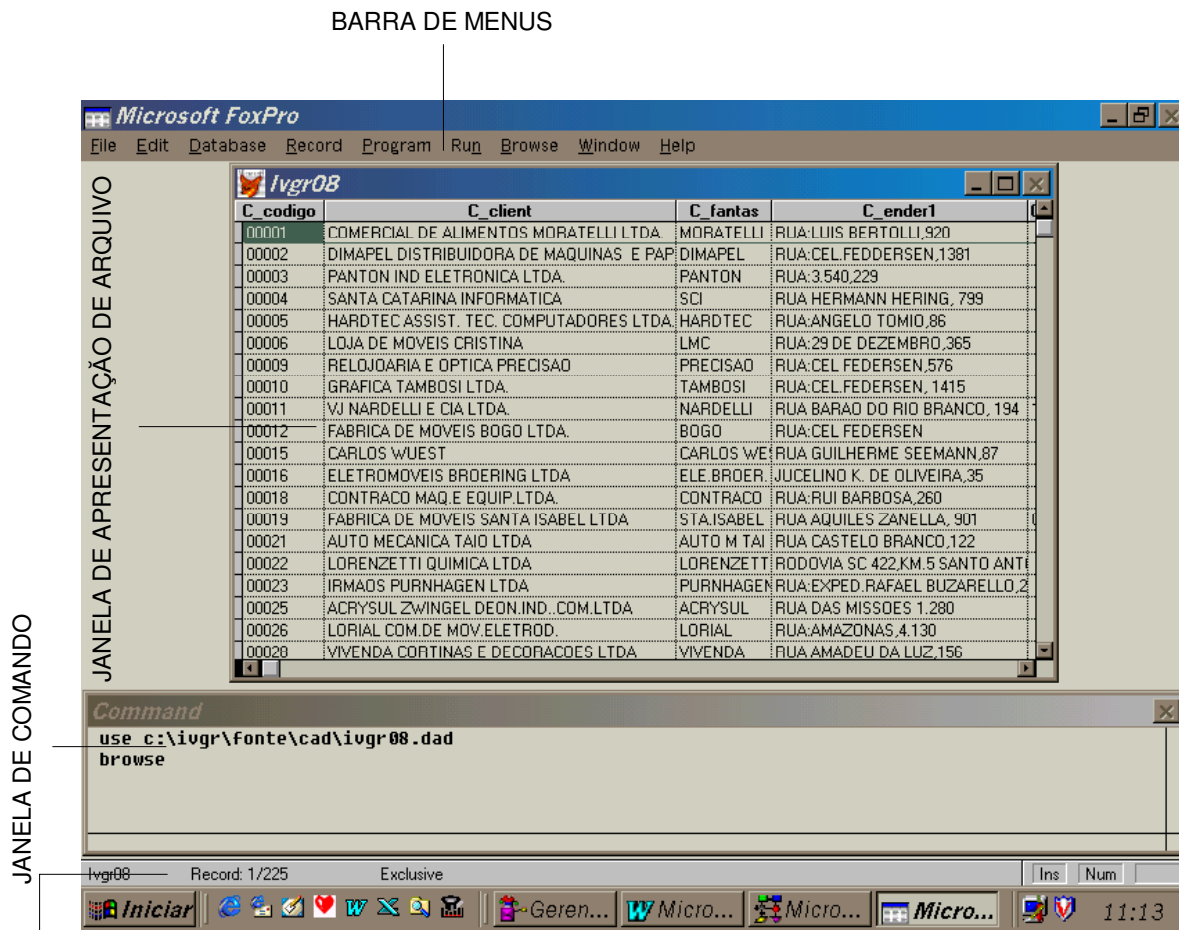
Figura 17: Micrografix FlowCharter sendo executado sob o Windows 95.

## 5.2 FOXPRO 2.6

O ferramenta escolhida para a manipulação da base de dados a ser utilizada no protótipo deste trabalho é o FoxPro na versão 2.6 para plataforma Windows, da Microsoft Corporation. Os principais elementos apresentados na figura 18 são [CHO95]:

- a) barra de menus - menu padrão estilo Windows;
- b) janela de comando - os comandos podem ser digitados na janela, ou selecionados no menu;
- c) barra de status - apresenta informações como o nome do arquivo e registro;

- d) janela de apresentação de arquivo - apresenta o arquivo selecionado no comando "use", quando executado o comando "browse".



BARRA DE STATUS

Fonte: adaptado de [CHO95].

Figura 18: FoxPro 2.6 executando sob o Windows 95.

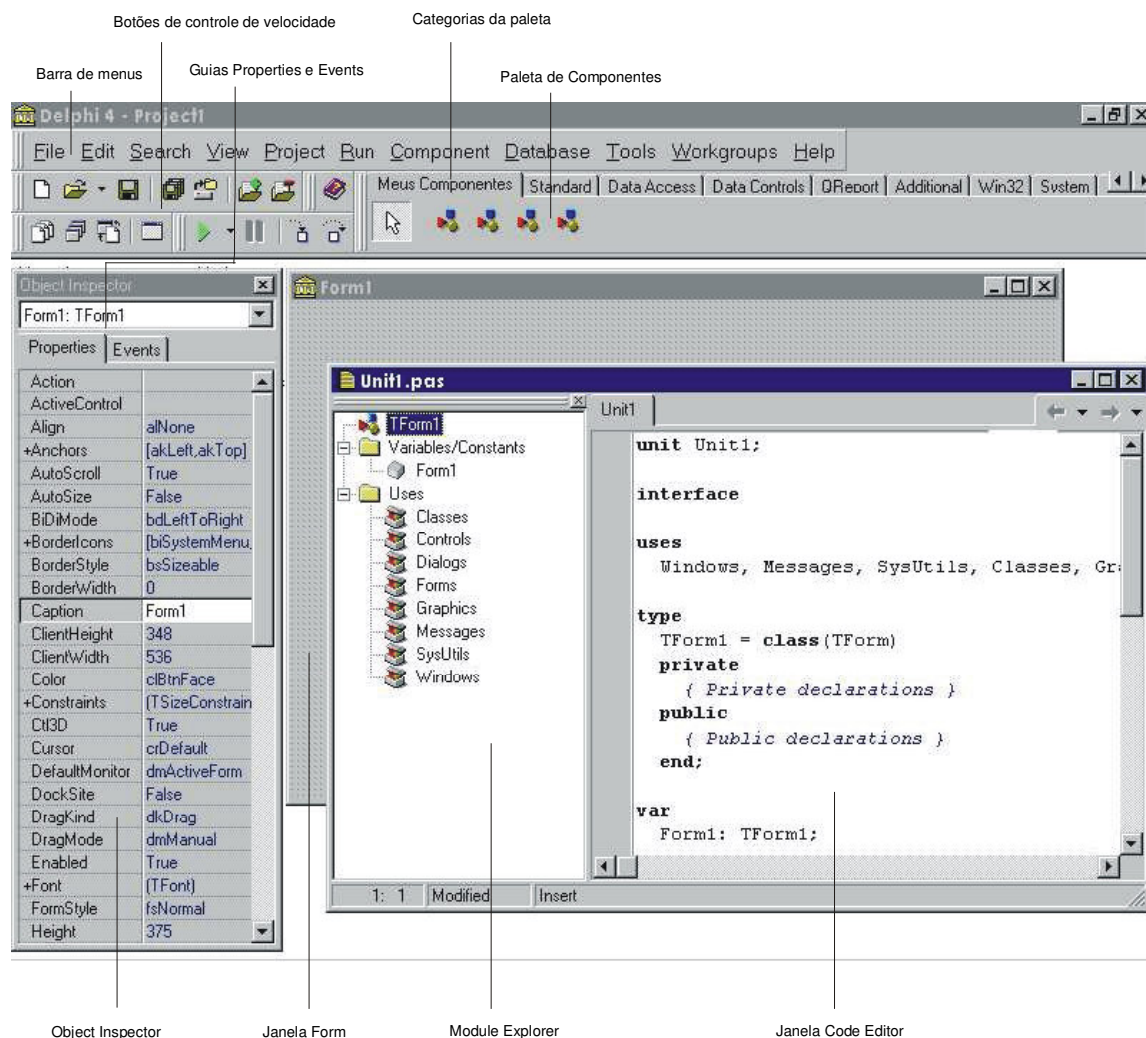
## 5.3 AMBIENTE VISUAL – DELPHI 4

O Delphi na versão 4, para plataforma Windows 95, da empresa Borland, é usado na implementação do protótipo deste trabalho. De acordo com [SWA99], o ambiente de desenvolvimento integrado do Delphi possui vários elementos principais. São eles:

- a) Botões de controle de velocidade – botões do tipo apontar-e-clicar em comando de menu selecionados. Exemplo: para salvar o projeto, tem o botão Save Project, que aparece com um conjunto de disquetes.

- b) Barra de menus - menu padrão estilo Windows.
- c) Paleta de componentes – contém ícones que representam os componentes VCL.
- d) Categorias de paleta – páginas que contém categorias de componentes. Clica-se nas guias de cada categoria, mostradas acima da paleta VCL para visualizar os componentes dessa categoria.
- e) Guias Properties e Events – no Object inspector há duas páginas. Uma com Properties, que apresenta as propriedades de um componente ou formulário. Outra com Events, que apresenta os eventos do componente ou formulário.
- f) Object Inspector – exibe todas as propriedades e os eventos para um ou mais componentes selecionados ou formulários.
- g) Janela Form – a representação gráfica de uma janela, o formulário.
- h) Module Explorer – mostra as classes do módulo atual, uma lista de outras unidades utilizadas por esta, variáveis, objetos, métodos e outras informações.
- i) Janela Code Editor – exibe o código-fonte do Pascal associado a cada formulário no aplicativo.

A figura 19 apresenta o Delphi sendo executado sob o Windows 95, apontando cada um dos elementos principais acima citados.



Object Inspector  
Fonte: [SWA99].

Janela Form

Module Explorer

Janela Code Editor

Figura 19: O Delphi sendo usado sob o Windows 95.

## **6 DESENVOLVIMENTO DO PROTÓTIPO DE SISTEMA DE CONTROLE DA UTILIZAÇÃO DE RECURSOS PARA PRODUÇÃO DAS INDÚSTRIAS DE MANUFATURA**

O protótipo tem como finalidade demonstrar a programação da produção para as indústrias de manufatura utilizando as técnicas de PERT-CPM. Para isso deverá ser analisada a disponibilidade de recursos (máquinas / operações e produtos). Das informações da programação resultarão ordens de fabricação, gráfico da programação, relatórios de necessidade de materiais, de ocupação de máquinas e de datas de entrega para os pedidos.

O protótipo usará do aplicativo iVgR os dados referentes aos produtos e pedidos de venda.

### **6.1 ESPECIFICAÇÃO DO SISTEMA**

De acordo com [RUM94], os documentos referentes à modelagem orientada a objetos são os seguintes: para o modelo estático, a descrição e o modelo de objetos; para o modelo dinâmico, o diagrama de estados, o diagrama de eventos e os cenários e o diagrama de fluxo de eventos; para o modelo funcional, o diagrama com valores de entrada e saída e o diagrama de fluxo de dados. Esses documentos são requeridos tanto na fase de análise de objetos quanto na de projeto de objetos, sendo que na fase de análise os documentos têm características mais parecidas com os objetos do mundo real e na fase de projeto os documentos têm características mais parecidas com a implementação em si. Os documentos da análise que já contemplem os passos necessários para a implementação são dispensados da fase de projeto. No trabalho o modelo de objetos será diferenciado na análise e projeto, e os outros documentos já estarão preparados para a implementação. A fase de projeto do sistema tem como documento a arquitetura do sistema.

#### **6.1.1 DESCRIÇÃO**

O protótipo será dividido em 3 subsistemas: o de Programação (PCP Programação), o de Apontamentos de Produção (PCP Apontamentos) e o de relatórios (PCP Relatórios). Os dados de Pedidos e Produtos e Operações são recuperados do aplicativo iVgR - Gerenciamento Empresarial.

