

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CAMPUS PATO BRANCO  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

**ADRIANO LINHAR  
ELQUIER SMANIOTTO LUZZATTO**

**UM PROTÓTIPO DE UM SISTEMA PARA DETERMINAÇÃO DA  
CAPACIDADE PRODUTIVA INSTALADA COM BASE EM ESTUDOS  
DE CRONOANÁLISE INDUSTRIAL PARA PEQUENAS EMPRESAS**

**TRABALHO DE CONCLUSÃO DE CURSO**

**PATO BRANCO  
2011**

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ  
CAMPUS PATO BRANCO  
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE  
SISTEMAS**

**ADRIANO LINHAR  
ELQUIER SMANIOTTO LUZZATTO**

**UM PROTÓTIPO DE UM SISTEMA PARA DETERMINAÇÃO DA  
CAPACIDADE PRODUTIVA INSTALADA COM BASE EM ESTUDOS  
DE CRONOANÁLISE INDUSTRIAL PARA PEQUENAS EMPRESAS**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, da Universidade Tecnológica Federal do Paraná, Campus Pato Branco, como requisito parcial para obtenção do título de Tecnólogo.

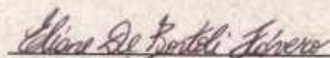
Orientador: Prof.<sup>a</sup> Msc. Eliane Maria De Bortoli Fávero.

**PATO BRANCO  
2011**

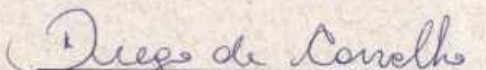
ATA Nº: 181

DEFESA PÚBLICA DO TRABALHO DE DIPLOMAÇÃO DOS ALUNOS ADRIANO LINHAR e ELQUIER SMANIOTTO LUZZATO.

Às 11:00 hrs do dia 5 de julho de 2011, Bloco S da UTFPR, Campus Pato Branco, reuniu-se a banca avaliadora composta pelos professores Eliane Maria de Bortoli Fávero (Orientadora), Diego de Carvalho (Convidado) e Cristiane T. Tartare de Carvalho (Convidada), para avaliar o Trabalho de Diplomação do aluno Adriano Linhar, matrícula 664936 e do aluno Elquier Smaniotto Luzzato, matrícula 749672, sob o título **Protótipo de um Sistema para Determinação da Capacidade Produtiva Instalada com Base em Estudos de Cronoanálise Industrial para Pequenas Empresas**; como requisito final para a conclusão da disciplina Trabalho de Diplomação do Curso Superior de Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas, Coordenação de Informática. Após a apresentação os candidatos foram entrevistados pela banca examinadora, e a palavra foi aberta ao público. Em seguida, a banca reuniu-se para deliberar considerando o trabalho **APROVADO**. Às 11:55 hrs foi encerrada a sessão.



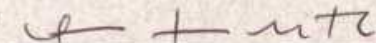
Prof. Eliane Maria de Bortoli Fávero, M.Sc.  
Orientadora



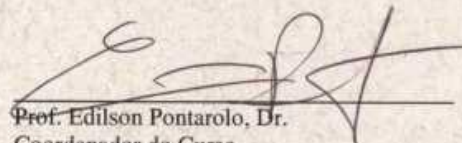
Prof. Diego de Carvalho, Esp.  
Convidado



Prof. Cristiane T. Tartare de Carvalho, Esp.  
Convidada



Prof. Omero Francisco Bertol, M.Sc.  
Coordenador do Trabalho de Diplomação



Prof. Edilson Pontarolo, Dr.  
Coordenador do Curso

## **AGRADECIMENTOS**

Agradecemos primeiramente a Deus, criador de tudo o que há, pela oportunidade de estar vivenciando e concluindo mais esta experiência, e que continua nos dando o dom da vida e iluminando nossos caminhos.

Aos amigos, pelo simples fato de existirem e não nos deixarem esmorecer pelo caminho, nos apoiando e alegrando com sua companhia quando necessário, e principalmente entendendo nossas eventuais ausências.

Às nossas famílias, que nos deram todo o apoio necessário, mostraram que é necessário sempre seguir em frente no nosso caminho, e ainda ajudaram a ter descontração e tranquilidade nos momentos mais críticos.

À nossa orientadora, professora Eliane De Bortoli Fávero, pela disposição, enorme paciência e compreensão, que foi fundamental para que pudéssemos concluir este trabalho.

## RESUMO

LINHAR, Adriano; LUZZATTO, Elquier Smaniotto. **Um Protótipo de um Sistema para Determinação da Capacidade Produtiva Instalada com Base em Estudos de Cronoanálise Industrial para Pequenas Empresas**. 2011. Monografia de Trabalho de Conclusão de Curso. Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas. Universidade Tecnológica Federal do Paraná, Campus Pato Branco. Pato Branco, 2011.

O presente trabalho apresenta o levantamento de requisitos, a análise e a implementação de um sistema de controle da capacidade produtiva, para pequenas empresas, com foco na produção em série, utilizando a cronoanálise industrial como ferramenta de apoio para a determinação da capacidade instalada. Além disso, o sistema proposto se caracteriza como um Sistema de Apoio à Decisão (SAD), a partir do qual é possível informar alguns parâmetros (ex. número de máquinas, número de homens) e simular a capacidade produtiva da empresa, possibilitando maior embasamento e facilidade na tomada de decisão por parte dos gestores da empresa. Para isso, foram estudadas as teorias que envolvem a análise de sistemas e a determinação da capacidade produtiva de uma empresa, bem como o levantamento de ferramentas necessárias ao desenvolvimento do sistema. Em seguida, foram levantados os requisitos para o sistema proposto; realizada a análise e modelagem orientada a objetos dos requisitos obtidos e implementado o sistema (plataforma Windows, banco de dados MySQL, linguagem JAVA). Afirma-se que os objetivos propostos inicialmente foram atendidos, com destaque para o cálculo do tempo padrão por operação, à determinação da capacidade produtiva instalada, geração de relatórios gerenciais no formato textual e gráfico e simulação de produção com base em parâmetros quantitativos de produção.

**Palavras-chave:** Capacidade Produtiva Instalada. Cronoanálise Industrial. Sistema de Apoio à Decisão. Desenvolvimento Java.

## LISTA DE TABELAS

Tabela 01 - Níveis de confiança utilizados na análise dos tempos de ciclo .....	34
Tabela 02 - Tolerância para alívio de fadiga.....	35
Tabela 03 - Cálculo da fadiga .....	35
Tabela 04 - Requisitos Funcionais.....	53
Tabela 05 - Requisitos Não-Funcionais.....	54
Tabela 06 - Descrição de Caso de Uso – Cadastrar produtos.....	57
Tabela 07 - Descrição de Caso de Uso – Calcular o tempo padrão .....	57
Tabela 08 - Descrição de Caso de Uso – Gerar Relatório de Operações por Período.....	58
Tabela 09 - Descrição de Caso de Uso – Consultar capacidade de produção dos produtos .....	58
Tabela 10 - Termos técnicos do diagrama de classes .....	61

## LISTA DE FIGURAS

Figura 01 - Exemplo de atributos .....	15
Figura 02 - Exemplo de métodos .....	16
Figura 03 - Exemplo de classe .....	16
Figura 04 - Um diagrama de caso de uso .....	21
Figura 05 - Exemplo de diagrama de classes .....	22
Figura 06 - Exemplo de Diagrama de Seqüência .....	23
Figura 07 - Diagrama de Classes no <i>JUDE Community</i> .....	44
Figura 08 - IDE do <i>NetBeans</i> 6.7.1 .....	45
Figura 09 - Gerenciamento do Banco de Dados MySQL através do <i>NetBeans</i> 6.7.1 .....	46
Figura 10 - Utilização do plug-in <i>iReport</i> através do <i>NetBeans</i> .....	49
Figura 11 - Diagrama de Casos de Uso do Sistema de determinação da capacidade produtiva.....	56
Figura 12 - Diagrama de Classes .....	60
Figura 13 - Diagrama de Entidade-Relacionamento .....	61
Figura 14 - Diagrama de Seqüência – Cadastrar Produtos.....	63
Figura 15 - Diagrama de Seqüência – Cadastrar Processos .....	63
Figura 16 - Diagrama de Seqüência – Calcular Tempo Padrão .....	64
Figura 17 - Diagrama de Seqüência – Gerar Relatório de Capacidade Produtiva .....	64
Figura 18 - Diagrama de Seqüência – Consultar Capacidade Produtiva.....	65
Figura 19 - Formulário de Login do sistema .....	66
Figura 20 - Tela Principal do Sistema .....	67
Figura 21 - Tela de Cadastro de Máquinas.....	68
Figura 22 - Formulário de Cadastro de Produtos .....	71
Figura 23 - Formulário de Cadastro de Operações .....	73
Figura 24 - Formulário de Cálculo de Fadiga .....	76
Figura 25 - Formulário de Cálculo de Necessidades Pessoais .....	79
Figura 26 - Formulário de Simulação de Produção .....	80
Figura 27 - Formulário para gerar gráfico de Operações dos Produtos.....	85
Figura 28 - Gráfico de Operações do Produto .....	87
Figura 29 - Formulário para filtrar o intervalo de datas para relatório.....	87
Figura 30 - Relatório de Operações por Período .....	88

## LISTA DE QUADROS

Quadro 1 - Método Listagem de Máquinas .....	69
Quadro 2 - Código de inserção de dados no Banco de dados.....	70
Quadro 3 - Código para buscar as informações dos produtos no Banco de dados. .	73
Quadro 4 - Código para calcular o tempo padrão da operação.....	76
Quadro 5 - Código para calcular o percentual de fadiga da operação. ....	78
Quadro 6 - Código para calcular o percentual de Necessidades Pessoais da operação. ....	80
Quadro 7 - Código para realizar a simulação da produção. ....	85
Quadro 8 - Código para Geração do Gráfico de Operações dos Produtos .....	86



## SUMÁRIO

1 INTRODUÇÃO .....	9
1.1 Objetivos .....	10
1.1.1 Objetivo Geral .....	10
1.1.2 Objetivos Específicos .....	10
1.2 Justificativa .....	11
2 REFERENCIAL TEÓRICO .....	13
2.1 Análise Orientada a Objetos .....	13
2.1.1 Objeto .....	14
2.1.1.1 Atributos .....	14
2.1.1.2 Métodos .....	15
2.1.2 Classes .....	16
2.1.3 Herança .....	17
2.1.4 Polimorfismo .....	17
2.1.5 Encapsulamento .....	18
2.2 <i>Unified Modeling Language</i> (UML) .....	18
2.2.1 Diagramas .....	19
2.2.1.1 Diagrama de Caso de Uso .....	20
2.2.1.2 Diagrama de Classes .....	21
2.2.1.3 Diagrama de Seqüência .....	22
2.3 Requisitos .....	24
2.3.1 Requisitos de Domínio .....	25
2.3.2 Requisitos de Usuário .....	25
2.4 <i>Rational Unified Process - RUP</i> .....	26
2.5 Capacidade Produtiva e Cronoanálise .....	27
2.5.1 Planejamento da capacidade .....	27
2.5.2 Capacidade e turnos de trabalho .....	30
2.5.3 Tempos cronometrados .....	31
2.5.4 Finalidade dos estudos de tempos .....	31
2.5.5 Cronoanálise industrial .....	32
2.5.6 Metodologia de Determinação do Tempo Padrão .....	33
2.5.7 Determinação do número de ciclos a serem cronometrados .....	34
2.5.8 Determinação das tolerâncias .....	34
2.5.9 Determinação do tempo padrão .....	36
3 PROCEDIMENTOS METODOLÓGICOS .....	40
3.1 Tipo de Pesquisa .....	40
3.2 População Amostra .....	41
3.3 Coleta dos Dados .....	42
3.3.1 Metodologia de Determinação do Tempo Padrão .....	42
3.4 Ferramentas Utilizadas .....	43
3.4.1 JUDE Community .....	43
3.4.2 <i>NetBeans</i> .....	44
3.4.3 Banco de Dados .....	45

3.4.3.1 MySQL .....	45
3.4.3.2 JPA - Java Persistence API .....	47
3.4.3.3 Provedor de Persistência - TopLink .....	48
3.4.4 Relatórios <i>JasperReports</i> com <i>iReport</i> .....	48
4 RESULTADOS E DISCUSSÕES .....	50
4.1 Descrição do Sistema informatizado para Determinação da Capacidade Produtiva .....	50
4.1.1 Descrição do Sistema Atual .....	51
4.2 Levantamento de Requisitos e Análise dos Dados .....	53
5 IMPLEMENTAÇÃO DO SISTEMA CAPACIDADE PRODUTIVA .....	66
5.1 Utilização do Protótipo.....	88
6 CONSIDERAÇÕES FINAIS .....	90
REFERÊNCIAS .....	92
ANEXOS .....	94
ANEXO A - planilha (EXCEL) utilizada pela empresa em estudo para a determinação da capacidade produtiva .....	95
ANEXO B - planilha (EXCEL) utilizada pela empresa em estudo para simular a produção .....	96

## 1 INTRODUÇÃO

Em muitas empresas, cujos processos eram executados manualmente, estes já estão sendo automatizados por meio de sistemas computacionais de menor custo e maior desempenho. Tendo em vista o crescimento da utilização de sistemas nas mais variadas áreas da economia, e a possibilidade de automatizar os processos nos mais variados setores, levando em consideração fatores como produtividade, ganho em escala e redução de custos, por exemplo, a automação vem sendo apontada como o caminho mais promissor a ser seguido.

Analisar o processo e capacidade efetiva de uma linha de produção, na maioria das vezes é uma tarefa complexa, especialmente quando o processo trabalha para atender diferentes produtos. Os administradores de produção freqüentemente visam encontrar a quantidade de produtos que se consegue produzir em determinado espaço de tempo.

Nas organizações, de um modo geral, quando as ações voltam-se para a máxima produção com o menor custo e qualidade desejada, elas ganham poder competitivo. Só assim podem adquirir maior participação no mercado e garantir os ganhos que possibilitarão novos investimentos.

É com o propósito de auxiliar a tomada de decisões a partir da determinação da capacidade produtiva que a análise para o desenvolvimento de um software que contemple tais necessidades se apóia e com tal foco será aqui estudada.

O presente trabalho apresentará o levantamento de requisitos, a análise e a implementação de um sistema de controle da capacidade produtiva, para pequenas empresas, com foco na produção em série, utilizando a cronoanálise industrial como ferramenta de apoio para a determinação da capacidade instalada. Além disso, o sistema proposto se caracteriza como um Sistema de Apoio à Decisão (SAD), a partir do qual é possível informar alguns parâmetros (ex. número de máquinas, número de homens) e simular a capacidade produtiva, possibilitando maior embasamento e facilidade na tomada de decisão por parte dos gestores da empresa

Assim, esse trabalho encontra-se estruturado da seguinte forma: o item 2 contém a justificativa seguida dos objetivos (item 3). O item 4 apresenta toda a

fundamentação teórica necessária para a elaboração do trabalho. No item 5 estão os procedimentos metodológicos adotados no trabalho, bem como a análise e modelagem do sistema em questão.

## **1.1 Objetivos**

### **1.1.1 Objetivo Geral**

Desenvolver um protótipo de um software para determinar a capacidade produtiva de pequenas empresas com base em estudos de cronoanálise industrial para pequenas empresas com produção em série.

### **1.1.2 Objetivos Específicos**

Para que seja possível alcançar o objetivo geral, o mesmo será dividido em etapas. As quais serão visualizadas como o resultado obtido ao final do trabalho. São elas:

- Fazer levantamento e adequação da bibliografia sobre a teoria que envolve a determinação da capacidade produtiva de uma empresa, análise de sistemas e ferramentas necessárias ao desenvolvimento do sistema;
- Levantar os requisitos funcionais e não-funcionais para o software proposto;
- Realizar a análise e modelagem orientada a objetos dos requisitos obtidos;
- Integrar as tecnologias que serão utilizadas (plataforma Windows, banco de dados MySQL, linguagem JAVA);
- Implementar o sistema tendo em vista as seguintes funcionalidades:

- Cadastrar máquinas;
  - Cadastrar operações;
  - Cadastrar ferramentas;
  - Cadastrar amostras de tempo;
  - Calcular tempo padrão por operação;
  - Determinar a capacidade produtiva.
  - Gerar relatório textual e gráfico de gargalos de operações dos produtos;
  - Possibilitar a simulação de produção com base em parâmetros quantitativos de produção, caracterizando assim um Sistema de Apoio à Decisão (SAD).
- Realizar testes do sistema em ambiente real, visando atender aos requisitos previamente identificados.

## **1.2 Justificativa**

O desenvolvimento do presente trabalho se justifica devido à necessidade de se conhecer a capacidade produtiva de uma empresa, visando aumentar a produtividade a um custo reduzido, aumentando o índice de certeza quanto aos prazos de produção dos produtos, permitindo identificar os gargalos de produção, para que seja possível eliminá-los, não esquecendo da qualidade.

Além disso, o sistema proposto se caracteriza como um Sistema de Apoio à Decisão (SAD), a partir do qual é possível informar alguns parâmetros (ex. número de máquinas, número de homens) e projetar a capacidade produtiva. Um sistema como esse permite à gerência melhor planejar as tomadas de decisões que envolvem o sistema produtivo de uma empresa.

Atualmente, os softwares existentes com a função de auxiliar na determinação da capacidade produtiva, em sua maioria são muito complexos, difíceis de serem utilizados, além de apresentarem um custo muito elevado e serem direcionados para empresas de médio e grande porte, limitando o acesso à pequena empresa. Sendo assim, se faz necessário o desenvolvimento de um sistema para essa finalidade, o qual seja acessível e de fácil entendimento e utilização, permitindo

às empresas de pequeno porte obter acesso a esse tipo de sistema, permitindo um melhor planejamento da produção, por meio de processos produtivos de qualidade e menor custo.

Sendo a cronoanálise uma ferramenta que auxilia na determinação da capacidade produtiva instalada em empresas com linhas de produção em série, as empresas que se beneficiariam com este tipo de software seriam justamente as empresas que possuem uma ou mais linhas de produção em série, pois são as empresas onde o modelo de levantamento da capacidade através da cronoanálise melhor se aplica.

A determinação da capacidade produtiva, assim como a capacidade de planejar a produção é um problema muito comum em organizações. Outro fator que está diretamente ligado à determinação da capacidade produtiva é o levantamento de custos de produção. Assim, o sistema proposto poderá auxiliar no fornecimento desses dados, e conseqüentemente, na geração do custo de mão-de-obra.

Outra grande vantagem é ter conhecimento do total de equipamentos utilizados e seu índice de utilização, além de permitir também que previsões sobre os recursos de capacidade instalada sejam realizadas, conforme mencionado anteriormente.

Com um sistema informatizado haverá um ganho de tempo no processamento dessas informações e será possível fornecer os indicadores e as informações precisas, oportunas e acessíveis, facilitando e proporcionando a realização de uma administração mais eficaz.

Este trabalho tem como objetivo principal desenvolver um protótipo de software protótipo de um software para determinar a capacidade produtiva de pequenas empresas com base em estudos de cronoanálise industrial para pequenas empresas com produção em série., fundamentado nos resultados obtidos com o estudo dirigido realizado durante o trabalho de estágio supervisionado. O desenvolvimento deste protótipo de sistema permitirá colocar em prática os conhecimentos adquiridos na graduação, com relação a Análise de Dados, Documentação de Sistemas, Desenvolvimento de Software e manipulação de Banco de Dados.

## 2 REFERENCIAL TEÓRICO

O presente trabalho foi desenvolvido com base na Análise Orientada a Objetos, utilizando, portanto a UML - *Unified Modeling Language* (Linguagem de Modelagem Unificada), que segundo Booch (2000), é uma linguagem padrão para elaboração da estrutura de projetos de software, podendo ser empregada para visualização, especificação, construção e documentação dos artefatos de sistemas de software.

Como a Programação Orientada a Objetos consiste em utilizar estruturas de dados que simulem o comportamento dos objetos, torna-se mais fácil a sua implementação, por ser uma metodologia unificada para análise e programação (CORREA, 2006). Sendo assim, pode-se utilizar a mesma estrutura definida para os objetos durante a modelagem na implementação do sistema, pois os conceitos utilizados são os mesmos.

Foi realizado um estudo específico sobre a área de domínio do sistema a ser trabalhado, tratando, portanto da Capacidade Produtiva Instalada e da Cronoanálise Industrial, além de um estudo do sistema atual para essa finalidade. Foi realizado também um levantamento bibliográfico sobre requisitos e a metodologia de análise de sistemas orientada a objetos, o que proporcionou embasamento à posterior coleta dos requisitos e análise do sistema, além de discorrer sobre as ferramentas utilizadas.

### 2.1 Análise Orientada a Objetos

Segundo Correa (2006), pode-se dizer que a Análise Orientada a Objetos consiste basicamente na abstração de objetos do mundo real simplificando-os em Objetos Computacionais, representando o mais fielmente possível as situações do mundo real nos Sistemas Computacionais.

Objetos Computacionais, assim como os objetos no mundo real, possuem comportamentos próprios, ou seja, ele próprio deve conter as informações necessárias sobre como reagir diante de determinada situação. Por exemplo, se for derrubada uma moeda no chão, ela irá picar, além de produzir um som peculiar, não porque alguém quis, mas porque as características da moeda a faz ter esse comportamento diante desse contexto.

### **2.1.1 Objeto**

Um objeto pode ser entendido como uma “coisa” física, como uma porta, uma maçã, um sapato, por exemplo, ou uma coisa mental, como uma conta bancária, que não existe fisicamente a ponto de podermos sentir, mas mesmo assim temos “noção” de como ela é e como se comporta (CORREA, 2006).

Já no ponto de vista computacional, segundo Martin (1995, p. 18) “um objeto é qualquer coisa, real ou abstrata, a respeito da qual armazenamos dados e os métodos que os manipulam”.

Ao fazer a abstração de um objeto real para um objeto computacional, são definidos seus atributos e métodos, que serão melhor explanados em seguida.

#### *2.1.1.1 Atributos*

Os atributos correspondem às características que descrevem um objeto real. Utilizando como, por exemplo, uma porta, suas características principais podem ser descritas da seguinte forma:

- Cor;
- Altura;
- Base;
- Espessura;
- Material (madeira, ferro);

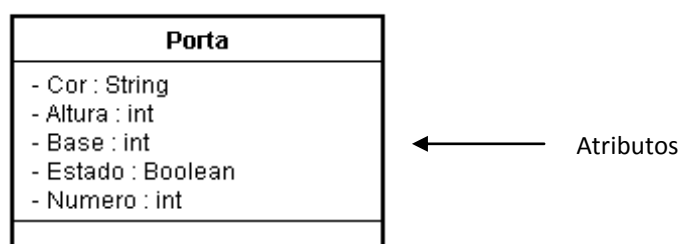


- Estado (aberta, fechada).

Como cada porta tem essas características alteradas de acordo com o local onde será utilizada, pode-se dizer que essas características variam de uma porta para outra.

Da mesma forma no campo computacional, segundo Correa (2006, p.15) “os atributos dos objetos são ‘variáveis’ ou ‘campos’ que armazenam os diferentes valores que as características dos objetos podem conter”.

