



**UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC**  
**CENTRO DE EDUCAÇÃO DO PLANALTO NORTE – CEPLAN**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM DESENVOLVIMENTO DE *SOFTWARE***

TRABALHO DE CONCLUSÃO DE CURSO DE  
ESPECIALIZAÇÃO

**DESENVOLVIMENTO DE UM  
*SOFTWARE* OTIMIZADOR DE  
CORTES PARA CHAPAS DE  
MADEIRA: OTIMIZACUT**

**AISLAN NADROWSKI**

**SÃO BENTO DO SUL, SC, 2013**

**AISLAN NADROWSKI**

**DESENVOLVIMENTO DE UM *SOFTWARE* OTIMIZADOR DE  
CORTES PARA CHAPAS DE MADEIRA: OTIMIZACUT**

Monografia apresentada ao curso de Pós-graduação em Desenvolvimento de *Software* do Centro de Educação do Planalto Norte, da Universidade do Estado de Santa Catarina como requisito parcial para obtenção do título de Especialista em Desenvolvimento de *Software*.

Orientador: Fabio Fernando Kobs,  
MSc.

**SÃO BENTO DO SUL, SC  
2013**

**AISLAN NADROWSKI**

**DESENVOLVIMENTO DE UM *SOFTWARE* OTIMIZADOR DE  
CORTES PARA CHAPAS DE MADEIRA: OTIMIZACUT**

Monografia apresentada ao curso de Pós-graduação em Desenvolvimento de *Software* como requisito parcial para obtenção do título de Especialista em Desenvolvimento de *Software*.

**Banca examinadora**

Orientador:

---

(Prof. MSc. Fabio Fernando Kobs)  
Universidade do Estado de Santa Catarina

Membro:

---

(Prof. Dr. Nilson Ribeiro Modro)  
Universidade do Estado de Santa Catarina

Membro:

---

(Prof. MSc. Fábio Manoel Caliori)  
Universidade do Estado de Santa Catarina

**São Bento do Sul, 03/10/2013**

Dedico a minha mãe Everilda, a  
meu irmão Elisan e a minha amada  
Fernanda.

## AGRADECIMENTOS

O desenvolvimento de uma pesquisa acadêmica demanda um esforço cujo qual não seria possível sem o auxílio e apoio de outras pessoas. Gostaria de agradecer aos professores do curso de pós-graduação em Desenvolvimento de *Software* do CEPLAN da UDESC, principalmente ao professor MSc. Fabio Fernando Kobs que orientou-me durante a realização do trabalho sempre com disposição para ajudar e sugerir melhorias e alterações.

Também agradeço a minha querida Fernanda Fleischmann pelo apoio na busca por referencial teórico utilizado para desenvolver o trabalho e pela sua pronta ajuda em reler o trabalho em busca de correções e melhorias.

Outra pessoa importante foi a amiga Géssica Maíra Arbigaus que auxiliou-me a sanar dúvidas no início da pesquisa, ainda quando a mesma possuía outro enfoque. Também gostaria de agradecer ao amigo Roberto Lopes no auxílio com o desenvolvimento do logotipo do *software*.

Por fim, é importante lembrar dos colegas de trabalho cujo os quais passa-se horas do dia lado-a-lado e que auxiliam no amadurecimento profissional, em particular, gostaria de agradecer a Eliseu da Silva por sua paciência em ensinar-me os conceitos sobre planos de corte e máquinas seccionadoras.

Obrigado!

"Seja gentil, porque cada pessoa que  
você encontra esta travando uma dura  
batalha."

John Watson

## RESUMO

NADROWSKI, Aislan. **Desenvolvimento de um *software* otimizador de cortes para chapas de madeira:** OtimizaCut. Monografia (Especialização em Desenvolvimento de *Software*) – Universidade do Estado de Santa Catarina. Programa de Pós-graduação em Desenvolvimento de *Software*, São Bento do Sul, 2013.

Este trabalho documenta o desenvolvimento de um *software* otimizador de cortes, nomeado OtimizaCut. Seu uso é destinado especificamente para o setor moveleiro no auxílio à realização de cortes em chapas de madeira (*MDF*) para produção de um móvel. O problema da pesquisa que dá embasamento ao trabalho é o de se é possível elaborar um *software* para auxílio na gestão de matéria-prima e redução de tempo de produção de um móvel. Trata-se de uma pesquisa de natureza aplicada. Aborda o problema de maneira qualitativa com fins exploratórios. Foi empregada a pesquisa de campo e pesquisa bibliográfica durante a realização da pesquisa. O objetivo geral da pesquisa é desenvolver um *software* otimizador de cortes que busque uma resposta ao problema da pesquisa. A linguagem de programação utilizada para desenvolvimento do *software* é o *C#* da Microsoft. Para o armazenamento dos dados manipulados pelo *software* são utilizados arquivos *XML*. A conclusão da pesquisa viabilizou o desenvolvimento do *software* otimizador de cortes. A proposta desenvolvida auxiliou na geração dos planos de corte e possibilitou a gestão da matéria-prima e ganho no tempo de produção de um móvel, pois não fica mais a cargo do operador da máquina seccionadora criar tais planos sem a utilização de um *software* específico.

**Palavras-chave:** Plano de corte de chapas. Linguagem de programação *C#*. Linguagem de marcação *XML*.

## LISTA DE ILUSTRAÇÕES

Figura 1 - Controles do <i>WPF</i> 1. ....	13
Figura 2 - Controle <i>TabControl</i> . ....	13
Figura 3 - Controles do <i>WPF</i> 2. ....	13
Figura 4 - Controle <i>DataGrid</i> . ....	14
Figura 5 - Controle <i>Listview</i> . ....	15
Figura 6 - Exemplo de plano de corte. ....	19
Figura 7 - Móvel renderizado. ....	23
Figura 8 - Móvel renderizado com as portas abertas. ....	24
Figura 9 - Plano de corte do material Carvalho 6mm. ....	27
Figura 10 - Plano de corte do material Carvalho 15mm. ....	28
Figura 11 - Plano de corte do material Carvalho 18mm. ....	29
Figura 12 - Seccionadora Fit 2.9. ....	31
Figura 13 - Janela de acesso aos parâmetros de configuração. ....	44
Figura 14 - Parâmetros do relatório. ....	45



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>9</b>
<b>2</b>	<b>REFERENCIAL TEÓRICO .....</b>	<b>12</b>
2.1	LINGUAGEM DE PROGRAMAÇÃO C#.....	12
2.2	LINGUAGEM DE MARCAÇÃO XML .....	16
2.3	PLANO DE CORTE DE CHAPAS.....	18
<b>3</b>	<b>METODOLOGIA DA PESQUISA .....</b>	<b>20</b>
3.1	NATUREZA DA PESQUISA.....	20
3.2	ABORDAGEM DO PROBLEMA .....	20
3.3	FINS DA PESQUISA .....	20
3.4	MEIOS DE INVESTIGAÇÃO.....	21
3.5	PROCEDIMENTOS METODOLÓGICOS .....	21
<b>4</b>	<b>EMPREGABILIDADE DO SOFTWARE .....</b>	<b>23</b>
4.1	PROJETO .....	23
4.2	OTIMIZAÇÃO.....	26
4.3	CORTE, USINAGEM E MONTAGEM.....	30
<b>5</b>	<b>IMPLEMENTAÇÃO DO SOFTWARE .....</b>	<b>33</b>
5.1	MÓDULOS DO SOFTWARE .....	33
5.1.1	Cadastro de Matéria-prima e Lote.....	33
5.1.2	Cálculo do Plano de Corte .....	40
5.1.3	Visualização de Planos de Corte.....	40
5.2	CONFIGURAÇÕES .....	41
5.3	RELATÓRIO .....	44
<b>6</b>	<b>CONCLUSÃO .....</b>	<b>47</b>
	<b>REFERÊNCIAS .....</b>	<b>50</b>
	<b>APÊNDICE A – XML DE UM PLANO OTIMIZADO .....</b>	<b>51</b>
	<b>APÊNDICE B – CÓDIGO SALVAR PARÂMETROS .....</b>	<b>56</b>
	<b>APÊNDICE C – CÓDIGO PARA GERAR RELATÓRIO..</b>	<b>59</b>
	<b>APÊNDICE D – MANUAL DO USUÁRIO .....</b>	<b>73</b>

## 1 INTRODUÇÃO

Este trabalho documenta o desenvolvimento de um *software* otimizador de cortes, nomeado OtimizaCut, destinado especificamente para o setor moveleiro. Um *software* otimizador de cortes é utilizado dentro da indústria moveleira com o objetivo de reduzir o desperdício de matéria-prima e também é um agente responsável por reduzir o tempo do processo de fabricação de um móvel. Existem diversos fluxos de trabalho dentro de uma marcenaria. Um dos fluxos iniciais é a fase de projeto de um ambiente (entende-se por ambiente uma cozinha, um dormitório, entre outros). O projeto é elaborado através da utilização de um *software* de desenho de ambientes, onde existe a possibilidade de gerar uma renderização, ferramenta essa que permite apresentar o ambiente ao cliente de maneira mais sofisticada. Após a aprovação dessa primeira etapa do fluxo, dá-se início a próxima fase, onde é criado um lote dentro do *software* otimizador de cortes a fim de gerar os planos de corte. Concluída essa fase, basta realizar o corte desses planos de corte na máquina seccionadora e fazer as eventuais usinagens ou aplicações de acabamentos superficiais nas peças, para que seja realizada a montagem dos móveis que compõem o ambiente.

Normalmente a máquina utilizada para cortar chapas de MDF (*Medium-Density Fiberboard*, ou seja, placa de fibra de madeira de média densidade) é a seccionadora, que é produzida por diversos fabricantes no Brasil e no exterior. Basicamente, essas seccionadoras são denominadas manuais ou automáticas. A principal característica da máquina manual é que ela é comandada inteiramente pelo operador, não existem ciclos automáticos de trabalho. É para esse tipo de máquina que o *software* OtimizaCut foi desenvolvido, onde o mesmo gera o plano de corte e permite que sejam impressos relatórios com os planos de corte para que o operador, com base nesses relatórios, execute os cortes da chapa na máquina. Já a principal característica da máquina automática é que o operador alimenta a mesma com matéria-prima e apenas inicia o corte, pois os ciclos de corte são realizados automaticamente e existe uma integração direta entre o otimizador de cortes e a seccionadora.

O *software* OtimizaCut permite ao usuário fazer o cadastrado de chapas de MDF, com suas respectivas características (como o comprimento, largura, espessura, cor, se possui veio ou não, código, descrição e tipo), criar lotes contendo tamanhos variados de peças – para formar um móvel ou mais – e posteriormente otimizar esse lote, gerando

assim um ou mais planos de corte dependendo da quantidade de peças inseridas dentro do lote. Após a otimização do lote, é possível imprimir relatórios contendo dados estatísticos acerca dos respectivos planos de corte, bem como imprimir o plano de corte para que o operador possa executar os cortes na máquina seccionadora manual.

O *software* otimizador de cortes não é para ser utilizado pelo operador da máquina de maneira direta, isso quer dizer que, ele apenas utiliza os relatórios gerados pelo *software* para execução dos cortes realizados na máquina. Geralmente, dependendo da estrutura da empresa que utiliza esse tipo de *software*, o mesmo é instalado em um computador situado no setor de planejamento, programação e controle de produção (PPCP) onde um funcionário responsável pela gestão da produção, através da utilização do *software*, tem o controle de peças a cortar, quantidade de matéria-prima utilizada que se transformou em peças úteis para a produção dos móveis e também a quantidade de matéria-prima que virou desperdício, através das sobras e refilos gerados nas chapas. Outra aplicação para o *software* é a utilização do mesmo em computadores instalados no balcão de empresas que revendem chapas de MDF. Essas empresas normalmente oferecem chapas de diversos tipos e cores a seus clientes e muitos destes clientes são marceneiros que não possuem uma máquina seccionadora para fazer o corte dessas chapas, nesse caso, o marceneiro informa as dimensões que deseja para as peças necessárias para montar seu móvel e esses cortes são realizados na empresa de vendas de chapas. O funcionário faz o cadastro das peças desejadas e otimiza o plano de corte, através dessa otimização é possível gerar os relatórios para ver quantos cortes são necessários para realização do trabalho (tento em vista que esse tipo de empresa normalmente cobra por corte realizado), bem como é possível ver quantas chapas são necessárias para montar determinado móvel, evitando assim compras desnecessárias por parte do cliente, através de um orçamento claro do serviço realizado.

O problema da pesquisa que dá embasamento ao trabalho é: **É possível elaborar um *software* para auxílio na gestão de matéria-prima e redução de tempo de produção de um móvel?**

O objetivo geral dessa pesquisa é: **Desenvolver um *software* otimizador de cortes a fim de reduzir desperdício de matéria-prima e diminuir o tempo do processo de fabricação de um móvel.** Os

objetivos específicos que auxiliam na realização do objetivo geral da pesquisa são:

- Estudo da linguagem de programação C#;
- Estudo de arquivos *XML* utilizados para armazenamento e manipulação das informações do *software*;
- Levantamento de requisitos para o *software* através da observação do autor dentro do ambiente industrial;
- Codificação do *software*.

O desenvolvimento do *software* é realizado utilizando a linguagem de programação orientada à objetos C# da Microsoft. A escolha da linguagem ocorreu por dois motivos: o desenvolvedor do *software* possuía grande interesse em estudar e aplicar essa linguagem de programação e pela sua rapidez de execução, comparada a algumas outras linguagens disponíveis no mercado atualmente. A velocidade de execução de um *software* otimizador de cortes é de grande valia e é um diferencial de mercado, pois dependendo da complexibilidade do cálculo a ser realizado para geração dos planos de corte nota-se uma grande diferença de desempenho. Já para o armazenamento e manipulação de dados do *software* foram empregados arquivos *XML* (*eXtensible Markup Language*) devido a sua flexibilidade e agilidade de utilização.

Outro fator importante à ser comentado é que neste documento não são divulgadas as regras utilizadas para fazer o posicionamento das peças no plano de corte (regras do cálculo), pois elas constituem no segredo de negócio. O restante do *software* é comentado, como o cadastro de materiais e lotes, visualização de planos de corte, entre outros.

A divisão de capítulos no trabalho é dada da seguinte forma: o capítulo 2 traz os principais conceitos para o entendimento dos elementos da pesquisa, proporcionando um alinhamento entre a teoria e o problema apresentado na pesquisa. Já o capítulo 3 apresenta as informações com relação à classificação da pesquisa dentro da metodologia da pesquisa. O capítulo 4 mostra a aplicabilidade do *software* OtimizaCut no mercado e o capítulo 5 documenta o desenvolvimento do mesmo. Por fim, o capítulo 6 apresenta as considerações finais, bem como sugestões para trabalhos futuros.

## 2 REFERENCIAL TEÓRICO

Este capítulo trás conceitos acerca dos temas abordados no desenvolvimento da pesquisa. A linguagem de programação utilizada para implementação do *software* foi o *C#* desenvolvido pela Microsoft. Já a linguagem de marcação utilizada para armazenamento dos dados do *software* foi a *Extensible Markup Language (XML)* e por fim é abordado o conceito de plano de corte.