O iVgR Gerenciamento Empresarial é um aplicativo desenvolvido pela Fonte Sistemas de Informática, de Rio do Sul, que controla as rotinas gerenciais, financeiras e administrativas da empresa.

O subsistema de Programação deverá detectar quando um pedido for incluído ou excluído no aplicativo, e a partir disso fazer a programação da produção planejando a utilização de máquinas e as necessidades de materiais baseada na técnica de redes PERT-CPM. Deverá detectar quando um pedido for incluído ou excluído no aplicativo.

Quando um pedido for incluído, deve fazer o cálculo de como serão distribuídos os processos referente aos produtos do pedido, e gerará ordens de fabricação. O pedido estará incluído na programação da produção na última posição.

Quando um pedido for excluído, o pedido deve ser tirado da programação da produção, e os pedidos subsequentes deverão ser reprogramados.

Deverá ainda permitir que o usuário mude a seqüência da fabricação de determinado pedido. Para isso o protótipo deverá fazer a reprogramação.

O subsistema de Apontamentos deverá emitir ordens de fabricação com os dados do subsistema de Programação e receber dados de quando uma etapa foi concluída e marcar a programação como concluída.

O subsistema de Relatórios deverá emitir relatórios de necessidade de materiais, de utilização de máquinas, de datas de entrega e apresentar o Gráfico de Gantt com os dados da programação.