Porém, esses atributos só podem ser modificados através de estímulos que provoquem a transição desses estados no objeto, podendo ser internos ou externos. Por exemplo, uma porta que esteja fechada só irá abrir se alguma coisa ou alguém provocar essa mudança.



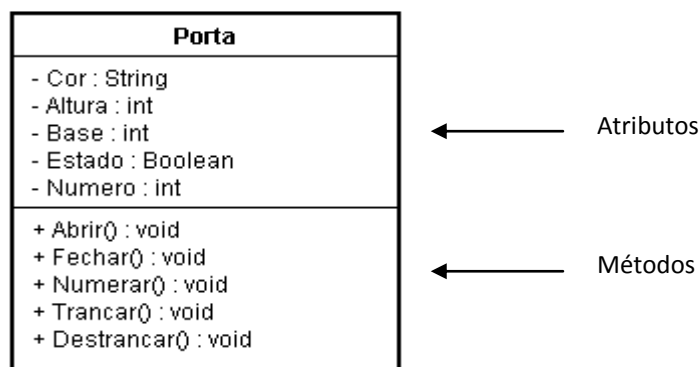
**Figura 01 - Exemplo de atributos**

Fonte: Elaborado pelo autor

### 2.1.1.2 Métodos

Segundo Correa (2006, p.16) “os métodos são procedimentos ou funções que realizam as ações próprias do objeto”. Ou seja, toda e qualquer ação que um objeto possa realizar, inclusive interagir com outros objetos, será realizada através de seus métodos.

Seguindo com o exemplo da porta, pode-se definir como seus métodos abrir, fechar, trancar, destrancar e enumerar, por exemplo.



**Figura 02 - Exemplo de métodos**

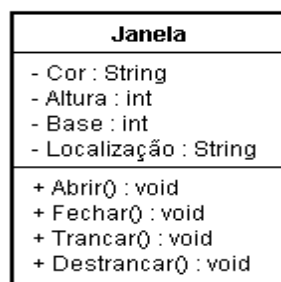
Fonte: Elaborado pelo autor

### 2.1.2 Classes

Segundo Casimiro (2010), uma classe pode ser definida como todos os atributos e métodos que servem para um conjunto de objetos. Dessa forma, a classe é como se fosse uma matriz que especifica objetos e seus atributos, mas não pode ser acessada diretamente.

Graficamente, é representada como um retângulo, e seu nome deve ser diferente das demais classes, podendo ser composto por texto, números e alguns sinais de pontuação (exceto dois pontos).

A Figura 03 abaixo mostra um exemplo da classe Janela, com atributos e métodos comuns que permitem a criação de vários objetos do tipo Janela.



**Figura 03 - Exemplo de classe**

Fonte: Elaborado pelo autor

### 2.1.3 Herança

Segundo Correa (2006, p.35) “Herança” significa que todos os atributos e métodos programados na classe ancestral já estarão automaticamente presentes em suas classes descendentes sem necessidade de reescrevê-los.

Observando-se, por exemplo, cachorros e gatos, pode-se perceber que são animais distintos, com características semelhantes (quatro patas, duas orelhas), mas que pertencem a um grupo maior, o dos mamíferos. É possível então atribuir as características comuns ao cachorro e ao gato à classe mamíferos, de modo que eles permaneçam somente com seus atributos e métodos próprios, como por exemplo, miar (gato) e latir (cachorro).

De acordo com o exemplo, na hierarquia das classes a classe Mamíferos seria a *superclasse*, por generalizar as classes Cachorro e Gato, que, portanto passariam a ser as *subclasses*.

Como uma subclasse “herda” as características e métodos programados na superclasse, se uma alteração for realizada em uma superclasse, todas as subclasses terão essa alteração disponível, evitando-se assim a necessidade de reprogramação, sendo esta a uma das principais vantagens da programação orientada a objetos (CORREA, 2006).

### 2.1.4 Polimorfismo

Segundo Medina (2010), polimorfismo significa ter ‘muitas formas’, ou seja, um único nome representando códigos diferentes, selecionados por algum mecanismo automático. “Um nome, vários comportamentos”. Em outras palavras, é um código que pode ser aplicado a várias classes de objetos.

Em analogia ao mundo real, pode-se dizer que o método *Abrir* é polimórfico, uma vez que pode ser aplicados a portas, janelas, latas, etc. sendo que cada um desses objetos irá reagir à sua maneira a esse método, mas ambos irão executá-lo.

### 2.1.5 Encapsulamento

Segundo Macoratti (2010), encapsulamento pode ser definido como o ato de empacotar ao mesmo tempo dados e objetos. O objeto esconde seus dados de outros objetos e permite que os dados sejam acessados somente por intermédio de seus próprios métodos. Isso é chamado de ocultação de informações (*information hiding*).

Dessa forma, os dados de um objeto ficam protegidos contra acessos indevidos ou alterações indevidas ou mesmo não-intencionais, além de ocultar do usuário os detalhes de sua implementação.

O objeto fica à disposição com toda a sua funcionalidade, sem ser necessário que se saiba como ele funciona internamente. Além disso, permite que sejam feitas modificações internas em um objeto, acrescentando métodos, sem que os outros componentes do sistema que utilizam esse objeto sejam afetados (CORREIA, 2006).

## 2.2 Unified Modeling Language (UML)

A *Unified Modeling Language* (UML) ou Linguagem Unificada de Modelagem é uma linguagem gráfica utilizada para visualização, elaboração, especificação e documentação de projetos de software, permitindo que os desenvolvedores do projeto visualizem o seu trabalho em diagramas padronizados.

Vários tipos de sistemas podem ser beneficiados com o uso da UML, como por exemplo, softwares corporativos, aplicações baseadas na WEB, aplicações *desktop* e até sistemas de tempo real, devido à sua expressividade, que abrange todas as visões necessárias ao desenvolvimento e implantação desses sistemas.

A UML pode representar as diferentes partes de um sistema de software através de seus muitos componentes, que são utilizados na criação de diagramas que podem representar um modelo conceitual ou físico do sistema. No entanto, ela é

apenas uma parte de um método para o desenvolvimento de um software. Ainda assim, pode ser utilizada independentemente do processo de produção a ser utilizado.

### 2.2.1 Diagramas

Um diagrama, segundo Booch (2000, p. 92), “é a apresentação gráfica de um conjunto de elementos, geralmente representada como um gráfico conectado de vértices (itens) e arcos (relacionamentos)”.

Os diagramas são a parte fundamental da UML. Por ser uma linguagem visual, permite aos desenvolvedores visualizarem o projeto de produção de software de vários ângulos diferentes que se interligam através de seus diagramas. Estes ainda se dividem em estruturais e comportamentais. Os diagramas estruturais são representados pelos seguintes diagramas (BOOCH, 2000):

- Diagrama de classes
- Diagrama de objetos
- Diagrama de componentes
- Diagrama de implantação

Já os diagramas comportamentais são representados pelos seguintes diagramas (BOOCH, 2000):

- Diagrama de caso de uso
- Diagrama de seqüência
- Diagrama de colaboração
- Diagrama de gráfico de estados
- Diagrama de atividades

O presente projeto dará ênfase aos diagramas de casos de uso, de classes, e de seqüência (interação).

### 2.2.1.1 Diagrama de Caso de Uso

É um diagrama utilizado para visualizar, especificar e documentar o comportamento de um sistema, subsistema ou de uma classe, para que usuários possam entender como utilizar esse elemento e os desenvolvedores possam implementá-lo (BOOCH, 2000).

Um diagrama de caso de uso é composto por atores, casos de uso e seus relacionamentos. Tais componentes serão apresentados nos itens a seguir:

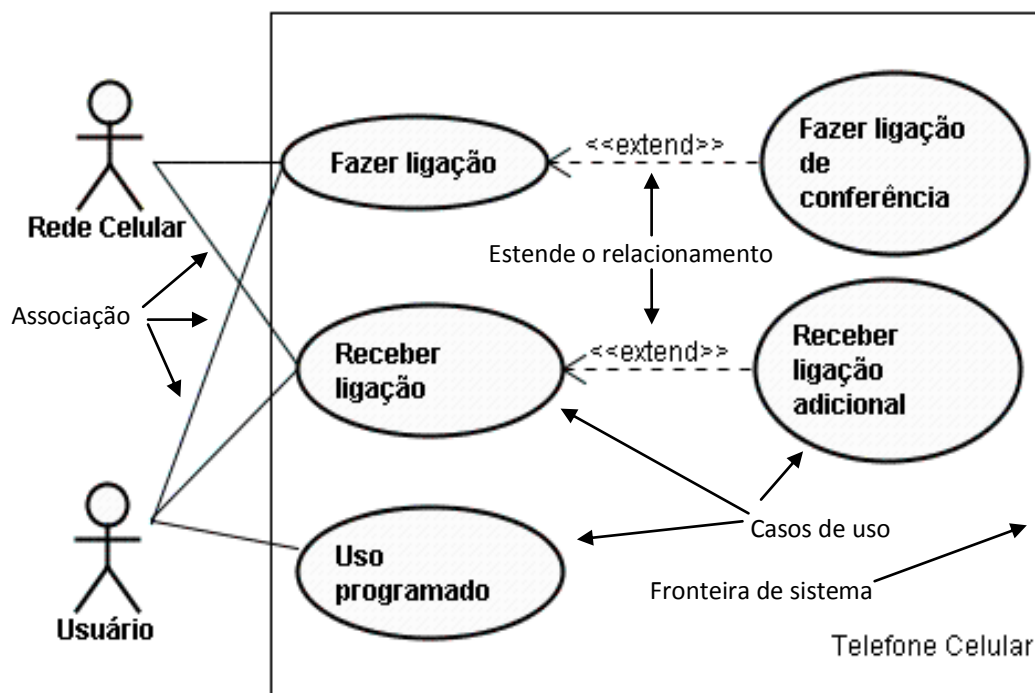
#### **a) Caso de Uso**

Um caso de uso, segundo Booch (2000), “é uma descrição de um conjunto de seqüência de ações, inclusive variantes, que um sistema executa para produzir um resultado de valor observável por um ator”, sendo esse conjunto representado graficamente como uma elipse, contendo o nome do caso de uso dentro dela conforme a Figura 05. Um nome de caso de uso pode ser qualquer texto, números e a maioria dos sinais de pontuação, sendo que na prática são utilizadas expressões verbais ativas.

#### **b) Atores**

Um ator representa uma entidade externa que utiliza ou interage com o sistema podendo, dessa forma, ser um usuário, um dispositivo de hardware ou até mesmo outro sistema que venha a interagir com o sistema em desenvolvimento.

Graficamente são representados por figuras que remetem à imaginação de pessoas, e são ligados aos casos de uso somente por meio de relacionamentos, que permitem a sua comunicação, com a possibilidade de receber e enviar mensagens (BOOCH, 2000). Os relacionamentos mais importantes na UML são três, que podem representar dependência, generalização e associação. A Figura 04 abaixo mostra graficamente um exemplo de caso de uso, com atores e relacionamentos.



**Figura 04 - Um diagrama de caso de uso**

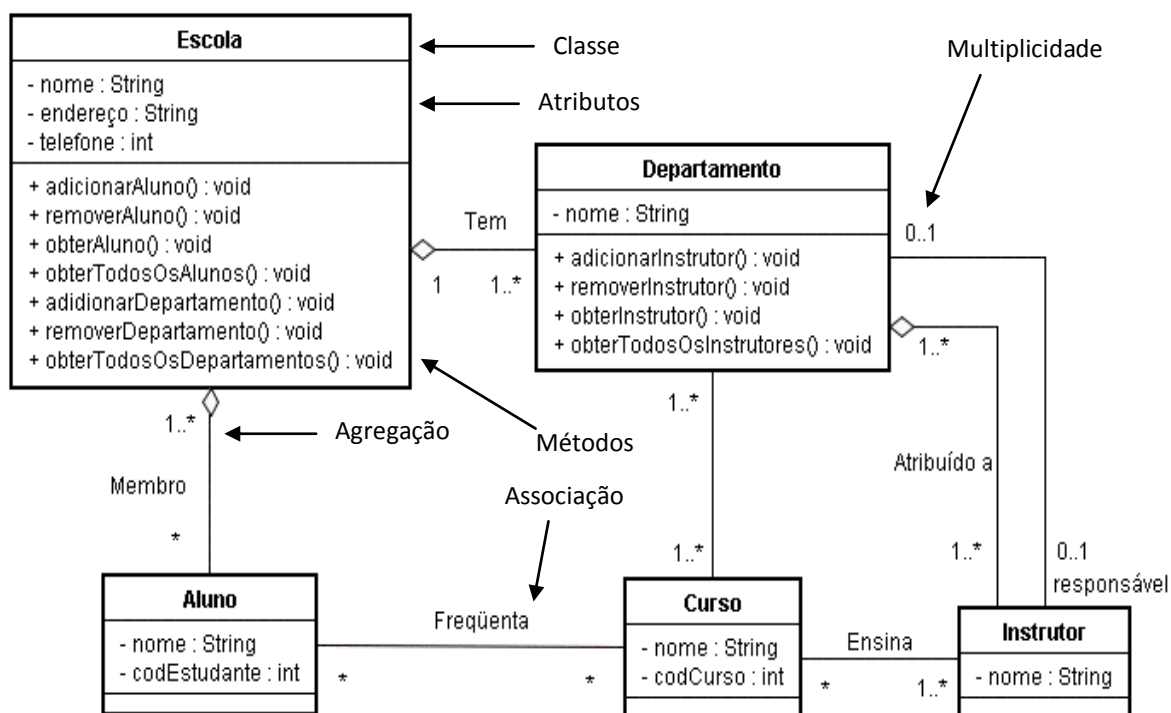
Fonte: BOOCH, 2000, p.232

### 2.2.1.2 Diagrama de Classes

Os diagramas de classes são utilizados para fazer a modelagem da visão estática de um sistema e é base para os diagramas de componentes e os diagramas de implantação. É composto pelas classes envolvidas no projeto e seus relacionamentos, que podem ser de associação, dependência, localização, agregação, restrição e generalização. Além disso, em alguns casos, se faz importante definir a quantidade de objetos que podem ser conectados pela instância de uma associação, sendo essa quantidade chamada multiplicidade, e expressada por um intervalo de valores ou até mesmo um valor explícito (BOOCH, 2000).

A classe é representada por um quadrado/retângulo, repartido com duas linhas de modo a gerar três partes, sendo uma para o nome da classe, outra para os atributos da classe, e a outra para os métodos da classe.

A Figura 05 a seguir mostra um exemplo de diagrama de classes e alguns relacionamentos.



**Figura 05 - Exemplo de diagrama de classes**

Fonte: BOOCH, 200, p.111

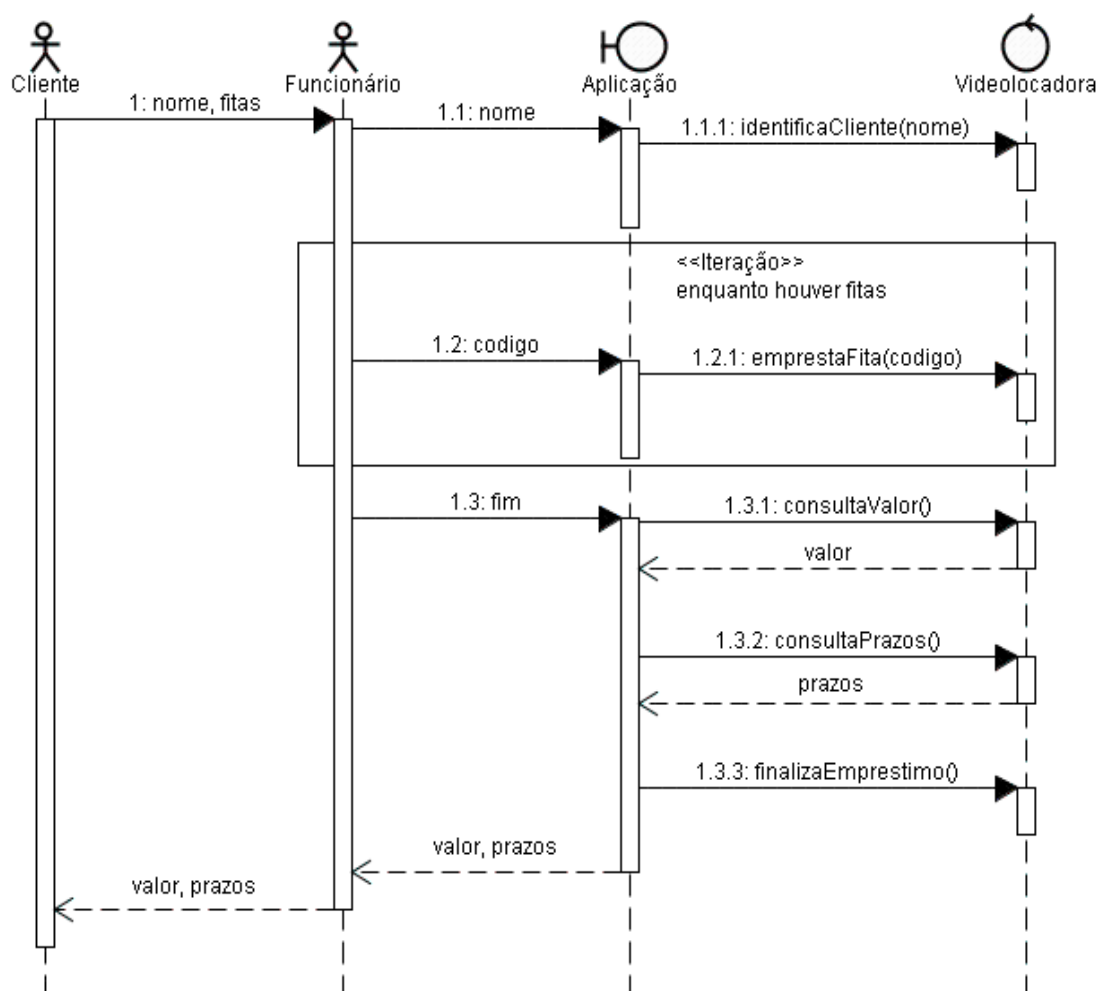
### 2.2.1.3 Diagrama de Seqüência

Os diagramas de seqüência descrevem, como o próprio nome sugere, a seqüência de comunicações entre objetos de um processo ao longo da linha do tempo. Porém, como em um sistema computacional podem existir vários processos, é sugerido que sejam construídos a partir dos casos de uso, ou seja, para cada caso de uso seria construído um diagrama de seqüência, descrevendo as seqüências normais de comunicação entre objetos e diagramas complementares com seqüência alternativas e tratamentos de erros (MACORATTI, 2010).



Graficamente o diagrama é composto por atores, como no diagrama de caso de uso, e instancias de objetos constituintes do sistema. Cada um dos componentes terá uma linha de tempo, representada por uma linha vertical, que estará tracejada quando o sistema ou o ator estiverem inativos, e cheia quando estiverem ativos (WASLAWICK, 2004).

A Figura 06 abaixo demonstra um exemplo de diagrama de seqüência, de um caso de uso de uma vídeo-locadora.



**Figura 06 - Exemplo de Diagrama de Seqüência**

Fonte: WAZLAWICK, 2004, p. 94

## 2.3 Requisitos

Os requisitos de um sistema, segundo Sommerville (2007), são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais, sendo que a imprecisão na especificação de requisitos é motivo de muitos problemas de engenharia de software.

Segundo Wazlawick (2004), a análise de requisitos é fundamental para o desenvolvimento de sistemas, pois trata justamente de descobrir o que o cliente espera do sistema. Ela está associada ao processo de descobrir quais são as características que o sistema deve apresentar, em termos de como o sistema deve ser, além de quais funções o sistema deve realizar e quais são as restrições que existem sobre estas funções. Assim, os requisitos podem ser classificados em:

- **Funcionais:** requisitos funcionais correspondem à listagem de todas as coisas que o sistema deve fazer, serviços que o sistema deve fornecer, como o sistema deve se comportar em determinadas situações, como por exemplo: cadastrar itens, realizar cálculos, gerar relatórios.
- **Não funcionais:** requisitos não-funcionais são restrições que se coloca sobre como o sistema deve realizar seus requisitos funcionais. Portanto, podem especificar desempenho, proteção, disponibilidade, tempo de resposta, espaço de armazenamento e outras propriedades exigidas para o sistema.

Os requisitos não funcionais não estão relacionados apenas com o sistema de software a ser desenvolvido. Alguns deles podem restringir o processo que deve ser usado para desenvolver o sistema, como por exemplo, uma especificação dos padrões de qualidade (SOMMERVILLE, 2007).

Os requisitos não-funcionais ainda podem ser divididos em subcategorias (SOMMERVILLE, 2007):

- **Requisitos de produto** – são aqueles que visam especificar como o produto deve “ser”. Como por exemplo, desempenho, rapidez, armazenamento em disco e memória necessária.
- **Requisitos organizacionais** – os requisitos organizacionais dizem respeito às metas da empresa do cliente e do desenvolvedor, suas políticas e estratégias

adotadas e os seus relacionamentos. Como exemplo, pode-se citar a linguagem de programação a ser utilizada, os padrões de processo que devem ser utilizados, requisitos de entrega estipulando como e quando o produto e sua documentação devem ser entregues.

- Requisitos externos – são aqueles que contemplam exigências externas ao sistema e seu processo de desenvolvimento, como por exemplo, uma certificação que deve ser atendida, a necessidade de interagir com outros sistemas já implantados, requisitos éticos e aspectos legais que devem ser seguidos.

### **2.3.1 Requisitos de Domínio**

Os requisitos de domínio são aqueles que são extraídos do domínio de aplicação do sistema que se está desenvolvendo, podendo fazer referência a conceitos ou terminologias específicas ao campo de atuação do sistema, em vez das necessidades dos usuários do sistema, podendo ser novos requisitos funcionais, restringir os requisitos funcionais existentes ou estabelecer como cálculos específicos do sistema devem ser resolvidos (SOMMERVILLE, 2007). Geralmente são difíceis de serem extraídos devido à dificuldade de compreender como eles estão relacionados a outros requisitos do sistema.

### **2.3.2 Requisitos de Usuário**

São os requisitos funcionais e não funcionais descritos em uma linguagem compreensível aos usuários do sistema. Devem ser escritos de forma simples, sem jargões de software ou técnicos demais, notações estruturadas ou formais, ou ainda descrever por meio da implementação do sistema. Devem-se usar preferencialmente

tabelas e formulários simples e diagramas intuitivos, evitando-se assim, problemas como falta de clareza e confusão de requisitos (SOMMERVILLE, 2007).

## **2.4 Rational Unified Process - RUP**

Booch (2004) define processo como “um conjunto de passos parcialmente ordenados com a intenção de atingir uma meta”. A UML é amplamente independente de processo, podendo ser utilizada com os vários processos da engenharia de software.