### 2.1 LINGUAGEM DE PROGRAMAÇÃO C#

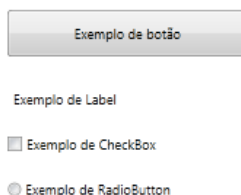
Para Sharp (2008) o *C#* é uma linguagem de programação orientada à objetos que compõem o *framework* da Microsoft chamado .NET (lê-se dot net). Possui muitas similaridades com outras linguagens de programação como o *C*, *C++* e o *Java*. A Microsoft desenvolveu um ambiente de programação poderoso chamado Visual Studio, é distribuída uma versão gratuita aos desenvolvedores chamada Visual Studio Express.

A linguagem de programação *C#* fornece uma importante ferramenta para desenvolvimento de interfaces gráficas de *software* chamada *Windows Presentation Foundation – WPF*. Utilizando o Visual Studio pode-se criar aplicativos *WPF* com bastante facilidade através do posicionamento de controles dentro da janela (*window*) apenas arrastando tais controles para dentro da janela ou através da escrita e manipulação do código *Application Markup Language – XAML*. Basicamente, toda janela e controles que são inseridos nela consistem em um código *XAML*. Os principais controles disponíveis no *WPF* são:

- **Button:** trata-se de um botão, onde é associado um evento para realização de uma determinada tarefa (figura 1);
- **Label:** é um texto que é utilizado dentro das janelas (figura 1);
- **CheckBox:** é um controle utilizado para marcação de verdadeiro ou falso (figura 1);
- **RadioButton:** é um controle utilizado para marcação de verdadeiro ou falso (figura 1);

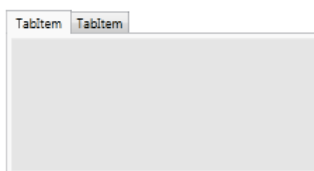
- **TabControl:** é um controle que possui abas superiores onde cada aba é visualizada individualmente (figura 2);
- **ComboBox:** é uma lista de itens em que o usuário pode selecionar apenas uma opção por vez (figura 3);
- **Datagrid:** trata-se de uma tabela para edição de dados (figura 4);
- **ListBox:** trata-se de uma tabela para visualização de dados (figura 5);
- **TextBox:** é um controle onde é o usuário pode inserir texto (figura 3);

Figura 1- Controles do WPF 1.



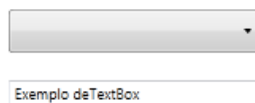
Fonte: produção do próprio autor.

Figura 2 - Controle *TabControl*.



Fonte: produção do próprio autor.

Figura 3 - Controles do WPF 2.



Fonte: produção do próprio autor.

Figura 4 - Controle *Datagrid*.

Codigo	Descricao	Comprimento	Largura	Quantidade	Rotacionar
1	Fundo	263	663	1	<input checked="" type="checkbox"/>
2	Traseira Gaveta	200,6	101,25	4	<input checked="" type="checkbox"/>
3	Lat esq gav	350	101,25	4	<input checked="" type="checkbox"/>
4	Lat dir gav	350	101,25	4	<input checked="" type="checkbox"/>
5	Base	300	400	2	<input type="checkbox"/>
6	Lateral	663	400	2	<input type="checkbox"/>
					<input type="checkbox"/>

Fonte: produção do próprio autor.

Figura 5 - Controle *Listview*.

Código	Descrição	Comprimento	Largura	Espessura	Cor	Tipo	Tem veio
1	MDF 18 mm Branco	1500	1500	18	Branco	MDF	Não
10	MDF Carvalho 6mm	2750,0	1850,0	6,0	Carvalho	MDF	Sim
11	MDF Carvalho 15mm	2750,0	1850,0	15,0	Carvalho	MDF	Sim
12	MDF Carvalho 18mm	2750,0	1850,0	18,0	Carvalho	MDF	Sim
2	MDF 10 mm Branco	2750,0	1850,0	10,0	Branco	MDF	Não
5	MDF 6 mm Branco	2750,0	1850,0	6,0	Branco	MDF	Não

Fonte: produção do próprio autor.



Ainda segundo Sharp (2008) o *C#* possui suporte a outros itens importantes a uma linguagem de programação moderna, dentre eles pode-se citar:

- **Estruturas de decisão:** a linguagem possui comandos como *if*, *else* e *switch*;
- **Estruturas de iteração:** utilizadas para criar *loops* dentro do código como o *for*, *while* e *do*;
- **Gerenciamento de erros e exceções:** a linguagem permite através da utilização dos comandos *try* e *catch* captar algum erro no momento de execução de determinada parte do código e lançar uma exceção, ou seja, avisar ao usuário que tal erro ocorreu e fazer o tratamento desse erro;
- **Coleções e arrays:** a linguagem suporta a utilização de coleções de objetos e valores;
- **Orientação a objetos:** a linguagem permite trabalhar com orientação a objetos, possuindo suporte ao gerenciamento de classes e objetos, herança, entre outros.

## 2.2 Linguagem de Marcação XML

Segundo Almeida (2002), em dezembro de 1997 foi publicada a primeira versão da linguagem de marcação chamada *Extensible Markup Language*, cuja sigla é *XML*. Esta primeira versão da linguagem foi proposta pelo consorcio W3 com o objetivo de produzir uma linguagem simplificada do *Standard Generalized Markup Language (SGML)*.

Dente os objetivos estabelecidos para a linguagem *XML* constam as seguintes atribuições:

- Possuir um código legível por humanos;
- Ser diretamente utilizável na Internet;
- Possibilitar um protocolo para troca de informações entre diferentes empresas.

A sintaxe básica da linguagem *XML* consiste em elemento, atributo e *Document Type Definitions (DTD)*. Um elemento é constituído por uma marcação inicial e uma marcação final, dentro destas marcações é inserido o texto bruto. O exemplo abaixo é composto por três elementos – Pessoa, Nome e Idade:

```

<Pessoa>
  <Nome>Aislan</Nome>
  <Idade>24</Idade>
</Pessoa>

```

Outro componente do *XML* é o atributo, que é associado ao elemento com o intuito de atribuir características ou propriedades a esse elemento. No desenvolvimento do *software* otimizador de cortes não foram empregados os atributos, pois não se fez necessário. No entanto, segue um exemplo de aplicação do atributo “identificador”, cujo valor é 1:

```

<Pessoa identificador = “1”>
  <Nome>Aislan</Nome>
  <Idade>24</Idade>
</Pessoa>

```

Ainda segundo Almeida (2002) o terceiro componente da sintaxe do *XML* é o *DTD* e trata-se de uma gramática dos elementos definidos pelo desenvolvedor dentro do arquivo *XML*.

```

<Documento>
  <Pessoa>
    <Nome>Aislan</Nome>
    <Idade>24</Idade>
  </Pessoa>
</Documento>

```

Considere o exemplo acima, o *DTD* equivalente é exibido abaixo. Onde a primeira linha diz qual é o elemento raiz do documento, no caso <Documento>. As próximas quatro linhas são marcações que mostram a possibilidade, através do asterisco (\*), do documento possuir quantidade ilimitada de elementos <Pessoa> contendo os elementos <Nome> e <Idade>, cujo quais possuem apenas informações e não mais elementos dentro deles.

```

<DOCTYPE Documento [
  <!ELEMENT DOCUMENTO(Pessoa*)>
  <!ELEMENT Pessoa (Nome, Idade)>
  <!ELEMENT Nome (#PCDATA)>
  <!ELEMENT Idade (#PCDATA)>
>

```

## 2.3 PLANO DE CORTE DE CHAPAS

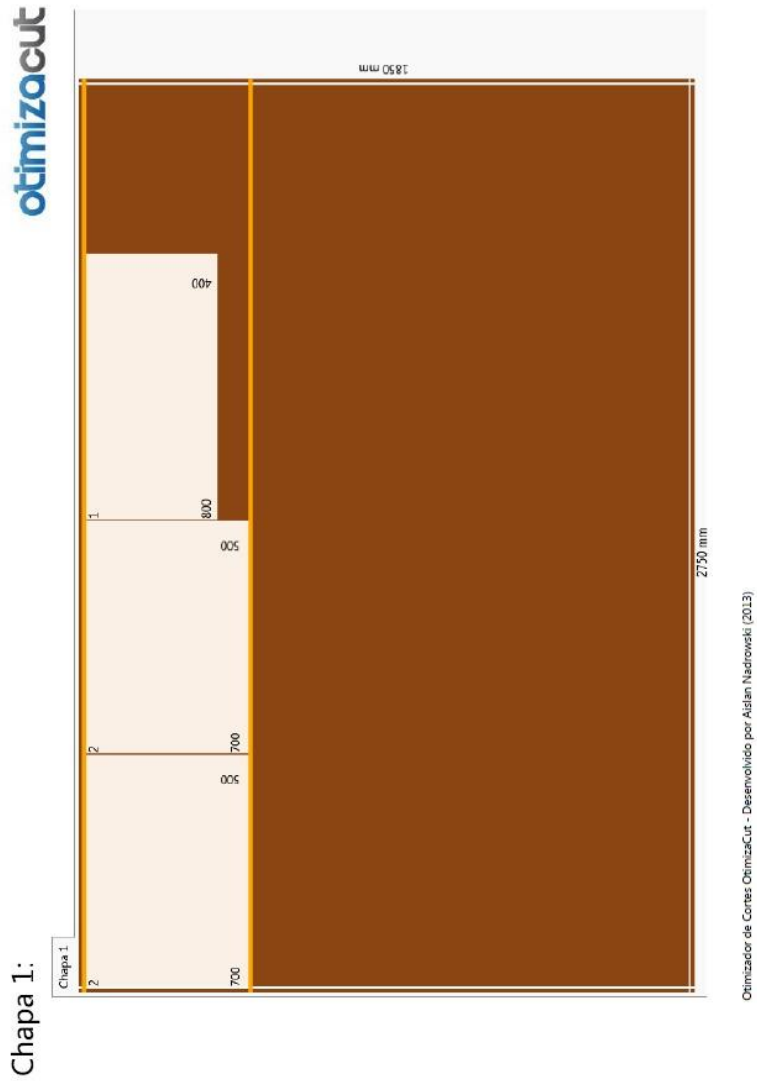
Segundo Oficina em Casa (2013) antes de iniciar os cortes em uma chapa de madeira é interessante fazer um estudo acerca do posicionamento das peças dentro da chapa, a fim de evitar desperdícios de matéria-prima.

Para isso pode-se criar tais planos de corte com auxílio de um *software* otimizador de cortes ou através de cálculos realizados pelo próprio marceneiro. Na figura 6 é possível ver um exemplo de plano de corte gerado pelo *software* OtimizaCut, que foi desenvolvido durante a execução da pesquisa.

Alguns elementos que compõem um plano de corte são:

- Representação das peças dentro de uma chapa;
- Representação dos cortes realizados na chapa em uma máquina seccionadora;
- Representação das dimensões (comprimento, largura e espessura) das peças dentro do plano de corte;
- Dados estatísticos, tais como: aproveitamento da chapa, quantidade de cortes realizados, entre outros.

Figura 6 - Exemplo de plano de corte.



Fonte: produção do próprio autor.

### **3 METODOLOGIA DA PESQUISA**

Este capítulo apresenta as informações com relação à classificação da pesquisa dentro da metodologia da pesquisa. Os tipos de classificação são: natureza da pesquisa, abordagem do problema, fins da pesquisa e meios de investigação. Ainda neste capítulo são apresentados os procedimentos metodológicos da pesquisa.

#### **3.1 NATUREZA DA PESQUISA**

A natureza seguida é a modalidade de pesquisa aplicada. Segundo Moresi (2003) essa natureza de pesquisa tem como objetivo gerar conhecimento com aplicabilidade prática com o intuito de solucionar problemas específicos.

Para Marconi e Lakatos (2009) uma pesquisa de natureza aplicada tem interesse prático, isto é, os resultados obtidos nesse tipo de pesquisa são utilizados para solução de problemas reais.

#### **3.2 ABORDAGEM DO PROBLEMA**

A abordagem da pesquisa é qualitativa, pois para Moresi (2003), há uma relação dinâmica entre o sujeito da pesquisa e o mundo real. Outro fator importante é a interpretação dos fenômenos e a atribuição de significados, onde a fonte de coleta de dados é o ambiente natural.

Traçando um paralelo com a pesquisa atual, o ambiente natural são os clientes em que o autor realizou observações diretas. As regras de cálculo são fundamentadas em atribuições de significados mais e menos importantes as vistas do cliente.

#### **3.3 FINS DA PESQUISA**

A classificação da pesquisa é exploratória com base nos objetivos seguidos para execução da mesma. Segundo Gil (2007) esse tipo de pesquisa tem como finalidade proporcionar aproximação com o problema estudado, com o propósito de torná-lo mais explícito ou para construção de hipóteses. O autor ainda comenta que esse tipo de pesquisa envolve as seguintes etapas:

- 1) Levantamento bibliográfico;

- 2) Entrevista com pessoas envolvidas no problema;
- 3) Análise de exemplo que auxiliem na compreensão do problema.

### 3.4 MEIOS DE INVESTIGAÇÃO

Um dos meios de investigação utilizado na pesquisa é a pesquisa de campo, pois foi utilizada pelo autor a observação direta durante visitas e conversações com clientes.

Para Moresi (2003), uma pesquisa de campo possui ferramentas como entrevista, testes, aplicação de questionários e observação participante ou não. Esse meio de pesquisa é realizado no local onde determinado fenômeno ocorre, não limitando-se a apenas pesquisas em laboratórios.

Outro meio de investigação utilizada é a pesquisa bibliográfica. Ainda segundo Moresi (2003), esse tipo de pesquisa possui como base publicações impressas e digitais, como livros, revistas, artigos e outros trabalhos acadêmicos. Normalmente este tipo de pesquisa cria uma base crítica para elaboração da pesquisa científica.

### 3.5 PROCEDIMENTOS METODOLÓGICOS

Para a realização do levantamento de requisitos do sistema utilizou-se da observação direta em ambientes industriais de fabricação de móveis, através da vivência com clientes desse tipo de mercado.

Após a finalização do levantamento de requisitos iniciou-se a busca por conteúdo bibliográfico para aprofundamento do aprendizado da linguagem de programação orientada à objetos *C#* da *Microsoft* e da linguagem de marcação *XML* utilizada para armazenamento de informação do *software*, caracterizando assim este trabalho também como uma pesquisa bibliográfica.

O desenvolvimento do *software* ocorreu em fases, ou seja, primeiramente foi desenvolvido o módulo de cadastro de materiais, seguido do cadastro de lotes. Posteriormente foi desenvolvido o módulo de visualização dos planos de corte e por fim o módulo de cálculo de otimização dos lotes.

Os testes foram realizados a cada finalização de implementação dos módulos, assim, foram realizados testes gradativos ao longo do desenvolvimento do *software* – algo continuamente utilizado na

programação ágil. Após a finalização da implementação do *software* foi realizada uma bateria de testes a fim de garantir o funcionamento do mesmo como um todo.

Até o presente momento, o *software* desenvolvido foi testado apenas em laboratório, ou seja, não foi vendido ou dada a cessão de uso a um cliente final.

## 4 EMPREGABILIDADE DO SOFTWARE

O objetivo deste capítulo é exemplificar a empregabilidade do *software* otimizador de cortes, mostrando sua aplicação dentro de um processo de fabricação de um móvel.