### 6.1.2 MODELO DE OBJETOS DA ANÁLISE

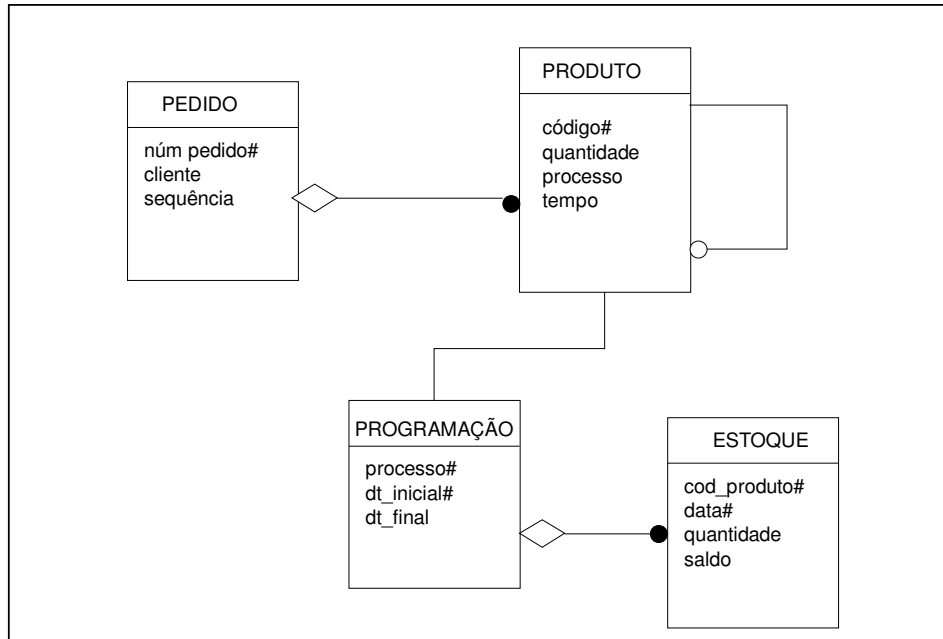


Figura 20: Modelo de Objeto do Protótipo

### 6.1.3 DIAGRAMA DE ESTADOS DA ANÁLISE - SUBSISTEMA PROGRAMAÇÃO DA PRODUÇÃO

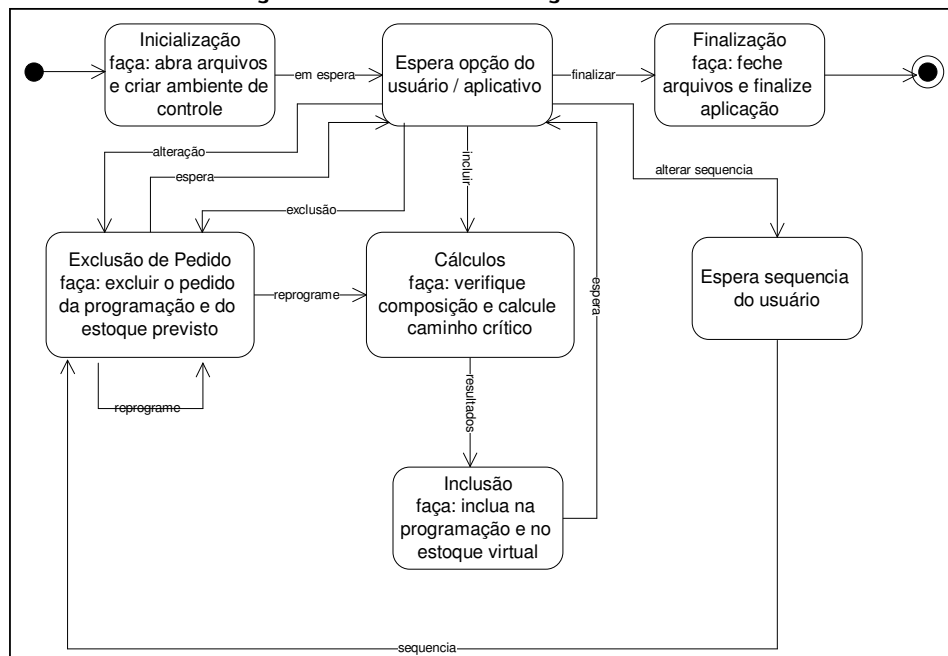


Figura 21: Diagrama de Estados - subsistema programação

### 6.1.4 DIAGRAMA DE ESTADOS DA ANÁLISE -

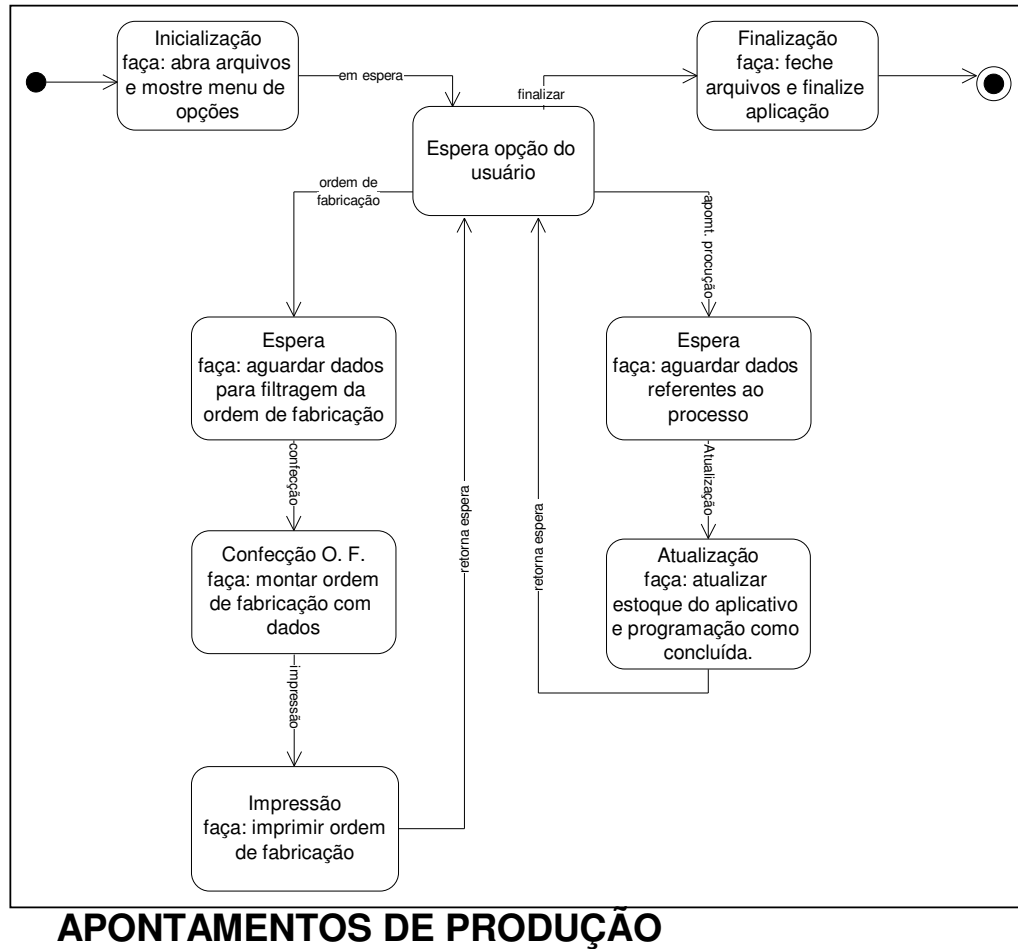


Figura 22: Diagrama de estados - subsistema apontamentos

### 6.1.5 DIAGRAMA DE ESTADOS DA ANÁLISE - SUBSISTEMA RELATÓRIOS

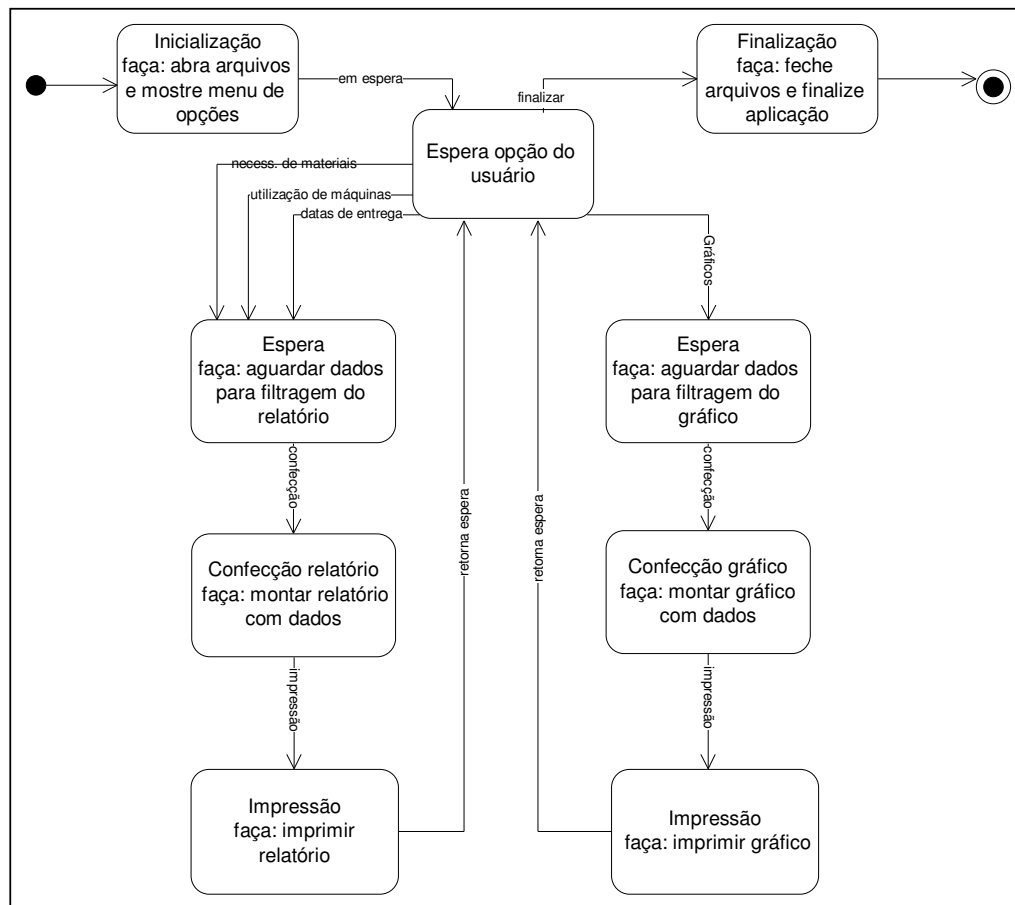


Figura 23: Diagrama de estados - subsistema relatórios

### 6.1.6 DIAGRAMA DE EVENTOS DA ANÁLISE

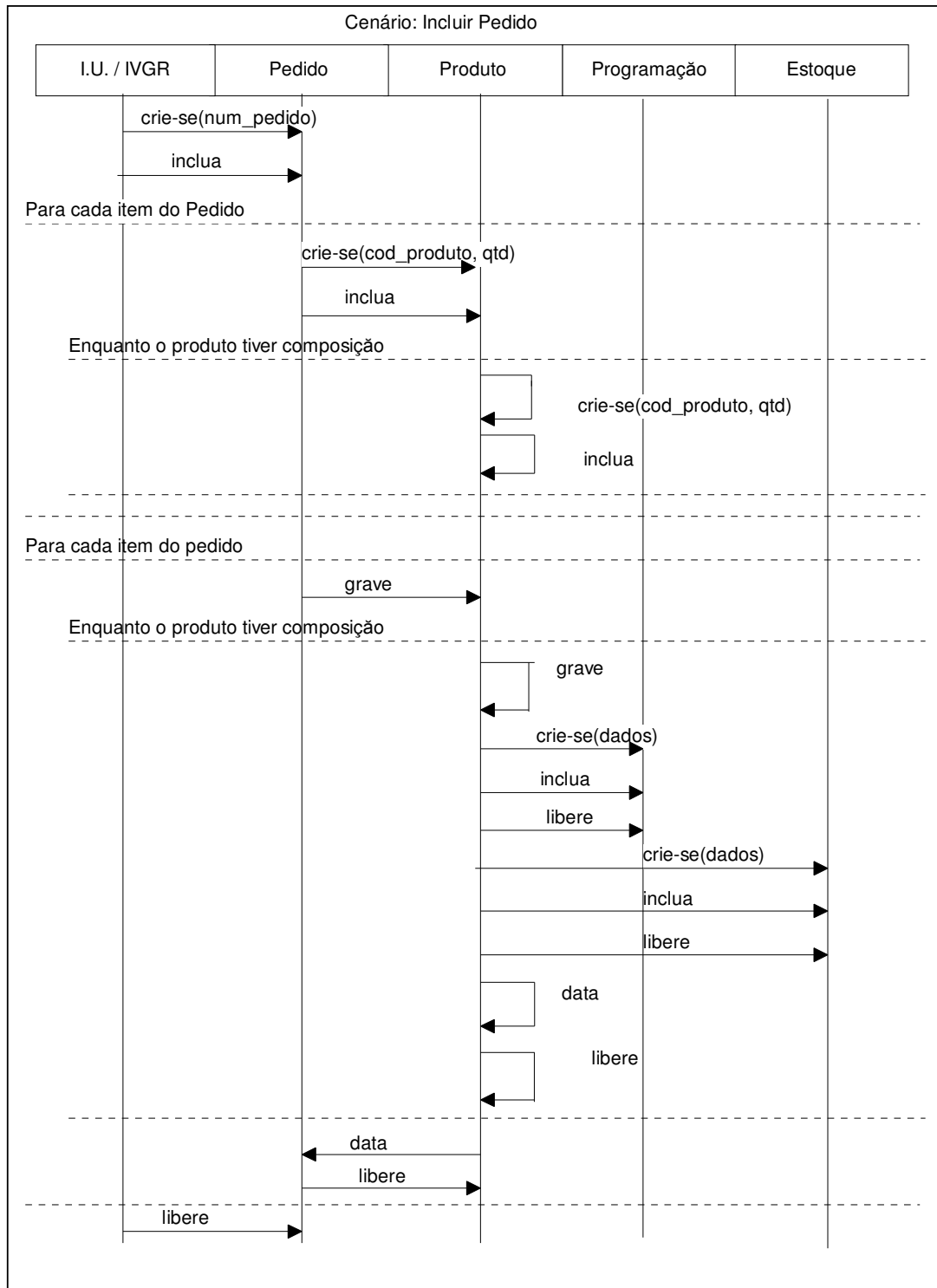


Figura 24: Diagrama de eventos - subsistema programação

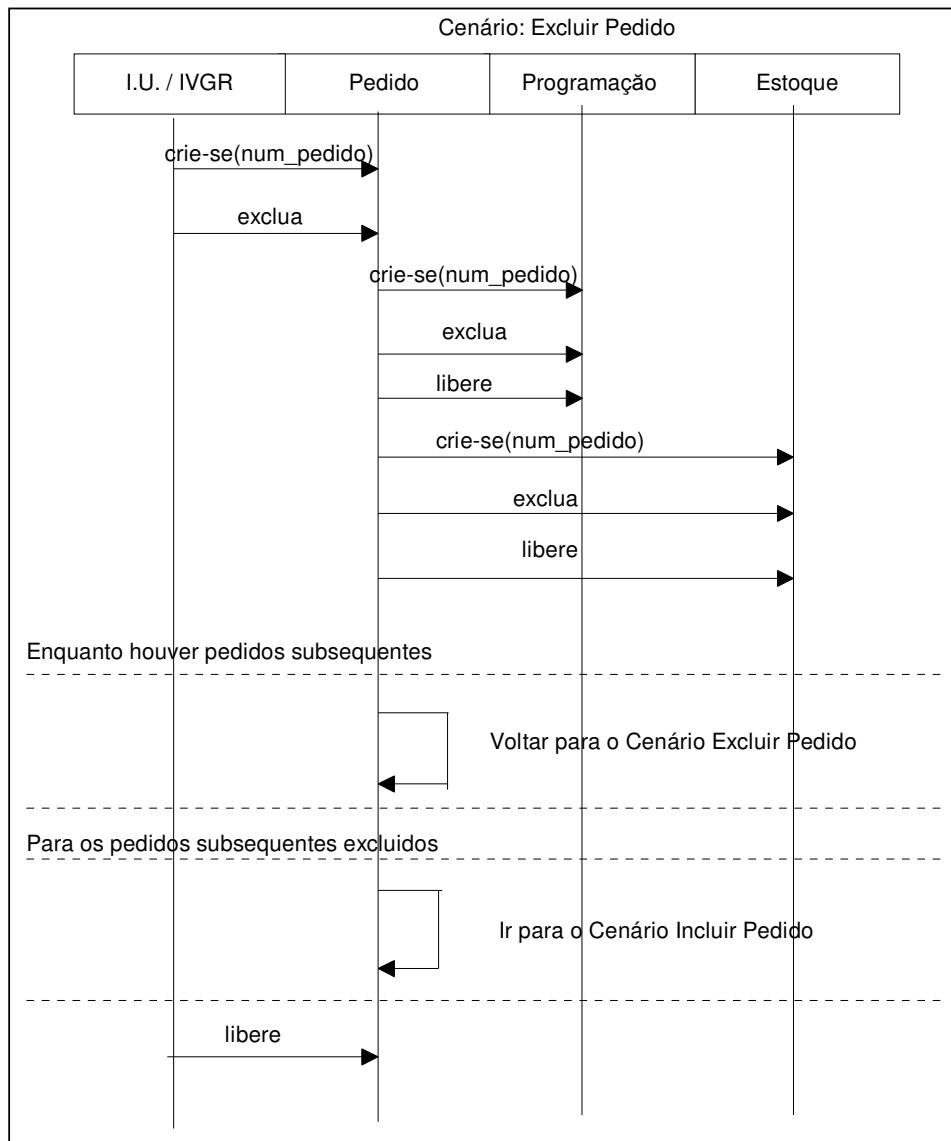


Figura 25: Diagrama de eventos - subsistema programação (continuação)

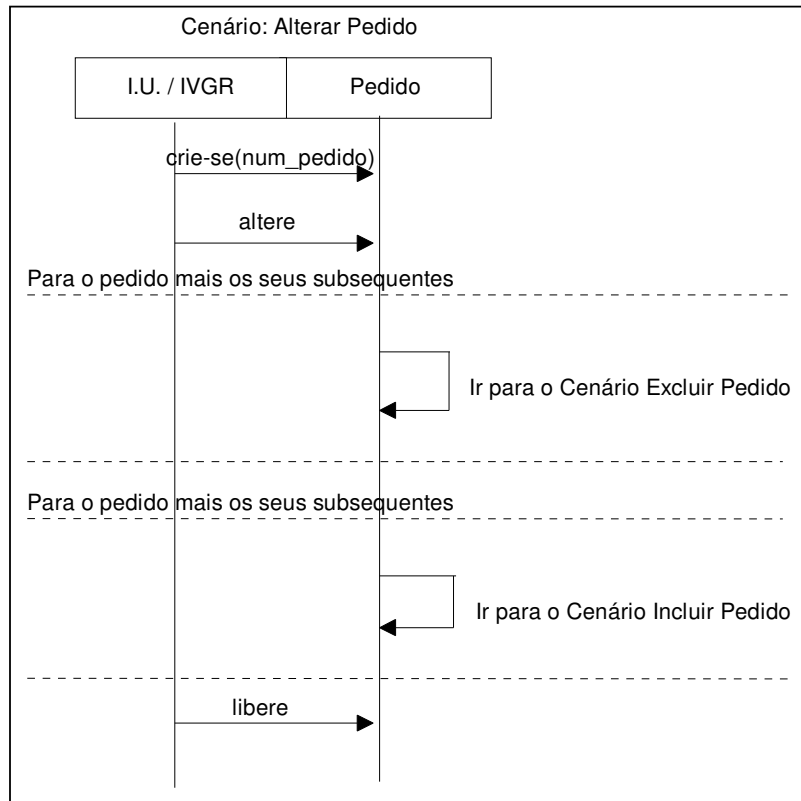


Figura 26: Diagrama de eventos - subsistema programação (continuação)

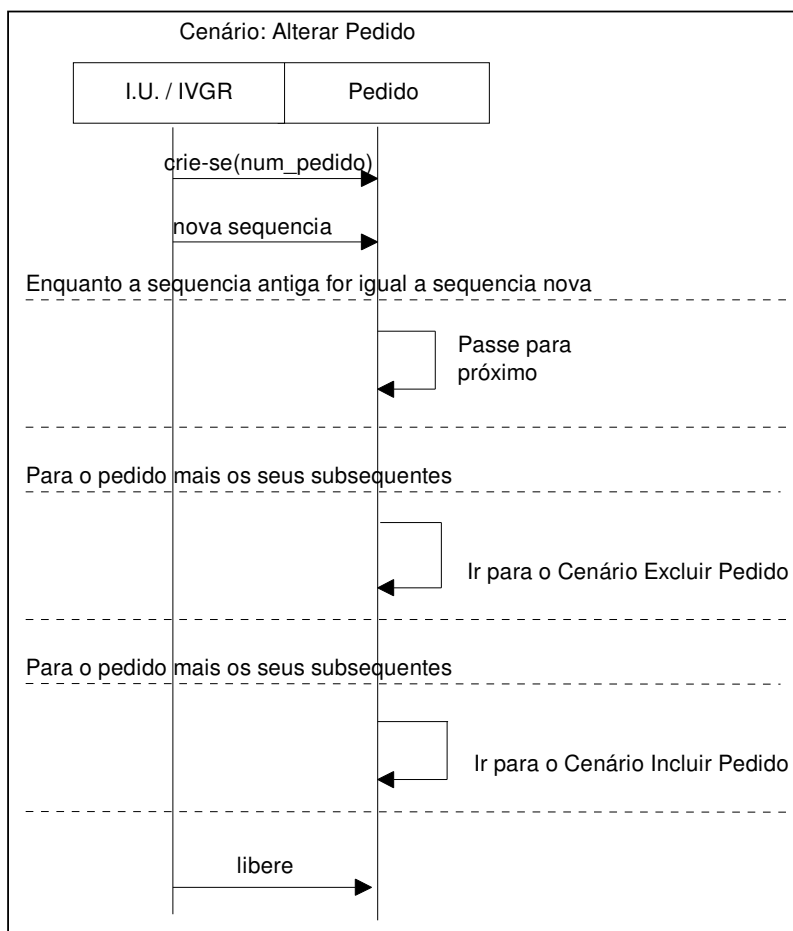


Figura 27: Diagrama de eventos - subsistema programação (continuação)