O RUP (Processo Racional Unificado) é um processo *iterativo*, ou seja, permite aperfeiçoamentos sucessivos e o desenvolvimento incremental de uma solução efetiva em vários ciclos, e é composto por quatro fases: Concepção, Elaboração, Construção e Transição, conforme descritas a seguir (BOOCH, 2004):

### **1. Concepção**

Nessa fase é delimitado o escopo do projeto, incluindo critérios de sucesso, avaliação de riscos e avaliação de recursos necessários, e é estabelecido o caso de negócio. Também é comum a criação de um protótipo, para testes e melhor compreensão do caso de negócio.

### **2. Elaboração**

A fase de elaboração tem como objetivo realizar a análise do domínio do problema, para que se possa desenvolver um plano do projeto de modo a eliminar os elementos de mais alto risco. Isso implica uma descrição da maioria dos requisitos do sistema. Como resultado, é possível examinar o escopo, os objetivos e os riscos do sistema, embasando a decisão de continuar ou não com o projeto.

### **3. Construção**

A construção é realizada de forma iterativa e incremental, sendo então coletados os requisitos restantes, estabelecido critérios de aceitação, concluindo a

implementação e o teste do software, até o momento em que o software, o ambiente e os usuários estejam todos prontos para a implantação do novo sistema.

#### **4. Transição**

Durante a fase de transição, é disponibilizada aos usuários finais uma versão beta do sistema, para que se possa ajustar o sistema e corrigir possíveis problemas identificados, para que depois seja substituída pelo sistema de produção.

No final da fase de transição, é verificado se os objetivos do ciclo de vida foram atingidos, ou se um novo ciclo se faz necessário, até o momento em que o software esteja concluído.

### **2.5 Capacidade Produtiva e Cronoanálise**

Nesta seção serão abordados os dois conceitos principais que oferecem embasamento ao projeto, que são a Capacidade Produtiva e a Cronoanálise, as quais se subdividem conforme os itens a seguir.

#### **2.5.1 Planejamento da capacidade**

A indústria só pode planejar dentro dos limites produtivos de suas máquinas, e o administrador, obrigatoriamente, tem que conhecer o potencial dos equipamentos dos quais dispõe.

Prover a capacidade produtiva para satisfazer à demanda atual e futura é uma responsabilidade fundamental da administração da produção. O equilíbrio adequado entre capacidade e demanda pode gerar lucros e clientes satisfeitos, enquanto que o mau equilíbrio levará à conseqüências problemáticas.

A importância do planejamento industrial de uma empresa está em conhecer cada fase das metas a serem atingidas, saber como proceder para alcançá-las e ter conhecimento dos recursos necessários. O planejamento consiste em selecionar uma alternativa de ação futura, onde a visão das coisas é antecipada, onde se tem a capacidade de prever antecipadamente os fatos (DAVIS, 2001).

A transformação de materiais em produtos, através de operações industriais é conhecida como processo produtivo. Os processos produtivos modificam materiais nas suas formas e/ou estruturas, dando-lhes novas funções, acrescentando valores e utilidades, isto é, gerando bens de consumo ou de capital para atender às demandas de mercado.

Esses processos são fenômenos físicos e requerem conhecimentos técnicos derivados das ciências físico-químicas e suas aplicações tecnológicas (ASTEC, 1999).

Pode-se chamar de capacidade a quantidade de produtos e serviços que podem ser produzidos em uma unidade produtiva, num dado intervalo de tempo. Por unidade produtiva entende-se uma fábrica, um departamento, um armazém, uma loja.

A projeção de demanda fornece estimativas de necessidade ao longo do tempo e dá informações sobre a capacidade a produzir, considerando: a estimativa da demanda que se pretende atingir, e avaliação financeira (DAVIS, 2001).

A administração da produtividade é um processo formal de gestão, envolvendo todos os níveis de gerência e colaboradores, com o objetivo último de reduzir custos de manufatura, distribuição e venda de um produto ou serviço através da integração das quatro fases do Ciclo da Produtividade, ou seja, medida, avaliação, planejamento e melhoria (MARTINS, 1998).

Existem muitos fatores dos quais depende a capacidade de uma unidade produtiva. Alguns dos fatores mais importantes influentes na capacidade são os seguintes (MARTINS, 1998):

1. Instalações – O tamanho da unidade produtiva é obviamente importante. Sempre que possível, ao projetar a unidade, tenta-se deixar um espaço vago para expansões futuras. O arranjo físico das instalações pode restringir a capacidade ou favorecê-la, quando bem elaborado pode resolver um problema imediato de capacidade. Outros fatores como aquecimento, iluminação e ruído também exercem

influência positiva ou negativa, dependendo de como atuam no funcionário de forma apropriada ou não.

2. Composição dos produtos ou serviços – em geral, a diversidade reduz a capacidade. Produtos uniformes dão oportunidade para a padronização de métodos e materiais, reduzindo o tempo de operação e aumentando a capacidade. Produtos diferentes exigem constantes alterações de processo e preparação de máquinas quando passa de um produto para outro e essas alterações, muitas vezes, exigem que as máquinas fiquem por algum tempo paradas, assim como os funcionários devem abastecer seus postos de trabalho ou ainda aguardar a preparação e dessa forma acaba reduzindo a capacidade.

3. O projeto do processo – em teoria, os projetos de processo variam desde totalmente manuais até os totalmente automatizados cada um com seus graus de complexidade e produtividade. A decisão sobre o modelo do projeto de processo varia conforme o segmento, escala de produção, complexidade de produção, espaço físico e assim por diante.

4. Fatores humanos – dada certa quantidade de recursos técnicos, o quadro e a habilidade dos funcionários pode aumentar a capacidade. O capital humano pode ser melhorado através de treinamento, aumento da habilidade dos funcionários e a experiência. Outro fator é a motivação, que é muito importante porque é ligada à satisfação com a companhia, com o ambiente de trabalho, com a variedade e os desafios impostos e pelas tarefas, com o nível salarial entre outros.

5. Fatores Operacionais – mais ligados à rotina de trabalho dos setores produtivos da empresa, podem ser organizados de forma a conduzir a capacidades maiores ou menores, ou pelo menos de maneira a facilitar ou dificultar o aproveitamento da capacidade existente em potencial. Havendo diferenças sensíveis na capacidade de processamento de um equipamento para outro, observar-se-á que os equipamentos ou setores mais lentos acabarão por determinar a velocidade dos demais.

6. Fatores externos – a capacidade pode, também, ser afetada por fatores que nascem foras das fronteiras da própria empresa, mas que nem por isso deixam de exercer sua influência, às vezes até de forma mais marcante que os fatores internos. Um bom exemplo são os padrões de qualidade e desempenho exigidos

dos produtos por parte dos clientes. Tais exigências podem acabar se constituindo numa barreira ao aumento da capacidade ou mesmo ao uso da capacidade atual.

Ao instalar seus parques fabris, as indústrias compram equipamentos que lhes permitem produzir um determinado volume de produtos. As máquinas, ao serem compradas, vêm com uma capacidade registrada nos seus catálogos, denominada capacidade nominal – aquela conseguida quando os equipamentos trabalham a plena carga e sem interrupção ou com eficiência 100%. A soma das capacidades nominais dos diversos conjuntos de máquinas forma a capacidade instalada da indústria. Diz-se conjunto porque tanto pode existir uma máquina que inicia e conclui um produto, como ele pode ser originado por diversas máquinas que trabalham em conjunto, uma dependendo da outra.

Capacidade efetiva é a capacidade real que a fábrica tem, considerando-se determinada eficiência sobre a capacidade nominal. É a capacidade efetivamente conseguida no trabalho normal dos equipamentos. Se a indústria não utiliza toda sua capacidade instalada, diz-se que há ociosidade. Capacidade ociosa é aquela disponível na empresa e que não esteja sendo utilizada. Se tiver capacidade 100 e trabalha a nível 60, utilizando 60% do que pode produzir, opera com 40% de ociosidade. Sua capacidade ociosa é de 40%.

Capacidade ideal de uma indústria é aquela na qual se obtém custo mínimo, e ela é conseguida, por exemplo, por meio da produção de lotes econômicos (RITZMAN, 2004).

### **2.5.2 Capacidade e turnos de trabalho**

Para a determinação da capacidade de produção não é suficiente somente a análise das vendas por período. Deve-se tomar um conjunto de decisões com relação à capacidade. Por exemplo, para atender demandas futuras, é necessário utilizar a capacidade efetiva, a capacidade máxima ou outro valor de capacidade. Além disso, é preciso decidir sobre o número de turnos de trabalho, número de



pessoal por turno, entre outras variáveis. Tais decisões devem ser analisadas com relação à capacidade financeira da empresa.

Somente após a determinação da capacidade e da quantidade de turnos de trabalhos a serem utilizados bem como da capacidade financeira da organização é que podem ser iniciados os procedimentos para a implantação do *layout* das máquinas na fábrica. A capacidade de produção da empresa depende dos gargalos, isto é, dos processos ou dos equipamentos que limitam a capacidade de produção e que devem ser identificados (RITZMAN, 2004).

### **2.5.3 Tempos cronometrados**

A cronometragem é um dos métodos mais empregados na indústria para medir o trabalho. Desde a época em que F. W. Taylor estruturou a administração científica e o estudo de tempos cronometrados, objetivando medir a eficiência individual, essa metodologia continua sendo muito utilizada para que sejam estabelecidos padrões para a produção e para os custos industriais (MARTINS, 1998).

A tomada de tempos é obtida através da utilização de cronômetros que são utilizados para medir a duração de uma operação. Mais adiante será tratado sobre a obtenção dos tempos por operação no item 4.6.6.

### **2.5.4 Finalidade dos estudos de tempos**

A eficiência dos tempos padrões de produção é influenciada pelo tipo de fluxo de materiais dentro da empresa, processos escolhidos, tecnologias utilizadas e características do trabalho que está sendo analisado.

Quanto maior a intervenção humana na produção, maior é a dificuldade de se medir corretamente os tempos, uma vez que cada operador tem habilidades, força e vontades diferentes. Os dados extraídos na medida do tempo são importantes para (MARTINS, 1998):

- Estabelecer padrões para os programas de produção para permitir o planejamento de fábrica, utilizando com eficácia os recursos disponíveis e, também, para analisar o desempenho da produção em relação ao padrão existente.
- Fornecer dados para a determinação dos custos padrões, para levantamento dos custos de fabricação, determinação de orçamentos e estimativa do custo de um produto novo.
- Fornecer dados para estudo de balanceamento de estruturas de produção, comparar roteiros de fabricação e analisar o planejamento de capacidade.

### **2.5.5 Cronoanálise industrial**

A cronoanálise tem como função a determinação dos tempos operacionais e de outras atividades industriais necessárias aos processos produtivos.

Os tempos extraídos da cronoanálise regem a Administração da Produção. Esses dados tornam palpáveis as funções administrativas, que são diretamente relacionadas à produtividade e à eficácia da produção na indústria (Martins, 1998).

Outras funções da cronoanálise industrial são utilizadas no levantamento de dados operacionais de produção: tempos, nº de homens, nº de máquinas, nº de operações simultâneas, descrição das operações, elaboração de cálculos estatísticos para a obtenção do tempo padrão e roteiro de fabricação, dentre outros.

Dentre as principais utilizações dos tempos destacam-se (ASTEC, 2000):

- Medida objetiva da produtividade;
- Implantação de incentivos salariais;
- Cálculo dos custos operacionais teóricos e reais;
- Cálculo de carga de trabalho;
- Determinação da capacidade produtiva;

- Estudos de métodos e processos industriais;
- Em diversas análises de produção;

A ferramenta conhecida como cronoanálise industrial usa a cronometragem para apurar a medição do tempo, aliada à estatística que determina o número de medições exigidas e o grau de confiabilidade para obter um tempo puro de cada operação, para a definição do tempo previsto, analisando o tempo medido para avaliar o ritmo do operador e conseqüentemente a capacidade produtiva instalada.

### **2.5.6 Metodologia de Determinação do Tempo Padrão**

As principais etapas para aplicação da técnica de determinação do tempo padrão consistem inicialmente em discutir com todos os envolvidos o tipo de trabalho que será executado, procurando obter a colaboração dos encarregados e dos operadores do setor. Em seguida deve-se definir o método da operação e dividi-la em elementos. O operador que irá realizar a operação deve ser treinado para executá-la, conforme o estabelecido.

Para auxílio e registro, deve-se elaborar um desenho esquemático do local de trabalho, anotando também todos os dados adicionais necessários. Após estes preparativos, realiza-se uma cronometragem preliminar para obter dados necessários à determinação do número necessário de cronometragens ou ciclos. Com as cronometragens, determina-se o tempo médio. O estudo deve ainda avaliar o fator ritmo ou velocidade da operação, tempo normal, tolerâncias para fadiga, e para necessidades pessoais. Finalmente o tempo padrão da operação é obtido (ASTEC, 2000).

### 2.5.7 Determinação do número de ciclos a serem cronometrados

Para se determinar o tempo padrão de uma peça ou de uma operação, o número de amostras deve ser determinado a partir do nível de confiança a que se pretende obter. Contudo a maneira mais correta para determinar o número de ciclos a serem cronometrados é deduzida da expressão do intervalo de confiança da distribuição por amostragem da média de uma variável distribuída normalmente, resultando na expressão (MARTINS, 1998):

$$n = (z \cdot R / E_r \cdot d_2 \cdot X)^2$$

Em que:

n = número de ciclos a serem cronometrados.

z = coeficiente de distribuição normal padrão para uma probabilidade determinada.

R = amplitude da amostra.

$E_r$  = erro relativo.

$d_2$  = coeficiente em função do número de cronometragens realizadas preliminarmente.

X = média da amostra.

A Tabela 01 apresenta os níveis de confiança normalmente utilizados.

**Tabela 01 - Níveis de confiança utilizados na análise dos tempos de ciclo**

NC ( % )	99,73	99,00	98,00	95,45	95,00	90,00	68,27
Zc	3,00	2,58	2,33	2,00	1,96	1,65	1,00

Fonte: Martins, 1998 p.88

### 2.5.8 Determinação das tolerâncias

Não é possível esperar que uma pessoa trabalhe sem interrupções o dia inteiro, devem ser previstas interrupções no trabalho para que sejam atendidas as

denominadas necessidades pessoais e para proporcionar um descanso, aliviando os efeitos da fadiga no trabalho. Para a tolerância no atendimento às necessidades pessoais, considera-se suficiente um tempo entre 10 e 25 minutos (5% aproximadamente) por dia de trabalho de 8 horas.

Na tolerância para alívio de fadiga deve ser observado não somente o trabalho realizado, mas também das condições ambientais do local de trabalho. Em função dos diferentes fatores que dificultam o trabalho haverá muita diferença no tempo destinado ao descanso. As tolerâncias concedidas para fadiga têm valor entre 10 e 50%, dependendo do grau de trabalho (MARTINS, 1998). Conforme a Tabela 02.

**Tabela 02 - Tolerância para alívio de fadiga**

Condições de Conforto	% sobre duração do trabalho	
	homens	mulheres
locais confortáveis	3%	5%
locais desconfortáveis	4,50%	7%
locais com poeira, sujeira e ruídos excessivos	12%	20%

Fonte: Elaborado pelo autor.

Adicionam-se ao tempo normal da cronometragem os percentuais de necessidades pessoais, atrasos imprevisíveis e fadiga, conforme a Tabela 03, o qual auxilia no cálculo da fadiga.

**Tabela 03 - Cálculo da fadiga**

NÍVEIS DE ESFORÇOS			CONDIÇÕES		MONOTONIA DO		
NÍVEIS	MENTAL	VISUAL	FÍSICO	AMBIENTAIS		TRABALHO	
	%	%	%	TIPO	%	CICLO (minuto)	%
MUITO LEVE	1	1	3	A	0	até 0,5	5
LEVE	2	2	5	B	2	de 0,6 a 1,0	4
MÉDIO	4	4	8	C	4	de 1,1 a 1,5	3
PESADO	7	7	12	D	6	acima de 1,5 , utilizar tipo A	
MUITO PESADO	10	10	18	E	8		

Fonte: ASTEC, 2000

Normalmente costuma-se adotar o fator de tolerância variando entre 1,1 e 1,20 para trabalho em unidades industriais com boas condições ambientais e trabalhos com nível de fadiga intermediário.

Para calcular os níveis de fadiga é necessário analisar a intensidade dos fatores para cada tipo de departamento e/ou atividade, estes estão descritos abaixo:

1. Tipo A

- Sem ruídos, poeira ou umidade. Calor ou frio normal.
- Iluminação e ventilação adequadas
- Sem riscos de acidentes ou doenças profissionais

2. Tipo E

- Ruidoso, muita poeira e/ou umidade. Excesso de calor ou frio artificial.
- Iluminação e ventilação péssimas
- Razoável risco de acidentes e doenças profissionais

Então para as condições ambientais tipo B, C, e D, considerar que são intermediárias as do tipo A e E.

Para a análise das intensidades dos níveis de esforços e condições ambientais, o ideal é que o Analista de Métodos e Processos calcule os fatores da fadiga junto com o Engenheiro de Segurança (ASTECC, 2000).

### **2.5.9 Determinação do tempo padrão**

Tempo Padrão (TP) é o tempo necessário para realizar uma operação dentro de um método estabelecido, em condições determinadas, realizada por um operário apto e treinado, possuindo habilidade média e trabalhando com esforço médio. Com base nesta definição a operação deve ser executada (ASTECC, 2000):

- Dentro de um método estabelecido - se houver mudança nos métodos ou no tipo de equipamentos, dispositivos, ferramentas ou materiais utilizados, haverá condições diferentes e o TP deverá ser revisto;

- Por um operário apto e treinado, ou seja, não se deve cronometrar um aprendiz, nem pessoas sem aptidões para a operação;
- Por um operário com habilidade e esforço médio - qualquer operação requer certa habilidade e esforço para ser executada. Mesmo que tenhamos dois operários considerados normais para um serviço, encontram-se sensíveis variações de habilidade e esforço (ou seja, a atuação pode ser boa, média ou fraca). Caberá ao cronoanalista efetuar o julgamento do esforço e habilidade demonstrados pelo operário.

A normalização dos tempos, talvez seja a parte mais delicada da cronometragem, pois pela intervenção do julgamento ocorre o fato de diferentes observadores formularem avaliações também desiguais. A experiência tem demonstrado que uniformidade no julgamento pode ser fator fundamental para estabelecer um tempo padrão correto (MARTINS, 1998).

A determinação do TP se faz por meio da cronometragem da operação utilizando-se cronômetros industriais e requer pessoal especializado (cronoanalista) para a sua elaboração. Uma vez que as tolerâncias ou suplementos para cada elemento foram estudados e expressos percentualmente, é estabelecido o tempo padrão para cada elemento, pelo acréscimo destes suplementos ao tempo estimado (ASTECC, 2000).

A determinação do TP pode então ser definida como uma série de elementos que compõem o tempo cronometrado, agregando fatores externos à operação em si. Fatores como o cansaço do operador, necessidades fisiológicas e outros fatores imprevisíveis, que podem atrasar o término da operação, são adicionados ao tempo cronometrado para garantir um tempo normal apesar dos atrasos.

## **2.6 Sistema de Apoio à Decisão (SAD)**

Em um sistema de manufatura, toda vez que são formulados objetivos, é necessário formular planos de como atingi-los, organizar recursos humanos e físicos necessários para a ação, dirigir a ação dos recursos humanos sobre os recursos

físicos e controlar esta ação para a correção de eventuais desvios. No âmbito da administração da produção, este processo é realizado pela função de Planejamento e Controle da Produção (PCP).

Zacarelli (1979) denomina o PCP como Programação e Controle da Produção, definindo-o como "... um conjunto de funções inter-relacionadas que objetivam comandar o processo produtivo e coordená-lo com os demais setores administrativos da empresa".

Sendo assim, pode-se considerar o PCP como um elemento central na estrutura administrativa de um sistema de manufatura, passando a ser um elemento decisivo para a integração da manufatura.

Logo, a tomada de decisão para o que produzir, quanto produzir e quando produzir, deve estar baseada em uma fonte de informações necessárias para a formação de um modelo que dê parâmetros e argumentos suficientes ao gestor.

Quando uma organização não possui sistemas de informação que auxiliem o processo de tomada de decisão, as decisões são baseadas em dados históricos gerados manualmente, muitas vezes inconsistentes e em experiências individuais. Quando existe um Sistema de Apoio à Decisão (SAD) apoiando esse processo, as informações fornecidas por ele são incorporadas aos dados históricos e experiências individuais, possibilitando melhores condições para a tomada de decisão.

Em qualquer das situações citadas, a saída ou a mudança de um colaborador poderá causar grande impacto na organização. Isto acontecerá devido à perda da história de como as decisões foram tomadas.

Decisão é uma escolha entre as alternativas existentes através de estimativas dos pesos destas alternativas. Apoio à decisão significa auxiliar nesta escolha gerando estas estimativas, a evolução ou comparação e escolha. Finlay (1994) e outros autores definem o SAD de um modo geral como "um sistema computacional que auxilia o processo de tomada de decisão".

Quando se fala em auxiliar o processo de tomada de decisão, isso não significa somente fornecer informações para apoio à decisão, mas, também, analisar alternativas, propor soluções, pesquisar o histórico das decisões tomadas, simular situações, etc.

Muitas decisões podem ser tomadas através de modelos simulados em computadores que servem para analisar e avaliar um amplo conjunto de problemas



do mundo real. As alternativas de decisão podem ser analisadas e validadas através de simulação antes que a decisão seja tomada (FITZ, 1993).

LAUDON (2004) afirma que um SAD tem por objetivo auxiliar o processo de decisão gerencial, combinando dados, ferramentas e modelos analíticos sofisticados e software amigável ao usuário em um único e poderoso sistema que pode dar suporte à tomada de decisão semi-estruturada e não-estruturada. Além disso, um SAD fornece aos usuários um conjunto flexível de ferramentas e capacidades para analisar dados importantes.

A simulação é uma forma de imitar a realidade sem correr os riscos, os custos e o tempo que resultariam se fosse necessário experimentar.

### **3 PROCEDIMENTOS METODOLÓGICOS**

A modelagem de um sistema é um processo que requer um prévio estudo dos métodos que são adotados atualmente, para que se possa identificar seus componentes e como eles se relacionam, dentro de um modelo específico de negócio.

Para a modelagem do sistema foi utilizado o referencial disponível de diversos autores da área, bem como os conhecimentos adquiridos nas disciplinas de análise de sistemas.

#### **3.1 Tipo de Pesquisa**

Para o desenvolvimento deste trabalho foram utilizadas três modalidades de pesquisa: a pesquisa Bibliográfica, Documental e Estudo de Caso.

A pesquisa bibliográfica, segundo CARVALHO (2010), é preciso ser feita na fase inicial de um desenvolvimento de investigação com o intuito de saber: Saber se alguém já publicou as respostas às questões propostas e decidir se é interessante repetir a investigação com os mesmos objetivos; saber quais os métodos utilizados em investigações similares e averiguar o melhor para ser aplicado; enquadrar o nosso estudo em um modelo de causalidade, diferenciando a variável resposta e as variáveis interferentes, facilitando assim, a identificação dos meios para controlá-las logo no início, não deixando que estas confundam nosso resultado.