### 4.1 PROJETO

O primeiro passo na fabricação de um móvel, dentro desse contexto, é elaborar o projeto do mesmo. Atualmente existem diversas ferramentas que permitem projetar um móvel de maneira dinâmica. Uma ferramenta muito utilizada nesse meio é a modelagem em três dimensões e o processo de renderização (figuras 7 e 8). O processo de renderização permite criar imagens realistas do móvel projetado, facilitando na venda de um móvel ao cliente final.

Figura 7 - Móvel renderizado.



Fonte: *Software Promob.*



Figura 8 - Móvel renderizado com as portas abertas.



Fonte: *Software Promob.*

Normalmente as ferramentas utilizadas para projetar os móveis possuem a função de gerar listas de peças necessárias para montar os respectivos móveis, onde são informadas as dimensões exatas de cada peça que compõem o móvel. Com base nessa lista são realizados os cortes das chapas de *MDF* na máquina seccionadora. No entanto, com a busca pela redução de desperdícios, baixa dos custos de produção e ganho em produtividade, surgiram os *softwares* otimizadores de corte que são empregados entre o *software* de desenho (projeto) e a máquina seccionadora.

O *software* otimizador de cortes é alimentado com as peças necessárias para montar determinado móvel e o mesmo através de cálculos avançados, faz o posicionamento das peças dentro das chapas de *MDF* com o objetivo de reduzir desperdícios e ganhar tempo na produção, pois a escrita de um plano de corte em uma máquina seccionadora, dependendo da complexibilidade do mesmo pode tornar-se uma operação complicada e demorada.

Neste exemplo, foi projetado um criado mudo que utiliza três tipos de materiais diferentes (com o mesmo acabamento na superfície, porém com espessuras diferentes) assim serão criadas três listas de peças a fim alimentar o otimizador de cortes com as peças necessárias para produção do criado mudo.

Os três materiais utilizados para montagem do criado mudo são:

- MDF Madeirado Carvalho de 18 milímetros de espessura;
- MDF Madeirado Carvalho de 15 milímetros de espessura;
- MDF Madeirado Carvalho de 6 milímetros de espessura.

O termo “madeirado” neste contexto significa que o acabamento da chapa possui veio e esse veio segue sempre em paralelo com o lado maior da chapa. Essa informação é importante, pois dependendo do posicionamento de uma determinada peça no móvel deve-se seguir o veio, por questões de beleza. Um exemplo desta regra dos veios são as tampas das gavetas do criado mudo, onde (conforme é possível ver nas figuras 7 e 8) o veio sempre está na vertical.

As listas de peças necessárias para montagem do móvel, separadas por tipo de material, são apresentadas nos quadros 1, 2 e 3.

Quadro 1 – MDF Madeirado Carvalho 6mm.

<b>MDF Madeirado Carvalho – 6mm</b>				
Descrição	Comprimento	Largura	Quantidade	Rotaciona
Fundo	210,6	341,5	4	Sim

Fonte: produção do próprio autor.

Quadro 2 – MDF Madeirado Carvalho 15mm.

<b>MDF Madeirado Carvalho – 15mm</b>				
Descrição	Comprimento	Largura	Quantidade	Rotaciona
Painel	136,75	296	4	Não

Fonte: produção do próprio autor.

Quadro 3 – MDF Madeirado Carvalho 18mm.

<b>MDF Madeirado Carvalho – 18mm</b>				
Descrição	Comprimento	Largura	Quantidade	Rotaciona
Fundo	263	663	1	Sim
Tras. Gav	200,6	101,25	4	Sim
L. esq. G	350	101,25	4	Sim
L. dir. G	350	101,25	4	Sim
Base	300	400	2	Não
Lateral	663	400	2	Não

Fonte: produção do próprio autor.

## 4.2 OTIMIZAÇÃO

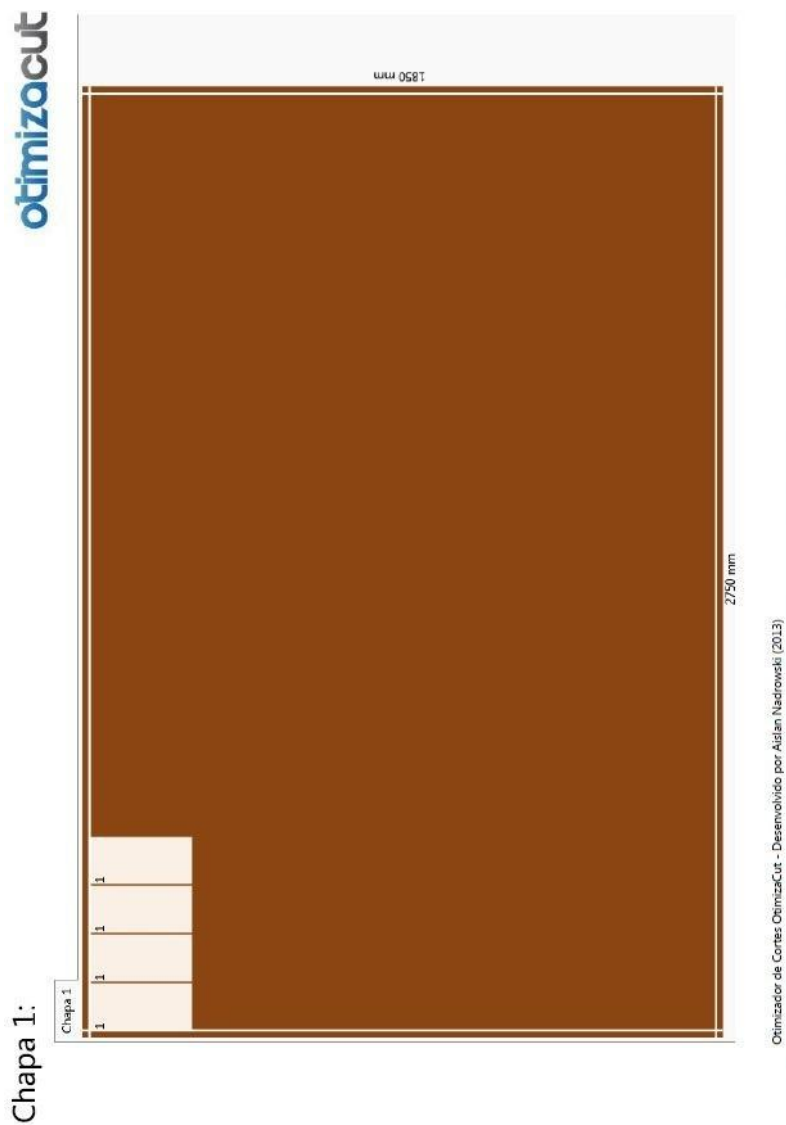
Após a geração das listas de peças é empregada outra ferramenta do processo: o otimizador de cortes. Nele são criados os lotes de peças e posteriormente são realizados os cálculos de posicionamento de peças dentro das chapas de MDF (são gerados os planos de corte). Com os lotes otimizados (calculados) é possível que o usuário emita o relatório de cortes de determinado lote, possibilitando:

- Utilizar as folhas do relatório que contenham os planos de corte como instrumento de auxílio ao operador da máquina seccionadora no momento do corte;
- Identificar os dados estatísticos detalhados por plano de corte a fim de gerar um orçamento ou gerenciar a matéria-prima do estoque.

As figuras 9, 10 e 11 representam os planos de cortes gerados pelo *software* OtimizaCut para fabricação do criado mudo representado na figura 7.



Figura 10 - Plano de corte do material Carvalho 15mm.





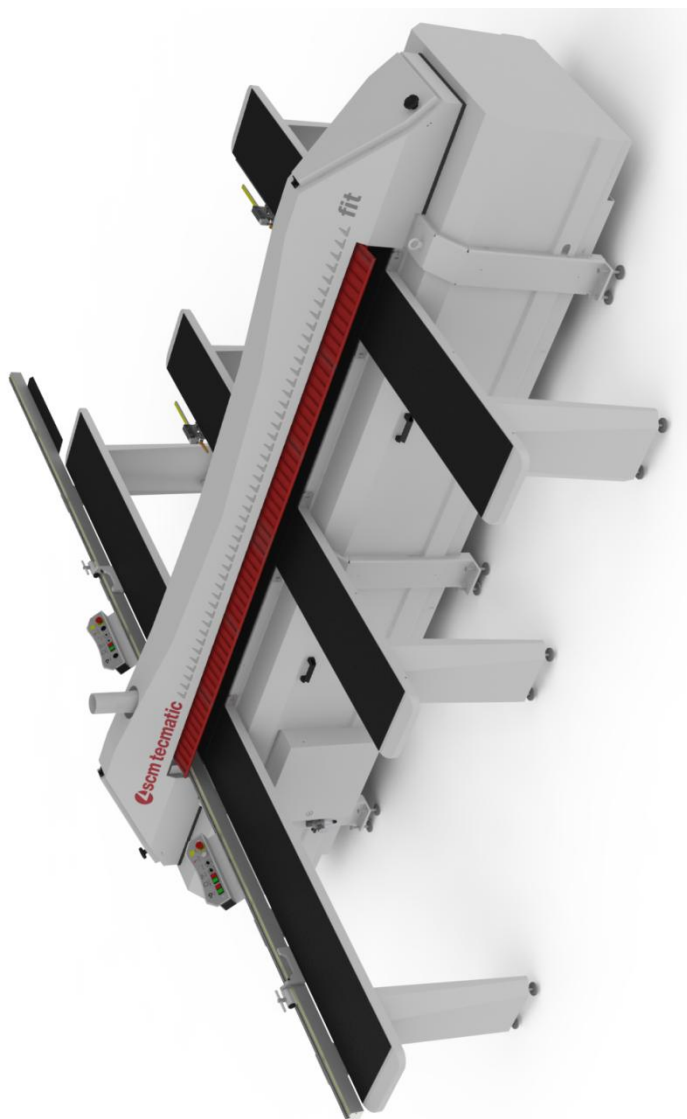
### 4.3 CORTE, USINAGEM E MONTAGEM

O corte de uma chapa de *MDF* pode ser realizado em uma máquina chamada seccionadora (figura 12). Esse tipo de máquina é classificada basicamente como manual ou automática. O modelo representado na figura 12 é um modelo manual produzido pela SCM Tecmatic, cujo modelo é a Fit 2.9. A principal característica da máquina manual é que o operador realiza os cortes de maneira manual, ou seja, ele é responsável por fazer o posicionamento da chapa em cima da mesa da máquina e pressionar o botão para avançar a serra com o intuito de realizar o corte. Já na seccionadora automática, o procedimento de posicionamento da chapa em cima da mesa é realizado pela máquina de maneira automática, bastando ao operador apenas pressionar o botão para iniciar o corte.

O OtimizaCut foi desenvolvido para trabalhar com máquinas seccionadoras manuais, onde o operador da máquina toma posse do plano de corte gerado pelo *software* e realiza os cortes na chapa. O plano de corte gerado pelo *software* apresenta algumas características importantes que facilitam ao operador no momento de corte:

- ✓ Cria o plano de corte pronto para o operador cortar na máquina seccionadora, sem a necessidade do mesmo perder tempo realizando cálculos de posicionamento das peças;
- ✓ Mostra o código em cima da peça com o intuito de facilitar a busca de uma determinada peça dentro do plano de corte;
- ✓ Exibe o comprimento e largura das peças dentro do plano de corte;
- ✓ Evidencia os primeiros cortes realizados na máquina (mostra se os cortes são horizontais ou verticais);
- ✓ Exibe as dimensões do material utilizado.

Figura 12 - Seccionadora Fit 2.9.



Fonte: SCM Tecmatic.



Após o término dos cortes das chapas de *MDF*, o processo pode seguir por diversos caminhos até a finalização do móvel. Por exemplo:

- ✓ Colar fita de borda nas laterais das peças;
- ✓ Usinar as peças para realizar trabalhos específicos;
- ✓ Fazer furação para auxílio na montagem.

## 5 IMPLEMENTAÇÃO DO SOFTWARE

Este capítulo documenta o desenvolvimento do *software* otimizador de cortes OtimizaCut. Não é exibida toda a codificação do *software* neste capítulo, a fim de não tornar o capítulo extenso e maçante ao leitor. Em alguns casos optou-se por explicar o funcionamento do algoritmo através de listas, ao invés de exibir os respectivos códigos desenvolvidos na linguagem de programação *C#*.

### 5.1 MÓDULOS DO SOFTWARE

Com intuito de facilitar o desenvolvimento e manutenção, o *software* é dividido em três módulos:

- Cadastro de matéria-prima e lote;
- Cálculo do plano de corte;
- Visualização de planos de corte.

O fluxo de dados entre esses módulos é feito através da leitura e escrita de arquivos *XML*. Para cada material cadastrado é gerado um arquivo *XML* com suas definições e da mesma forma para cada lote cadastrado é criado um arquivo *XML* com as definições desse lote. Ao editar um material ou lote é editado o respectivo arquivo *XML* utilizado como ferramenta de banco de dados do *software*. Após o término do cálculo é gerado um novo arquivo *XML* com as coordenadas de posicionamento das peças em cima da matéria-prima (chapa) e esse arquivo é lido pelo módulo visualizador de planos de corte, permitindo ao usuário visualizar o plano de corte e gerar o relatório de cortes.

#### 5.1.1 Cadastro de Matéria-prima e Lote

O conteúdo de um arquivo *XML* utilizado no cadastro de uma matéria-prima é exibido abaixo:

```
<?xml version="1.0"?>
<!--Arquivo gerado pelo Software OtimizaCut, desenvolvido
por Aislan Nadrowski - Ano: 2013-->
<material>
  <codigo>10</codigo>
  <descricao>MDF Carvalho 6mm</descricao>
```

```

<comprimento>2750,0</comprimento>
<largura>1850,0</largura>
<espessura>6,0</espessura>
<cor>Carvalho</cor>
<tipo>MDF</tipo>
<rotaciona>Sim</rotaciona>
</material>

```

Os elementos que compõem este arquivo são:

- **<código>**: é um valor alfanumérico que representa o código do material. O valor desse campo é inserido no momento do cadastro do material, durante a edição do material não é possível alterá-lo;
- **<descricao>**: é um valor alfanumérico utilizado para descrever a matéria-prima. É possível editar esse campo após o cadastro da matéria-prima através da tela de edição de materiais;
- **<comprimento>**: é um valor numérico que expressa o comprimento da matéria-prima (medida maior da chapa, que é paralela ao veio do material). É possível editar esse campo após o cadastro da matéria-prima através da tela de edição de materiais;
- **<largura>**: é um valor numérico que expressa a largura da matéria-prima (medida menor da chapa, que é perpendicular ao veio do material). É possível editar esse campo após o cadastro da matéria-prima através da tela de edição de materiais;
- **<espessura>**: é um valor numérico que expressa a espessura da matéria-prima. É possível editar esse campo após o cadastro da matéria-prima através da tela de edição de materiais;
- **<cor>**: é um valor alfanumérico que expressa a cor da matéria-prima. É possível editar esse campo após o

cadastro da matéria-prima através da tela de edição de materiais;

- **<tipo>**: é um valor alfanumérico que expressa o tipo da matéria-prima. É possível editar esse campo após o cadastro da matéria-prima através da tela de edição de materiais;
- **<rotaciona>**: este campo informa se a matéria-prima possui veio ou não (Sim/Não).