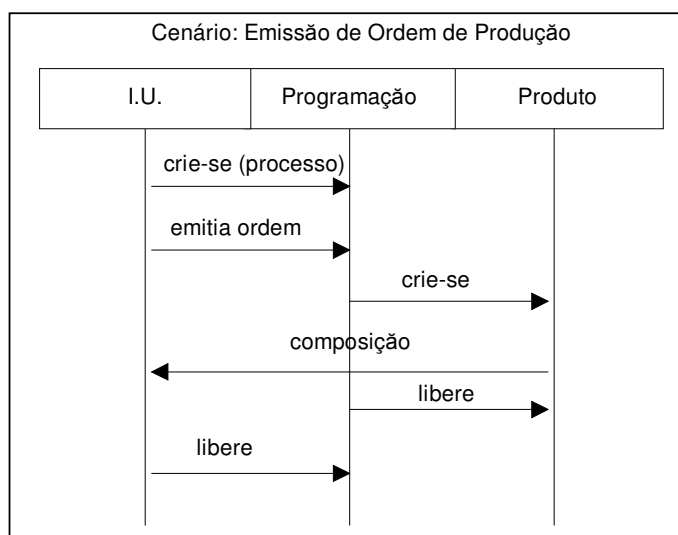


Figura 28: Diagrama de eventos - subsistema apontamentos

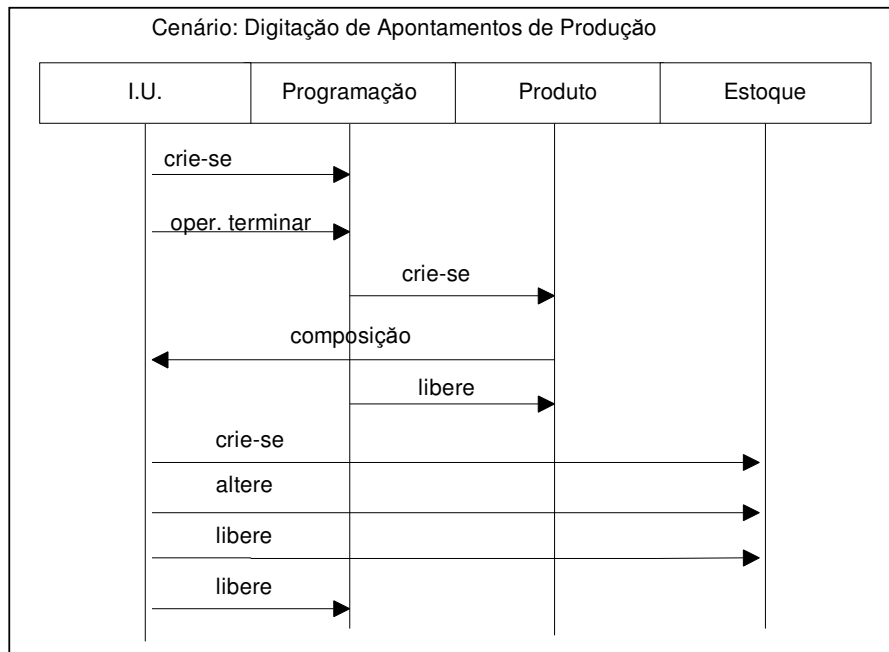


Figura 29: Diagrama de eventos - subsistema apontamentos (continuação)

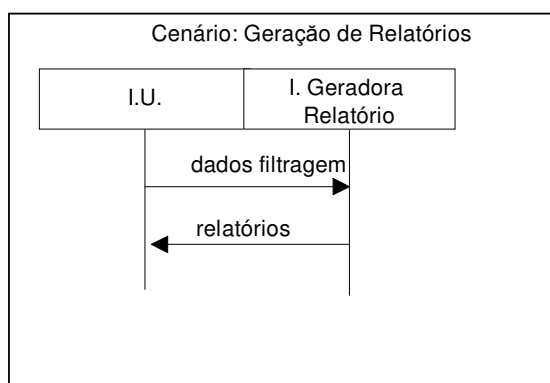


Figura 30: Diagrama de eventos - subsistema gerador

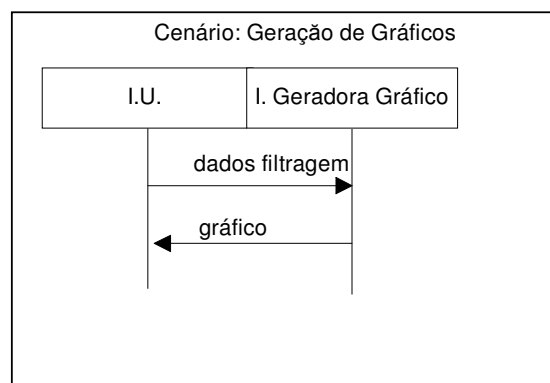


Figura 31: Diagrama de eventos - subsistema gerador (continuação)