Por se tratar de uma análise de um sistema que tem como objetivo apresentar a capacidade produtiva com base em cronoanálise industrial foi realizada uma pesquisa bibliográfica acerca desses assuntos, a fim de dar o devido embasamento à posterior modelagem do sistema.

A pesquisa bibliográfica foi feita mediante consultas ao acervo da biblioteca da UTFPR, campus Pato Branco e através de artigos e materiais publicados na Internet.

Segundo Barroso (2010), o estudo de caso é uma metodologia de investigação utilizada para explorar ou descrever acontecimentos e contextos complexos, nos quais estão envolvidos simultaneamente vários fatores. Esse tipo de abordagem não representa um método por si só, mas uma estratégia de pesquisa que permite o uso de métodos qualitativos e quantitativos, uma vez que o estudo de caso fica intimamente ligado ao contexto ou processo estudado (CASTILLO, 2010).

Portanto, a presente pesquisa caracteriza-se ainda como um estudo de caso, pois analisa o caso de uma pequena empresa do setor industrial, localizada na cidade de Pato Branco, a qual se utiliza de um sistema para geração da capacidade produtiva com base na cronoanálise industrial.

A pesquisa documental também se fez necessária, pois segundo SILVA (*apud* Figueiredo, 2010), documentos são utilizados como fontes de informações, indicações e esclarecimentos que trazem seu conteúdo para elucidar determinadas questões e servir de prova para outras, de acordo com o interesse do pesquisador.

Neste trabalho foram analisados os documentos e planilhas gerados pela empresa utilizada como caso de estudo, por meio dos quais foi possível analisar o sistema já existente – atualmente disponível em planilhas do Excel (conforme o Anexo A), a fim de levantar os requisitos funcionais e não-funcionais necessários ao sistema. Além disso, foram obtidas informações junto a um consultor da área de produção, visando melhor esclarecer os requisitos necessários ao sistema.

### **3.2 População Amostra**

Não foi calculada uma amostra para a pesquisa realizada, já que se tratou de um estudo de caso de uma empresa do setor industrial.

### **3.3 Coleta dos Dados**

A coleta de dados foi realizada através de entrevistas e reuniões com consultores da área, durante o período de desenvolvimento do projeto, analisando documentos e planilhas do Excel gerado pela empresa estudada, que fazem parte do sistema atual adotado pela empresa.

#### **3.3.1 Metodologia de Determinação do Tempo Padrão**

A determinação do tempo padrão consiste inicialmente em discutir com todos os envolvidos o tipo de trabalho que será executado, procurando obter a colaboração dos encarregados e dos operadores do setor. Em seguida deve-se definir o método da operação e dividi-la em elementos. O operador que irá realizar a operação deve ser treinado para executá-la. Para auxílio e registro, deve-se elaborar um desenho esquemático do local de trabalho, anotando também todos os dados adicionais necessários.

Após estes preparativos, determina-se uma quantidade de amostras a serem coletadas através do índice de confiança desejado pela empresa. Por exemplo, para um índice de confiança de 95%, serão necessárias 20 amostras. Então se realizam 20 cronometragens da operação em questão para se obter os dados necessários à determinação do tempo médio. Além das amostras de tempo, o estudo deve ainda coletar o fator ritmo ou velocidade da operação, tempo normal, tolerâncias para fadiga, e para necessidades pessoais, para que, através de fórmula específica, o tempo padrão da operação seja obtido (ASTEC, 2000).

### 3.4 Ferramentas Utilizadas

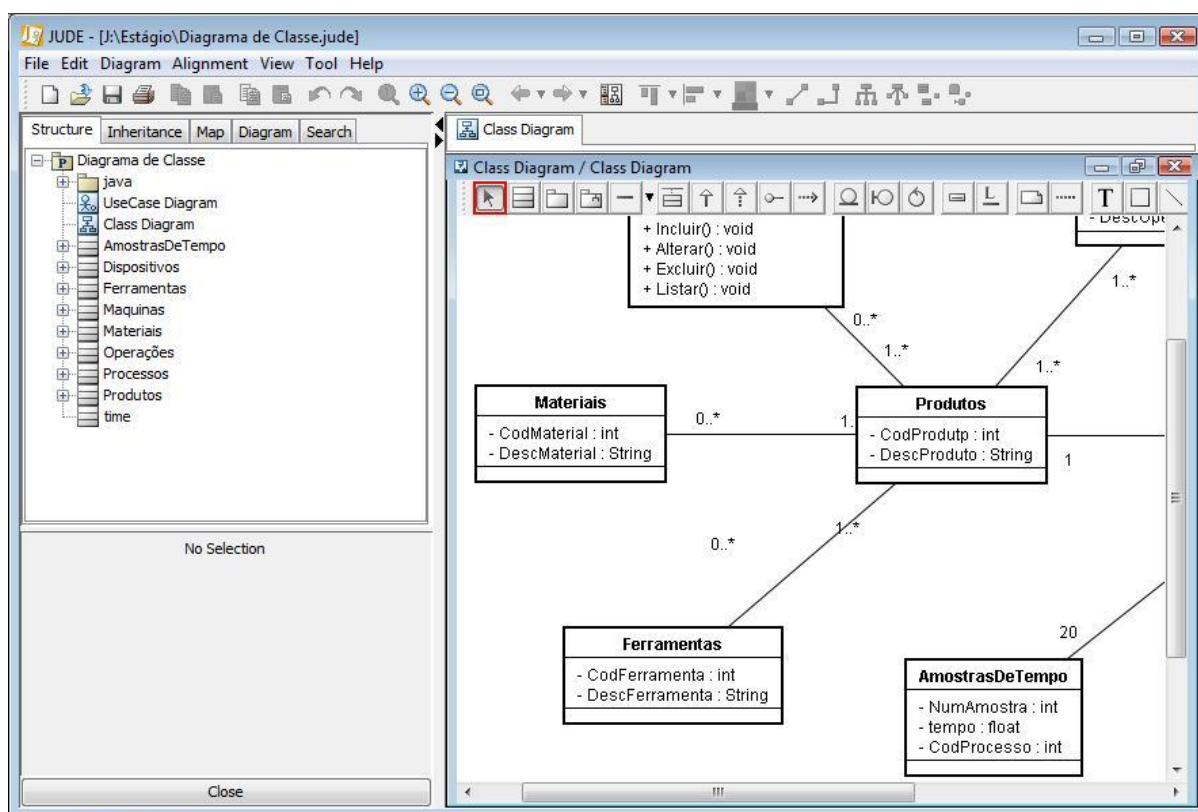
Durante o projeto foram utilizadas algumas ferramentas para a modelagem e implementação do sistema, as quais serão apresentadas a seguir.

#### 3.4.1 JUDE Community

Durante o processo de análise e modelagem deste trabalho, foi utilizada uma ferramenta CASE – *Computer Aided Software Engineering* (Engenharia de Software Auxiliada por Computador) – que como o próprio nome sugere, é uma ferramenta, um software com a finalidade de auxiliar na produção de softwares.

Segundo Carlos (2010), um dos componentes indispensáveis de uma ferramenta CASE é a modelagem visual, ou seja, a possibilidade de representar, através de modelos gráficos, o que está sendo definido.

A ferramenta CASE escolhida foi o JUDE Community – Java and UML Developer Environment (Ambiente de Desenvolvimento Java e UML) por ser uma ferramenta própria para análise, além de sua facilidade de uso e intuitividade. A Figura 07 mostra o desenvolvimento do diagrama de classes desenvolvido no JUDE.



**Figura 07 - Diagrama de Classes no JUDE Community**

Fonte: Elaborado pelo autor

### 3.4.2 NetBeans

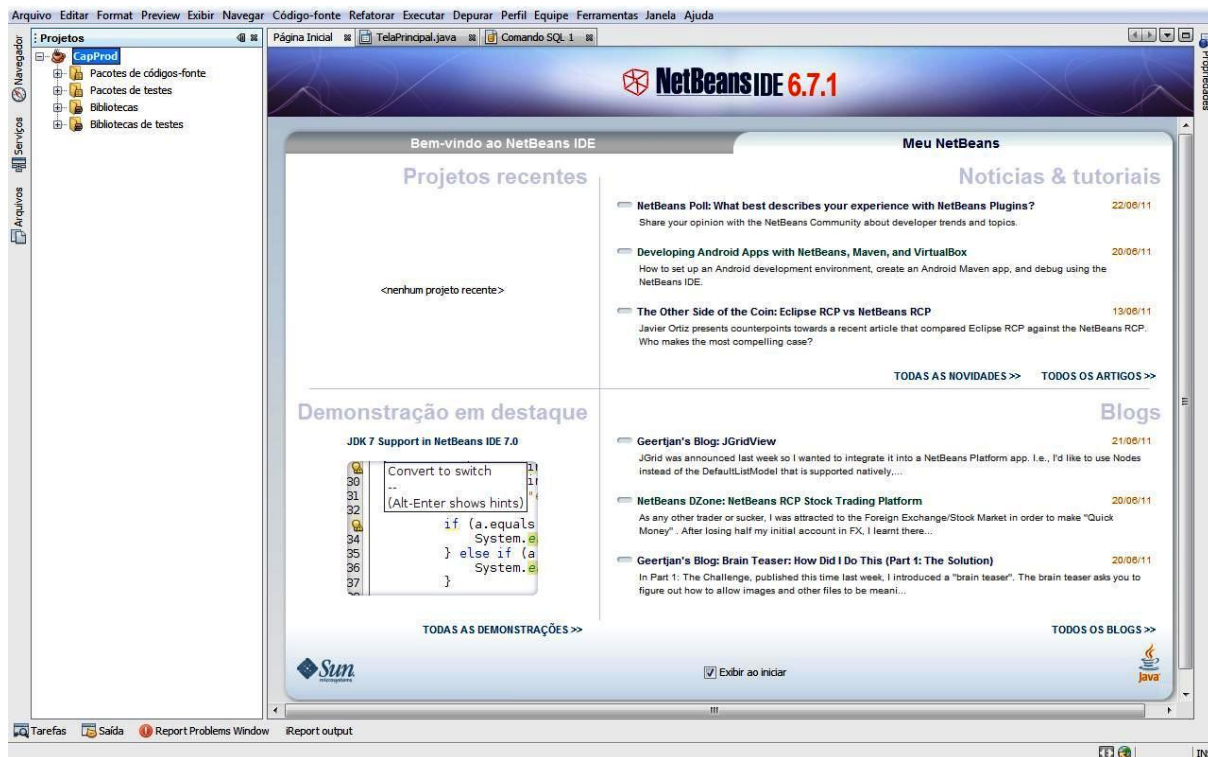
Para a codificação do projeto foi utilizado o *Netbeans*, por ser um ambiente integrado de desenvolvimento (IDE) que facilita ao programador criar programas através de recursos gráficos, além de apresentar uma interface tradicional com o desenvolvedor, com uso de menus, barras de ferramentas e outros componentes, além de editores para aplicações visuais ou web.

O *NetBeans* IDE além de ser um produto livre, sem restrições à sua forma de utilização, é escrita em Java, ou seja, não depende de plataforma, funcionando em qualquer sistema operacional que suporte a máquina virtual Java (JVM).

Além disso, outra vantagem utilizada é o fato de possuir um gerenciador de banco de dados integrado com interface gráfica, facilitando o processo de criação do

banco de dados e a posterior manipulação dos dados da aplicação, além de permitir a criação de uma instancia de um servidor de banco de dados.

A Figura 08 demonstra a IDE do *NetBeans* 6.7.1.



**Figura 08 - IDE do *NetBeans* 6.7.1**

Fonte: Elaborado pelo autor.

### 3.4.3 Banco de Dados

#### 3.4.3.1 MySQL

O banco de dados utilizado na aplicação foi o MySQL devido à algumas características, como:

- É um Software Livre com base na GPL;
- Pouco exigente quanto a recursos de hardware;
- Portabilidade (suporta praticamente qualquer plataforma atual);
- Replicação facilmente configurável.

Para gerenciar o banco de dados, foi criada uma instancia do servidor MySQL no próprio *NetBeans*.

A Figura 09 a seguir mostra o gerenciamento do banco de dados através do *NetBeans*.

NetBeans IDE 6.7.1

Arquivo Editar Format Preview Exibir Navegar Código-fonte Refatorar Executar Depurar Perfil Equipe Ferramentas Janela Ajuda

<config. padrão>

Serviços

- Banco de dados
  - Servidor MySQL em localhost:3306 [root] ← Instancia do servidor MySQL
  - Java DB
  - Drivers
  - jdbc:derby://localhost:1527/AULA [aluno em ALUNO]
  - jdbc:derby://localhost:1527/exemploProva [elquier em ELQUIER]
  - jdbc:derby://localhost:1527/prova [root em ROOT]
  - jdbc:derby://localhost:1527/sample [app em APP]
  - jdbc:mysql://localhost:3306/CapacidadeProdutiva [root em Esquema padrão] ← Banco de dados da aplicação
- capacidadeprodutiva
  - Tabelas ← Tabelas do banco
    - amostras
    - cronoanalistas
    - dispositivos
    - dispositivosdosprodutos
    - ferramentas
    - ferramentasdosprodutos
    - maquinas
    - maquinasdasoperacoes
    - materiais
    - materiaisdosprodutos
    - operacoes
    - operacoesdosprodutos
    - produtos
    - usuarios
  - Visualizações
  - Procedimentos
  - information\_schema
  - mysql
  - test
  - teste
  - jdbc:mysql://localhost:3306/information\_schema [root em Esquema padrão]

Conexão: jdbc:mysql://localhost:3306/CapacidadeProd...

```
1 select * from capacidadeprodutiva.produtos
```

select \* from capacidadedep

Tamanho da página: 20 | Total de linhas: 7

#	codProduto	nomeProduto	capacidadePorHora
1		1 Puxador Inox	3866
2		2 Grade 2x2 m	4342
3		3 Grade 2x4 m	4342
4		4 Dobradiga 10x10 inox	1361
5		5 Puxador Alumínio	181
6		6 Tubo Gás Comp Dir 4bc	6139
7		7 Teste	3767

Interface gráfica

**Figura 09 - Gerenciamento do Banco de Dados MySQL através do *NetBeans* 6.7.1**

Fonte: Elaborado pelo autor.



### 3.4.3.2 JPA - Java Persistence API

Para acessar um banco de dados relacional, o *NetBeans* utiliza uma camada intermediária que abstrai os códigos SQL, dessa maneira o programador pode fazer a manipulação dos registros, como a inclusão, exclusão e alteração sem conhecer a sintaxe SQL.

A camada intermediária de persistência utilizada na aplicação é o JPA - *Java Persistence API* (API de Persistência Java), que abstrai um banco de dados relacional para um modelo orientado a objetos. Assim, para armazenar um registro no banco de dados é necessário criar um objeto referente a esse registro, valorizar as propriedades do objeto e só então fazer a persistência (BORSOI; BRITTO, 2010).

Dentre as principais classes utilizadas pelo JPA na aplicação, destacam-se:

a) Operações – Essa classe representa um registro na tabela Operações. É instanciado um objeto dessa classe com o registro que se deseja incluir, excluir ou alterar.

b) EntityManagerFactory – Essa classe é responsável pela conexão com o banco de dados. Também é responsabilidade dessa classe carregar todas as estruturas de tabelas e fazer os mapeamentos dos dados do banco de dados para serem utilizados como objetos (BORSOI; BRITTO, 2010).

c) EntityManager – Essa classe é responsável pela execução dos comandos no banco de dados. Normalmente é criado um EntityManager para cada comando que se deseja executar no banco. Para a inclusão e alteração de registros é utilizado o método persist(), e para a exclusão o método remove(). Se for necessário fazer uma pesquisa no banco, o método createQuery() é utilizado. O mesmo espera um comando SQL e pode retornar um ou mais registros (BORSOI; BRITTO, 2010).

### 3.4.3.3 Provedor de Persistência - TopLink

O *TopLink* é um dos possíveis provedores de persistência para a JPA. Ele provê alto desempenho e escalabilidade, produtividade para o desenvolvedor e flexibilidade. Além de possuir um bom desempenho quando utilizado com o *NetBeans*, o *TopLink* trabalha em conjunto com qualquer banco de dados, incluindo bancos de dados não-relacionais, e com qualquer servidor de aplicação.

O *TopLink* provê também recursos como (DALLACQUA, 2011):

- Um rico framework que provê opções de fácil uso para formular *queries* dinâmicas e sofisticadas. Onde os desenvolvedores podem definir *queries* usando qualquer expressão, como EJB QL, SQL e *Stored Procedures*;
- Um framework de transação que provê uma transação em nível de objetos. O framework de transação suporta gerenciamento direto das transações, e também gerenciamento externo das transações através de JTA/JTS;
- Um sistema avançado de *cache* que melhora o desempenho do sistema, ao não requerer que os dados mais recentemente ou mais freqüentemente acessados sejam recuperados do banco de dados, mas sim da memória do servidor.

### 3.4.4 Relatórios *JasperReports* com *iReport*

Para a criação dos relatórios do projeto, foi adicionado ao *NetBeans* o *plug-in iReport*, que é um programa de código livre, com a capacidade de criar visualmente relatórios estáticos e dinâmicos de diferentes complexidades para aplicações Java, utilizando o *JasperReports*.

*JasperReports* é um *framework open-source* que possibilita criar qualquer tipo de relatório de forma simples e rápida, devido à sua interface gráfica intuitiva, fazendo uso de XML, possibilitando fácil manipulação pois evita modificações diretas no código fonte, minimizando assim o tempo de criação dos mais complexos relatórios (GONÇALVES, 2008).

A Figura 10 ilustra a utilização do *plug-in iReport* através do *NetBeans*.

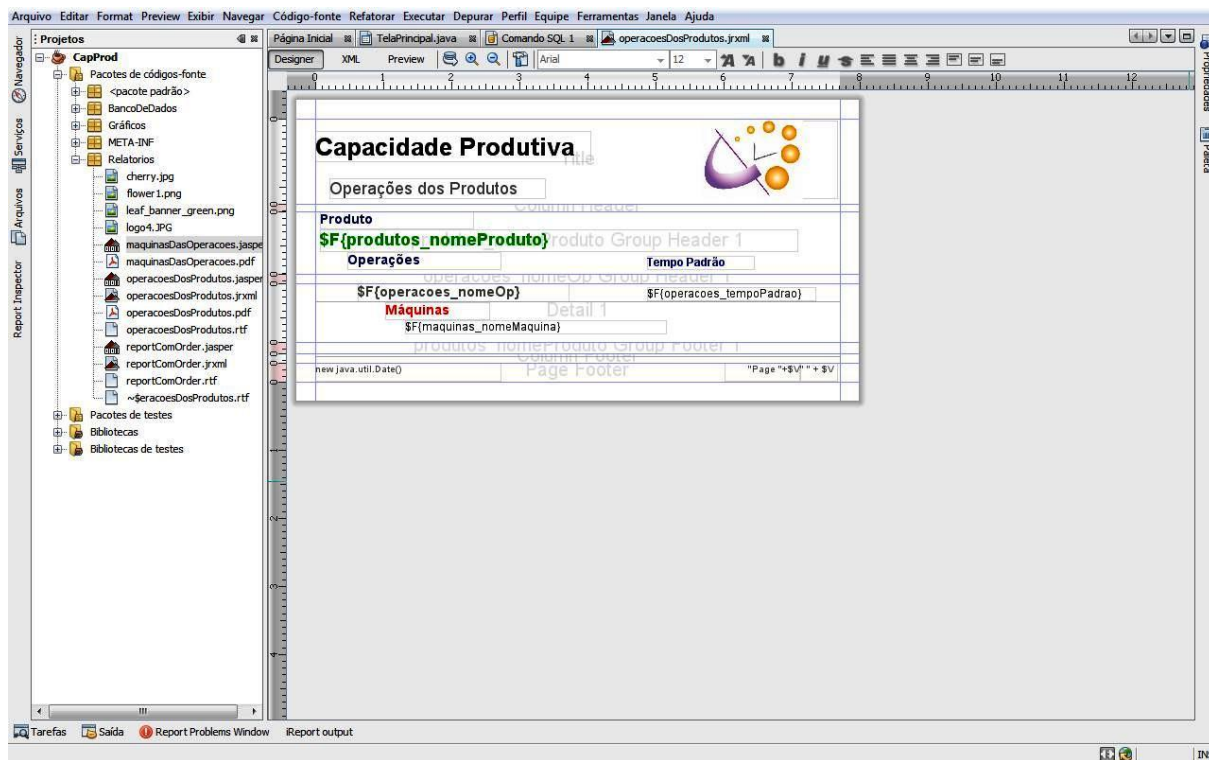


Figura 10 - Utilização do *plug-in iReport* através do *NetBeans*

Fonte: Elaborado pelo autor.

## **4 RESULTADOS E DISCUSSÕES**

Neste capítulo serão demonstradas as etapas do desenvolvimento do sistema, bem como os resultados obtidos. Primeiramente será apresentada uma descrição do sistema, o ambiente onde vai ser implantando e as informações relacionadas, na seqüência a modelagem do sistema, descrição dos requisitos e diagramas, e posteriormente uma descrição de toda a implementação do sistema, demonstrando sua interface e funcionamento.

### **4.1 Descrição do Sistema informatizado para Determinação da Capacidade Produtiva**

O sistema proposto neste trabalho é destinado para empresas que possuam produção em série total ou parcial na fabricação de seus produtos, tendo a função de auxiliar a gerência ou responsável pela engenharia de produção da empresa, na determinação da capacidade produtiva de determinada linha de produção.

A elaboração do projeto do sistema foi baseado no conhecimento dos autores deste trabalho juntamente com os dados obtidos no levantamento de requisitos.

A seguir será apresentada uma breve descrição do sistema atual da empresa em estudo, suas dificuldades e necessidades de informatização.

#### 4.1.1 Descrição do Sistema Atual

Este sistema visa possibilitar aos gestores de produção, por meio da determinação dos tempos padrões por operação, baseada na metodologia da cronoanálise industrial, calcular os tempos de ciclos por operações de um processo em linhas de produção em série. Além disso, possibilitar a elaboração de controles que, a partir de uma análise do controlador, irão sugerir, uma capacidade produtiva muito próxima da real, de forma a resultar em um máximo aproveitamento da mão-de-obra e em um nível mínimo de ociosidade da mesma. A capacidade produtiva pode ser o nível máximo de atividade de valor adicionado que pode ser conseguido, em condições normais de operação e por um determinado período de tempo.

Sendo assim, visando obter tais resultados, atualmente a empresa se utiliza de uma planilha eletrônica, assim como o lançamento dos dados para análise, conforme descritos abaixo.

No método atual, a determinação do tempo padrão envolve várias etapas, onde, o processo entre a criação do arquivo até a conclusão do estudo, exige um nível de conhecimento bastante avançado do operador, tanto em relação à metodologia, quanto acerca da tecnologia aplicada. O conhecimento do operador é requerido principalmente pelo fato de haver operações de inclusão, exclusão e formatação de campos conforme descrito em seguida.