Abaixo segue o código desenvolvido em C# para cadastro do arquivo XML referente a matéria-prima:

```
private void salvar_Click(object sender, RoutedEventArgs e)
{
    try
    {
        string caminho = "c:\\dados\\materiais\\" + codigoMat.Text +
        ".xml";

        XmlTextWriter writer = new XmlTextWriter(caminho, null);
        writer.Formatting = Formatting.Indented;

        writer.WriteStartDocument();
        writer.WriteComment("Arquivo gerado pelo Software
        OtimizaCut, desenvolvido por Aislan Nadrowski - Ano: 2013");

        writer.WriteStartElement("material");

        writer.WriteElementString("codigo", codigoMat.Text);
        writer.WriteElementString("descricao", descricaoMat.Text);
        writer.WriteElementString("comprimento",
        comprimentoMat.Text);
        writer.WriteElementString("largura", larguraMat.Text);
        writer.WriteElementString("espessura", espessuraMat.Text);
        writer.WriteElementString("cor", corMat.Text);
        writer.WriteElementString("tipo", tipoMat.Text);

        if (rotacionaMat.SelectedIndex.ToString().Equals("1"))
```

```

        writer.WriteString("rotaciona", "Não");

        if (rotacionaMat.SelectedIndex.ToString().Equals("0"))
            writer.WriteString("rotaciona", "Sim");

        writer.WriteEndElement();
        writer.Close();

        MessageBox.Show("Material cadastrado com sucesso!");
        this.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

O código acima trata-se de um evento que ocorre ao clicar no botão “Cadastrar”, após o preenchimento dos campos relacionados com a matéria-prima. Seguem os passos realizados pelo algoritmo:

- 1) Cria-se um objeto *XmlTextWriter*;
- 2) Aplica-se um tratamento na formatação do documento para que o mesmo seja indentado ao ser criado, através do comando *writer.Formatting = Formatting.Indented*;
- 3) São escritos os elementos dentro do arquivo;
- 4) Exibe uma mensagem de aviso ao usuário, informando que o cadastro foi executado com sucesso.

A rotina de cadastro está inserida dentro de um bloco *try* a fim de verificar se o algoritmo é executado com sucesso. Se ocorrer alguma avaria que impeça esse processo é lançada uma exceção através do bloco *catch*:

```

    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }

```

O algoritmo utilizado para a edição da matéria-prima possui a seguinte sequência de execução após o usuário selecionar o material e clicar em “Editar”:

- 1) Abre-se o arquivo *XML* daquele material;
- 2) Faz-se uma pesquisa dentro deste arquivo e preenche-se os campos da tela (*textbox*);
- 3) O usuário faz as alterações necessárias e clica no botão “Salvar”;
- 4) As informações contidas nos campos de tipo *textbox* são salvas dentro do arquivo *XML*.

A manipulação de um lote possui muitos elementos semelhantes a manipulação de matéria-prima. Abaixo é exibido o conteúdo de um arquivo de lote que contém seis peças cadastradas:

```
<?xml version="1.0"?>
<!--Arquivo gerado pelo Software OtimizaCut, desenvolvido por Aislan
Nadrowski - Ano: 2013 [LOTE FILE]-->
<lote>
  <cabecalho>
    <codigoLote>5</codigoLote>
    <descricaoLote>Criado Mudo - Carv 18mm</descricaoLote>
  </cabecalho>
  <material>
    <codigo>12</codigo>
    <descricao>MDF Carvalho 18mm</descricao>
    <comprimento>2750,0</comprimento>
    <largura>1850,0</largura>
    <espessura>18,0</espessura>
    <cor>Carvalho</cor>
    <tipo>MDF</tipo>
    <rotacional>Sim</rotacional>
  </material>
  <pecas>
    <peca>
      <codigoP>1</codigoP>
      <descricaoP>Fundo</descricaoP>
```

```

    <comprimentoP>263</comprimentoP>
    <larguraP>663</larguraP>
    <quantidadeP>1</quantidadeP>
    <rotacionalP>True</rotacionalP>
  </peca>
  <peca>
    <codigoP>2</codigoP>
    <descricaoP>Traseira Gaveta</descricaoP>
    <comprimentoP>200,6</comprimentoP>
    <larguraP>101,25</larguraP>
    <quantidadeP>4</quantidadeP>
    <rotacionalP>True</rotacionalP>
  </peca>
  <peca>
    <codigoP>3</codigoP>
    <descricaoP>Lat esq gav</descricaoP>
    <comprimentoP>350</comprimentoP>
    <larguraP>101,25</larguraP>
    <quantidadeP>4</quantidadeP>
    <rotacionalP>True</rotacionalP>
  </peca>
  <peca>
    <codigoP>4</codigoP>
    <descricaoP>Lat dir gav</descricaoP>
    <comprimentoP>350</comprimentoP>
    <larguraP>101,25</larguraP>
    <quantidadeP>4</quantidadeP>
    <rotacionalP>True</rotacionalP>
  </peca>
  <peca>
    <codigoP>5</codigoP>
    <descricaoP>Base</descricaoP>
    <comprimentoP>300</comprimentoP>
    <larguraP>400</larguraP>
    <quantidadeP>2</quantidadeP>
    <rotacionalP>False</rotacionalP>
  </peca>
  <peca>
    <codigoP>6</codigoP>
    <descricaoP>Lateral</descricaoP>

```

```

<comprimentoP>663</comprimentoP>
<larguraP>400</larguraP>
<quantidadeP>2</quantidadeP>
<rotacionalP>False</rotacionalP>
</peca>
</pecas>
</lote>

```

Os elementos que compõem este arquivo são:

- **<cabeçalho>**: dentro deste elemento existem outros dois elementos que caracterizam o lote, são eles: **<codigoLote>** e **<descricaoLote>**. Tais campos são informados no momento do cadastro do lote e durante a edição do lote não é possível alterar estas duas informações.
- **<material>**: este elemento possui os mesmos elementos que compõem um arquivo de cadastro de material. Quando o usuário atribui determinado material para o lote em edição, essas informações são salvas automaticamente dentro do arquivo de lote.
- **<peças>**: dentro deste elemento estão as peças cadastradas no lote. Cada peça possui os seguintes elementos característicos:
  - **<codigoP>**: código da peça;
  - **<descricaoP>**: descrição da peça;
  - **<comprimentoP>**: comprimento da peça;
  - **<larguraP>**: largura da peça;
  - **<quantidadeP>**: quantidade da peça;
  - **<rotacionalP>**: informação se a peça pode rotacionar dentro do plano de corte ou não.



A execução do algoritmo de manipulação de um lote é realizada da seguinte maneira:

- 1) O usuário informa qual material deseja utilizar no lote através de um controle do *C#* chamado *combobox*. Neste controle são carregados os materiais cadastrados anteriormente pelo usuário;
- 2) O usuário cadastra as peças que compõem o lote utilizando um controle do *C#* chamado *Datagrid*;
- 3) Para finalizar, basta clicar em “Salvar” para que o *software* crie o arquivo *XML* do lote;
- 4) Após salvar o lote, o usuário pode otimizar o mesmo clicando em “Calcular!”;
- 5) Se o cálculo do plano de corte for realizado com sucesso será exibida uma nova janela com os respectivos planos de corte daquele lote.

### 5.1.2 Cálculo do Plano de Corte

O cálculo de um plano de corte é realizado por uma classe do *software* chamada *Calculo*. Esta classe recebe um parâmetro que é o código do lote a ser cadastrado, busca o arquivo *XML* referente a esse lote e realiza os cálculos de posicionamento de peças com base em alguns fatores:

- Espessura da serra;
- Refilos laterais;
- Tamanho da matéria-prima disponível.

Após o término do cálculo, essa classe cria um novo arquivo *XML* com as coordenadas para posicionamento das peças na chapa (ou para as diversas chapas, dependendo da quantidade de peças informadas durante a edição do lote).

Os cálculos realizados dentro da classe *Calculo* não serão documentados neste trabalho, pois constituem o coração do *software*, sendo requerido sigilo empresarial.

### 5.1.3 Visualização de Planos de Corte

Ao finalizar o cálculo do lote, o *software* cria um arquivo *XML* contendo as coordenadas da matéria-prima e das peças que compõem os

planos de cortes gerados. Este arquivo é lido pelo módulo visualizar de cortes.

O conteúdo do arquivo de lote otimizado, pronto para abertura no visualizador de planos de corte encontra-se no apêndice A.

Os elementos que compõem este arquivo são:

- **<material>**: contém as informações a respeito da matéria-prima utilizada no(s) plano(s) de corte.
- **<planos>**: possui o elemento **<plano>**. Para cada plano de corte é criado um elemento **<plano>** dentro de **<planos>** e para cada peça dentro de um plano é criado um elemento **<peça>**. O conteúdo do elemento **<peça>** é:
  - **<codigoP>**: código da peça;
  - **<descricaoP>**: descrição da peça;
  - **<comprimentoP>**: comprimento da peça dentro do plano de corte (se a peça rotacionar, a largura vira comprimento);
  - **<larguraP>**: largura da peça dentro do plano de corte (se a peça rotacionar, o comprimento vira largura);
  - **<rotacionalP>**: informa se a peça rotacionou dentro do plano de corte durante o cálculo de otimização;
  - **<quantidadeP>**: quantidade de peças cadastradas com determinado código;
  - **<coordenadaPrimeirosCortes>**: elemento utilizado para sinalizar os primeiros cortes da chapa;
  - **<coordenadaPrimeirosCortes2>**: elemento também utilizado para sinalizar os primeiros cortes da chapa;
  - **<coordenadaXP>**: coordenada X de posicionamento da peça dentro do plano de corte;
  - **<coordenadaYP>**: coordenada Y de posicionamento da peça dentro do plano de corte.

## 5.2 CONFIGURAÇÕES

A manipulação dos parâmetros de configuração é realizada através da janela do *software* exibida na figura 13. Os valores dos

parâmetros são salvos dentro de um arquivo chamado *configurações.xml*. Ao carregar essa janela é realizada uma pesquisa dentro deste arquivo e os valores dos respectivos elementos são setados para cada parâmetro. Já o processo de salvamento dos parâmetros após alguma alteração feita pelo usuário do *software* consiste em salvar os valores dentro do arquivo *configurações.xml*.

O conteúdo do arquivo *configurações.xml* é exibido abaixo:

```
<configuracoes>
  <serra>4,2</serra>
  <refilos>15,0</refilos>
  <mostraCompLarg>True</mostraCompLarg>
  <mostraLinhasRefilos>True</mostraLinhasRefilos>
  <mostraCortesIniciais>True</mostraCortesIniciais>
  <orientacaoCortesIniciaisDaChapa>Horizontais</orientacaoCor
tesIniciaisDaChapa>

  <mostraDesenhoNoRelatorio>True</mostraDesenhoNoRelato
rio>

  <mostraEstatisticasNoRelatorio>True</mostraEstatisticasNo
Relatorio>
</configuracoes>
```

Os elementos que compõem este arquivo são:

- **<serra>**: indica a espessura da serra. É informado em milímetros;
- **<refilos>**: indica a largura dos refilos laterais da chapa. É informado em milímetros;
- **<mostraCompLarg>**: se o valor informado for *True* é exibido o comprimento e largura das peças dentro do plano de corte. Se o valor informado for *False* essas informações não são exibidas;
- **<mostraLinhasRefilos>**: se o valor informado for *True* são exibidas as linhas dentro do plano de corte que indicam os

refilos laterais da chapa. Caso contrário, se o valor informado for *False* essas linhas não são exibidas;

- **<mostraCortesIniciais>**: se o valor informado for *True* são exibidas as linhas que indicam os cortes iniciais da chapa. Caso contrário, se o valor informado for *False* essas linhas não são exibidas;
- **<orientacaoCortesIniciaisDaChapa>**: se o valor informado for Horizontais, os primeiros cortes da chapa serão horizontais e se o valor informado por Verticais os primeiros cortes da chapa serão verticais;
- **<mostraDesenhoNoRelatorio>**: se o valor informado for *True* ao gerar um relatório de corte são criadas as páginas com os planos de corte. Caso contrário, se o valor informado for *False* essas páginas não são criadas;
- **<mostraEstatisticasNoRelatorio>**: se o valor informado for *True* ao gerar um relatório de corte são criadas as páginas com os dados estatísticos. Caso contrário, se o valor informado for *False* essas páginas não são criadas.

A rotina desenvolvida em *C#* para salvar os parâmetros dentro do arquivo configurações.xml encontra-se no apêndice B.

Figura 13 - Janela de acesso aos parâmetros de configuração.

OtimizaCut - Aislan Systems

Materiais Lotes **Configurações** Sobre

Parâmetros de configuração

Espessura da serra: 4,2 (mm)

Refilos laterais: 15,0 (mm)

☒ Exibir comprimento e largura no plano de corte

☒ Exibir linhas de refilos laterais no plano de corte

☒ Exibir apenas os cortes do primeiro ciclo (cortes verticais/cortes horizontais)

Orientação dos cortes iniciais da chapa

☒ Horizontais

☐ Verticais

Configurações para geração do relatório de cortes

☒ Exibir desenho do plano de corte no relatório

☒ Exibir estatísticas do plano de corte

Salvar

Fonte: produção do próprio autor.

### 5.3 RELATÓRIO

Esta seção mostra o código implementado para geração do relatório. Basicamente, o relatório divide-se em três partes:

- Capa;
- Páginas de planos de corte;
- Páginas de estatísticas.

Primeiramente é realizada uma busca no arquivo configurações.xml para verificar se os parâmetros indicados na figura 14 estão marcados, através do código-fonte exibido abaixo.

Figura 14 - Parâmetros do relatório.

Configurações para geração do relatório de cortes

- ☒ Exibir desenho do plano de corte no relatório
- ☒ Exibir estatísticas do plano de corte

Fonte: produção do próprio autor.