### 6.1.7 DIAGRAMA DE FLUXO DE EVENTOS DA ANÁLISE

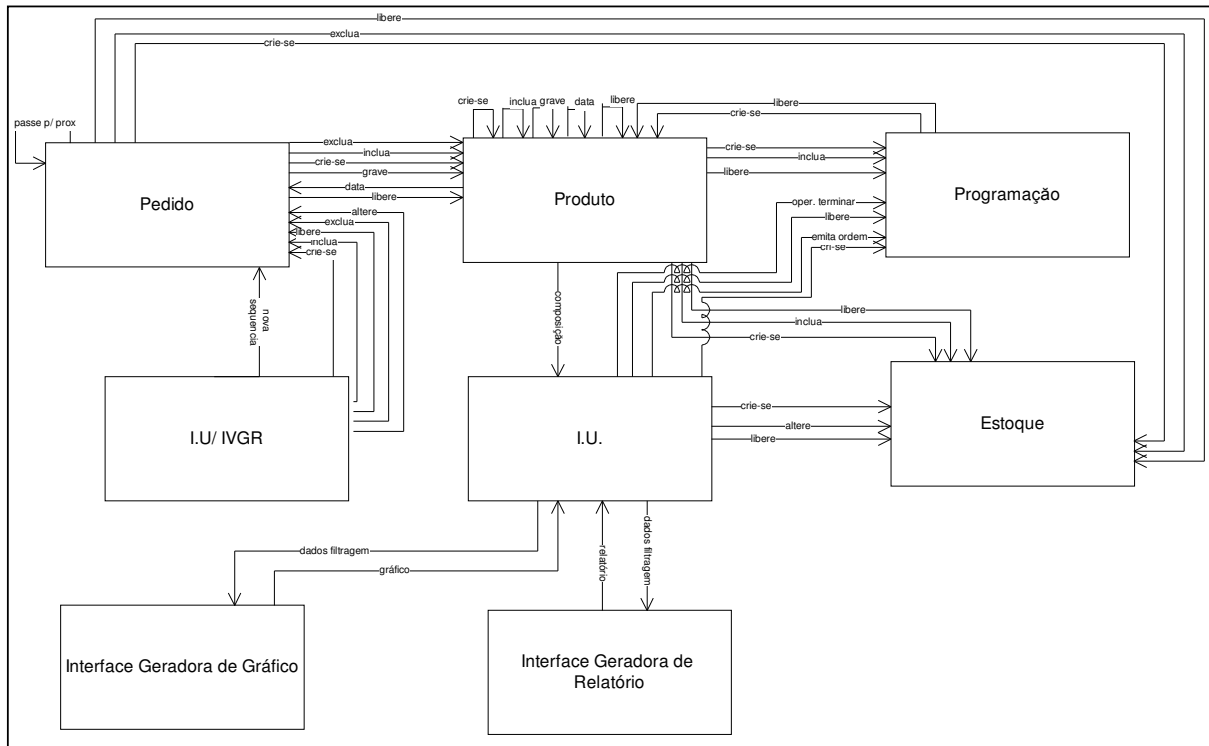


Figura 32: Diagrama de fluxo de eventos do protótipo

## 6.1.8 VALORES DE ENTRADA E SAÍDA DA ANÁLISE

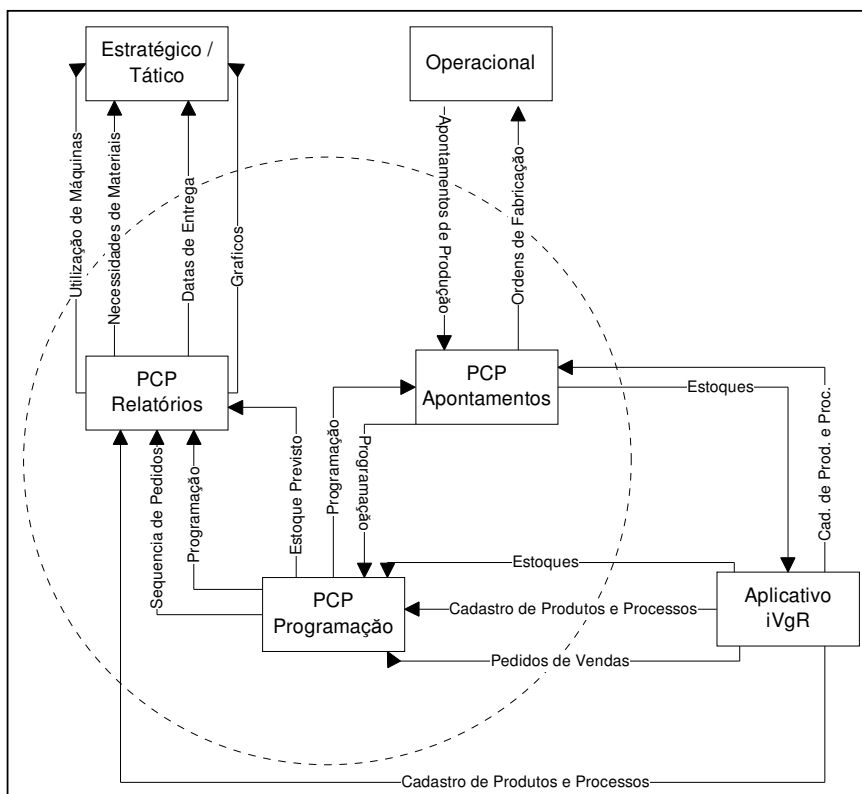


Figura 33: Valores de entrada e saída do protótipo

### 6.1.9 DIAGRAMA DE FLUXO DE DADOS DA ANÁLISE

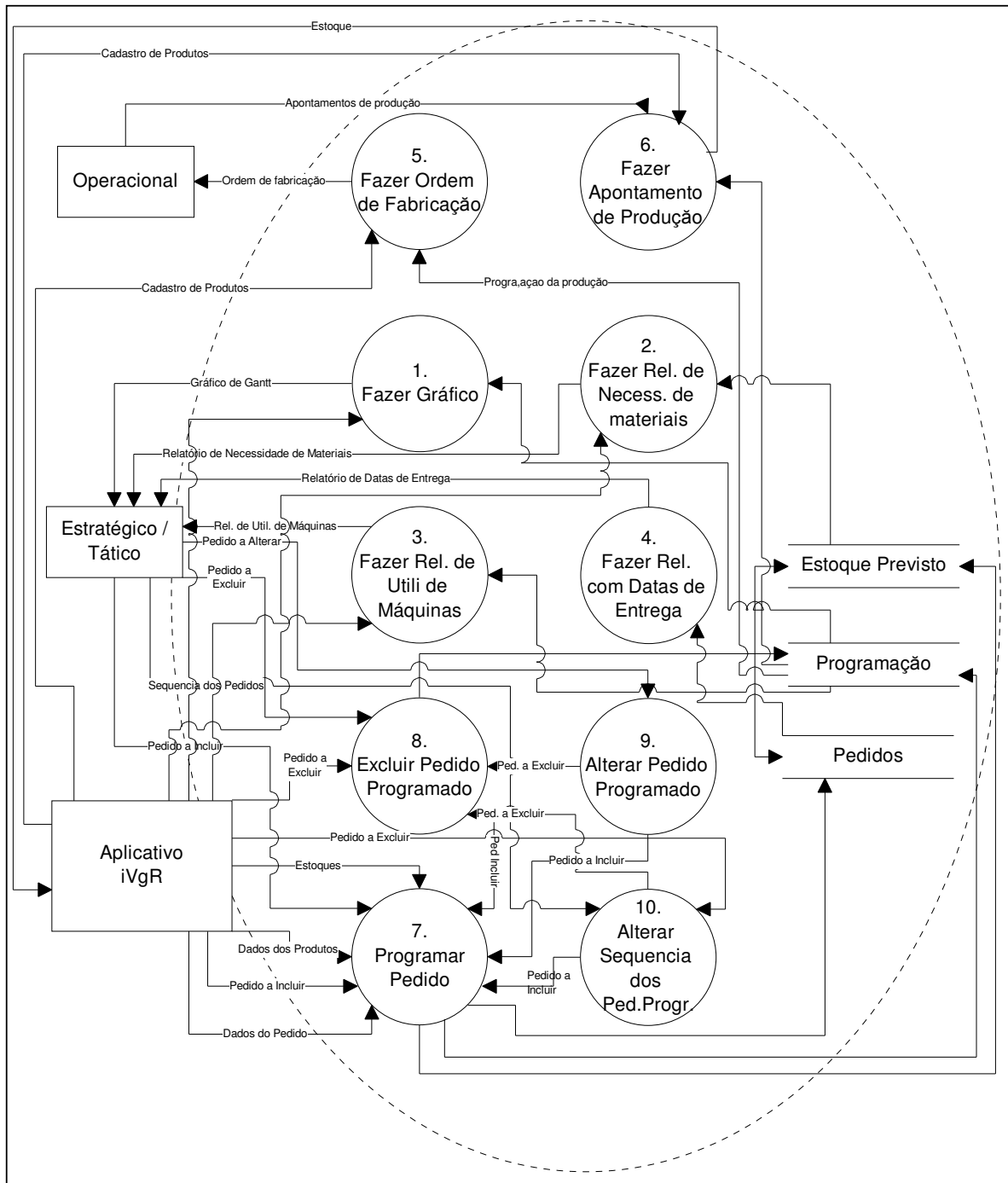


Figura 34: Diagrama de fluxo de dados do protótipo.

### 6.1.10 ARQUITETURA DO SISTEMA - PROJETO DO SISTEMA

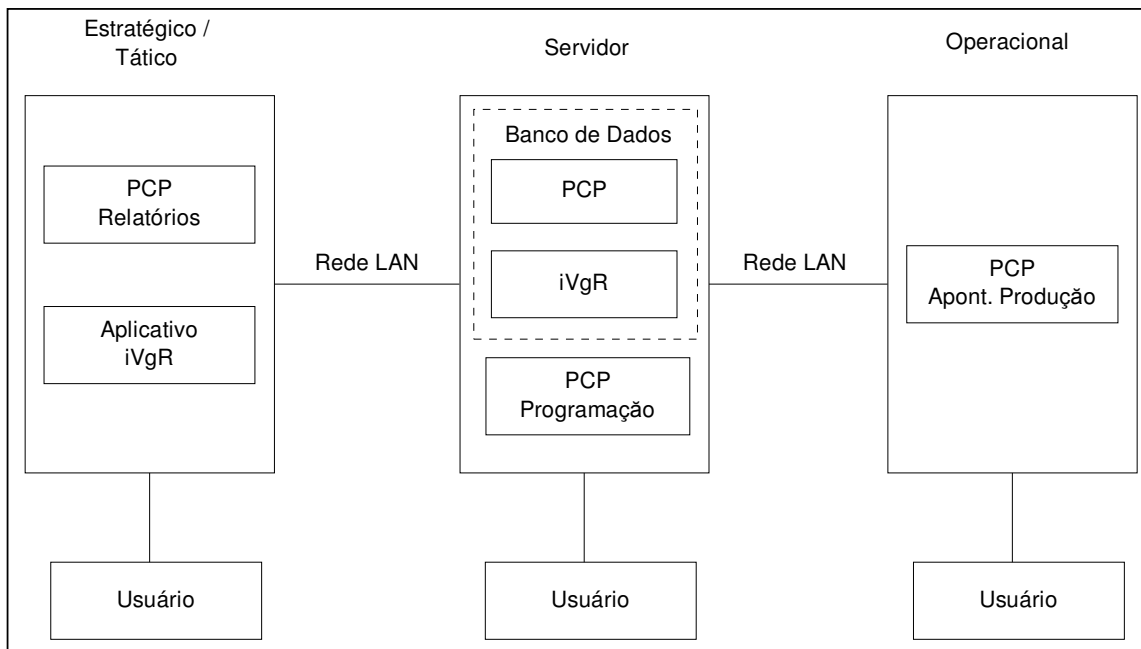


Figura 35: Arquitetura do sistema do protótipo, do projeto de sistema.

### 6.1.11 MODELO DE OBJETOS - PROJETO DE OBJETOS

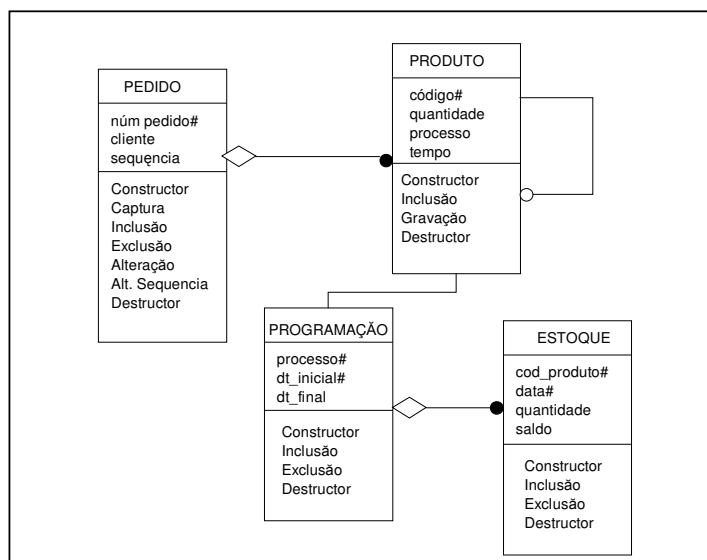


Figura 36: Modelo de objetos do projeto do protótipo.

Para os demais modelos do projeto, os modelos desenvolvidos na análise já contemplam os passos necessários para a implementação.

## 6.1.12 IMPLEMENTAÇÃO DAS CLASSES DE OBJETOS

Unit Tipos;

interface

uses

Classes, DB, DMlvgr, DMDados, SysUtils;

Type

```

TProduto = Class(TObject)
  CodRed: Real;
  Tipo: String[1];
  Quant: Real;
  Processo: Real;
  QuantUP: Real;
  UPsAcum: Real;
  Composicao: TList;
  Constructor construtor(CR: Real; QT: Real; TP: String);
  Procedure Incluir;
  Procedure Excluir;
  Function Gravar(TempoInicial: Real; Doc: String): Real;
  Procedure CalculaUPs;
private
public
end;
TPedido = class(TObject)
  SerNum: String[8];
  Produtos: TList;
  Cliente: String[50];
  Constructor Construtor(Ped: String);
  Destructor Destrutor;
  Procedure Incluir(CR: Real; QT: Real);
  Function Gravar: Real;
  Procedure ExcluiPedido;
  Procedure CapturaPedido;
  Procedure AlterarPedido;
private
public
end;

```

implementation

```

constructor TProduto.Construtor(CR: Real; QT: Real; TP: String);
begin
  Codred := CR;
  if TP = 'M' then

```

```

        Tipo := '0'
    else
        Tipo := TP;
    Quant := QT;
    Composicao := Tlist.Create;
    Processo := 0;
    UPsAcum := 0;
end;

```

```

Procedure TProduto.Excluir;
var
    cont: integer;
    Prod: TProduto;
begin
    if Composicao.Count > 0 then
        begin
            for cont := 0 to Composicao.Count -1 do
                begin
                    Prod := Composicao[cont];
                    Prod.Excluir;
                    Prod.Free
                end;
            end;
            Composicao.Free;
        end;
end;

```

```

procedure TProduto.Incluir;
var
    Cont: integer;
    Prod: TProduto;
    Registro: TBookmark;
begin
    Ivgr.TComposto.SetKey;
    Ivgr.TComposto.FieldName('C_CODRED').Value := CodRed;
    if Ivgr.TComposto.GotoKey then
        begin
            repeat
                Registro := Ivgr.TComposto.GetBookmark;
                Ivgr.TProdutos.SetKey;
                Ivgr.TProdutos.FieldName('C_CODRED').Value :=
Ivgr.TComposto.FieldName('C_CODREC').Value;
                If Ivgr.TProdutos.GotoKey then
                    begin
                        if Ivgr.TProdutosC_TIPPRO.Value = 'S' then
                            begin
                                Processo := Ivgr.TComposto.FieldName('C_CODREC').Value;

```

```

QuantUP := Ivgr.TComposto.FieldByName('C_QUANTI').Value *
Quant;
    end
    else
    begin
    Prod := Produto.construtor ( Ivgr.TComposto.FieldByName
        ('C_CODREC').Value;
        Ivgr.TComposto.FieldByName('C_QUANTI').Value * Quant;
        Ivgr.TProdutos.FieldByName('C_TIPPRO').Value);
    Prod.Incluir;
    Composicao.Add(Prod);
    end;
    end;
    Ivgr.TComposto.GotoBookmark(Registro);
    Ivgr.TComposto.Next;
    until (Ivgr.TComposto.FieldByName('C_CODRED').Value <> CodRed) or
(Ivgr.TComposto.Eof);
    end;
end;

```

```

Function TProduto.Gravar(tempoInicial: Real; Doc: String): Real;
var
    TempoMenor, TempoMaior, TempoAux: Real;
    cont: integer;
    Prod: TProduto;
    Pronto: Boolean;
begin
    if composicao.Count > 0 then
    begin
        TempoMaior := TempoInicial;
        for cont := 0 to Composicao.Count -1 do
        begin
            prod := composicao[cont];
            TempoAux := prod.Gravar(TempoInicial, Doc);
            if TempoAux > TempoMaior then
            begin
                TempoMaior := TempoAux;
            end;
        end;
        TempoMenor := TempoMaior;
        TempoMaior := TempoMenor + QuantUp;
        Dados.Programacao.SetKey;
        Dados.Programacao.FieldByName('CODMAQ').Value := Processo;
        Dados.Programacao.FieldByName('TEMPOINI').Value := TempoMaior;
        Dados.Programacao.GotoNearest;
        Pronto := false;
    repeat

```

```

if Dados.ProgramacaoCODMAQ.Value = Processo then
begin
  if not Dados.Programacao.Bof then
  begin
    Dados.Programacao.Prior;
    if Dados.ProgramacaoCODMAQ.Value = Processo then
    begin
      if Dados.ProgramacaoTEMPOFIM.Value < TempoMenor then
      begin
        Pronto := True;
      end
      else
      begin
        Dados.Programacao.Next;
        TempoMenor := Dados.ProgramacaoTEMPOFIM.Value;
        TempoMaior := TempoMenor + QuantUP;
        if Dados.Programacao.Eof then
          Pronto := True
        else
          Dados.Programacao.Next;
        end;
      end
      else
      begin
        Pronto := True;
      end;
    end
    else
    begin
      if Dados.ProgramacaoTEMPOINI.Value < TempoMaior then
      begin
        pronto := true;
      end
      else
      begin
        TempoMenor := Dados.ProgramacaoTEMPOFIM.Value;
        TempoMaior := TempoMenor + QuantUP;
        Dados.Programacao.Next;
      end;
    end;
  end
  end
  else
  begin
    Pronto := True
  end;
until (Pronto = True);
Dados.Programacao.append;
Dados.ProgramacaoCODRED.Value := CodRed;