Em um primeiro momento, faz-se a criação de um arquivo específico para cada processo de fabricação de um único produto, onde o mesmo é arquivado em uma pasta do disco local na máquina do operador. Nesta etapa existe o risco de perda de arquivos, deleção acidental, incompatibilidade de tecnologia, ou ainda, de o arquivo ser corrompido por inúmeros fatores. A realização de *backups*, para eventual recuperação de arquivos, pode acarretar na recuperação de um arquivo desatualizado que pode comprometer o processo de tomada de decisão.

Em seguida, são lançados dados referentes à identificação do produto em estudo, assim como uma relação de matéria-prima utilizada, equipamentos, operações e dispositivos necessários para a fabricação do produto.

Então, são coletados e registrados os dados referentes ao processo de produção em questão, período, duração, setor e o responsável.

O cálculo para a determinação do nível de confiança, erro relativo e determinação de número de amostras, é feito pelo operador na aplicação das fórmulas específicas em processo externo à planilha eletrônica, onde o cálculo é feito manualmente em calculadora e posteriormente lançados na planilha ou em recurso da própria planilha eletrônica.

Esta etapa exige, do operador, conhecimento avançado, no processo do cálculo estatístico, para levantamento dos fatores diretamente ligados à determinação do tempo padrão de cada operação do processo.

No processo de inclusão de linhas na planilha eletrônicas, necessárias para o lançamento de dados de cada operação, o código referente à seqüência da operação é lançado manualmente, pois a planilha não reordena automaticamente, além disso, a descrição da operação, que antecede o lançamento das amostras dos tempos, possui campo limitado, o que prejudica o entendimento do início e fim da operação. Só então, é informada a quantidade de peças resultantes da execução da operação.

A quantidade de campos para lançamento dos dados é relativamente fixa, são 20 campos onde a alteração na quantidade resulta em uma nova formatação que prejudica o padrão de exibição, o que pode confundir o operador e induzir ao erro na leitura e análise dos resultados.

Em seguida são informados dados referentes ao ritmo do operador, número de homens participantes da operação, quantidade de operações simultâneas e grupo de máquinas utilizado.

O lançamento dos coeficientes de correção do tempo padrão, como necessidades pessoais, atrasos imprevisíveis e fadiga, além de serem calculados pelo operador em ambiente externo à planilha eletrônica, se misturam à área de cálculos possibilitando o esquecimento do lançamento destes, assim como o lançamento do número de peças resultantes desta operação.

Por conseguinte, a análise da capacidade produtiva e levantamento de gargalos são feitos de maneira visual no próprio ambiente de lançamento da planilha eletrônica, que não possui geração de relatórios dinâmicos.

Desta forma, considerando o número de variáveis e o número de cálculos necessários a todo esse processo, pode-se perceber que trata-se de um sistema bastante suscetível a erros, sendo feito da forma descrita acima, fazendo uso de

uma planilha eletrônica de cálculo e envolvendo tantos procedimentos manuais. Outra desvantagem deste tipo de sistema é a impossibilidade de emissão de relatórios sistematizados dos dados de produção e por período, os quais possam auxiliar efetivamente a tomada de decisão por parte dos gestores.

## 4.2 Levantamento de Requisitos e Análise dos Dados

Durante a fase de levantamento dos requisitos do sistema para determinação da capacidade produtiva, foram definidos os requisitos funcionais e os requisitos não funcionais - de interface, de hardware e de software – conforme mostram as Tabelas 04 e 05 abaixo.

**Tabela 04 - Requisitos Funcionais**

<b>Identificação</b>	<b>Descrição</b>	<b>Importância</b>
<b>[R1] Cadastrar Produtos</b>	O sistema deve permitir que sejam cadastrados produtos, mediante inclusão de seus dados. O sistema deve permitir que os produtos sejam alterados e excluídos caso necessário.	Essencial
<b>[R2] Cadastrar Materiais</b>	O sistema deve permitir que sejam cadastrados os materiais que são utilizados na fabricação dos produtos. O sistema deve permitir que dados de materiais sejam alterados e excluídos caso não sejam mais utilizados.	Essencial
<b>[R3] Cadastrar Ferramentas</b>	O sistema deve permitir que sejam cadastradas ferramentas, mediante inclusão de seus dados. O sistema deve permitir que os dados das ferramentas sejam alterados e excluídos, caso uma ferramenta que não seja mais utilizada.	Essencial
<b>[R4] Cadastrar Dispositivos</b>	O sistema deve permitir que sejam cadastrados os dispositivos que são utilizados na empresa para a realização dos processos. O sistema deve permitir que os dados dos dispositivos sejam alterados e excluídos os	Essencial

	dispositivos que não são mais utilizados.	
<b>[R5] Cadastrar Operações</b>	<p>O sistema deve permitir que sejam cadastradas as operações envolvidas na produção dos produtos, cadastrando suas respectivas informações</p> <p>O sistema deve permitir que sejam alterados os dados de operações e excluídas operações caso seja necessário.</p>	Essencial
<b>[R7] Cadastrar Amostras de Tempo</b>	O sistema deve permitir que sejam cadastradas as amostras de tempo cronometradas, alteradas quando houver mudança nos processos, e excluídas quando não forem mais utilizadas.	Essencial
<b>[R8] Cadastrar Cronoanalistas</b>	O sistema deve permitir que sejam cadastrados os cronoanalistas que irão realizar as coletas das amostras de tempo.	Essencial
<b>[R9] Cadastrar Máquinas</b>	<p>O sistema deve permitir que sejam cadastradas as máquinas que são utilizadas na produção dos produtos.</p> <p>O sistema deve permitir que as máquinas cadastradas sejam alteradas quando necessário e excluídas do sistema quando houver essa necessidade.</p>	Essencial
<b>[R10] Calcular o tempo padrão por operação</b>	O sistema deve calcular o tempo padrão por operação, com base nas fórmulas de cronoanálise industrial.	Essencial
<b>[R11] Gerar relatório de gargalo de produção dos produtos</b>	O sistema deve gerar um relatório contendo o gargalo de produção de cada produto.	Essencial
<b>[R12] Consultar capacidade de produção dos produtos</b>	O sistema deve permitir que sejam feitas consultas da capacidade de produção dos produtos, informando se há capacidade ou não.	Essencial

Fonte: Elaborado pelo autor.

Tabela 05 - Requisitos Não-Funcionais

<b>Requisitos do Produto</b>		
<b>Identificação</b>	<b>Descrição</b>	<b>Importância</b>
<b>[R1] Confiabilidade</b>	O sistema deve possuir mecanismos que garantam que o usuário não perca informações.	Essencial



	<p>O sistema deve oferecer recursos que possibilitem que o usuário recupere a quantidade de peças programadas caso alguma anomalia ocorra, como erro de execução do aplicativo, queda de energia, etc.</p>	
	O sistema deve oferecer uma ferramenta de backup.	
<b>[R2] Desempenho</b>	O tempo de resposta para gerar os relatórios deve ser menor que 10 segundos.	Desejável
<b>[R3] Reusabilidade</b>	O sistema deve permitir que seus componentes possam ser utilizados de forma independente.	Desejável
<b>[R4] Segurança</b>	O sistema deve manter um <i>log</i> (registro) contendo todas as operações dos usuários.	Desejável
<b>[R5] Usabilidade</b>	O sistema deve possibilitar o uso de pelo menos 60% das funcionalidades da ferramenta após treinamento de no máximo 4 horas.	Essencial
	O sistema deverá oferecer ferramentas de apoio aos usuários "novatos", como tutoriais, ajuda, exemplos, imagens, etc. para que o usuário que esteja aprendendo a utilizar o sistema aprenda sem grandes dificuldades.	Desejável
<b>Requisitos de Interface</b>		
<b>Identificação</b>	<b>Descrição</b>	<b>Importância</b>
<b>[R1] Cadastro e edição</b>	O sistema deverá oferecer uma interface para o cadastramento e alteração dos requisitos em no máximo duas janelas.	Essencial
<b>[R2] Preenchimento da capacidade desejada</b>	O sistema deverá oferecer uma interface para preenchimento da capacidade e condições de tempo desejadas.	Essencial
<b>[R3] Consulta de produtos</b>	O sistema deverá oferecer uma interface para consulta dos produtos.	Essencial
<b>Requisitos de Hardware</b>		
<b>Identificação</b>	<b>Descrição</b>	<b>Importância</b>
<b>[R1] Cronômetro</b>	Hardware para fazer a coleta das amostras de tempo dos processos de produção.	Essencial

<b>[R2] Microcomputador</b>	O sistema precisará de uma máquina, com no mínimo 512 MB de RAM, com processador de 2,8 GHz ou superior, HD de no mínimo 80 GB.	Essencial
<b>Requisitos de Software</b>		
<b>[R1] Plataformas</b>	O sistema deverá ser executado nas plataformas operacionais Microsoft Windows 98, Windows XP, Vista e 7.	Essencial

Fonte: Elaborado pelo autor.

Após realizar a coleta dos requisitos do sistema, fez-se necessário organizá-los em grupos correlacionados. Foi então elaborado o diagrama de casos de uso do Sistema para controle da capacidade produtiva, ilustrado na Figura 11.

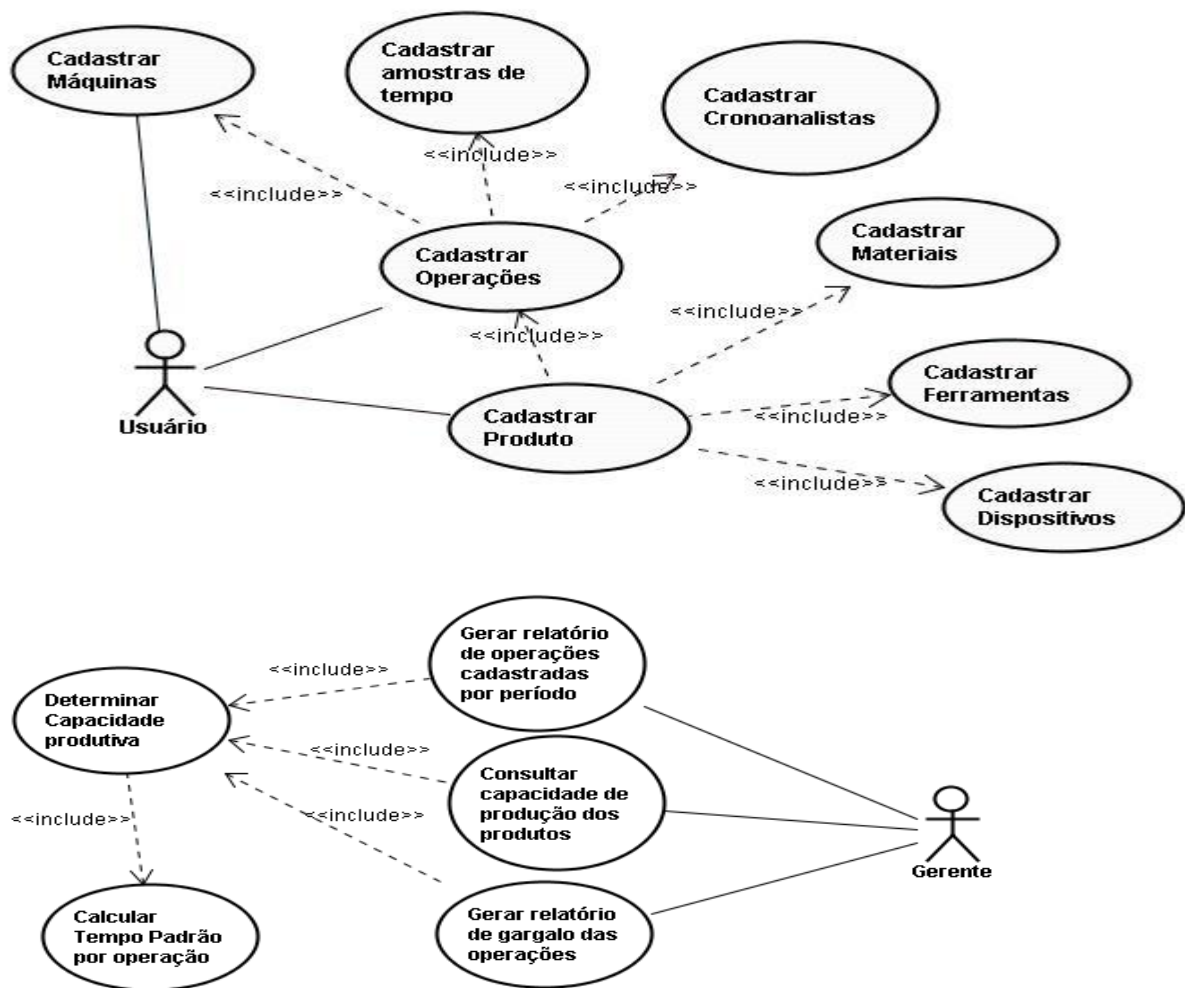


Figura 11 - Diagrama de Casos de Uso do Sistema de determinação da capacidade produtiva

Fonte: Elaborado pelo autor.

Posteriormente, para um melhor entendimento, foram feitas as descrições dos principais casos de uso do sistema, apresentadas a seguir nas tabelas de 06 à 10.

**Tabela 06 - Descrição de Caso de Uso – Cadastrar produtos**

<b>Caso de Uso:</b>	Cadastrar produtos
<b>Atores:</b>	Usuário
<b>Objetivo:</b>	Cadastrar os produtos a serem produzidos
<b>Tipo:</b>	Essencial
<b>Ativação:</b>	O caso de uso começa quando o usuário clica na guia de Cadastros.
<b>Fluxo Normal</b>	
<b>Fluxo típico de eventos</b>	
<ol style="list-style-type: none"> <li>1. Na tela de Cadastro de Produtos, o usuário cadastra o produto, informando o nome do produto, os processos, as ferramentas, os materiais e os dispositivos relacionados ao produto.</li> <li>2. O produto é incluído no banco de dados e o sistema emite a mensagem: “Produto Cadastrado com Sucesso”.</li> <li>3. O sistema volta para a tela de Cadastro de Produtos com os campos em branco para novo cadastro.</li> <li>4. O caso de uso termina.</li> </ol>	
<b>Fluxo Alternativo (Produto já cadastrado)</b>	
Linha 2: Se o produto já constar no banco de dados, ele não será incluído.	

Fonte: Elaborado pelo autor.

**Tabela 07 - Descrição de Caso de Uso – Calcular o tempo padrão**

<b>Caso de Uso:</b>	Calcular tempo padrão por processo
<b>Atores:</b>	Gerente
<b>Objetivo:</b>	Calcular o tempo padrão de cada processo
<b>Tipo:</b>	Essencial
<b>Ativação:</b>	Clique no botão gerar relatório.
<b>Fluxo Normal</b>	
<b>Fluxo típico de eventos</b>	
<ol style="list-style-type: none"> <li>1. O gerente escolhe qual o produto e o tipo de período para o qual quer o relatório (horas, dias, semanas, mês)</li> <li>2. O sistema realiza o cálculo de tempo normal (<math>T_n</math>) para cada processo através da fórmula: <ol style="list-style-type: none"> <li>a) <math>T_n = ((T_m/A)/e)+A_t</math>, onde:  <math>T_m</math> = Tempo médio (soma dos tempos/<math>n^o</math> de tempos);  <math>A</math> = Amplitude dos tempos;  <math>e</math> = erro relativo  <math>A_t</math> = Acréscimo de tempo relativo ao <math>e</math>;</li> <li>b) O acréscimo de tempo ocorre somente se o valor de <math>e</math> for maior que 2,5, e é calculado pela seguinte fórmula:  <math>(e / 100 + 1) * T_m / 1,05 - T_m</math></li> </ol> </li> </ol>	

<p>E caso o resultado der negativo, multiplica-se por (-1) para obter o módulo, pois não existe tempo negativo.</p> <p>c) Com o tempo normal, é calculado o tempo padrão (Tp):  <math display="block">Tp = Tn * (NP + AI + F) / 100 + Tn</math>, onde:  NP = percentual de Necessidades Pessoais;  AI = percentual de Atrasos Imprevisíveis;  F = percentual de Fadiga.</p> <p>d) Após o cálculo do tempo padrão por processo, são somados os tempos padrão de todos os processos do produto, gerando assim, o tempo padrão de produção unitário do produto.</p> <p>e) Após encontrar o tempo padrão unitário, divide-se o tempo requerido pelo relatório pelo tempo unitário, gerando assim, a capacidade produtiva do produto selecionado no período de tempo estipulado pelo gerente.</p> <p>3. O relatório é mostrado na tela.</p> <p>4. O caso de uso termina.</p>
<b>Fluxo Alternativo (Falta de dados)</b>
<p>Linha 2: Se algum dos dados necessários para a realização dos cálculos não existir, o sistema emitirá uma mensagem informando qual dado está faltando.</p>

Fonte: Elaborado pelo autor.

**Tabela 08 - Descrição de Caso de Uso – Gerar Relatório de Operações por Período**

<b>Caso de Uso:</b>	Gerar Relatório de Operações por Período
<b>Atores:</b>	Gerente
<b>Objetivo:</b>	Gerar um relatório com as operações cadastradas em um determinado período.
<b>Tipo:</b>	Essencial
<b>Ativação:</b>	O caso de uso começa quando o Gerente clica na guia de Relatórios.
<b>Fluxo Normal</b>	
<b>Fluxo típico de eventos</b>	
<ol style="list-style-type: none"> <li>1. O Gerente clica na guia de Relatórios.</li> <li>2. Em seguida, no item Operações por Período.</li> <li>3. Na tela de opções do relatório, o gerente seleciona o intervalo de datas desejado.</li> <li>4. O usuário clica em Gerar Relatório.</li> <li>5. Aparece a tela de opção de visualização, onde o gerente escolhe se quer visualizar o relatório na tela, ou se deseja imprimir o relatório.</li> <li>6. O sistema emite o relatório com as seguintes informações: <ol style="list-style-type: none"> <li>a) Nome da operação.</li> <li>b) Tempo padrão da operação.</li> <li>c) Data de cadastro da operação.</li> </ol> </li> <li>7. O caso de uso termina.</li> </ol>	

Fonte: Elaborado pelo autor.

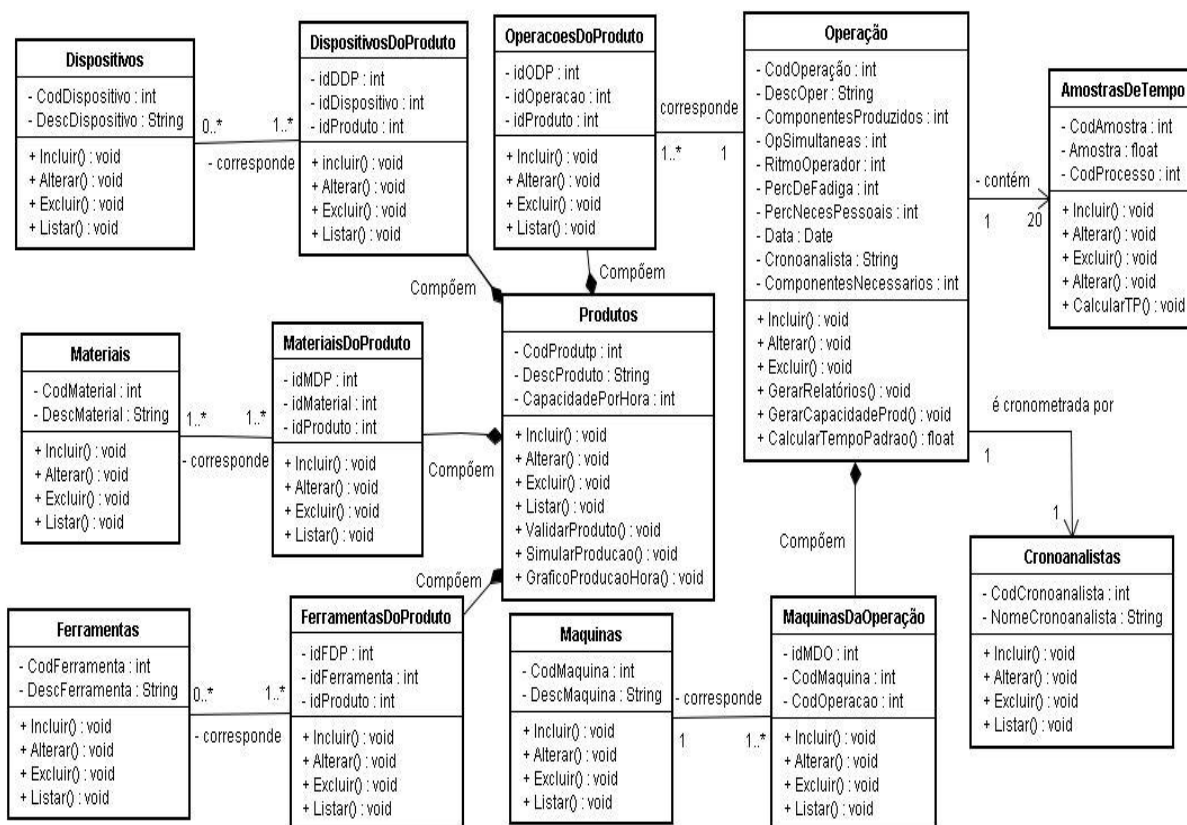
**Tabela 09 - Descrição de Caso de Uso – Consultar capacidade de produção dos produtos**

<b>Caso de Uso:</b>	Consultar capacidade de produção dos produtos
---------------------	---

<b>Atores:</b>	Gerente
<b>Objetivo:</b>	Consultar a capacidade de produção de um produto
<b>Tipo:</b>	Essencial
<b>Ativação:</b>	O caso de uso começa quando o Gerente clica na guia de Consultas.
<b>Fluxo Normal</b>	
<b>Fluxo típico de eventos</b>	
<ol style="list-style-type: none"> <li>1. O Gerente clica na guia de Consultas.</li> <li>2. Em seguida, no item Capacidade por Produto.</li> <li>3. Na tela de consulta, o gerente informa qual produto que devem ser consultado e a quantidade desejada desse produto.</li> <li>4. O usuário clica em Verificar.</li> <li>5. O sistema faz a consulta no banco de dados fazendo os cálculos necessários.</li> <li>6. O sistema retorna com as seguintes informações: <ol style="list-style-type: none"> <li>a) Nome do produto.</li> <li>b) Nome das máquinas utilizadas.</li> <li>c) Tempo necessário para cada máquina produzir a quantidade desejada.</li> <li>d) Tempo disponível de cada máquina.</li> <li>e) Se há ou não capacidade.</li> </ol> </li> <li>7. O caso de uso termina.</li> </ol>	

Fonte: Elaborado pelo autor.