```
private void gerarPlano(object sender, RoutedEventArgs e)
{
    try
    {
        //Carrega valores de configuracoes

        string caminhoConf =
"C:\\dados\\configuracoes\\configuracoes.xml";

        XmlTextReader arqConf = new
XmlTextReader(caminhoConf);

        bool mostraPlanosDeCorteNoRelatorioBool = false;
        bool mostraEstatisticasNoRelatorioBool = false;

        while (arqConf.Read())
        {
            if (arqConf.NodeType == XmlNodeType.Element)
            {
                if (arqConf.Name == "mostraDesenhoNoRelatorio")
                {
                    if (arqConf.ReadString().Equals("True"))
                        mostraPlanosDeCorteNoRelatorioBool = true;

                    if (arqConf.ReadString().Equals("False"))
                        mostraPlanosDeCorteNoRelatorioBool = false;
                }
            }
        }
    }
}
```

```

    if (arqConf.Name == "mostraEstatisticasNoRelatorio")
    {

        if (arqConf.ReadString().Equals("True"))
            mostraEstatisticasNoRelatorioBool = true;

        if (arqConf.ReadString().Equals("False"))
            mostraEstatisticasNoRelatorioBool = false;
    }
}

arqConf.Close();

```

Posteriormente são carregados os valores referentes ao material e lote calculado, para gerar a capa do relatório.

Se a variável `mostraPlanosDeCorteNoRelatorioBool` for igual a `true` significa que o parâmetro “Exibir desenho do plano de corte no relatório” está habilitado (figura 14), assim, a lógica responsável pela função é executada criando as páginas do relatório com os planos de corte gerados. De maneira semelhante, se a variável `mostraEstatisticasNoRelatorioBool` for igual a `true` significa que o parâmetro “Exibir estatísticas do plano de corte” está habilitado (figura 14), assim, a lógica responsável é executada criando as páginas do relatório com os planos de corte gerados. Tais códigos são exibidos no apêndice C.

## 6 CONCLUSÃO

A elaboração desta pesquisa resultou no desenvolvimento de um *software* otimizador de cortes. As funções implementadas neste *software* foram o cadastro de matéria-prima e cadastro de lote, cálculo do lote para geração dos planos de corte e desenvolvimento do módulo de visualização de planos de corte. Tais implementações foram viabilizadas através da execução dos objetivos específicos propostos no início da pesquisa, que foram: estudo da linguagem de programação orientada à objetos *C#* da Microsoft, estudo acerca de arquivos *XML* que foram utilizados para armazenamento dos dados do *software* e levantamento dos requisitos do sistema através da observação direta do autor dentro do ambiente industrial moveleiro.

Quanto às maiores dificuldades encontradas pelo autor durante o processo de pesquisa foram: aprendizado da linguagem de programação *C#*, pois o mesmo não possuía conhecimento nesta linguagem antes de iniciar a pesquisa e com relação às regras de cálculo para posicionamento das peças dentro da matéria-prima, pois são diversas situações que influenciam no posicionamento de uma peça, como:

- Obedecer o sentido de veio ou não (rotacionar a peça dentro do plano de corte ou não);
- Criar sequências de corte que sejam possíveis de ser realizadas na máquina seccionadora;
- Regras gerais para optar se a sobra de uma determinada tira deve ser preferencialmente vertical ou horizontal;
- Regras para posicionamento de peças de dimensões idênticas de maneira agrupada dentro do plano de corte ou não, dependendo da situação.

Optou-se por não divulgar os códigos-fonte implementados na linguagem *C#* responsáveis por realizar o cálculo de posicionamento das peças, pois trata-se do coração do *software* e consequentemente, de um segredo comercial. De fato, certas situações são matematicamente previsíveis e fáceis de concluir qual é o melhor método de posicionamento de uma peça dentro do plano de corte, no entanto, existem alguns fatores externos que influenciam nesse posicionamento,



como o tipo da máquina que o usuário irá utilizar para efetuar os cortes e até mesmo o tipo de móvel que o cliente produz.

Existem algumas funções cuja implementação é sugerida para trabalhos futuros. Tais funções são:

- Integração do *software* otimizador de cortes com máquinas seccionadoras automáticas;
- Possibilidade de edição do plano de corte manualmente após o cálculo do lote;
- Módulo para cálculo de orçamento, com base na quantidade de cortes executados em determinado plano de corte e preço da chapa utilizada.

Os testes do *software* foram realizados em laboratório durante o desenvolvimento do mesmo. Buscou-se executar tais testes utilizando lotes de peças que produzem móveis reais e chapas com dimensões de padrões de mercado, com o intuito de aproximar-se ao máximo da utilização deste tipo de sistema em um ambiente industrial moveleiro. As funções do *software* anteriormente mencionadas foram implementadas de maneira separada, ou seja, primeiramente foi codificado o módulo de cadastro de materiais e lotes, posteriormente o módulo de visualização de planos de corte e o módulo de cálculo de otimização. Ao término de cada implementação de um módulo foram realizadas baterias de testes com o intuito de corrigir possíveis problemas antes do término da codificação do *software* como um todo. Após a finalização da implementação dos módulos foram realizados testes de fluxos de dados entre os três módulos do *software*.

Com relação aos pontos positivos da pesquisa pode-se citar: aprendizado da linguagem de programação *C#* e manipulação de arquivos *XML*, possibilitando desenvolver trabalhos futuros utilizando esta tecnologia. Outro fator positivo é que existe mercado para os serviços de tecnologia da informação mesmo dentro de pequenas indústrias de móveis ou a serviço de pequenos marceneiros e a demanda vem crescendo constantemente com a pressão que o mercado exerce sobre esse tipo de produtor com relação à redução de custos e redução de tempo de processo para fabricação de determinado item. Desta maneira, a atual pesquisa possibilitou o desenvolvimento de um *software* que pode vir a ser comercializado num futuro próximo.

Portanto, com relação ao problema da pesquisa que foi identificado no início deste trabalho, conclui-se que sim, é possível desenvolver um *software* para gestão de matéria-prima e que seja um agente redutor de tempo no processo de produção de determinado móvel. No entanto, são inúmeras as variáveis que podem influenciar neste resultado e principalmente com relação ao cálculo de posicionamento de peças conclui-se, que não é possível afirmar que determinado plano de corte ficou ruim, bom ou ótimo para todos os clientes usuários de um *software* otimizador de cortes. Pode-se buscar o bom ou ótimo na grande maioria dos clientes que utilizam o sistema, mas as variáveis externas ao *software* e inerentes ao processo de fabricação de cada cliente também ajudam a definir o nível de satisfação deste cliente.

## REFERÊNCIAS

ALMEIDA, Maurício Barcellos. **Uma introdução ao XML, sua utilização na Internet e alguns conceitos complementares**. Ci. Inf., Brasília , v. 31, n. 2, Aug. 2002 . Available from <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S0100-19652002000200001&lng=en&nrm=iso](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0100-19652002000200001&lng=en&nrm=iso)>. access on 25 Aug. 2013. <http://dx.doi.org/10.1590/S0100-19652002000200001>.

GIL, A.C. **Como elaborar projetos de pesquisa**. 4ª Ed. São Paulo. Editora Atlas S.A. 2007.

MARCONI, M. A.; LAKATOS, E. M. **Técnicas de pesquisa**. 7ª Ed. São Paulo. Editora Atlas S.A. 2009.

MORESI, E. **Metodologia da pesquisa**. Dissertação. Universidade Católica de Brasília – UCB. Brasília. 2003.

OFICINA DE CASA. **Como montar um plano de corte**. Disponível em: <http://oficinadecasa.com.br/saber/como-que-e-isso/como-montar-um-plano-de-corte/>. Acesso em: 25 ago. 2013.

SHARP, J. **Microsoft Visual C# 2008: passo a passo**. Tradução: Edson Furmankiewicz. 1ª Ed. Porto Alegre. Editora Bookman. 2008.

## APÊNDICE A – XML DE UM PLANO OTIMIZADO

```

<?xml version="1.0"?>
<!--Arquivo gerado pelo Software OtimizaCut, desenvolvido por Aislan Nadrowski -
Ano: 2013 [PLANO OTIMIZADO]-->
<otimizado>
  <material>
    <codigo>12</codigo>
    <descricao>MDF Carvalho 18mm</descricao>
    <comprimento>2750,0</comprimento>
    <largura>1850,0</largura>
    <espessura>18,0</espessura>
    <cor>Carvalho</cor>
    <tipo>MDF</tipo>
    <rotaciona>Sim</rotaciona>
  </material>
  <planos>
    <plano>
      <peca>
        <codigoP>6</codigoP>
        <descricaoP>Lateral</descricaoP>
        <comprimentoP>663</comprimentoP>
        <larguraP>400</larguraP>
        <rotacionaP>Nao</rotacionaP>
        <quantidadeP>2</quantidadeP>
        <coordenadaXP>15</coordenadaXP>
        <coordenadaYP>15</coordenadaYP>
        <coordenadaPrimeirosCortes>15</coordenadaPrimeirosCortes>
        <coordenadaPrimeirosCortes2>415</coordenadaPrimeirosCortes2>
      </peca>
      <peca>
        <codigoP>6</codigoP>
        <descricaoP>Lateral</descricaoP>
        <comprimentoP>663</comprimentoP>
        <larguraP>400</larguraP>
        <rotacionaP>Nao</rotacionaP>
        <quantidadeP>2</quantidadeP>
        <coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
        <coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
        <coordenadaXP>15</coordenadaXP>
        <coordenadaYP>682,2</coordenadaYP>
      </peca>
      <peca>
        <codigoP>1</codigoP>
        <descricaoP>Fundo</descricaoP>
        <comprimentoP>263</comprimentoP>

```

```

<larguraP>663</larguraP>
<rotacionalP>Nao</rotacionalP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>15</coordenadaXP>
<coordenadaYP>1349,4</coordenadaYP>
</peca>
<peca>
<codigoP>5</codigoP>
<descricaoP>Base</descricaoP>
<comprimentoP>300</comprimentoP>
<larguraP>400</larguraP>
<rotacionalP>Nao</rotacionalP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>15</coordenadaXP>
<coordenadaYP>1349,4</coordenadaYP>
</peca>
<peca>
<codigoP>5</codigoP>
<descricaoP>Base</descricaoP>
<comprimentoP>300</comprimentoP>
<larguraP>400</larguraP>
<rotacionalP>Nao</rotacionalP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>15</coordenadaXP>
<coordenadaYP>1349,4</coordenadaYP>
</peca>
<peca>
<codigoP>3</codigoP>
<descricaoP>Lat esq gav</descricaoP>
<comprimentoP>350</comprimentoP>
<larguraP>101,25</larguraP>
<rotacionalP>Nao</rotacionalP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>419,2</coordenadaXP>
<coordenadaYP>1349,4</coordenadaYP>
</peca>
<peca>
<codigoP>3</codigoP>
<descricaoP>Lat esq gav</descricaoP>

```

```

<comprimentoP>350</comprimentoP>
<larguraP>101,25</larguraP>
<rotacionaP>Nao</rotacionaP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>524,65</coordenadaXP>
<coordenadaYP>1349,4</coordenadaYP>
</peca>
<peca>
  <codigoP>3</codigoP>
  <descricaoP>Lat esq gav</descricaoP>
  <comprimentoP>350</comprimentoP>
  <larguraP>101,25</larguraP>
  <rotacionaP>Nao</rotacionaP>
  <quantidadeP>2</quantidadeP>
  <coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
  <coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
  <coordenadaXP>15</coordenadaXP>
  <coordenadaYP>1703,6</coordenadaYP>
</peca>
<peca>
  <codigoP>3</codigoP>
  <descricaoP>Lat esq gav</descricaoP>
  <comprimentoP>350</comprimentoP>
  <larguraP>101,25</larguraP>
  <rotacionaP>Nao</rotacionaP>
  <quantidadeP>2</quantidadeP>
  <coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
  <coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
  <coordenadaXP>120,45</coordenadaXP>
  <coordenadaYP>1703,6</coordenadaYP>
</peca>
<peca>
  <codigoP>4</codigoP>
  <descricaoP>Lat dir gav</descricaoP>
  <comprimentoP>350</comprimentoP>
  <larguraP>101,25</larguraP>
  <rotacionaP>Nao</rotacionaP>
  <quantidadeP>2</quantidadeP>
  <coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
  <coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
  <coordenadaXP>225,9</coordenadaXP>
  <coordenadaYP>1703,6</coordenadaYP>
</peca>
<peca>
  <codigoP>4</codigoP>

```

```

<descricaoP>Lat dir gav</descricaoP>
<comprimentoP>350</comprimentoP>
<larguraP>101,25</larguraP>
<rotacionalP>Nao</rotacionalP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>15</coordenadaXP>
<coordenadaYP>2057,8</coordenadaYP>
</peca>
<peca>
<codigoP>4</codigoP>
<descricaoP>Lat dir gav</descricaoP>
<comprimentoP>350</comprimentoP>
<larguraP>101,25</larguraP>
<rotacionalP>Nao</rotacionalP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>120,45</coordenadaXP>
<coordenadaYP>2057,8</coordenadaYP>
</peca>
<peca>
<codigoP>4</codigoP>
<descricaoP>Lat dir gav</descricaoP>
<comprimentoP>350</comprimentoP>
<larguraP>101,25</larguraP>
<rotacionalP>Nao</rotacionalP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>225,9</coordenadaXP>
<coordenadaYP>2057,8</coordenadaYP>
</peca>
<peca>
<codigoP>2</codigoP>
<descricaoP>Traseira Gaveta</descricaoP>
<comprimentoP>200,6</comprimentoP>
<larguraP>101,25</larguraP>
<rotacionalP>Nao</rotacionalP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>15</coordenadaXP>
<coordenadaYP>2412</coordenadaYP>
</peca>
<peca>

```

```

<codigoP>2</codigoP>
<descricaoP>Traseira Gaveta</descricaoP>
<comprimentoP>200,6</comprimentoP>
<larguraP>101,25</larguraP>
<rotacionalP>Nao</rotacionalP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>120,45</coordenadaXP>
<coordenadaYP>2412</coordenadaYP>
</peca>
<peca>
<codigoP>2</codigoP>
<descricaoP>Traseira Gaveta</descricaoP>
<comprimentoP>200,6</comprimentoP>
<larguraP>101,25</larguraP>
<rotacionalP>Nao</rotacionalP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>225,9</coordenadaXP>
<coordenadaYP>2412</coordenadaYP>
</peca>
<peca>
<codigoP>2</codigoP>
<descricaoP>Traseira Gaveta</descricaoP>
<comprimentoP>200,6</comprimentoP>
<larguraP>101,25</larguraP>
<rotacionalP>Nao</rotacionalP>
<quantidadeP>2</quantidadeP>
<coordenadaPrimeirosCortes>0</coordenadaPrimeirosCortes>
<coordenadaPrimeirosCortes2>0</coordenadaPrimeirosCortes2>
<coordenadaXP>15</coordenadaXP>
<coordenadaYP>2616,8</coordenadaYP>
</peca>
</plano>
</planos>
</otimizado>

```



## APÊNDICE B – CÓDIGO SALVAR PARÂMETROS

```
string caminho = @"c:\dados\configuracoes\configuracoes.xml";

XmlDocument doc = new XmlDocument();
doc.Load(caminho);

XmlNodeList trocarValorEspessuraSerra =
doc.GetElementsByTagName("serra");

for (int i = 0; i < trocarValorEspessuraSerra.Count; i++)
{
    trocarValorEspessuraSerra[i].InnerText = espessuraSerra.Text;
}

XmlNodeList trocarValorRefilosLaterais =
doc.GetElementsByTagName("refilos");

for (int i = 0; i < trocarValorRefilosLaterais.Count; i++)
{
    trocarValorRefilosLaterais[i].InnerText = refilosLaterais.Text;
}

XmlNodeList mostraComprimentoLargura =
doc.GetElementsByTagName("mostraCompLarg");

for (int i = 0; i < mostraComprimentoLargura.Count; i++)
{
    if (mostraComLarg.IsChecked == true)
        mostraComprimentoLargura[i].InnerText = "True";

    if (mostraComLarg.IsChecked == false)
        mostraComprimentoLargura[i].InnerText = "False";
}

XmlNodeList mostraRefilosLaterais =
doc.GetElementsByTagName("mostraLinhaRefilos");

for (int i = 0; i < mostraRefilosLaterais.Count; i++)
{
    if (mostraRefilos.IsChecked == true)
        mostraRefilosLaterais[i].InnerText = "True";
```