```



```

    Dados.ProgramacaoQUANT.Value := Quant;
    Dados.ProgramacaoSITUACAO.Value := 'P';
    Dados.ProgramacaoCODMAQ.Value := Processo;
    Dados.ProgramacaoDOCUMENTO.Value := Doc;
    Dados.ProgramacaoTEMPOINI.Value := TempoMenor;
    Dados.ProgramacaoTEMPOFIM.Value := TempoMaior;
    Dados.Programacao.Post;
    for cont := 0 to composicao.Count -1 do
    begin
        Prod := Composicao[cont];
        Dados.AlterarEstoque(Prod.CodRed, Doc, (Prod.Quant * -1),
TempoMenor, Prod.Tipo);
        end;
        Dados.AlterarEstoque(CodRed, Doc, Quant, TempoMaior, Tipo);
        Gravar := TempoMaior;
    end
    else
    begin
        Gravar := TempoInicial;
    end;
end;

Procedure TProduto.CalculaUPs;
Var
    Prod: TProduto;
    Acumulador: Real;
    cont: integer;
Begin
    Acumulador := 0;
    if Composicao.Count > 0 then
    begin
        for cont := 0 to Composicao.Count -1 do
        begin
            Prod := Composicao[cont];
            If Prod.Composicao.Count > 0 then
            begin
                Prod.CalculaUPs;
            end;
            Acumulador := Acumulador + Prod.UPsAcum + Prod.QuantUP;
        end;
        UPsAcum := Acumulador;
    end;
end;

Constructor TPedido.construtor(Ped: String);
begin

```

```

    SerNum := Ped;
    Produtos := TList.Create;
end;

```

```

Destructor TPedido.Destrutor;
var
    cont: integer;
    Prod: Tproduto;
begin
    if Produtos.Count > 0 then
        for cont := Produtos.Count - 1 downto 0 do
            begin
                Prod := Produtos[cont];
                Prod.Excluir;
                Prod.Free;
            end;
        Produtos.Free;
    end;
end;

```

```

Procedure TPedido.Incluir(CR: Real; QT: Real);
Var
    cont: Integer;
    Prod, ProdAux: TProduto;
begin
    Prod := TProduto.construtor(CR, QT, 'A');
    Prod.Incluir;
    Produtos.Add(Prod);
end;

```

```

Function TPedido.Gravar: Real;
var
    tempo, tempomax: Real;
    cont: integer;
    prod: Tproduto;
begin
    tempomax := 0;
    if Produtos.Count > 0 then
        begin
            for cont := 0 to Produtos.Count -1 do
                begin
                    prod := Produtos[cont];
                    tempo := Prod.Gravar(now, SerNum);
                    if Tempo > Tempomax then
                        begin
                            Tempomax := Tempo;
                        end;
                end;
            end;
        end;
end;

```

```

        end
    end;
end;
Gravar := TempoMax;
end;

Procedure TPedido.CapturaPedido;
var
    final: Real;
    ultimo: Real;
begin
    Ivgr.TVendas.SetKey;
    Ivgr.TVendas.FieldName('C_SERNUM').Value := SerNum;
    if Ivgr.TVendas.GotoKey then
    begin
        Cliente := Ivgr.TVendas.fieldbyname('C_CLIENT').Value;
    end;
    Ivgr.TProvenda.IndexName := 'NOTA';
    Ivgr.TProvenda.SetKey;
    Ivgr.TProvenda.FieldName('C_SERNUM').Value := SerNum;
    Ivgr.TProvenda.FieldName('C_ORDENA').Value := '01';
    if Ivgr.TProvenda.GotoKey then
    begin
        if Dados.Sequencia.IsEmpty then
        begin
            Ultimo := 1;
        end
        else
        begin
            Dados.Sequencia.Last;
            Ultimo := Dados.SequenciaORDEM.value + 1;
        end;
        While (Ivgr.TProvenda.FieldName('C_SERNUM').Value = SerNum) and
not (Ivgr.TProvenda.Eof) do
        begin
            Incluir(Ivgr.TProvenda.FieldName('C_CODRED').Value,
Ivgr.TProvenda.FieldName('C_QUANTI').Value);
            Ivgr.TProvenda.Next;
        end;
        Final := Gravar;
        Dados.Sequencia.append;
        Dados.SequenciaSERNUM.Value := SerNum;
        Dados.SequenciaCLIENTE.Value := Cliente;
        Dados.SequenciaORDEM.Value := Ultimo;
        Dados.SequenciaFINAL.Value := Datetimetostr(Final);
        Dados.Sequencia.Post;
    end;
end;

```

end;

Procedure TPedido.ExcluiPedido;

Var

Pedido: Real;

SN: ^String;

Pedidos : TList;

cont: integer;

begin

Dados.Sequencia.IndexName := 'SERNUM';

Dados.Sequencia.SetKey;

Dados.Sequencia.FieldName('SERNUM').Value := SerNum;

If Dados.Sequencia.GotoKey then

begin

Pedido := Dados.SequenciaORDEM.Value;

Dados.Sequencia.IndexName := 'ORDEM';

Dados.Sequencia.SetKey;

Dados.Sequencia.FieldName('ORDEM').Value := Pedido;

If Dados.Sequencia.GotoKey then

begin

if not Dados.Sequencia.Eof then

begin

Dados.EstoqueDeletaPedido(Dados.SequenciaSERNUM.Value);

Dados.ProgramacaoDeletaPedido(Dados.SequenciaSERNUM.Value);

Dados.Sequencia.Delete;

if not dados.Sequencia.Eof then

begin

Pedidos := TList.Create;

While Dados.SequenciaORDEM.Value > Pedido do

begin

New(SN);

Dados.EstoqueDeletaPedido(Dados.SequenciaSERNUM.Value);

ados.ProgramacaoDeletaPedido(Dados.SequenciaSERNUM.Value);

SN^ := Dados.SequenciaSERNUM.Value;

Pedidos.Add(SN);

Dados.Sequencia.Delete;

end;

for cont := 0 to Pedidos.Count -1 do

begin

SN := Pedidos[cont];

SerNum := SN^;

CapturaPedido;

Dispose(SN);

end;

Pedidos.Free;

end;

end;

```

        end;
    end;
end;

Procedure TPedido.AlterarPedido;
Var
    Pedido: Real;
    SN: ^String;
    Pedidos : TList;
    cont: integer;
begin
    Dados.Sequencia.IndexName := 'SERNUM';
    Dados.Sequencia.SetKey;
    Dados.Sequencia.FieldName('SERNUM').Value := SerNum;
    If Dados.Sequencia.GotoKey then
    begin
        Pedido := Dados.SequenciaORDEM.Value;
        Dados.Sequencia.IndexName := 'ORDEM';
        Dados.Sequencia.SetKey;
        Dados.Sequencia.FieldName('ORDEM').Value := Pedido;
        If Dados.Sequencia.GotoKey then
        begin
            if not dados.Sequencia.Eof then
            begin
                Pedidos := TList.Create;
                While Dados.SequenciaORDEM.Value >= Pedido do
                begin
                    New(SN);
                    Dados.EstoqueDeletaPedido(Dados.SequenciaSERNUM.Value);
                    Dados.ProgramacaoDeletaPedido(Dados.SequenciaSERNUM.Value);
                    SN^ := Dados.SequenciaSERNUM.Value;
                    Pedidos.Add(SN);
                    Dados.Sequencia.Delete;
                end;
                for cont := 0 to Pedidos.Count -1 do
                begin
                    SN := Pedidos[cont];
                    SerNum := SN^;
                    CapturaPedido;
                    Dispose(SN);
                end;
                Pedidos.Free;
            end;
        end;
    end;
end;
end;
end;

```

end.

## 6.2 APRESENTAÇÃO DAS TELAS E OPERACIONALIDADE

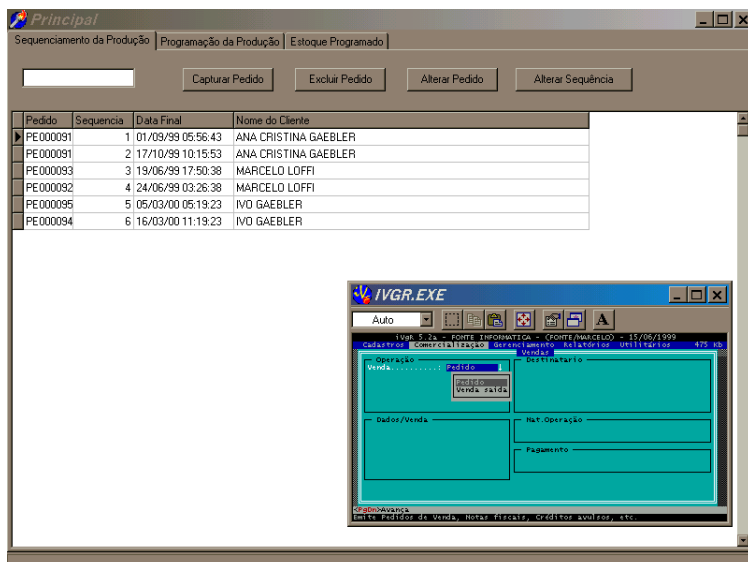
Tela de abertura, que é apresentada enquanto os subsistemas do protótipo são carregados (tela 1):



Tela 1: Tela de Apresentação

### SUBSISTEMA DE PROGRAMAÇÃO

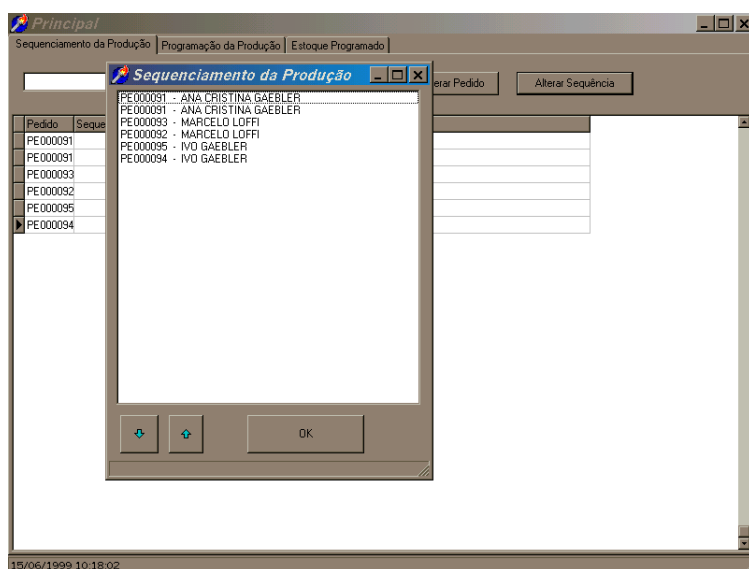
Página de Seqüenciamento da Produção (tela 2):



Tela 2: Página seqüenciamento do subsistema programação e iVgR

Digitado um pedido no aplicativo iVgR (janela menor da tela 2), o subsistema captura automaticamente o pedido e o inclui na programação, no último lugar da seqüência. Quando excluído um pedido no aplicativo o subsistema também detecta e exclui o pedido da programação. As rotinas de captura e exclusão podem ser feitas também manualmente, pressionando os botões capturar pedido e excluir pedido, tendo em vista que um pedido pode ser cancelado temporariamente sem precisar ser excluído do aplicativo. Há também as possibilidades de alterar pedido e alterar a seqüência. O primeiro campo da tabela tem os números dos pedidos, o segundo campo tem a seqüência em que os pedidos serão produzidos, o terceiro campo tem as datas em que os pedidos estarão prontos e o quarto tem o nome dos clientes referente aos pedidos.