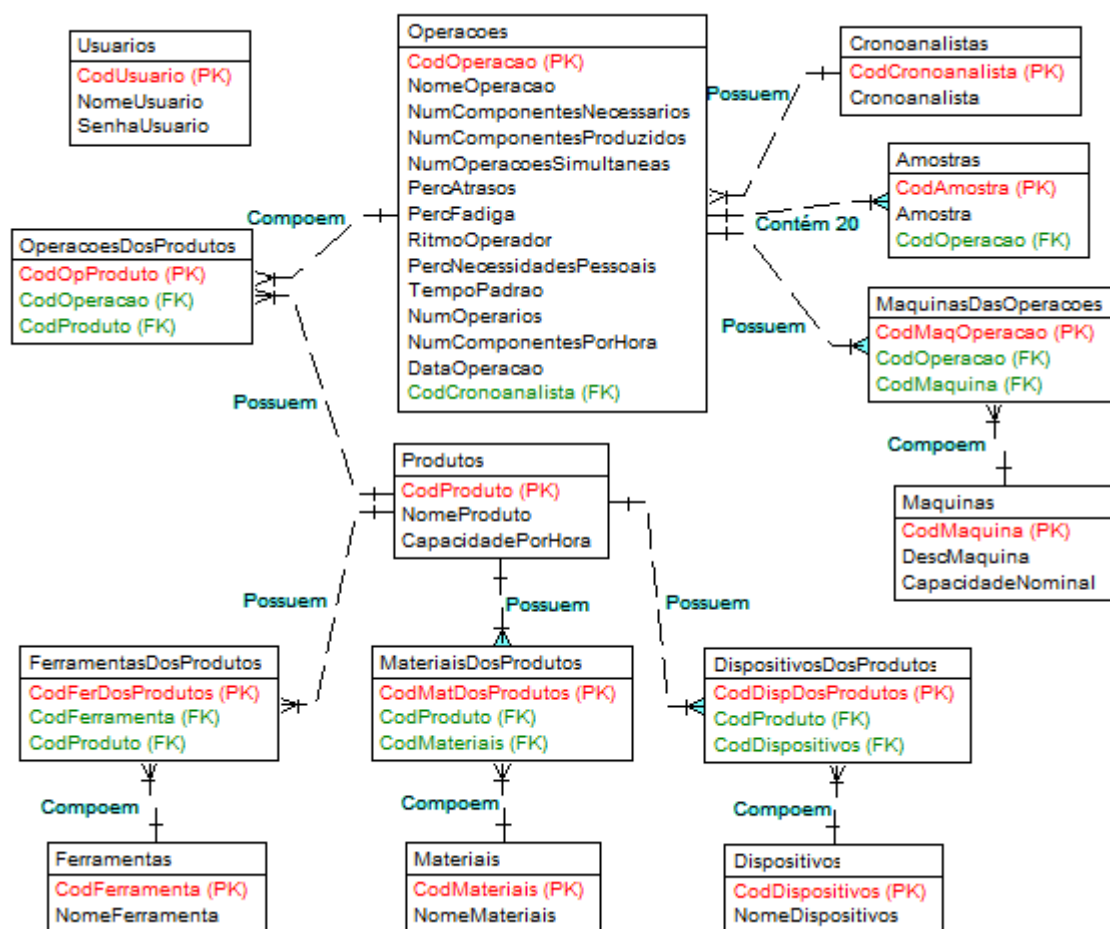
Com base em todos os dados disponíveis, foi possível elaborar o diagrama de classes, apresentado na Figura 12, e o diagrama de entidade-relacionamento apresentado na Figura 13 a seguir:



**Figura 12 - Diagrama de Classes**

Fonte: Elaborado pelo autor.

[1.1]



**Figura 13 - Diagrama de Entidade-Relacionamento**

Fonte: Elaborado pelo autor.

Para auxiliar na compreensão do diagrama de classes, a Tabela 11 foi elaborada com explicações sobre alguns dos termos técnicos utilizados na sua composição.

**Tabela 10 - Termos técnicos do diagrama de classes**

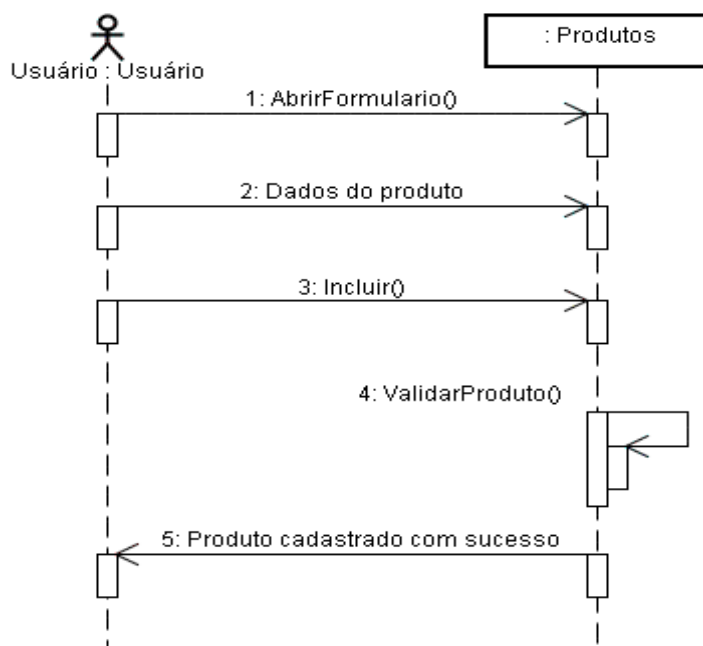
TERMO	EXPLICAÇÃO
Tempo Padrão (TP)	É o tempo necessário para realizar uma operação dentro de um método estabelecido,
Ritmo do Operário	O ritmo desempenhado pelo operário durante a cronometragem

	de uma operação. É expresso em porcentagem, sendo 100% o ritmo normal do operador.
Percentual de Fadiga	Percentual calculado conforme mostrado na tabela 03
Percentual de Necessidades Especiais	Percentual calculado conforme mostrado na tabela 03
Dispositivos	Exemplo de dispositivos: Bancada de solda, bancada de corte, bancada de dobra.
Materiais	Exemplo de materiais: Arames, tubos de aço, plástico.
Ferramentas	Exemplo de ferramentas: Chave de fenda, chave de boca, martelo.
Máquinas	Máquinas utilizadas. Exemplo: chapa de corte, politriz.
Operações Simultâneas	Número de operações que ocorrem ao mesmo tempo em uma operação.
Número de arames por peça	Cada operação é realizada para um arame de cada peça, ou seja, se houver dois arames em uma peça, o tempo de produção será dobrado.

Fonte: Elaborado pelo autor.

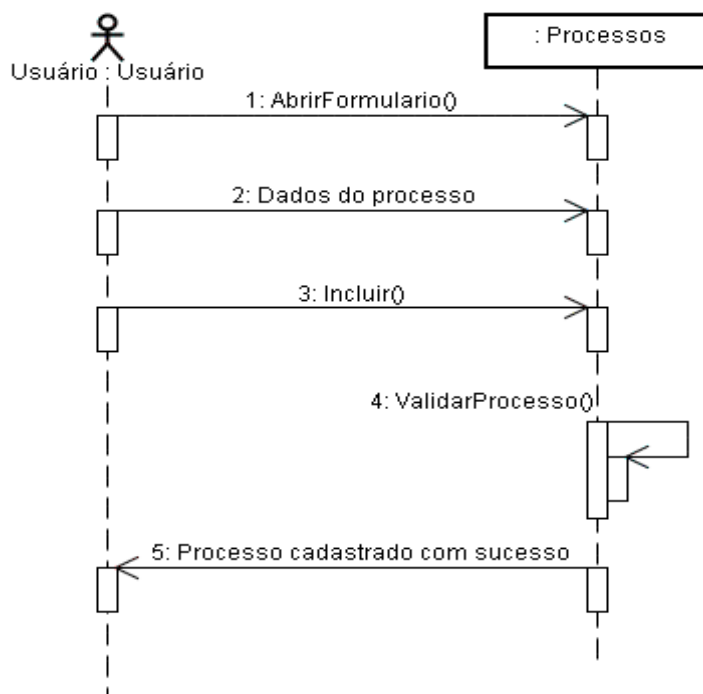
Após desenvolver o diagrama de classes, foram então desenvolvidos os diagramas de seqüência que representam os principais casos de uso mostrados anteriormente na figura 11. Os diagramas de seqüência são mostrados a seguir nas Figuras 14 a 18.





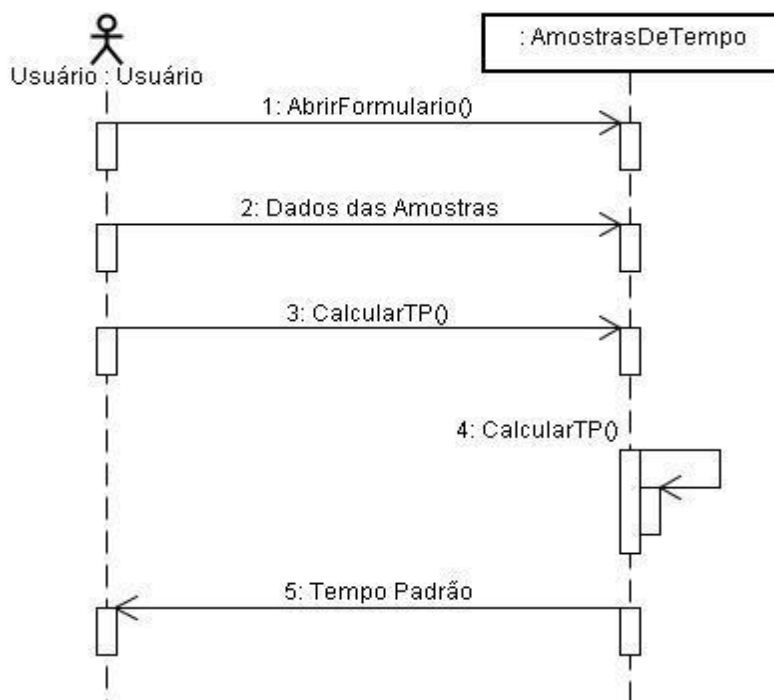
**Figura 14 - Diagrama de Seqüência – Cadastrar Produtos**

Fonte: Elaborado pelo autor.



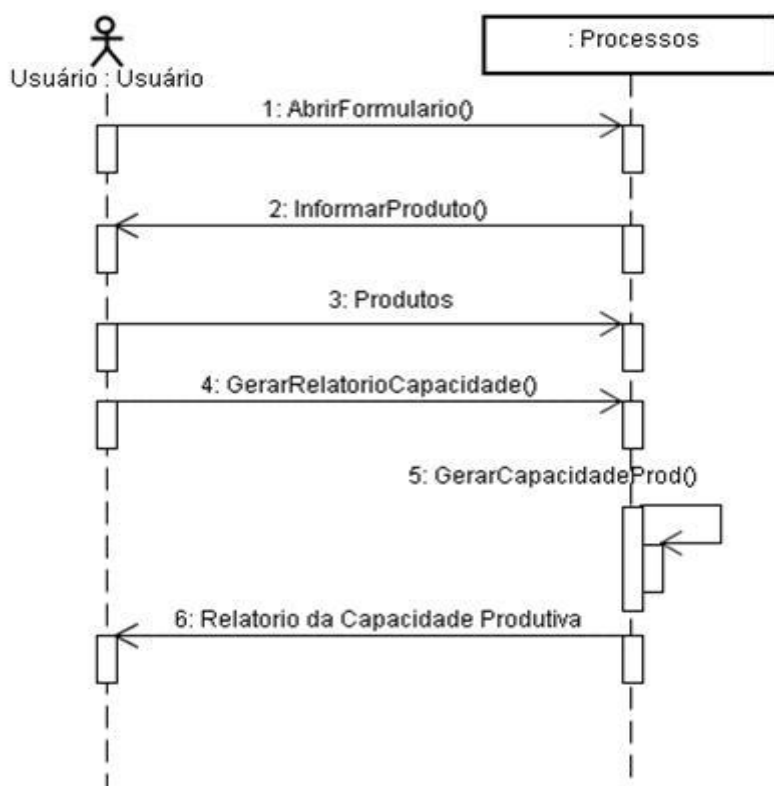
**Figura 15 - Diagrama de Seqüência – Cadastrar Processos**

Fonte: Elaborado pelo autor.



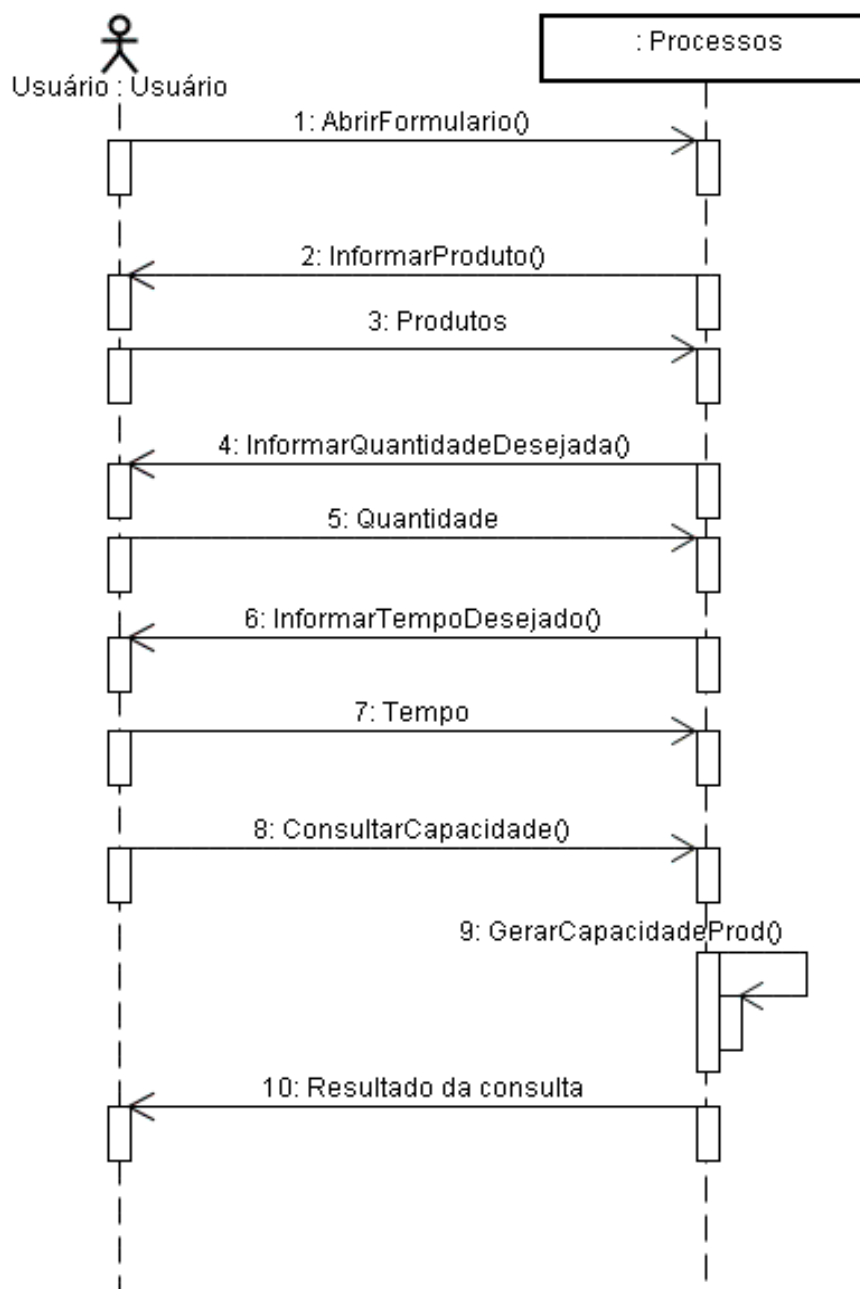
**Figura 16 - Diagrama de Seqüência – Calcular Tempo Padrão**

Fonte: Elaborado pelo autor.



**Figura 17 - Diagrama de Seqüência – Gerar Relatório de Capacidade Produtiva**

Fonte: Elaborado pelo autor.



**Figura 18 - Diagrama de Seqüência – Consultar Capacidade Produtiva**

Fonte: Elaborado pelo autor.

## 5 IMPLEMENTAÇÃO DO SISTEMA CAPACIDADE PRODUTIVA

Nesta sessão serão relatadas as funcionalidades do sistema proposto, o qual visa à informatização da determinação da Capacidade Produtiva instalada, baseando-se na cronoanálise industrial, com foco para pequenas empresas. Sendo assim, a seguir será apresentado como acessar e fazer uso das funcionalidades existentes no sistema, sendo para isso, apresentadas as telas relativas a cada funcionalidade e o código-fonte respectivo.

O acesso ao sistema é feito por meio da tela *login*, de acordo com os dados previamente cadastrados no banco de dados. A Figura 19 mostra o formulário de *login* do sistema. Nela existe um botão *Entrar* para confirmar os dados inseridos e um botão *Cancelar* para cancelar a entrada no sistema. Possui ainda dois campos para inserção dos dados de usuário e senha, que correspondem a dados armazenados na base de dados, a partir dos quais o usuário será validado ou não, quando da ativação do botão *Entrar*. Caso não haja coerência entre os dados inseridos e os que constam no banco, o sistema não permitirá o acesso ao sistema.



A imagem mostra uma janela de software com o título "Verificação de Usuário". O conteúdo principal da janela é o formulário de login para o sistema "CAPACIDADE PRODUTIVA", versão 1.0. À esquerda do formulário há uma imagem decorativa de engrenagens cinzas. O formulário contém dois campos de entrada: "Usuário" e "Senha". Abaixo dos campos, há dois botões: "Entrar" e "Cancelar".

**Figura 19 - Formulário de Login do sistema**

Fonte: Elaborado pelo autor.

Depois de realizado o *login*, é possível acessar as funções do sistema com base nos menus que se encontram no topo da Tela Principal do Sistema, mostrado na Figura 20. É por meio dessa tela que se pode selecionar a funcionalidade desejada: cadastros, consultas, alterações ou exclusões de dados, geração de relatórios ou ainda realizar uma simulação de produção.



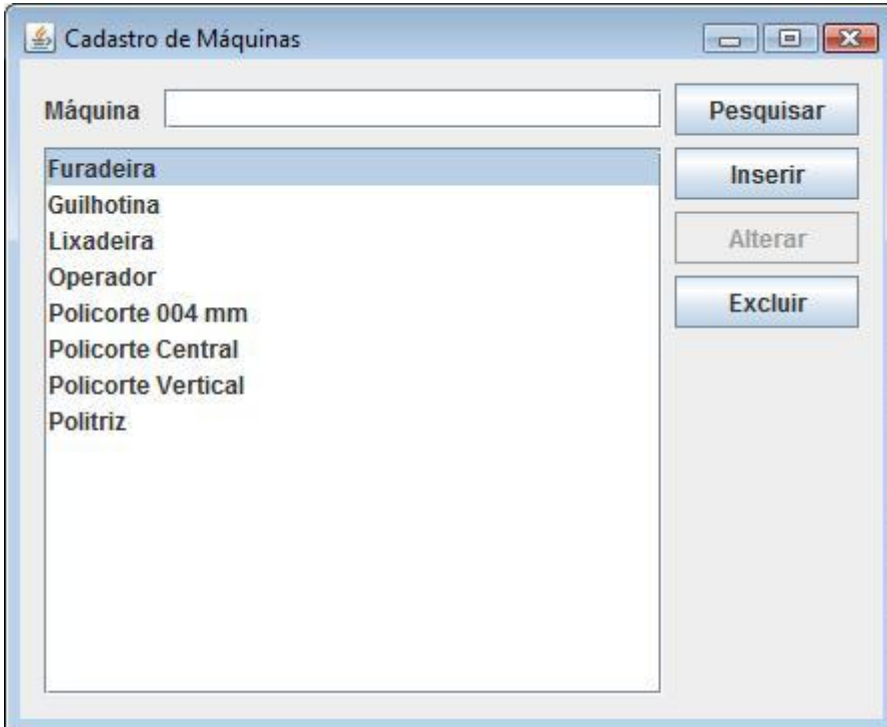
**Figura 20 - Tela Principal do Sistema**

Fonte: Elaborado pelo autor.

A partir do menu Cadastros podem ser acessados os formulários cadastro de materiais, máquinas, ferramentas e dispositivos, que seguem o mesmo padrão e serão a partir de agora denominados “formulários de cadastro padrão”, pois seus itens são tratados de forma semelhante, e os formulários para cadastro de operações e de produtos, que são singulares por sua apresentação ser um pouco mais complexa e diferente das anteriores. Ambas, além de listarem os itens já cadastrados, permitem sua inclusão, alteração e exclusão do banco de dados.

A Figura 21 apresenta o Cadastro de Máquinas, como um exemplo de formulário de cadastro padrão presentes no sistema. O novo formulário que se abre possui uma listagem das máquinas previamente cadastradas, uma barra de navegação para visualizar todos os dados cadastrados, e também botões padrão no canto superior direito do formulário, sendo estes:

- *Pesquisar*: para pesquisar uma nova máquina;
- *Inserir*: para inserir uma máquina no banco de dados;
- *Alterar*: para alterar uma máquina no banco de dados; e
- *Excluir*: para remover uma máquina do banco de dados.



A imagem mostra uma janela de software intitulada "Cadastro de Máquinas". No topo, há uma barra de título com ícones de minimizar, maximizar e fechar. Abaixo, há um campo de texto rotulado "Máquina" e um botão "Pesquisar". À esquerda, há uma lista de itens: "Furadeira", "Guilhotina", "Lixadeira", "Operador", "Policorte 004 mm", "Policorte Central", "Policorte Vertical" e "Politriz". À direita da lista, há três botões: "Inserir", "Alterar" e "Excluir".

**Figura 21 - Tela de Cadastro de Máquinas**

Fonte: Elaborado pelo autor.

Ao acessar os formulários de cadastro simples, o sistema cria uma lista vazia, faz então uma busca no banco de dados, a qual retorna todos os registros relativos ao formulário, preenchendo a lista em seguida, conforme código apresentado no Quadro 1.

```

private void preencheListDeMaquinas() {
    listaModelo.clear();
    int tamanholistaRegMaquinas = listaRegMaquinas.size();
    Maquinas maquina = new Maquinas();
    String maq;
    try {
        iniciarConexaoBanco();
        for( int i=0; i<tamanholistaRegMaquinas; i++){
            maquina = (Maquinas) listaRegMaquinas.get(i);
            maq = maquina.getNomeMaquina();
            listaModelo.add(i, maq);
        }
        listDeMaquinas.setModel(listaModelo);
        listDeMaquinas.setSelectionMode( ListSelectionMode.SINGLE_SELECTION );
        listDeMaquinas.setSelectedIndex(0);
        listDeMaquinas.addListSelectionListener(this);

        txtNomeMaquina.setText("");
        txtNomeMaquina.requestFocus();
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Erro: " +e.getMessage(), "Aviso", 1)
    }
}

```

#### Quadro 1 - Método Listagem de Máquinas

Fonte: Elaborado pelo autor.