```

        if (mostraRefilos.IsChecked == false)
            mostraRefilosLaterais[i].InnerText = "False";
    }

XmlNodeList mostraCortesIniciaisNaChapa =
doc.GetElementsByTagName("mostraCortesIniciais");

for (int i = 0; i < mostraCortesIniciaisNaChapa.Count; i++)
{
    if (mostraCortesIniciais.IsChecked == true)
        mostraCortesIniciaisNaChapa[i].InnerText = "True";

    if (mostraCortesIniciais.IsChecked == false)
        mostraCortesIniciaisNaChapa[i].InnerText = "False";
}

XmlNodeList orientacaoDosCortesIniciais =
doc.GetElementsByTagName("orientacaoCortesIniciaisDaChapa");

for (int i = 0; i < mostraCortesIniciaisNaChapa.Count; i++)
{
    if (corteHorizontalSetado.IsChecked == true)
        orientacaoDosCortesIniciais[i].InnerText = "Horizontais";

    if (corteVerticalSetado.IsChecked == true)
        orientacaoDosCortesIniciais[i].InnerText = "Verticais";
}

XmlNodeList mostraDesenhoNoRelatorio =
doc.GetElementsByTagName("mostraDesenhoNoRelatorio");

for (int i = 0; i < mostraCortesIniciaisNaChapa.Count; i++)
{
    if (mostraDesenho.IsChecked == true)
        mostraDesenhoNoRelatorio[i].InnerText = "True";

    if (mostraDesenho.IsChecked == false)
        mostraDesenhoNoRelatorio[i].InnerText = "False";
}

XmlNodeList mostraEstatisticasNoRelatorio =
doc.GetElementsByTagName("mostraEstatisticasNoRelatorio");

for (int i = 0; i < mostraCortesIniciaisNaChapa.Count; i++)
{

```

```
if (mostraEstatisticas.IsChecked == true)
    mostraEstatisticasNoRelatorio[i].InnerText = "True";

if (mostraEstatisticas.IsChecked == false)
    mostraEstatisticasNoRelatorio[i].InnerText = "False";
}

doc.Save(caminho);

MessageBox.Show("Parâmetros salvos com sucesso!");
```

## APÊNDICE C – CÓDIGO PARA GERAR RELATÓRIO

```
//Carrega valores do material e lote para mostrar na capa do relatório
LVData informacoesParaRelatorio = new LVData();
string descricaoLote = "";
string codigoLote = "";
string caminhoEst = "C:\\dados\\planos_otimizados\\estatisticas.xml";

XmlTextReader leitorCapa = new XmlTextReader(caminhoEst);

while (leitorCapa.Read())
{
    if (leitorCapa.NodeType == XmlNodeType.Element)
    {
        if (leitorCapa.Name == "codigo")
            informacoesParaRelatorio.Codigo = leitorCapa.ReadString();

        if (leitorCapa.Name == "descricao")
            informacoesParaRelatorio.Descricao =
                leitorCapa.ReadString();

        if (leitorCapa.Name == "comprimento")
            informacoesParaRelatorio.Comprimento =
                leitorCapa.ReadString();

        if (leitorCapa.Name == "largura")
            informacoesParaRelatorio.Largura =
                leitorCapa.ReadString();

        if (leitorCapa.Name == "espessura")
            informacoesParaRelatorio.Espessura =
                leitorCapa.ReadString();

        if (leitorCapa.Name == "cor")
            informacoesParaRelatorio.Cor = leitorCapa.ReadString();

        if (leitorCapa.Name == "tipo")
            informacoesParaRelatorio.Tipo = leitorCapa.ReadString();

        if (leitorCapa.Name == "descricaoLote")
            descricaoLote = leitorCapa.ReadString();

        if (leitorCapa.Name == "codigoLote")
```

```

        codigoLote = leitorCapa.ReadString();

    }

}

string codigoMatString = informacoesParaRelatorio.Codigo;
string descricaoMatString = informacoesParaRelatorio.Descricao;
string comprimentoMatString = informacoesParaRelatorio.Comprimento;
string larguraMatString = informacoesParaRelatorio.Largura;
string espessuraMatString = informacoesParaRelatorio.Espessura;
string corMatString = informacoesParaRelatorio.Cor;
string tipoMatString = informacoesParaRelatorio.Tipo;

leitorCapa.Close();

//INFORMAÇÕES DA CAPA DO RELATÓRIO.
// Cria objeto PrintDialog
PrintDialog pd = new PrintDialog();
if (pd.ShowDialog() != true) return;

// Cria documento
FixedDocument document = new FixedDocument();
document.DocumentPaginator.PageSize = new Size(pd.PrintableAreaWidth,
pd.PrintableAreaHeight);

//Cria pagina da capa
FixedPage page1 = new FixedPage();
page1.Width = document.DocumentPaginator.PageSize.Width;
page1.Height = document.DocumentPaginator.PageSize.Height;

//Titulo da capa
TextBlock tituloCapa = new TextBlock();
tituloCapa.Text = "Relatório de Cortes (OtimizaCut)";
tituloCapa.FontWeight = FontWeights.UltraBold;
tituloCapa.FontSize = 30;
tituloCapa.Margin = new Thickness(300, 30, 0, 0);
page1.Children.Add(tituloCapa);

TextBlock propriedadesMatCapa = new TextBlock();
propriedadesMatCapa.Text = "Propriedades do material utilizado: ";
propriedadesMatCapa.FontWeight = FontWeights.UltraBold;
propriedadesMatCapa.FontSize = 16;
propriedadesMatCapa.Margin = new Thickness(96, 100, 0, 0);
page1.Children.Add(propriedadesMatCapa);

TextBlock codigoMatCapa = new TextBlock();

```

```

codigoMatCapa.Text = "Código: " + codigoMatString;
codigoMatCapa.FontSize = 14;
codigoMatCapa.Margin = new Thickness(96, 120, 0, 0);
page1.Children.Add(codigoMatCapa);

```

```

TextBlock descricaoMatCapa = new TextBlock();
descricaoMatCapa.Text = "Descrição: " + descricaoMatString;
descricaoMatCapa.FontSize = 14;
descricaoMatCapa.Margin = new Thickness(96, 135, 0, 0);
page1.Children.Add(descricaoMatCapa);

```

```

TextBlock comprimentoMatCapa = new TextBlock();
comprimentoMatCapa.Text = "Comprimento: " + comprimentoMatString;
comprimentoMatCapa.FontSize = 14;
comprimentoMatCapa.Margin = new Thickness(96, 150, 0, 0);
page1.Children.Add(comprimentoMatCapa);

```

```

TextBlock larguraMatCapa = new TextBlock();
larguraMatCapa.Text = "Largura: " + larguraMatString;
larguraMatCapa.FontSize = 14;
larguraMatCapa.Margin = new Thickness(96, 165, 0, 0);
page1.Children.Add(larguraMatCapa);

```

```

TextBlock espessuraMatCapa = new TextBlock();
espessuraMatCapa.Text = "Espessura: " + espessuraMatString;
espessuraMatCapa.FontSize = 14;
espessuraMatCapa.Margin = new Thickness(96, 180, 0, 0);
page1.Children.Add(espessuraMatCapa);

```

```

TextBlock corMatCapa = new TextBlock();
corMatCapa.Text = "Cor: " + corMatString;
corMatCapa.FontSize = 14;
corMatCapa.Margin = new Thickness(96, 195, 0, 0);
page1.Children.Add(corMatCapa);

```

```

TextBlock tipoMatCapa = new TextBlock();
tipoMatCapa.Text = "Tipo: " + tipoMatString;
tipoMatCapa.FontSize = 14;
tipoMatCapa.Margin = new Thickness(96, 210, 0, 0);
page1.Children.Add(tipoMatCapa);

```

```

Image finalImage = new Image();
finalImage.Margin = new Thickness(850, 60, 0, 0);
BitmapImage logo = new BitmapImage();

```

```

logo.BeginInit();
logo.UriSource = new Uri("C:/dados/imagens/logo.jpg");
logo.EndInit();
finalImage.Source = logo;
page1.Children.Add(finalImage);

```

```

TextBlock propriedadesLoteCapa = new TextBlock();
propriedadesLoteCapa.Text = "Propriedades do lote otimizado: ";
propriedadesLoteCapa.FontWeight = FontWeights.UltraBold;
propriedadesLoteCapa.FontSize = 16;
propriedadesLoteCapa.Margin = new Thickness(96, 240, 0, 0);
page1.Children.Add(propriedadesLoteCapa);

```

```

TextBlock codigoLoteCapa = new TextBlock();
codigoLoteCapa.Text = "Código: " + codigoLote;
codigoLoteCapa.FontSize = 14;
codigoLoteCapa.Margin = new Thickness(96, 260, 0, 0);
page1.Children.Add(codigoLoteCapa);

```

```

TextBlock descricaoLoteCapa = new TextBlock();
descricaoLoteCapa.Text = "Descrição: " + descricaoLote;
descricaoLoteCapa.FontSize = 14;
descricaoLoteCapa.Margin = new Thickness(96, 275, 0, 0);
page1.Children.Add(descricaoLoteCapa);

```

```

TextBlock rodapeCapa = new TextBlock();
rodapeCapa.Text = "Otimizador de Cortes OtimizaCut - Desenvolvido por  
Aislan Nadrowski (2013)";
rodapeCapa.FontSize = 12;
rodapeCapa.Margin = new Thickness(96, 750, 0, 0);
page1.Children.Add(rodapeCapa);

```

```

// Adiciona página da capa no documento
PageContent page1Content = new PageContent();
((IAddChild)page1Content).AddChild(page1);
document.Pages.Add(page1Content);

```

Se a variável `mostraPlanosDeCorteNoRelatorioBool` for igual a `true` significa que o parâmetro “Exibir desenho do plano de corte no relatório” está habilitado (figura 51), assim, a lógica abaixo é executada criando as páginas do relatório com os planos de corte gerados:

```

//PLANOS DE CORTE (INSERÇÃO DOS DESENHOS NO RELATÓRIO).

```

```

if (mostraPlanosDeCorteNoRelatorioBool == true)

```

{

```
int quantidadeDeTabs = tabcontrolView.Items.Count;
```

```
for (int i = 0; i < quantidadeDeTabs; i++)
```

{

```
tabcontrolView.SelectedIndex = i;
```

```
string nomeArquivoImagem = i.ToString() + ".png";
```

```
int numeroDaChapa = i + 1;
```

```
string mensagemDeOk = "Coordenadas geradas para a chapa "  
+ numeroDaChapa.ToString() + " com sucesso. Pressione Ok!";
```

```
MessageBox.Show(mensagemDeOk);
```

```
SaveUsingEncoder(nomeArquivoImagem, tabcontrolView, new  
PngBitmapEncoder());
```

}

```
int quantidadeDeTabsMais1 = 0;
```

```
for (int i = 0; i < quantidadeDeTabs; i++)
```

{

```
quantidadeDeTabsMais1 = i + 1;
```

```
//Insere titulo na pagina do desenho
```

```
FixedPage page2 = new FixedPage();
```

```
page2.Width = document.DocumentPaginator.PageSize.Width;
```

```
page2.Height = document.DocumentPaginator.PageSize.Height;
```

```
TextBlock txt1 = new TextBlock();
```

```
txt1.Text = "Chapa " + quantidadeDeTabsMais1 + ":";
```

```
txt1.FontSize = 30;
```

```
txt1.Margin = new Thickness(20, 10, 0, 0);
```

```
page2.Children.Add(txt1);
```

```
PageContent page2Content = new PageContent();
```

```
((IAddChild)page2Content).AddChild(page2);
```

```
document.Pages.Add(page2Content);
```

```
string auxUriPlano = @"C:/dados/imagens/relatorio/" +  
i.ToString() + ".png";
```

```
Image planoDeCorteImage = new Image();
```

```
planoDeCorteImage.Margin = new Thickness(96, 40, 0, 0);
```

```
BitmapImage planoDeCorte1 = new BitmapImage();
```

```
planoDeCorte1.BeginInit();
```

```
planoDeCorte1.UriSource = new Uri(auxUriPlano);
```

```
planoDeCorte1.EndInit();
```

```
planoDeCorteImage.Source = planoDeCorte1;
```

```
page2.Children.Add(planoDeCorteImage);
```



```

Image logoEscrito = new Image();
logoEscrito.Margin = new Thickness(880, 10, 0, 0);
BitmapImage logoEscrito1 = new BitmapImage();
logoEscrito1.BeginInit();
logoEscrito1.UriSource = new
Uri("C:/dados/imagens/logo_escrito.jpg");
logoEscrito1.EndInit();
logoEscrito.Source = logoEscrito1;
page2.Children.Add(logoEscrito);
TextBlock rodapePaginaPlano = new TextBlock();
rodapePaginaPlano.Text = "Otimizador de Cortes OtimizaCut -
Desenvolvido por Aislan Nadrowski (2013)";
rodapePaginaPlano.FontSize = 12;
rodapePaginaPlano.Margin = new Thickness(96, 750, 0, 0);
page2.Children.Add(rodapePaginaPlano);
}
}

```

Se a variável `mostraEstatisticasNoRelatorioBool` for igual a `true` significa que o parâmetro “Exibir estatísticas do plano de corte” está habilitado (figura 51), assim, a lógica abaixo é executada criando as páginas do relatório com os planos de corte gerados:

//ESTATÍSTICAS DOS PLANOS.