Alteração de seqüência (tela3):



Tela 3: Alteração de seqüência de produção do subsistema programação

Selecionando o pedido, ao pressionar nas setas, ele é colocado para cima e para baixo, até que esteja na posição desejada. Pressionando o botão OK, o subsistema irá refazer a programação da produção, com a nova seqüência.

Página de Programação da Produção (tela 4):

Pedido	Descrição do Produto	Quantidade	S	Descrição da Máquina	Data Inicial	Data Final
PE000093	BASES PEQUENA	1	A	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 05
PE000093	BASES GRANDE	1	A	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 09
PE000092	MADEIRA 125*5*1 5 LIXADA	1500	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 22
PE000092	MADEIRA 195*5*2 LIXADA	200	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 00
PE000092	PE PEQUENO	200	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 05
PE000092	BASES GRANDE	100	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 04
PE000092	MADEIRA 200*3*3 LIXADA	200	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	11/06/1999 22
PE000092	MADEIRA 200*12*3 LIXADA	200	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 03
PE000092	BASES PEQUENA	100	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 01
PE000092	BASES GRANDE	100	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 04
PE000095	MADEIRA 125*5*1 5 LIXADA	75000	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	10/08/1999 17
PE000095	MADEIRA 195*5*2 LIXADA	10000	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	26/06/1999 17
PE000095	PE PEQUENO	10000	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	06/07/1999 17
PE000095	BASES GRANDE	5000	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	04/07/1999 05
PE000095	MADEIRA 200*3*3 LIXADA	10000	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	21/06/1999 17
PE000095	MADEIRA 200*12*3 LIXADA	10000	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	02/07/1999 17
PE000095	BASES PEQUENA	5000	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	29/06/1999 05
PE000095	BASES GRANDE	5000	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	04/07/1999 05
PE000094	MADEIRA 125*5*1 5 LIXADA	3750	A	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	14/06/1999 17
PE000094	MADEIRA 195*5*2 LIXADA	500	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 11
PE000094	PE PEQUENO	500	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 23
PE000094	BASES GRANDE	250	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 20
PE000094	MADEIRA 200*3*3 LIXADA	500	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 05
PE000094	MADEIRA 200*12*3 LIXADA	500	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 18
PE000094	BASES PEQUENA	250	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 14
PE000094	BASES GRANDE	250	P	LIXAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 20
PE000094	PIB GRANDES	4	D	TORNEAR MADEIRA PRE CORTADA	11/06/1999 17:	12/06/1999 18

Tela 4: Página de programação da produção do subsistema programação

Na tela 4 é apresentada a programação da produção, mostrando na seqüência especificada na tela 3, quais produtos devem ser produzidos e o pedido referente a eles, a quantidade, em que máquina / processo, a data e hora de início e a data e hora de término.

Página de Estoque Programado (tela 5):

Pedido	Código	Descrição do Produto	Quantidade	Saldo	Data
PE000094	220	PALITOS	1500	3000	12/06/1999 18:31:32
PE000093	220	PALITOS	-900	2100	15/06/1999 00:26:38
PE000092	220	PALITOS	-600	1500	20/06/1999 23:50:38
PE000095	220	PALITOS	30000	31500	02/07/1999 17:19:23
PE000095	220	PALITOS	-30000	1500	29/09/1999 17:19:23
PE000094	220	PALITOS	-1500	0	08/03/2000 14:19:23
PE000092	221	BASES PEQUENA	100	100	12/06/1999 01:38:47
PE000093	221	BASES PEQUENA	150	250	12/06/1999 05:50:38
PE000094	221	BASES PEQUENA	250	500	12/06/1999 14:19:32
PE000093	221	BASES PEQUENA	-150	350	15/06/1999 00:26:38
PE000092	221	BASES PEQUENA	-100	250	20/06/1999 23:50:38
PE000095	221	BASES PEQUENA	5000	5250	29/06/1999 05:19:23
PE000095	221	BASES PEQUENA	-5000	250	29/09/1999 17:19:23
PE000094	221	BASES PEQUENA	-250	0	08/03/2000 14:19:23
PE000092	222	BASES GRANDE	100	100	12/06/1999 04:02:47
PE000092	222	BASES GRANDE	100	200	12/06/1999 04:02:47
PE000093	222	BASES GRANDE	150	350	12/06/1999 09:26:38
PE000093	222	BASES GRANDE	150	500	12/06/1999 09:26:38
PE000093	222	BASES GRANDE	-150	350	12/06/1999 11:14:38
PE000094	222	BASES GRANDE	250	600	12/06/1999 20:19:32
PE000094	222	BASES GRANDE	250	850	12/06/1999 20:19:32
PE000093	222	BASES GRANDE	-150	700	15/06/1999 00:26:38
PE000092	222	BASES GRANDE	-100	600	20/06/1999 15:26:38
PE000092	222	BASES GRANDE	-100	500	20/06/1999 23:50:38
PE000095	222	BASES GRANDE	5000	5500	04/07/1999 05:19:23
PE000095	222	BASES GRANDE	5000	10500	04/07/1999 05:19:23
PE000095	222	BASES GRANDE	-5000	5500	06/07/1999 17:19:23
PE000095	222	BASES GRANDE	-5000	500	29/09/1999 17:19:23

Tela 5: Página de estoque programado do subsistema programação

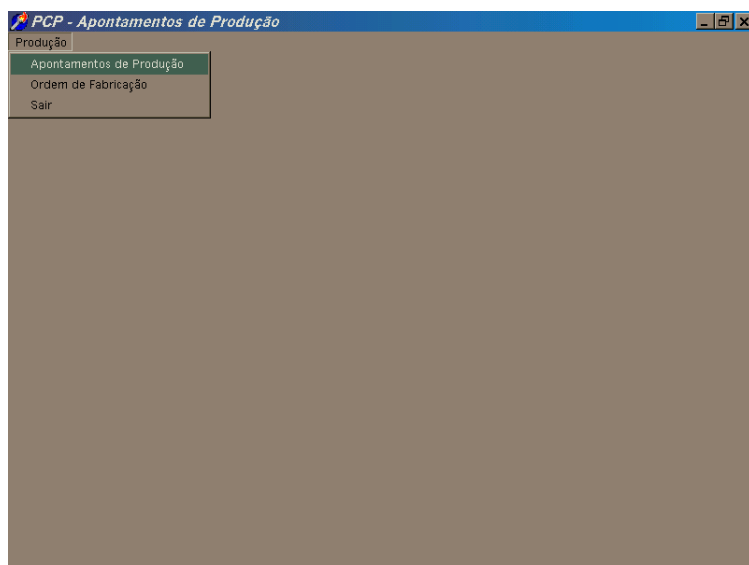
Nesta página podem ser visualizadas as baixas de matéria prima / produtos de composição e o acréscimo no produto produzido, sendo apresentados o número do pedido



referente àquele produto, o código e a descrição do produto, a quantidade do movimento, o saldo e a data referente à movimentação.

## SUBSISTEMA DE APONTAMENTOS

Tela de menu (tela 6):



Tela 6: Tela de menu do subsistema apontamentos

A tela 6 apresenta as opções Apontamentos de Produção, Ordens de Fabricação e Sair. Escolhida a opção Apontamentos de Produção o subsistema abrirá uma tela para apontamentos (tela 7), escolhida a opção Ordens de Fabricação o subsistema abrirá uma tela para emissão das ordens (tela 8) e escolhida a opção Sair, o subsistema finaliza.

Tela de Apontamentos da Produção (tela 7):

Máquina / Operação: TORNEAR MADEIRA PRE CORTADA

Produzido:

Código	Descrição	Quantidade
219	PIS GRANDES	10704
220	PALITOS	32104

Material consumido:

Código	Descrição	Quantidade
136	MAT. PRIMA 1	160
137	MAT. PRIMA 2	160
138	MAT. PRIMA 3	160
139	MAT. PRIMA 4	160
140	MAT. PRIMA 5	240
141	MAT. PRIMA 6	40
143	MAT. PRIMA 7	80
207	MAT. PRIMA 8	160
208	MAT. PRIMA 9	160

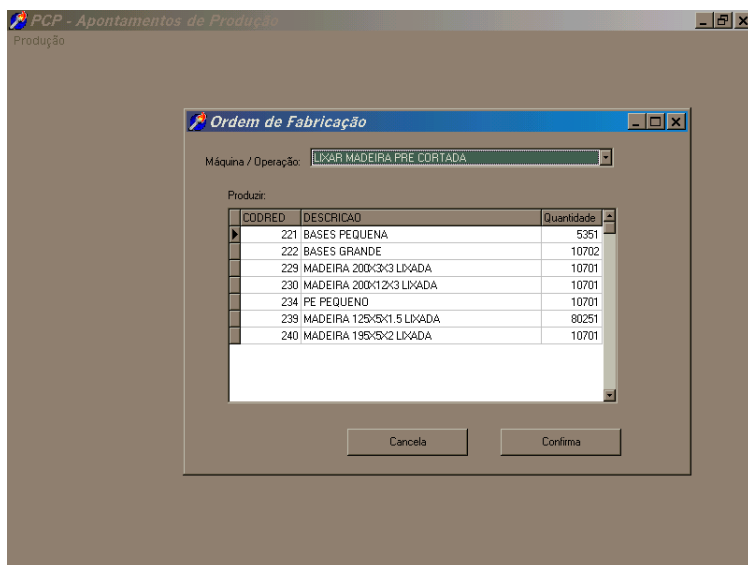
Recalcular

Cancela Confirma

Tela 7: Tela de apontamentos da produção do subsistema apontamentos

Na tela 7, selecionando o campo máquina / operação, serão apresentados os produtos que deveriam ter sido produzidos pela ordem de fabricação na primeira tabela e os produtos que deveriam ter sido utilizados para a produção dos mesmos na segunda tabela. As quantidades apresentadas podem ser alteradas manualmente, sendo que determinado produto pode ter ocupado mais matéria prima ou pode ter sido fabricado menos produtos do que estava previsto na ordem de fabricação por alguma razão externa. No segundo caso, quando a alteração for na quantidade de produtos produzidos, pressionando em Recalcular os valores dos materiais consumidos serão recalculados sem necessidade de alteração manual.

Tela de emissão de ordens de fabricação (tela 8):

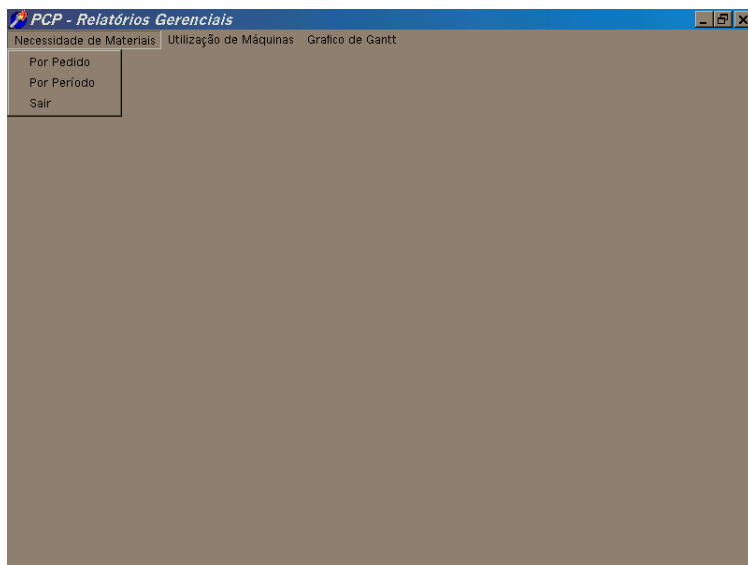


Tela 8: Tela de emissão de ordens de fabricação do subsistema apontamentos

Na tela 8, selecionando máquina / operação, serão apresentados na tabela os produtos que devem ser produzidos no dia. Pressionando o botão Confirma será impresso um relatório com a ordem de fabricação.