Ao clicar no botão *Inserir*, é verificado se o campo Máquina foi preenchido. Em seguida, se a nova máquina não consta no banco de dados, ela é inserida e incluída na lista de máquinas conforme mostra o Quadro 2. Ao clicar duas vezes em uma das máquinas da lista, a mesma é carregada no campo Máquinas e habilita o botão *Alterar*, que ao ser pressionado pede para confirmar a alteração, e, em caso positivo, realiza a alteração. Ao clicar no botão *Excluir*, é solicitada a confirmação da exclusão, e, novamente, em caso positivo, exclui a máquina do banco de dados e da lista de máquinas. O botão *Pesquisar* auxilia o usuário a encontrar uma determinada máquina, caso não esteja conseguindo visualizá-la na lista de máquinas. Ao ser clicado, uma nova janela aparece para que o usuário digite a máquina que está

procurando. Se ela existir no banco de dados, será selecionada na lista de máquinas e carregada no campo Máquinas.

```
private void inserirMaquina() {
    String novaMaquina = txtNomeMaquina.getText();
    Maquinas maquina = new Maquinas();

    if (novaMaquina.equals("")){
        JOptionPane.showMessageDialog(null, "Preencha o campo Nome", "Aviso", 1);
        txtNomeMaquina.requestFocus();
        return;
    }

    for (int i=0; i<listaModelo.size(); i++){
        if ( novaMaquina.equalsIgnoreCase( (String) listaModelo.elementAt(i)) ){
            JOptionPane.showMessageDialog(null, "Já existe uma máquina com este nome.",
"Aviso", 1);
            txtNomeMaquina.requestFocus();
            return;
        }
    }

    maquina.setNomeMaquina( (String) txtNomeMaquina.getText() );

    em.getTransaction().begin();
    try {
        em.persist(maquina);

        em.getTransaction().commit();

        iniciarConexaoBanco();

        preencheListDeMaquinas();

    } catch (Exception e) {
        javax.swing.JOptionPane.showMessageDialog( this,
"Erro na Inclusão: " + e.getMessage() );
        em.getTransaction().rollback();
    } finally {
        em.close();
    }
}
}
```

**Quadro 2 - Código de inserção de dados no Banco de dados.**

Fonte: Elaborado pelo autor.

Os formulários de cadastro de Materiais, Dispositivos e Ferramentas, seguem o mesmo modelo de implementação do formulário de cadastro de Máquinas.



**Figura 22 - Formulário de Cadastro de Produtos**

Fonte: Elaborado pelo autor.

O formulário para cadastro de Produtos, apresentado na Figura 22, trata dos serviços de inclusão, alteração e exclusão de produtos, juntamente com as informações relacionadas a eles. Nele há um campo para o nome do produto, além de quatro grupos distintos de elementos que compõem um produto, sendo eles: Materiais, Ferramentas, Dispositivos e Operações. Os três primeiros são informados nos formulários de cadastro padrão, como mostrado na Figura 21.

Em cada um dos quatro grupos encontra-se um *JComboBox* que contém as informações previamente cadastradas de itens do respectivo grupo, um botão *Novo* que direciona para o respectivo formulário de cadastro caso o item desejado não tenha sido previamente cadastrado, uma lista de itens do grupo, um botão *Adicionar* que adiciona o item selecionado no *JComboBox* na lista, e um botão *Remover*, para remover um item da lista se for o caso.

No canto superior do formulário ainda encontra-se um botão *Inserir*, que, após realizar as verificações de preenchimento de dados, informando ao usuário caso

haja alguma irregularidade, insere os dados do produto no banco de dados. Existe ainda um botão *Cancelar*, caso o usuário decida cancelar o cadastro corrente.

No canto inferior esquerdo da tela encontra-se uma lista com os nomes dos produtos cadastrados até então. Ao lado da lista há um botão *Carregar* que quando acionado busca os dados do produto selecionado no banco de dados, e preenche os campos com as respectivas informações, conforme mostra o Quadro 3. Há um botão *Alterar* que fica inicialmente desabilitado, sendo habilitado somente após o botão *Carregar* ser acionado, permitindo que todas as informações sobre o produto estejam disponíveis para alteração. Após clicar no botão *Alterar* e confirmar a intenção de alterar os dados do produto carregado é realizado a alteração do mesmo. Há ainda um botão *Excluir*, que realiza a exclusão do produto selecionado na lista, assim como os demais dados a ele vinculados.

```
private void carregarProduto() {
    EntityManagerFactory emf = Persistence.createEntityManagerFactory("CapProdPU");
    EntityManager em = emf.createEntityManager();

    if (listProdutos.getSelectedIndex() == -1){
        javax.swing.JOptionPane.showMessageDialog( this,
            "Não há operacao a ser carregada." );
        return;
    }
    String produtoDoList = (String) listProdutos.getSelectedValue();
    Query query = em.createQuery(
        "SELECT p FROM Produtos p WHERE p.nomeProduto = :nomeProduto" );
    query.setParameter("nomeProduto", produtoDoList );
    try {
        produtoCarregado = (Produtos) query.getSingleResult();

    } catch( javax.persistence.NoResultException e ) {
        javax.swing.JOptionPane.showMessageDialog( this, e.getMessage() );
    }

    limpaCampos();
    txtNomeProduto.setText( produtoCarregado.getNomeProduto() );

    // Preenche o List de MATERIAIS

    Materiais mat = new Materiais();
    String materialProList;
    try {
        materialListModel.clear();
        iniciarConexaoBanco();
        query = em.createQuery(
            "SELECT m.idMaterial FROM Materiaisdosprodutos m WHERE m.idDoProduto =
:idDoProduto");
```

```

query.setParameter("idDoProduto", produtoCarregado);
Vector resultado = (Vector) query.getResultList();

for (int i=0; i<resultado.size(); i++){
    mat = (Materiais) resultado.get(i);
    materialProList = mat.getNomeMaterial();
    materialListModel.add(i, materialProList);
}
listMateriais.setModel(materialListModel);

} catch ( javax.persistence.NoResultException e ) {
    javax.swing.JOptionPane.showMessageDialog( this, e.getMessage() );
} catch (Exception e) {
    e.printStackTrace();
}

// O código para preencher os demais Lists segue a mesma lógica, apenas mudando
// os nomes das variáveis de acordo com o grupo específico

```

### Quadro 3 - Código para buscar as informações dos produtos no Banco de dados.

Fonte: Elaborado pelo autor.

**Operação**

Nome:

1  N° de componentes produzidos

1  N° de componentes necessários

1  N° de Operações Simultaneas

1  N° de Operá...

100  Ritmo do Operador (%)

0  Fadiga (%)

0  Necessidades Pessoais (%)

**Máquina(s)**

Policorte Central

Guilhotina

**Dados das amostras**

Amostras da Operação (Segundos)

3.13	3.15	2.89	2.72	2.53
2.54	2.72	2.74	2.67	2.83
3.23	3.41	3.43	2.67	2.58
2.83	2.9	2.54	3.03	2.6

Data:  dd/MM/aaaa

Cronoanalista:

**Tempo padrão: 3.05 (s)**

**Operações Anteriores**

Corte Engate

Corte Trava Central

Corte Trava Vertical

Dobra e Engate

**Figura 23 - Formulário de Cadastro de Operações**

Fonte: Elaborado pelo autor.

O formulário de Cadastro de Operações, apresentado na Figura 23, é de vital importância para o funcionamento do sistema. Nele são informados os dados para a realização do cálculo do tempo padrão, que é a base para determinar a capacidade de uma determinada produção.

Neste formulário são cadastradas as diversas operações necessárias para produzir um determinado produto. No grupo “Operação”, localizado ao alto na esquerda, são informados: nome da operação, nº de componentes produzidos, nº de componentes necessários, nº de operações simultâneas, nº de operários, o ritmo do operador que está realizando a operação, o percentual de fadiga, que, assim como o percentual de necessidades pessoais, deve ser calculado em formulário próprio, que se abre ao clicar no botão *Calcular* ao lado do campo.

No canto inferior esquerdo encontra-se o grupo Máquinas, similar àqueles do formulário de cadastro de Produtos, porém este com informações sobre as máquinas a serem utilizadas na operação.

No grupo Dados das Amostras há uma tabela onde poderão ser digitadas até 20 amostras de tempo coletadas na realização da operação, um campo para digitar a data em que foram realizadas as coletas, e um *JComboBox* para escolher o cronoanalista que coletou as amostras.

Logo abaixo há um botão *Calcular Tempo Padrão* para que se possa calcular o tempo padrão, conforme mostra o Quadro 4, antes de inserir a operação no banco de dados, possibilitando verificar alguma informação incorreta.

No canto inferior direito há uma lista das operações previamente cadastradas com os botões *Carregar*, *Alterar* e *Excluir*, seguindo a mesma lógica da lista de produtos cadastrados do formulário de cadastro de produtos.

Por fim, há no canto superior direito há um botão *Inserir*, que, depois de verificar se todos os campos estão preenchidos de forma correta, insere a operação no banco de dados. Logo abaixo, há um botão *Cancelar*, que cancela o cadastro corrente, limpando os campos do formulário.

```
private void calculaTempoPadrao() {
    NumberFormat formatador = new DecimalFormat("0.00");
    float ritmo = 100;
    float necPessoais = 0;
    float fadiga = 0;
    float atrasos = 0;

    if (maquinaListModel.contains( "Operador" )){
        atrasos = 3;
    }else{
        atrasos = 5;
    }
}
```

```

ritmo = Float.parseFloat( txtRitmoOperador.getText() );
necPessoais = Float.parseFloat( txtNecPessoais.getText() );
fadiga = Float.parseFloat( txtFadiga.getText() );

if ( validaTabela() ){
    ritmo = ritmo/100;

    float media = 0, maior = 0;
    float menor = Float.parseFloat( poePonto( (String) tableAmostras.getValueAt(0, 0) ) );
    float amplitude, tempoNormal, erroRelativo, aux, valorAtual, acres;
    double constant = 0.0852, acrescimo = 0;
    String conversor = String.valueOf(constant);
    String sub;
    float constante = Float.parseFloat(conversor);

    try{
        for( int i=0; i<4; i++){
            for ( int j=0; j<5; j++){
                sub = String.valueOf(tableAmostras.getValueAt(i, j));
                sub = poePonto(sub);
                media = media + Float.parseFloat( sub );
                valorAtual = Float.parseFloat( poePonto( dadosTabela[i][j] ) );
                if ( valorAtual > maior){
                    maior = valorAtual;
                }
                if (valorAtual < menor){
                    menor = valorAtual;
                }
            }
        }
        media = Float.parseFloat( poePonto(formatador.format( media / 20 ) ) );
        amplitude = Float.parseFloat( poePonto(formatador.format( maior - menor ) ) );
        aux = Float.parseFloat( poePonto(formatador.format( amplitude / media ) ) );
        erroRelativo = Float.parseFloat( poePonto(formatador.format( aux / constante ) ) );

        if ( erroRelativo > 2.5 ){
            acrescimo = (erroRelativo / 100 + 1) * media / 1.05 - media;
            if (acrescimo < 0)
                acrescimo = acrescimo * (-1);
        }
        String s = String.valueOf(acrescimo);
        acres = Float.parseFloat(s);
        acres = Float.parseFloat( poePonto(formatador.format(acres) ) );
        tempoNormal = Float.parseFloat( poePonto(formatador.format( (media + acres) * ritmo ) ) );
    };

    TP = Float.parseFloat( poePonto( formatador.format(
        tempoNormal * ((necPessoais + atrasos + fadiga)/100) + tempoNormal ) ) );

    if( TP > 60 ){
        lblTempoPadrao.setText( "Tempo padrão: " + TP + " (min)" );
    }else{
        lblTempoPadrao.setText( "Tempo padrão: " + TP + " (s)" );
    }
}
catch(Exception e){
    JOptionPane.showMessageDialog(null,"Erro: " + e.getMessage(), "Aviso", 1);
}
}

```

```

}
}

```

#### Quadro 4 - Código para calcular o tempo padrão da operação.

Fonte: Elaborado pelo autor.

A seguir a Figura 24 apresenta o formulário para calcular o percentual de fadiga, invocado ao clicar sobre o botão *Calcular* ao lado do campo *Fadiga (%)* no formulário de operações.

**Cálculo da Fadiga**

**Níveis de Esforço**

Níveis de Esforço	Mental	Visual	Físico
Muito Leve	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Leve	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Médio	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pesado	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Muito Pesado	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Condições Ambientais**

Tipo de Ambiente	Seleção
A	<input type="checkbox"/>
B	<input type="checkbox"/>
C	<input type="checkbox"/>
D	<input type="checkbox"/>
E	<input type="checkbox"/>

Definições dos Tipos de Ambiente

A=0  
B=2  
C=4  
D=6  
E=8

**Monotonia do Trabalhador**

Duração da Operação (segundos)	Seleção
Até 30 s	<input type="checkbox"/>
De 31 à 60 s	<input type="checkbox"/>
De 61 à 90 s	<input type="checkbox"/>
Acima de 90 s	<input type="checkbox"/>

**Atualizar**    Percentual de fadiga: 0

Figura 24 - Formulário de Cálculo de Fadiga

Fonte: Elaborado pelo autor.

O formulário de Cálculo de Fadiga apresenta três tabelas, uma para cada área envolvida no cálculo de fadiga, sendo estas tabelas preenchidas com caixas de seleção, de forma que o usuário possa selecionar as opções da tabela que estejam de acordo com a realidade da empresa. A segunda tabela, que trata das condições ambientais, é a única que não se explica por si só, por isso foi colocado ao lado um componente *JTextArea* com as informações necessárias para que o usuário possa decidir qual opção da tabela que deve ser escolhida.

Há também um botão *Atualizar* para atualizar o percentual de fadiga antes do usuário fechar o formulário. Ao fechar o formulário, o valor resultante das opções marcadas pelo usuário é preenchido no campo *Fadiga* da tela de cadastro de operações, conforme o código apresentado no Quadro 5 abaixo.

```
private int calcularFadiga() {
    String[][] val = new String[5][4];
    String[][] val1 = new String[5][2];
    String[][] val2 = new String[4][2];

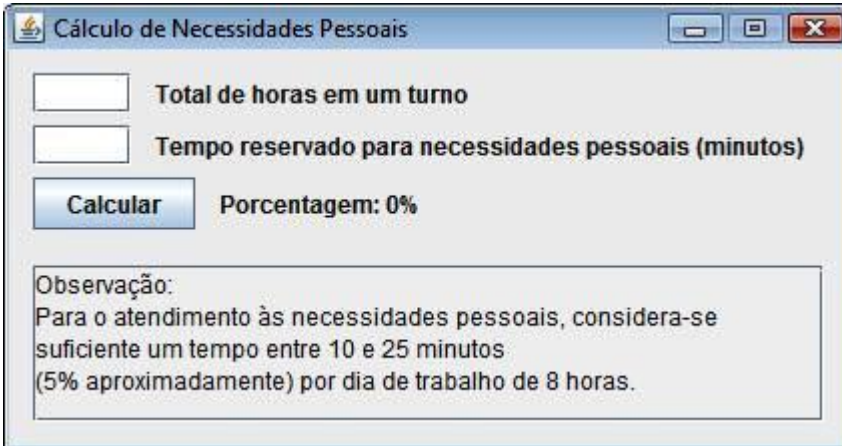
    fadiga = 0;

    try{
        for( int i=0; i<5; i++){
            for ( int j=1; j<4; j++){
                val[i][j] = String.valueOf(tabelaFadiga.getValueAt(i, j));
                if( i == 0){
                    switch (j){
                        case 1: if(val[i][j].equals("true")) fadiga = fadiga + 1;
                                break;
                        case 2: if(val[i][j].equals("true")) fadiga = fadiga + 1;
                                break;
                        case 3: if(val[i][j].equals("true")) fadiga = fadiga + 3;
                                break;
                    }
                }
            }
            if( i == 1){
                switch (j){
                    case 1: if(val[i][j].equals("true")) fadiga = fadiga + 2;
                            break;
                    case 2: if(val[i][j].equals("true")) fadiga = fadiga + 2;
                            break;
                    case 3: if(val[i][j].equals("true")) fadiga = fadiga + 5;
                            break;
                }
            }
            if( i == 2){
                switch (j){
                    case 1: if(val[i][j].equals("true")) fadiga = fadiga + 4;
                            break;
                }
            }
        }
    }
}
```





com os dados informados nos campos anteriormente citados, e um *JTextArea* com explicações sobre como preencher os campos, inclusive com sugestões.



**Figura 25 - Formulário de Cálculo de Necessidades Pessoais**

Fonte: Elaborado pelo autor.

Ao fechar o formulário, o campo *Necessidades Pessoais* da tela de cadastro de operações é preenchido com o valor calculado, conforme o código apresentado no Quadro 6 a seguir.

```
private void calculaNecessidades() {
    float horas = 0;
    int minutos = 25;
    float nec;
    if ( ValidaCampos() ){
        horas = Float.parseFloat( poePonto( txtHorasPorTurno.getText() ) );
        minutos = Integer.parseInt( txtTempoParaNecessidades.getText() );

        horas = horas * 60;
        nec = (minutos * 100) / horas;
        necessidade = (int) nec;

        lblPorcentagem.setText("Porcentagem: " + necessidade + "%");
    }
}
public void windowClosing(WindowEvent e) {

    cadOp.getTxtNecPessoais().setEnabled(true);
    cadOp.getTxtNecPessoais().setText( String.valueOf( necessidade ) );

    this.dispose();
}
```

}

#### Quadro 6 - Código para calcular o percentual de Necessidades Pessoais da operação.

Fonte: Elaborado pelo autor.

Na Figura 26 é apresentado o formulário de Simulação de Produção mostrado, que é utilizado para simular uma possível produção futura. Ele possui uma tabela com os campos: *Dias*, *Turnos* e *Horas por turno*, os quais geram o total de horas disponíveis a serem utilizadas; *Total de Operadores*, indicando o número de operadores que irão trabalhar na produção; e a *Capacidade a ser Utilizada* para simular a produção.

Em seguida há uma caixa de combinação com o nome de todos os produtos cadastrados, e um campo para a quantidade desejada do produto selecionado. Ao lado direito destes, há um botão *Adicionar*, que, quando acionado, adiciona os dados informados na lista localizada logo abaixo da caixa de combinação. Abaixo do botão *Adicionar* há um botão *Remove*, para remover da lista o produto que estiver selecionado e sua respectiva quantidade.

Logo abaixo há um botão *Simular*, que após verificar o preenchimento do formulário, realiza a simulação da produção de acordo com dados informados, preenchendo a tabela *Resultado*, que se encontra na parte inferior do formulário.

Dias	Turnos	Horas por turno	Nº de Operadores Disponíveis	Capacidade a ser Utilizada (%)
2	2	8	5	80

Produto: Dobradiça 10x10 inox    Quantidade Desejada: 2000   

**Resultado**

Máquinas/Operador	Horas Necessárias	Horas Disponíveis	Utilização (%)	Situação
Policorte Central	1.48	32.0	4	Tem capacidade
Guilhotina	1.62	32.0	5	Tem capacidade
Furadeira	0.14	32.0	0	Tem capacidade
Operador	1.75	32.0	5	Tem capacidade

Figura 26 - Formulário de Simulação de Produção

Fonte: Elaborado pelo autor.

O Quadro 7 a seguir apresenta o código utilizado para realizar a simulação.

```
private void simularProducao() {
    if ( validaTabela() ){
        if (listProducao.getLastVisibleIndex() == -1){
            JOptionPane.showMessageDialog(null,
                "A lista de produção está vazia", "Aviso", 1);
            return;
        }else{
            arrayMaquinas.clear();
            int dias = Integer.parseInt( dadosTabela[0][0] );
            int turnos = Integer.parseInt( dadosTabela[0][1] );
            float horasPorTurno = Float.parseFloat( poePonto( (String) dadosTabela[0][2] ) );
            int numOperadores = Integer.parseInt( dadosTabela[0][3] );
            int eficiencia = Integer.parseInt( dadosTabela[0][4] );
            int tamanhoList = produtoListModel.size();
            int contOp = 0;
            //Recuperar as operacoes de cada um dos produtos da lista
            iniciaConexaoBanco();
            Query query;

            Produtos prod = new Produtos();
            Operacoes op = new Operacoes();
            Maquinas maqDaOperacao = new Maquinas();
            String produtoNoArray;

            for ( int i=0; i<tamanhoList; i++ ){
                //pega o produto na posição 'i' do Array
                ProdutoProgramado pp = (ProdutoProgramado) arrayPP.get(i);
                produtoNoArray = pp.getProduto();

                //busca o produto no banco
                query = em.createQuery("SELECT p FROM Produtos p WHERE p.nomeProduto =
:nomeProduto");
                query.setParameter("nomeProduto", produtoNoArray );
                try {
                    prod = (Produtos) query.getSingleResult();
                }
            }
        }
    }
}
```

```

    } catch (Exception e) {
        javax.swing.JOptionPane.showMessageDialog( this, e.getMessage() );
    }
    //busca as operacoes do produto no banco
    try {
        query = em.createQuery("SELECT o.idOp FROM Operacoesdosprodutos o WHERE
o.idProd = :idProd");
        query.setParameter("idProd", prod );

        Vector resultado = (Vector) query.getResultList();
        //para cada operacao, pegar as maquinas dela
        for (int j=0; j<resultado.size(); j++){
            op = (Operacoes) resultado.get(j);
            contOp++;
        }
        //busca as maquinas no banco
        query = em.createQuery("SELECT m.idMaquina FROM Maquinasdasoperacoes m
WHERE m.idOperacao = :idOperacao");
        query.setParameter("idOperacao", op);

        Vector resultMaquinas = (Vector) query.getResultList();
        int qtdeMaquinas = resultMaquinas.size();
        float tempoDividido = ( (op.getTempoPadrao() * pp.getQuantidade())/3600 )
/qtdeMaquinas;

        try {
            for (int k=0; k<resultMaquinas.size(); k++){
                maqDaOperacao = (Maquinas) resultMaquinas.get(k);
                boolean novo = true;
                int ultimo = arrayMaquinas.size();

                for (int m=0; m<ultimo; m++){
                    MaquinaDaProgramacao MDP = new MaquinaDaProgramacao();
                    MDP = (MaquinaDaProgramacao) arrayMaquinas.get( m );

                    if (MDP.getNomeMaquina().equals( maqDaOperacao.getNomeMaquina()
))){

                        MDP.setTempoMaquina( MDP.getTempoMaquina() + tempoDividido );
                        novo = false;
                    }
                }
            }
        }
    }

```

```

    }
    if (novo){
        MaquinaDaProgramacao MDP = new MaquinaDaProgramacao();
        MDP.setNomeMaquina( maqDaOperacao.getNomeMaquina() );
        MDP.setTempoMaquina( tempoDividido );
        arrayMaquinas.add(ultimo, MDP);
    }
}
} catch (Exception e) {}
}
} catch( javax.persistence.NoResultException e ) {
    javax.swing.JOptionPane.showMessageDialog( this, e.getMessage() );
} catch (Exception e) {
    e.printStackTrace();
}
}
limpaTabelaResultado();

NumberFormat formatador = new DecimalFormat("0.00");
MaquinaDaProgramacao maq;
LinhaSimulacao linha = new LinhaSimulacao();
int numMaq = arrayMaquinas.size();
float horasDisponiveis = horasPorTurno * turnos * dias;
float horasNecessarias = 0;

for (int i=0; i<numMaq; i++){
    maq = new MaquinaDaProgramacao();
    maq = (MaquinaDaProgramacao) arrayMaquinas.get(i);
    //soma todos os tempos de todas as maquinas no operador
    horasNecessarias = horasNecessarias + maq.getTempoMaquina();
}
DefaultTableModel modeloTR = new DefaultTableModel();
modeloTR.addColumn("Máquinas/Operador");
modeloTR.addColumn("Horas Necessárias");
modeloTR.addColumn("Horas Disponíveis");
modeloTR.addColumn("Utilização (%)");
modeloTR.addColumn("Situação");

for (int j=0; j<numMaq; j++){

```

```

maq = new MaquinaDaProgramacao();
maq = (MaquinaDaProgramacao) arrayMaquinas.get(j);
linha.setNomeMaquina( maq.getNomeMaquina() );
if ( maq.getNomeMaquina().equalsIgnoreCase( "Operador" )){
    linha.setTempoNecessario( horasNecessarias / numOperadores );
    linha.setTempoDisponivel( horasDisponiveis );
    float util = (linha.getTempoNecessario() * 100) / horasDisponiveis;
    int utilizacao = (int) util;
    linha.setUtilizacao(utilizacao);
    String situacao;
    if ( (linha.getTempoNecessario() <= horasDisponiveis) && (utilizacao <= eficiencia))
        situacao = "Tem capacidade";
        else situacao = "Não tem capacidade";
        linha.setSituacao( situacao );
    }else{
        linha.setTempoNecessario( maq.getTempoMaquina() );
        linha.setTempoDisponivel(horasDisponiveis);
        float util = (maq.getTempoMaquina() * 100) / horasDisponiveis;
        int utilizacao = (int) util;
        linha.setUtilizacao(utilizacao);
        String situacao;
        if ( (linha.getTempoNecessario() <= horasDisponiveis) && (utilizacao <= eficiencia)
)
            situacao = "Tem capacidade";
            else situacao = "Não tem capacidade";
            linha.setSituacao( situacao );
        }
String nome = linha.getNomeMaquina();
float TN = Float.parseFloat( poePonto( formatador.format(linha.getTempoNecessario()
) ) );

float TD = linha.getTempoDisponivel();
int ut = linha.getUtilizacao();
String sit = linha.getSituacao();
Object[] dados = {nome, TN, TD, ut, sit} ;
modeloTR.addRow( dados );
}
tabelaResultado.setModel( modeloTR );
}
}

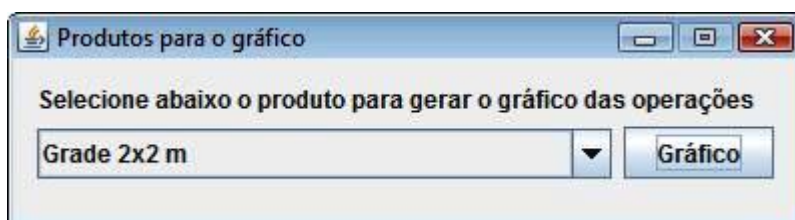
```

}

**Quadro 7 - Código para realizar a simulação da produção.**

Fonte: Elaborado pelo autor.