```

if (mostraEstatisticasNoRelatorioBool == true)
{
    string caminhoPlano = "c:\\dados\\planos_otimizados\\" +
    codigoLote + ".xml";
    XmlDocument doc = new XmlDocument();
    doc.Load(caminhoPlano);

    int contadorChapasParaRelatorio = 0;
    int contadorPecasNaChapa = 0;

    double areaDaPeca = 0;
    double areaTotalDasPecasNaChapa = 0;

    XmlNodeList listaPlano =
    doc.SelectNodes("otimizado/planos/plano");

    foreach (XmlNode node1 in listaPlano)
    {

```

```
FixedPage page3 = new FixedPage();
page3.Width = document.DocumentPaginator.PageSize.Width;
page3.Height = document.DocumentPaginator.PageSize.Height;
contadorChapasParaRelatorio++;
```

```
TextBlock codigoListaPecas = new TextBlock();
codigoListaPecas.Text = "CÓDIGO";
codigoListaPecas.FontSize = 12;
codigoListaPecas.Margin = new Thickness(96, 130, 0, 0);
codigoListaPecas.FontWeight = FontWeights.UltraBold;
page3.Children.Add(codigoListaPecas);
```

```
TextBlock descricaoListaPecas = new TextBlock();
descricaoListaPecas.Text = "DESCRIÇÃO";
descricaoListaPecas.FontSize = 12;
descricaoListaPecas.Margin = new Thickness(180, 130, 0, 0);
descricaoListaPecas.FontWeight = FontWeights.UltraBold;
page3.Children.Add(descricaoListaPecas);
```

```
TextBlock comprimentoListaPecas = new TextBlock();
comprimentoListaPecas.Text = "COMPRIMENTO";
comprimentoListaPecas.FontSize = 12;
comprimentoListaPecas.Margin = new Thickness(350, 130, 0, 0);
comprimentoListaPecas.FontWeight = FontWeights.UltraBold;
page3.Children.Add(comprimentoListaPecas);
```

```
TextBlock larguraListaPecas = new TextBlock();
larguraListaPecas.Text = "LARGURA";
larguraListaPecas.FontSize = 12;
larguraListaPecas.Margin = new Thickness(500, 130, 0, 0);
larguraListaPecas.FontWeight = FontWeights.UltraBold;
page3.Children.Add(larguraListaPecas);
```

```
TextBlock quantidadeListaPecas = new TextBlock();
quantidadeListaPecas.Text = "QUANTIDADE";
quantidadeListaPecas.FontSize = 12;
quantidadeListaPecas.Margin = new Thickness(610, 130, 0, 0);
quantidadeListaPecas.FontWeight = FontWeights.UltraBold;
page3.Children.Add(quantidadeListaPecas);
```

```
TextBlock areaDaPecaListaPecas = new TextBlock();
areaDaPecaListaPecas.Text = "ÁREA (M²)";
areaDaPecaListaPecas.FontSize = 12;
areaDaPecaListaPecas.Margin = new Thickness(750, 130, 0, 0);
```

```

areaDaPecaListaPecas.FontWeight = FontWeights.UltraBold;
page3.Children.Add(areaDaPecaListaPecas);

XmlNodeList listaPeca = node1.SelectNodes("peca");

var listaDePecaParaRelatorio = new List<PecaParaPlano>
{
};

var lista = new List<PecaParaPlano>
{
};

int posLinha = 145;

foreach (XmlNode node in listaPeca)
{
    XmlElement elementoPeca = (XmlElement)node;

    string codigoP =
        elementoPeca.GetElementsByTagName("codigoP")[0].InnerText;
    string descricaoP =
        elementoPeca.GetElementsByTagName("descricaoP")[0].InnerText;
    string comprimentoP =
        elementoPeca.GetElementsByTagName("comprimentoP")[0].InnerText;
    string larguraP =
        elementoPeca.GetElementsByTagName("larguraP")[0].InnerText;
    string rotacionalP =
        elementoPeca.GetElementsByTagName("rotacionalP")[0].InnerText;
    string quantidadeP =
        elementoPeca.GetElementsByTagName("quantidadeP")[0].InnerText;
    string coordenadaXP =
        elementoPeca.GetElementsByTagName("coordenadaXP")[0].InnerText;
    string coordenadaYP =
        elementoPeca.GetElementsByTagName("coordenadaYP")[0].InnerText;
}

```

```
string primeirosCortesP =
elementoPeca.GetElementsByTagName("coordenadaPrimeiro
sCortes")[0].InnerText;
```

```
double lar = Convert.ToDouble(larguraP);
double comp = Convert.ToDouble(comprimentoP);
double coorX = Convert.ToDouble(coordenadaXP);
double coorY = Convert.ToDouble(coordenadaYP);
```

```
areaDaPeca = lar * comp;
areaTotalDasPecasNaChapa = areaTotalDasPecasNaChapa
+ areaDaPeca;
contadorPecasNaChapa++;
```

```
PecaParaPlano pecaNaLista = new PecaParaPlano();
pecaNaLista.Codigo = codigoP;
pecaNaLista.Descricao = descricaoP;
pecaNaLista.Comprimento = comprimentoP;
pecaNaLista.Largura = larguraP;
pecaNaLista.Rotacionar = false;
pecaNaLista.Quantidade = quantidadeP;
```

```
listaDePecaParaRelatorio.Add(pecaNaLista);
lista.Add(pecaNaLista);
```

```
}
```

```
List<PecaParaPlano> list = lista.GroupBy(a =>
a.Codigo).Select(g => g.First()).ToList();
```

```
int tamanhoDaLista = list.Count();
```

```
for (int i = 1; i <= tamanhoDaLista; i++)
{
    int repeticao = 0;
    foreach (PecaParaPlano p in listaDePecaParaRelatorio)
    {
        if (p.Codigo.Equals(list[i-1].Codigo))
            repeticao++;
    }
}
```

```
TextBlock codigoListaPecasInfo = new TextBlock();
codigoListaPecasInfo.Text = list[i-1].Codigo;
codigoListaPecasInfo.FontSize = 12;
```

```

codigoListaPecasInfo.Margin = new Thickness(96, posLinha,
0, 0);
page3.Children.Add(codigoListaPecasInfo);

```

```

TextBlock descricaoListaPecasInfo = new TextBlock();
descricaoListaPecasInfo.Text = list[i-1].Descricao;
descricaoListaPecasInfo.FontSize = 12;
descricaoListaPecasInfo.Margin = new Thickness(180,
posLinha, 0, 0);

```

```

page3.Children.Add(descricaoListaPecasInfo);

```

```

TextBlock comprimentoListaPecasInfo = new TextBlock();
comprimentoListaPecasInfo.Text = list[i-1].Comprimento;
comprimentoListaPecasInfo.FontSize = 12;
comprimentoListaPecasInfo.Margin = new Thickness(350,
posLinha, 0, 0);

```

```

page3.Children.Add(comprimentoListaPecasInfo);

```

```

TextBlock larguraListaPecasInfo = new TextBlock();
larguraListaPecasInfo.Text = list[i-1].Largura;
larguraListaPecasInfo.FontSize = 12;
larguraListaPecasInfo.Margin = new Thickness(500,
posLinha, 0, 0);
page3.Children.Add(larguraListaPecasInfo);

```

```

TextBlock quantidadeListaPecasInfo = new TextBlock();
quantidadeListaPecasInfo.Text = repeticao.ToString();
quantidadeListaPecasInfo.FontSize = 12;
quantidadeListaPecasInfo.Margin = new Thickness(610,
posLinha, 0, 0);

```

```

page3.Children.Add(quantidadeListaPecasInfo);

```

```

double areaPeca2 = ( (Convert.ToDouble(list[i-
1].Comprimento) * Convert.ToDouble(list[i-1].Largura) *
repeticao ) ) / 1000000;
TextBlock areaListaPecasInfo = new TextBlock();
areaListaPecasInfo.Text = areaPeca2.ToString("0.00");
areaListaPecasInfo.FontSize = 12;
areaListaPecasInfo.Margin = new Thickness(750, posLinha,
0, 0);
page3.Children.Add(areaListaPecasInfo);
posLinha = posLinha + 15;

```

```

    repeticao = 0;
}

```

*//Insere titulo na pagina de estatistica*

```

TextBlock tituloPagEst = new TextBlock();
tituloPagEst.Text = "Chapa " + contadorChapasParaRelatorio
+ " - Estatísticas:";
tituloPagEst.FontSize = 30;
tituloPagEst.Margin = new Thickness(20, 10, 0, 0);
page3.Children.Add(tituloPagEst);
PageContent page3Content = new PageContent();
((IAddChild)page3Content).AddChild(page3);
document.Pages.Add(page3Content);

```

```

Image logoEscrito = new Image();
logoEscrito.Margin = new Thickness(880, 10, 0, 0);
BitmapImage logoEscrito1 = new BitmapImage();
logoEscrito1.BeginInit();
logoEscrito1.UriSource = new
Uri("C:/dados/imagens/logo_escrito.jpg");
logoEscrito1.EndInit();
logoEscrito.Source = logoEscrito1;
page3.Children.Add(logoEscrito);

```

```

TextBlock rodapePaginaPlano = new TextBlock();
rodapePaginaPlano.Text = "Otimizador de Cortes OtimizaCut -
Desenvolvido por Aislan Nadrowski (2013)";
rodapePaginaPlano.FontSize = 12;
rodapePaginaPlano.Margin = new Thickness(96, 750, 0, 0);
page3.Children.Add(rodapePaginaPlano);

```

```

SolidColorBrush corDoSeparador = new SolidColorBrush();
corDoSeparador.Color = Colors.Black;

```

```

Line separador = new Line();
separador.X1 = 2750;
separador.Y1 = 60;
separador.Y2 = 60;
separador.StrokeThickness = 1;
separador.Stroke = corDoSeparador;
page3.Children.Add(separador);

```

```

Line separador1 = new Line();
separador1.X1 = 2750;

```

```

separador1.Y1 = 120;
separador1.Y2 = 120;
separador1.StrokeThickness = 1;
separador1.Stroke = corDoSeparador;
page3.Children.Add(separador1);

```

```

TextBlock infoMatRel = new TextBlock();
infoMatRel.Text = "Material: ";
infoMatRel.FontSize = 12;
infoMatRel.Margin = new Thickness(10, 65, 0, 0);
infoMatRel.FontWeight = FontWeights.UltraBold;
page3.Children.Add(infoMatRel);

```

```

TextBlock infoMatRelInfo = new TextBlock();
infoMatRelInfo.Text = comprimentoMatString + "mm x " +
larguraMatString + "mm x " + espessuraMatString + "mm";
infoMatRelInfo.FontSize = 12;
infoMatRelInfo.Margin = new Thickness(65, 65, 0, 0);
page3.Children.Add(infoMatRelInfo);

```

```

TextBlock quantidadePecas = new TextBlock();
quantidadePecas.Text = "Quantidade de peças: ";
quantidadePecas.FontSize = 12;
quantidadePecas.Margin = new Thickness(10, 80, 0, 0);
quantidadePecas.FontWeight = FontWeights.UltraBold;
page3.Children.Add(quantidadePecas);

```

```

TextBlock quantidadePecasInfo = new TextBlock();
quantidadePecasInfo.Text = contadorPecasNaChapa.ToString();
quantidadePecasInfo.FontSize = 12;
quantidadePecasInfo.Margin = new Thickness(135, 80, 0, 0);
page3.Children.Add(quantidadePecasInfo);
contadorPecasNaChapa = 0;

```

```

TextBlock areaMatRel = new TextBlock();
areaMatRel.Text = "Área total da chapa: ";
areaMatRel.FontSize = 12;
areaMatRel.Margin = new Thickness(300, 65, 0, 0);
areaMatRel.FontWeight = FontWeights.UltraBold;
page3.Children.Add(areaMatRel);

```

```

double areaMaterial =
(Convert.ToDouble(comprimentoMatString) *
Convert.ToDouble(larguraMatString)) / 1000000;

```

```

TextBlock areaMatRelInfo = new TextBlock();
areaMatRelInfo.Text = areaMaterial.ToString() + "m²";
areaMatRelInfo.FontSize = 12;
areaMatRelInfo.Margin = new Thickness(417, 65, 0, 0);
page3.Children.Add(areaMatRelInfo);

```

```

TextBlock aproveitamentoMatRel = new TextBlock();
aproveitamentoMatRel.Text = "Área total de peças na chapa: ";
aproveitamentoMatRel.FontSize = 12;
aproveitamentoMatRel.Margin = new Thickness(300, 80, 0, 0);
aproveitamentoMatRel.FontWeight = FontWeights.UltraBold;
page3.Children.Add(aproveitamentoMatRel);

```

```

areaTotalDasPecasNaChapa = areaTotalDasPecasNaChapa /
1000000;

```

```

TextBlock aproveitamentoMatRelInfo = new TextBlock();
aproveitamentoMatRelInfo.Text =
areaTotalDasPecasNaChapa.ToString() + "m²";
aproveitamentoMatRelInfo.FontSize = 12;
aproveitamentoMatRelInfo.Margin = new Thickness(469, 80, 0,
0);
page3.Children.Add(aproveitamentoMatRelInfo);

```

```

TextBlock aproveitamentoMatRelPorcentagem = new
TextBlock();
aproveitamentoMatRelPorcentagem.Text = "Aproveitamento da
chapa: ";
aproveitamentoMatRelPorcentagem.FontSize = 12;
aproveitamentoMatRelPorcentagem.Margin = new
Thickness(300, 95, 0, 0);
aproveitamentoMatRelPorcentagem.FontWeight =
FontWeights.UltraBold;

```

```

page3.Children.Add(aproveitamentoMatRelPorcentagem);

```

```

double aproveitamentoChapa = areaTotalDasPecasNaChapa /
areaMaterial * 100;

```

```

TextBlock aproveitamentoMatRelPorcentagemInfo = new
TextBlock();
aproveitamentoMatRelPorcentagemInfo.Text =
aproveitamentoChapa.ToString("0.00") + "%";
aproveitamentoMatRelPorcentagemInfo.FontSize = 12;

```



```

        aproveitamentoMatRelPorcentagemInfo.Margin = new
        Thickness(453, 95, 0, 0);
        page3.Children.Add(aproveitamentoMatRelPorcentagemInfo);

        areaTotalDasPecasNaChapa = 0;

    }

}

// Imprime documento
string nomeParaSalvar = "Mat " + codigoMatString + " - " +
descricaoMatString + " (Lote " + codigoLote + " - " + descricaoLote +
")";
pd.PrintDocument(document.DocumentPaginator, nomeParaSalvar );

}

catch (Exception ex)
{
    MessageBox.Show(ex.Message, "Erro ao imprimir plano de
corte.", MessageBoxButton.OK, MessageBoxImage.Error);
}