## SUBSISTEMA RELATÓRIOS

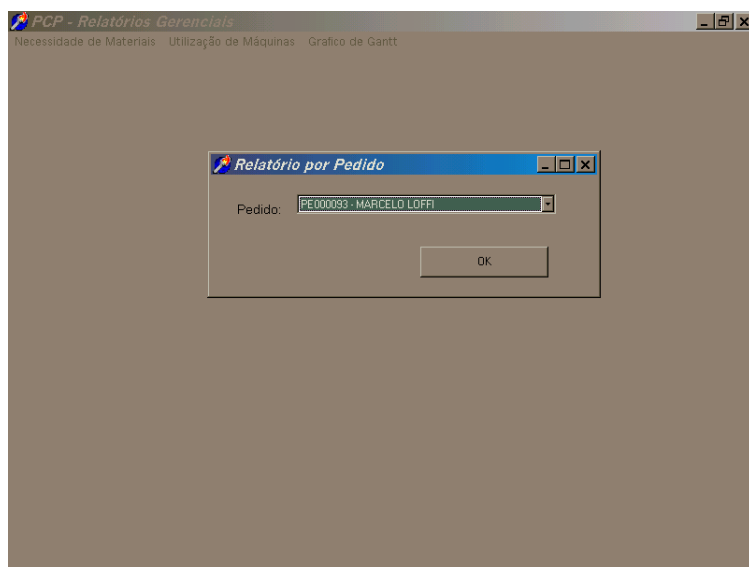
Tela de menu (tela 9):



Tela 9: Tela de menu do subsistema relatórios

A tela 9 apresenta como opções o relatório de Necessidade de Materiais (por pedido e por período), o relatório de Utilização de Máquinas (por máquina), o Gráfico de Gantt (por pedido e por período) e Sair que finaliza o subsistema.

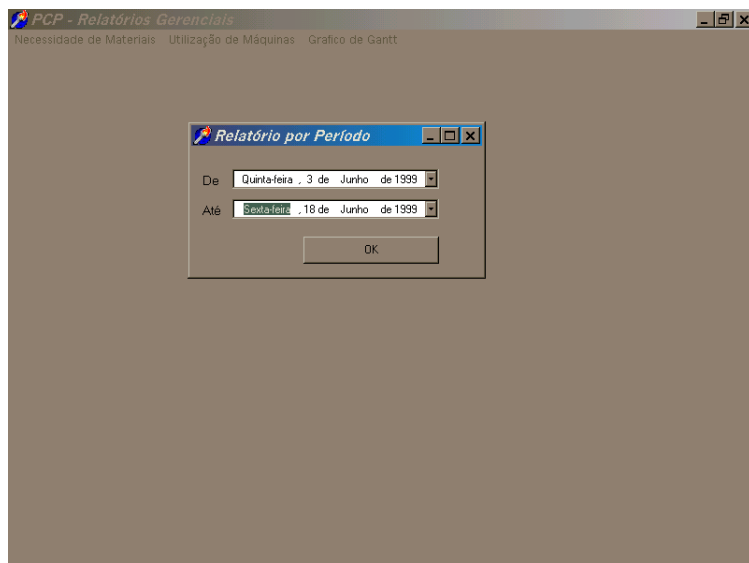
Tela de filtro para relatório de Necessidade de Materiais por pedido (tela 10):



Tela 10: Filtro para relatório de necessidade de materiais por pedido do subsistema relatórios

Na tela 10, selecionando o pedido, pressionando no botão OK, será impresso o relatório de necessidade de materiais do pedido selecionado.

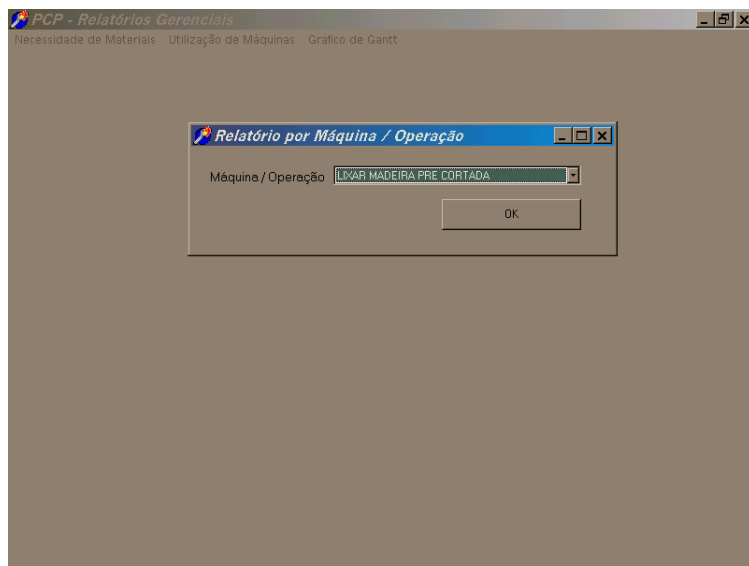
Tela de filtro para relatório de Necessidade de Materiais por período (tela 11):



Tela 11: Filtro para relatório de necessidade de materiais por período do subsistema relatórios

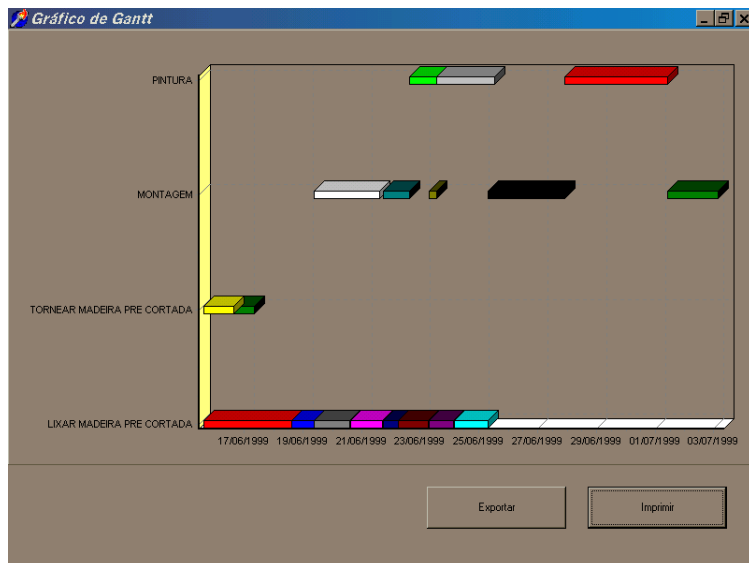
Na tela 11, selecionando a data inicial e a data final, pressionando o botão OK, será impresso o relatório de necessidade de materiais para o intervalo das datas selecionadas.

Tela de filtro para relatório de Utilização de Máquinas por máquina (tela 12):



Tela 12: Filtro para relatório de utilização de máquinas por máquina do subsistema relatórios.

As telas de filtro para o gráfico de Gantt são iguais às de necessidade de materiais. O Gráfico de Gantt é apresentado conforme a tela 13, com os dados referente ao filtro:



Tela 13: Gráfico de Gantt dos dados filtrados do subsistema relatórios

Conforme mostra a tela 13, o Gráfico de Gantt mostra a seqüência em que as atividades devem ser realizadas. O botão Exportar, exporta o gráfico para outros aplicativos, salvando no formato Bitmap do Windows (BMP) ou Windows Meta File (WMF), ou copiando para a área de transferência. O botão Imprimir imprime o gráfico.

## 7 CONCLUSÃO

Através do planejamento/programação da produção, o protótipo gera ordens de fabricação para a produção, evitando assim perda de tempo em cálculos manuais para o planejamento. Gera relatórios para os executivos de produção de prazos de entrega de pedidos e de necessidades de materiais por pedido, evitando a perda de clientes por promessas de prazos de entrega errados, resultantes de planejamentos errados ou falta de material por não ter sido previsto. Gera relatórios de utilização de máquinas, para que o executivo possa programar manutenções preventivas das máquinas periodicamente. Há assim economia de mão-de-obra, conseqüentemente abaixando os custos da produção além da empresa ganhar com a satisfação do cliente.

A maior dificuldade encontrada durante a confecção do trabalho foi a falta de material didático a respeito do ERP, mas como o módulo enfatizado foi o de planejamento/programação da produção, foi encontrado material que tratava do assunto isoladamente.

Como sugestão para futuros trabalhos, o desenvolvimento da parte de manutenção de máquinas, onde cada máquina poderá armazenar de quanto em quanto tempo é necessária a manutenção e quais os materiais que serão necessários para esta manutenção, e prevendo também a manutenção na hora da programação da produção. Na utilização da metodologia de orientação a objetos, a sugestão é utilizar outras técnicas de modelagem.

## REFERÊNCIAS BIBLIOGRÁFICAS

- [BIN94] BINDER, Fábio Vinícius. **Sistemas de apoio a decisão**. São Paulo : Érica, 1994.
- [BON71] BONINI, Edmundo Eboli. **CPM - PERT - e outros métodos - "Técnicas do Caminho Crítico"** . São Paulo : Andreotti, 1971.
- [CHI90] CHIAVENATO, Idalberto. **Iniciação ao planejamento e controle da produção**. São Paulo : McGraw-Hill, 1990.
- [CHO95] CHOMEN, Marcos. **FoxPro 2.6 for Windows: aprenda desenvolvendo**. São Paulo : Érica, 1995.
- [COR97] CORRÊA, Henrique L.; GIANESI, Irineu G.N.; CAON, Mauro. **Planejamento, programação e controle da produção - MRP II / ERP: conceitos, uso e implantação**. São Paulo : Atlas, 1997.
- [DAL98] DALFOVO, Oscar. **Desenho de um modelo de sistema de informação estratégico para a tomada de decisão nas pequenas e médias empresas do setor têxtil de Blumenau**. Dissertação de Mestrado - Centro de Ciências Sociais e Aplicadas. Blumenau : FURB, 1998.
- [FUR94] FURLAN, José Davi; MOTTA, Ivonido da; AMARAL, Francisco Piedade. **Sistemas de informação executiva**. São Paulo : Makron books do Brasil, 1994.
- [INS96] INSO CORPORATION. 1996. **Ajuda do Micrografx FlowCharter**.
- [LOE99] LOECSH, Cláudio; HEIN, Nelson. **Pesquisa operacional: fundamentos e modelos**. Blumenau : Ed. Da FURB, 1999.
- [MAR95] MARTIN, James; ODELL, James J. **Análise e projeto orientados a objeto**. Tradução de José Carlos Barbosa dos Santos. Revisão técnica Ronald Stevis Cassiolato. São Paulo : Makron Books, 1995.



- [NOG99] NOGUEIRA, Adail Roberto. **Metodologia OMT**. 1999. Endereço Eletrônico : <http://www.filadelfia.br.adail/omt/omt.html>.
- [RUM94] RUMBAUGH, James et all. **Modelagem e projetos baseados em objetos**. Rio de Janeiro : Campus, 1994.
- [STO91] STONER, James A. F.; FREEMAN, R. Edward. **Administração**. 5<sup>a</sup>. ed. Rio de Janeiro : Prentice-hall do Brasil, 1991.
- [SWA99] SWAN, Tom. **Delphi 4: bíblia do programador**. Tradução de Adriana Kramer. São Paulo : Berkeley Brasil, 1999.
- [TUB97] TUBINO, Dalvio Ferrari. **Manual de planejamento e controle da produção**. São Paulo : Atlas, 1997.
- [WIN93] WINBLAD, Ann L.; EDWARDS, Samuel D.; KING, David R. **Software orientado a objeto**. São Paulo : Makron Books, 1993.
- [ZAC76] ZACCARELLI, Sérgio Batista. **Programação e controle da produção**. 4<sup>a</sup> ed. São Paulo : Pioneira, 1976.