Após serem realizados alguns cadastros, é possível ao sistema então gerar relatórios e gráficos. Como exemplo de gráfico, há o gráfico de *Operações dos Produtos*, gerado a partir do produto selecionado em um formulário, como mostra a Figura 27.

**Figura 27 - Formulário para gerar gráfico de Operações dos Produtos**

Fonte: Elaborado pelo autor.

No formulário apresentado na Figura 27, há uma caixa de combinação, que possui o nome de todos os produtos cadastrados até então, e ao lado direito há um botão *Gráfico*, que ao ser acionado gera um gráfico com as operações do produto selecionado, sendo apresentada cada operação com seu respectivo tempo padrão, conforme mostra o Quadro 8 a seguir.

```
public GraficoOperacoesDosProdutos( OperacoesDoGrafico opProGrafico ) {
    super( "Gráfico" );
    this.opProGrafico = opProGrafico;
    this.setResizable(false);
    this.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
    Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
    Dimension frameSize = this.getSize();
    setLocation((screenSize.width - frameSize.width) / 4, (screenSize.height - frameSize.height) /
4);

    //Cria um dataSet para inserir os dados que serão passados para a criação do gráfico tipo Pie
    DefaultPieDataset pieDataset = new DefaultPieDataset();
```

```

        //Adiciona os dados ao dataSet deve somar um total de 100%
Operacoes[] opes = opProGrafico.getOpes();
int numOp = opes.length;
for (int i=0; i<numOp; i++){
    pieDataset.setValue( opes[i].getNomeOp() + ": \n" +
        String.valueOf(opes[i].getTempoPadrao() + " s"),
            opes[i].getTempoPadrao() );
}

//Cria um objeto JFreeChart passando os seguintes parametros
JFreeChart grafico = ChartFactory.createPieChart3D(
    "Operações do Produto: " + opProGrafico.getProduto().getNomeProduto(), //Titulo
    pieDataset, //DataSet
    true, //Para mostrar ou não a legenda
    false, //Para mostrar ou não os tooltips
    false);

PiePlot3D plot = (PiePlot3D) grafico.getPlot();
plot.setDirection(Rotation.CLOCKWISE);
plot.setForegroundAlpha(0.5f);

Panel p = new Panel();
p.setLayout( new BorderLayout() );
p.add( new ChartPanel( grafico ), BorderLayout.NORTH);

this.add( p );
this.pack();
}

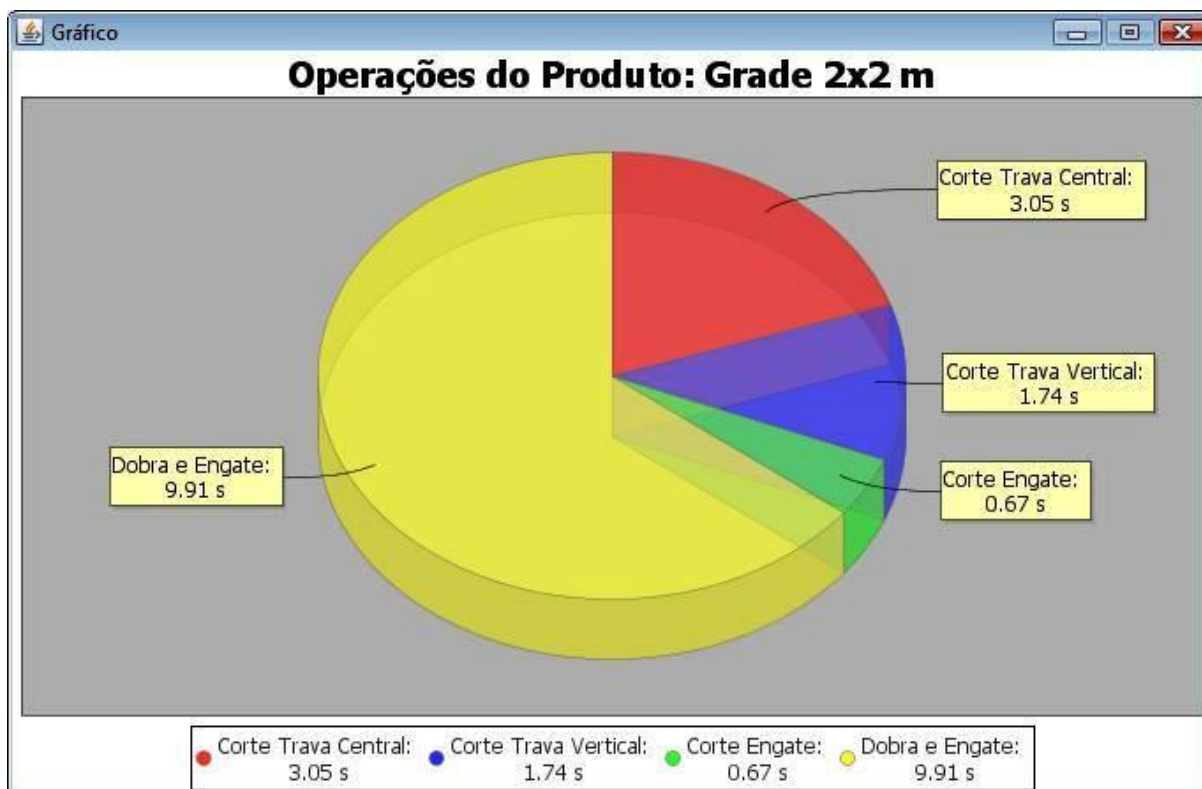
```

**Quadro 8 - Código para Geração do Gráfico de Operações dos Produtos**

Fonte: Elaborado pelo autor.

O resultado do código apresentado no Quadro 8 é apresentado na Figura 28.





**Figura 28 - Gráfico de Operações do Produto**

Fonte: Elaborado pelo autor.



Operações por Período

Data inicial: 05/06/2011

Data final: 20/06/2011

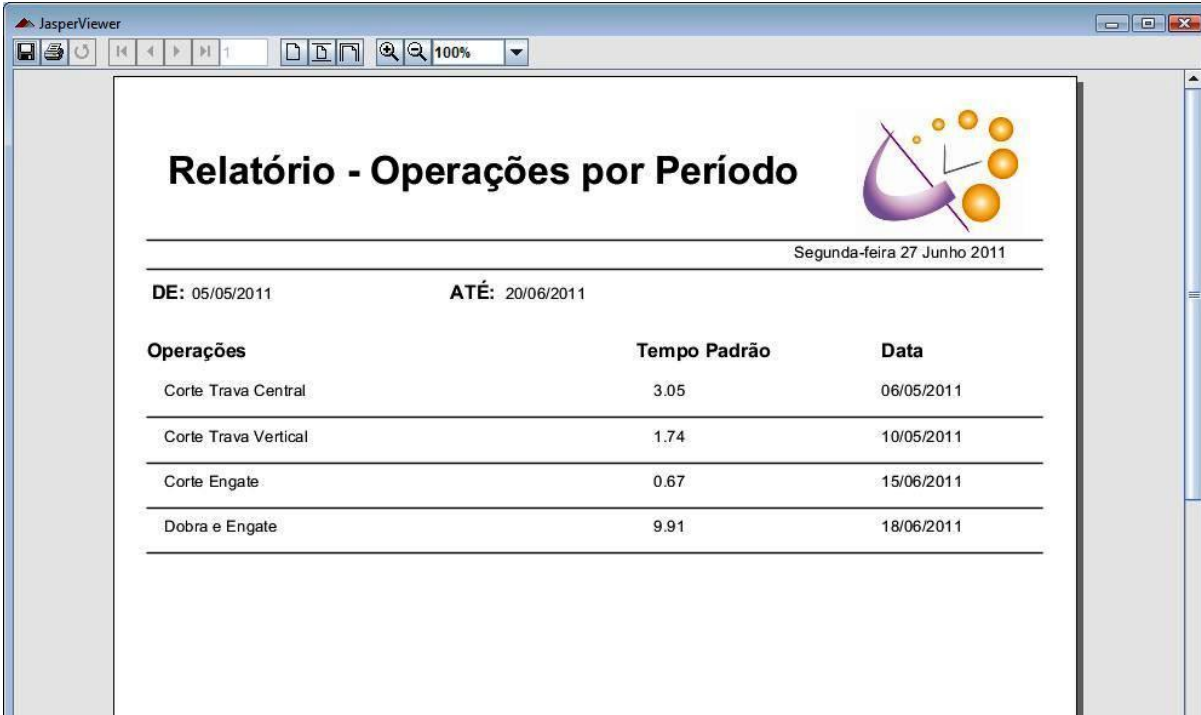
Relatório Cancelar

**Figura 29 - Formulário para filtrar o intervalo de datas para relatório**

Fonte: Elaborado pelo autor.

A Figura 29 apresenta um formulário que precede a apresentação do relatório *Operações por Período* possibilitando filtrar um intervalo de datas. O formulário possui um campo *Data inicial* e um campo *Data final* onde deve ser informado o intervalo de datas que será utilizado para construir o relatório, um botão

*Cancelar*, que ao ser acionado cancela a solicitação do relatório fechando o formulário, e um botão *Relatório*, que ao ser acionado verifica se as datas são válidas, e em caso positivo constrói o relatório, conforme apresentado na Figura 30.



Relatório - Operações por Período

Segunda-feira 27 Junho 2011

DE: 05/05/2011 ATÉ: 20/06/2011

Operações	Tempo Padrão	Data
Corte Trava Central	3.05	06/05/2011
Corte Trava Vertical	1.74	10/05/2011
Corte Engate	0.67	15/06/2011
Dobra e Engate	9.91	18/06/2011

**Figura 30 - Relatório de Operações por Período**

Fonte: Elaborado pelo autor.

## 5.1 Utilização do Protótipo

Ao acessar a tela principal do protótipo, após realizar o *login*, o usuário primeiramente deve efetuar os cadastros de Materiais, Ferramentas, Dispositivos e Máquinas que são utilizados na empresa, que se encontram no menu Cadastros da tela Principal. Em seguida, efetuar o cadastro de Operações que são realizadas na empresa, para, por último realizar o cadastro de Produtos produzidos na empresa. Caso o usuário prefira, ele pode ir direto ao formulário de cadastro de Produtos, e a partir deste, acessar os demais formulários de cadastros.

Após serem cadastrados alguns produtos, torna-se possível ao sistema gerar os relatórios e gráficos presentes no menu Relatórios da tela Principal, assim como realizar a simulação de produção dos produtos cadastrados, através da Simulação de Produção encontrada no menu Simulação da tela Principal.

Há também, na tela Principal, o menu Usuários, o qual pode ser acessado para incluir novos usuários no sistema, ou para que o usuário alterem a sua senha.

## 6 CONSIDERAÇÕES FINAIS

Esse trabalho objetivou desenvolver um sistema para determinar a capacidade produtiva de pequenas empresas com base em estudos de cronoanálise industrial para pequenas empresas com produção em série. Sendo assim, pode-se dizer que os objetivos foram atingidos, pois foi possível desenvolver um sistema para auxiliar na determinação da capacidade produtiva, além de atender a todos os objetivos específicos estipulados. Dentre os principais resultados estão os relatórios gerenciais, que auxiliarão as empresas do ramo produtivo a identificar possíveis melhorias em seus processos, a fim de expandir a capacidade produtiva da empresa.

Dentre os aspectos que geraram maior dificuldade, está o trabalho de codificação que resulta no tempo padrão, e a codificação que possibilita a simulação de produção, uma vez que o número de variáveis envolvidas nos cálculos é relativamente alto, sendo que as planilhas que eram utilizadas pela empresa para essa finalidade não possibilitavam um entendimento claro das variáveis e cálculos.

Ademais, o desenvolvimento do presente trabalho foi de muita importância para o aprendizado dos acadêmicos envolvidos acerca das técnicas de levantamento de requisitos, análise orientada a objetos e modelagem de sistemas, assim como técnicas de implementação em Java visual. Assim, foi possível aplicar os conceitos e técnicas estudados em sala de aula em um caso prático, o que é de muita validade para o aprendizado dos acadêmicos e para sua vida profissional.

Além disso, permitiu que fossem aprendidos conceitos e técnicas sobre capacidade produtiva instalada nas empresas, conhecimento necessário para o desenvolvimento do sistema em questão, proporcionando assim um diferencial aos acadêmicos envolvidos.

Para as pequenas empresas do ramo industrial, especialmente as que trabalham com produção em série, trata-se de um protótipo de sistema perfeitamente passível de ser implantado, proporcionando praticidade e eficiência na gestão de processos produtivos. Além disso, pode ser considerado um sistema

acessível financeiramente e tecnicamente, o qual certamente as auxiliará na melhoria da qualidade de seus processos, com produtividade e menores custos.

Futuramente o sistema pode ser aprimorado com a inclusão de novos módulos, como por exemplo, controle de estoque, o que permitirá além de saber se há capacidade para produzir, se há material suficiente para produzir, auxiliando assim no controle de custos da empresa.

## REFERÊNCIAS

ASTEC, **Treinamento Empresarial, Cronoanálise Industrial e Métodos e Processos**. Curitiba: 1999 e 2000.

BARROSO, Marta. **Definição de Estudo de Caso**. Disponível em <<http://claracoutinho.wikispaces.com/2-+Definição+do+Estudo+de+Caso+3>>. Acesso em 20/05/2011.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML Guia do Usuário**. Rio de Janeiro: Campus, 2000, 7ª edição.

BORSOI, Beatriz Terezinha; BRITTO, Robison Cris. **Linguagem de Programação Comercial**. Cuiabá: UFMT, 2010.

CASIMIRO, Anderson. **Orientação a Objetos – parte 1**. Disponível em <<http://duodraco.wordpress.com/2007/06/24/orientacao-a-objetos-parte-1/>>. Acesso em 11/05/2011.

CARLOS, João. **Ferramentas CASE**. Disponível em: <[http://imasters.uol.com.br/artigo/3048/uml/ferramentas\\_case/](http://imasters.uol.com.br/artigo/3048/uml/ferramentas_case/)>. Acesso em 13/05/2011.

CARVALHO, Daniel; SARTORATO, Eduardo; HELEN Fernanda; CARNEIRO, Rafael. Disponível em <<http://pesquisabibliografica.vilabol.uol.com.br/conceito.html>>. Acesso em 21/04/2011.

CASTILLO, Renata Almeida Fonseca del. **Aprendendo Sobre Pesquisas**. Disponível em <[http://www.ead.unicamp.br/trabalho\\_pesquisa/Pesq\\_estcaso.htm](http://www.ead.unicamp.br/trabalho_pesquisa/Pesq_estcaso.htm)>. Acesso em 20/05/2011.

CORREA, Carlos Henrique; TAFNER, Malcon Anderson. **Análise Orientada a Objetos**. Florianópolis: Visual Books, 2006.

DALLACQUA, Vinicius Teixeira. **Persistência de Dados em Java com JPA e Toplink.** Disponível em [http://www.portalct.com.br/blogs/tecnologia/administracao/files/files/Persistencia\\_de\\_Dados\\_em\\_Java\\_com\\_JPA\\_e\\_Toplink.pdf](http://www.portalct.com.br/blogs/tecnologia/administracao/files/files/Persistencia_de_Dados_em_Java_com_JPA_e_Toplink.pdf)>. Acesso em 30/06/2011.

DAVIS, Mark; AQUILANO, Nicholas; CHASE Richard, **Fundamentos da Administração da Produção.** Porto Alegre: Bookman, 2001.

MACORATTI, José Carlos. **Orientação a Objetos: Conceitos Básicos.** Disponível em <[http://www.macoratti.net/net\\_oocb.htm](http://www.macoratti.net/net_oocb.htm)>. Acesso em 11/05/2011.

MARTIN, James. **Análise e Projeto Orientados a Objeto.** São Paulo: Makron Books, 1995.

MARTINS, Petronio. **Administração da Produção.** São Paulo: Saraiva, 1998.

MEDINA, Roseclea Duarte. **Capítulo 4 – Polimorfismo.** Disponível em <[www-usr.inf.ufsm.br/~rose/curso3/cafe/cap4\\_Polimorfismo.pdf](http://www-usr.inf.ufsm.br/~rose/curso3/cafe/cap4_Polimorfismo.pdf)>. Acesso em 11/05/2011.

RITZMAN, Larry. **Administração da Produção e Operações.** São Paulo: Pearson, 2004.

## ANEXOS





## ANEXO B - PLANILHA (EXCEL) UTILIZADA PELA EMPRESA EM ESTUDO PARA SIMULAR A PRODUÇÃO

Descrição dos Produtos	Programação Diária	Nº de Arames por Peça	Número da Operação (cronoanálise)	Ação Aplicada Arames	Diâmetro, espessura e Acabamento	Temperatura Arame em mm	Nº de Arames para cortar	Capacidade e por operação	Grupo Máquina Nome	Tempo Padrão	Tempo Padrão (5% AI)	Tempo Setup (Seg)	Tempo +Setup (Seg)	Nº de Horas Neces.	
PEÇA A	1	1	10	ARMA	arm art 4,0		1	220		16,36	0,00		0,00	0,00	Capacidade dia = 1820 pcs com 6 operadores
		1	20	ARMA		1	232		15,54	0,00		0,00	0,00		
		1	30	ARMA		1	284		12,67	0,00		0,00	0,00		
		1	40	PONTEIA		1	248	PT002	14,52	0,00	0	0,00	0,00		
		1	50	REBARBA		1	201	RBP002	17,92	0,00		0,00	0,00		
		1	60	CALIBRA		1	230		15,62	0,00		0,00	0,00		
		1	70	CARREGA		1	390		9,22	0,00		0,00	0,00		
PEÇA B	1	1	10	LIXA	arm art 5,5 e 4,5		1	2635	LX001	1,37	0,00		0,00	0,00	Capacidade dia = 2700 pcs com 11 operadores
		1	20	LIXA		1	3896	LX001	0,92	0,00		0,00	0,00		
		1	30	ARMA		1	521		6,91	0,00		0,00	0,00		
		1	40	ARMA		1	410		8,78	0,00		0,00	0,00		
		1	50	ARMA		1	419		8,58	0,00		0,00	0,00		
		1	60	PONTEIA		1	391	PT001	9,22	0,00	0	0,00	0,00		
		1	70	REBARBA		1	616	PE008	5,85	0,00	0	0,00	0,00		
		1	80	ESTAMPA		1	523	PE002	6,89	0,00	0	0,00	0,00		
PEÇA C	1	1	10	ARMA	arm art 4,0		1	220		16,37	0,00		0,00	0,00	Capacidade dia = 1900 pcs com 8 operadores
		1	20	ARMA		1	232		15,54	0,00		0,00	0,00		
		1	30	ARMA		1	282		12,75	0,00		0,00	0,00		
		1	40	PONTEIA		1	254	PT002	14,16	0,00	0	0,00	0,00		
		1	50	REBARBA		1	201	RBP002	17,92	0,00		0,00	0,00		
		1	60	DOBRA		1	345	PE008	10,42	0,00	0	0,00	0,00		
		1	70	CALIBRA		1	248		14,50	0,00		0,00	0,00		
		1	80	CARREGA		1	455		7,91	0,00		0,00	0,00		
<b>Descrição do Grupo Máquina</b>		<b>Nº de Máquinas</b>		<b>Nº Turnos</b>	<b>Nº Horas por turno</b>	<b>Nº Horas Paradas</b>	<b>Nº Horas Úteis</b>	<b>Nº Horas Utilizadas</b>	<b>Nº Horas Disponíveis</b>		<b>Índice de Utilização</b>	<b>Situação</b>	<b>Eficiência (%)</b>		
PE002		1		1	9,5	1	8,5	0,00	8,5		0%	TEM CAPACIDADE	95%		
PE008		1		1	9,5	1	8,5	0,00	8,5		0%	TEM CAPACIDADE	95%		
PT001		1		1	9,5	1	8,5	0,00	8,5		0%	TEM CAPACIDADE	95%		
PT002		1		1	9,5	1	8,5	0,00	8,5		0%	TEM CAPACIDADE	95%		
RBP002		1		1	9,5	1	8,5	0,00	8,5		0%	TEM CAPACIDADE	95%		
LX001		1		1	9,5	1	8,5	0,00	8,5		0%	TEM CAPACIDADE	95%		
ZP		23		1	9,5	1	194,7	0,00	194,7		0%	TEM CAPACIDADE	95%		

Fonte: Elaborado pelos funcionários da empresa em questão.