}

```

## APÊNDICE D – MANUAL DO USUÁRIO



**Manual do Usuário**  
***Software Otimizador de Cortes OtimizaCut***

Versão: 1.1

## INTRODUÇÃO

O principal objetivo deste documento é auxiliar o usuário do *software* na utilização do mesmo. Neste documento são descritas de maneira detalhada as funções que estão disponíveis no *software*.

Histórico de revisões			
Nome	Descrição	Data	Revisão
Aislan	Criar manual.	Julho de 2013	1.0
Aislan	Revisão	Agosto de 2013	1.1

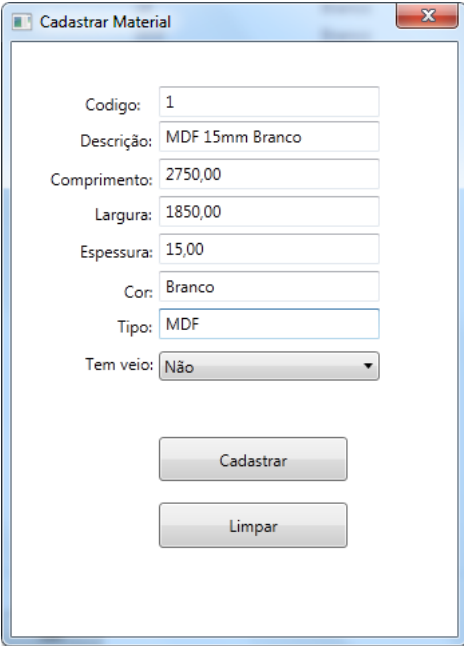
## FUNÇÕES

O *software* é composto por diversas funções que permitem ao usuário realizar a otimização de um lote de peças e emitir relatórios com base nessas otimizações, essas funções são descritas a seguir.

### BIBLIOTECA DE MATERIAIS

É possível cadastrar, editar e apagar um material. O cadastro de material é um procedimento essencial para que o *software* gere os resultados esperados.

A biblioteca de materiais cadastrados é visualizada em forma de tabela na aba “Materiais”, conforme é exibido na figura 02.



A imagem mostra uma janela de software intitulada "Cadastrar Material". Dentro da janela, há uma série de campos de entrada rotulados à esquerda:

- Código: 1
- Descrição: MDF 15mm Branco
- Comprimento: 2750,00
- Largura: 1850,00
- Espessura: 15,00
- Cor: Branco
- Tipo: MDF
- Tem veio: Não (menu suspenso)

Na parte inferior da janela, há dois botões de ação: "Cadastrar" e "Limpar".

**Figura 01 – Cadastro de material**

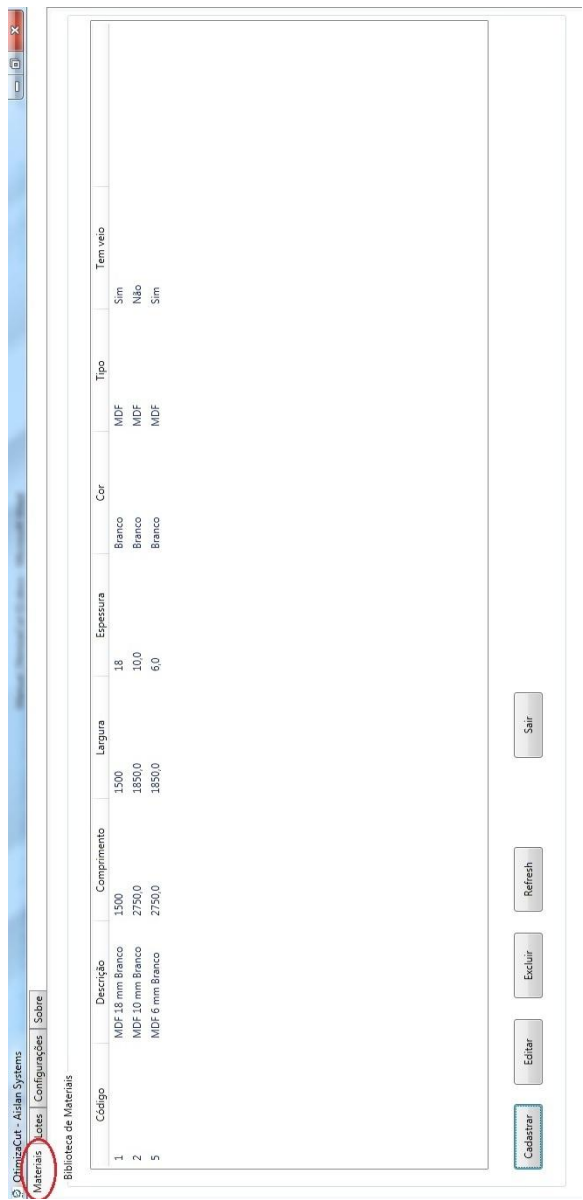
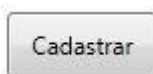


Figura 02 - Biblioteca de materiais

## **CADASTRO**

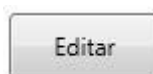
O cadastro de material é realizado clicando no botão “Cadastrar” – figura 03 - situado na parte inferior da janela exibida na figura 01. Para fazer o cadastro é necessário inserir as informações referentes ao material como: código, descrição, comprimento, largura, espessura, cor, tipo do acabamento e se possui veio ou não. Após a inserção das informações nos campos correspondentes basta clicar no botão “Cadastrar” – figura 03 - para concluir o processo.



**Figura 03 – Botão cadastrar material**

## **EDIÇÃO**

Para fazer a edição de um material cadastrado basta selecionar o material na tabela e clicar no botão “Editar” – figura 04 - situado na parte inferior da janela exibida na figura 02. Todos os campos de informações são editáveis, com exceção do campo “Código”.



**Figura 04 – Botão editar material**

## **EXCLUSÃO**

A exclusão de um material é realizada da seguinte maneira: deve-se selecionar o material desejado na tabela e clicar no botão “Excluir” – figura 05 - situado na parte inferior da tela exibida na figura 02.



**Figura 05 – Botão excluir material**

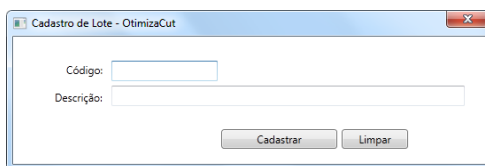
## BIBLIOTECA DE LOTES

O *software* permite fazer o cadastro e edição de lotes de peças. Ao editar um lote é possível:

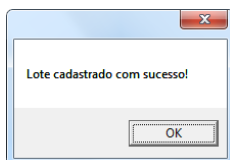
- ✓ Informar qual material deve ser utilizado para otimizar o plano;
- ✓ Adicionar peças de diferentes tamanhos e quantidades;
- ✓ Informar se as peças podem rotacionar ou não dentro do plano de corte com base no sentido dos veios da matéria-prima (lembrando que, de maneira geral, quando a opção “rotacionar” está habilitada, a chance de evitar desperdícios da chapa é maior).

### CADASTRO

Para realizar o cadastro de um lote basta acessar a biblioteca de lotes na aba “Lotes” conforme exibido na figura 08. Ao clicar no botão “Cadastrar” situado na parte inferior da tela é exibida a janela mostrada na figura 06, nela é possível informar o código e descrição do novo lote a ser criado. Se o processo de cadastro for concluído com sucesso irá mostrar a janela exibida na figura 07, bastante o usuário clicar no botão “Ok”.

A janela de software intitulada "Cadastro de Lote - OtimizaCut" possui um campo de entrada para "Código:" e um campo de entrada para "Descrição:". Abaixo dos campos, há dois botões: "Cadastrar" e "Limpar".

**Figura 06 – Cadastro de lote**



**Figura 07 – Cadastro com sucesso**

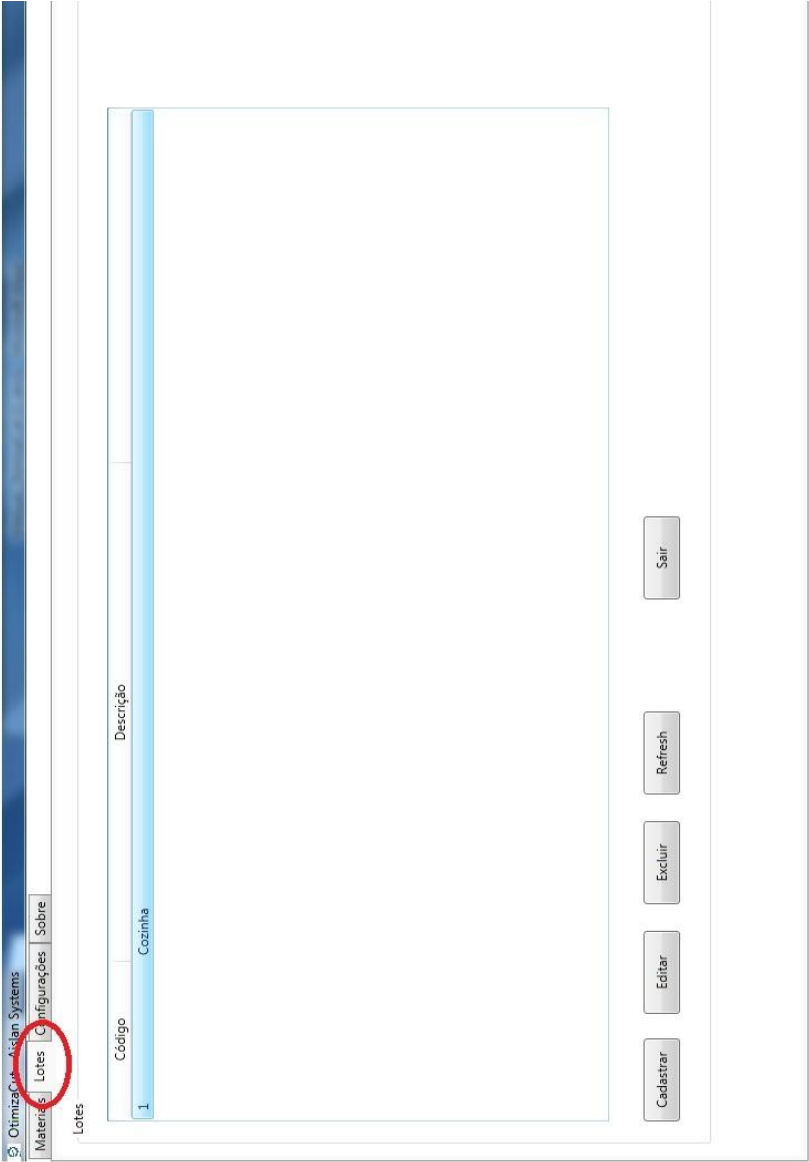


Figura 08 – Biblioteca de lotes



## EDIÇÃO

Para fazer a edição de um lote (entende-se por edição de lote o processo de adicionar peças ao lote e selecionar o material utilizado para produção das respectivas peças).

Para editar basta selecionar o lote na biblioteca de lotes e clicar no botão “Editar” situado na parte inferior da janela exibida na figura 08 que abrirá a janela exibida na figura 09.

**Identificação**

**Código:** 1      **Descrição:** Cozinha

**Escolha do material (matéria-prima)**

**Material:** 2 - MDF 10 mm Branco

Calcular!    Salvar    Sair

Excluir todas as peças do lote

Codigo	Descricao	Comprimento	Largura	Quantidade	Rotacionar	
1	Tampa	800	400	1	<input checked="" type="checkbox"/>	
2	Lateral	700	500	2	<input type="checkbox"/>	
					<input type="checkbox"/>	

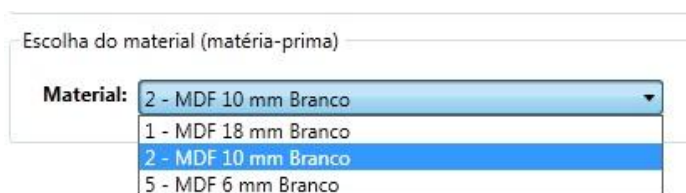
**Figura 09 – Janela para edição de lote**

A janela de edição de lote é dividida em três partes:

**Parte superior:** contém a identificação do lote aberto para edição (exibe o código e descrição);

**Parte central:** é possível selecionar o material que será utilizado para otimizar o lote (é possível selecionar os materiais que foram cadastrados na biblioteca de materiais), conforme exibido na figura 10;

**Parte inferior:** nessa tabela é possível inserir as peças que integram o lote, informando os seguintes dados: código, descrição, comprimento, largura, quantidade e se é permitida a rotação da peça dentro do plano de corte, desconsiderando o veio do material (com essa opção selecionada é aumentada a chance de melhor aproveitamento da matéria-prima).

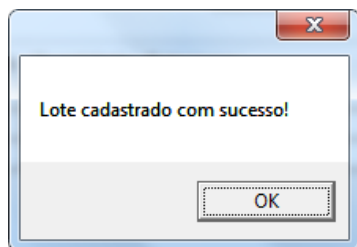


**Figura 10 – Seleção do material utilizado**

Após efetuar alguma alteração no lote é preciso salvá-lo para as modificações serem registradas no banco de dados. Para salvar é utilizado o botão “Salvar” – figura 11 - se o lote for salvo corretamente será exibida a janela mostrada na figura 12. Ainda na figura 11 é exibido o botão “Calcular!” que, ao ser clicado otimiza o lote e gera os planos de corte para visualização.



**Figura 11 – Botões Calcular!, Salvar e Sair**



**Figura 12 – Lote cadastrado com sucesso**

## **VISUALIZAÇÃO DE UM PLANO DE CORTE**

Após o cálculo de um lote, os planos de cortes gerados são exibidos automaticamente em uma nova janela (figura 14). Essa janela possui um menu superior (figura 13) que possibilita gerar um relatório com as informações estatísticas e com os planos de cortes a serem executados pelo operador da máquina seccionadora de chapa de MDF.

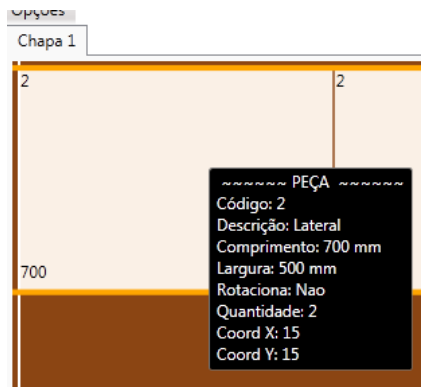


**Figura 13 – Geração de relatório de cortes**

No exemplo da figura 14, o lote otimizado gerou apenas uma chapa. Mas em casos que a quantidade de peças é maior consequentemente o número de chapas utilizadas será maior e para cada chapa é criada uma aba na parte superior do plano (Chapa 1, Chapa 2, Chapa 3, etc).



Figura 14 – Visualizador de planos de corte



**Figura 15 – Tooltip com informações da peça**

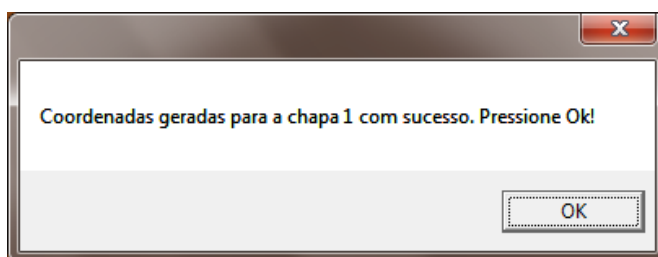
Ao posicionar o cursor do *mouse* sobre a peça é exibido um *tooltip* (retângulo com fundo preto) que contém informações referentes à respectiva peça (figura 15). As informações são:

- **Código:** Código da peça informado no cadastro do lote;
- **Descrição:** Descrição da peça informada no cadastro do lote;
- **Comprimento:** Comprimento da peça informado no cadastro do lote (em milímetros);
- **Largura:** Largura da peça informada no cadastro do lote (em milímetros);
- **Rotaciona:** Informa se a peça rotacionou dentro do plano de corte;
- **Quantidade:** Quantidade de peças do mesmo código informadas no cadastro do lote;
- **CoordX:** Coordenada X de posicionamento da peça dentro do plano de corte (em milímetros);
- **CoordY:** Coordenada Y de posicionamento da peça dentro do plano de corte (em milímetros).

## RELATÓRIO

É possível gerar um relatório com informações estatísticas sobre o plano de corte e com o desenho do plano de corte, com a finalidade de auxiliar no corte das chapas na máquina seccionadora e manter o controle da matéria-prima com ajuda dos dados estatísticos gerados.

Para gerar um relatório referente aos planos de cortes que estão atualmente em visualização basta clicar na opção “Gerar Relatório de Cortes” conforme exibido na figura 14 e confirmar pressionando “Ok” na janela exibida na figura 16.



**Figura 16 – Coordenadas geradas com sucesso**

O relatório é dividido basicamente em três partes:

- 1) **Capa:** contem a identificação do lote otimizado e informações sobre a matéria-prima utilizada no respectivo lote (figura 17);
- 2) **Páginas com os planos de corte:** contem os planos de corte originados na otimização do respectivo lote (figura 18);
- 3) **Páginas com os dados estatísticos:** contem os dados estatísticos referentes aos planos de corte gerados (figura 19).

## Relatório de Cortes (OtimizaCut)

### Propriedades do material utilizado:

Código: 2  
Descrição: MDF 10 mm Branco  
Comprimento: 2750,0  
Largura: 1850,0  
Espessura: 10,0  
Cor: Branco  
Tipo: MDF

### Propriedades do lote otimizado:

Código: 1  
Descrição: Cozinha



Figura 17 – Capa do relatório

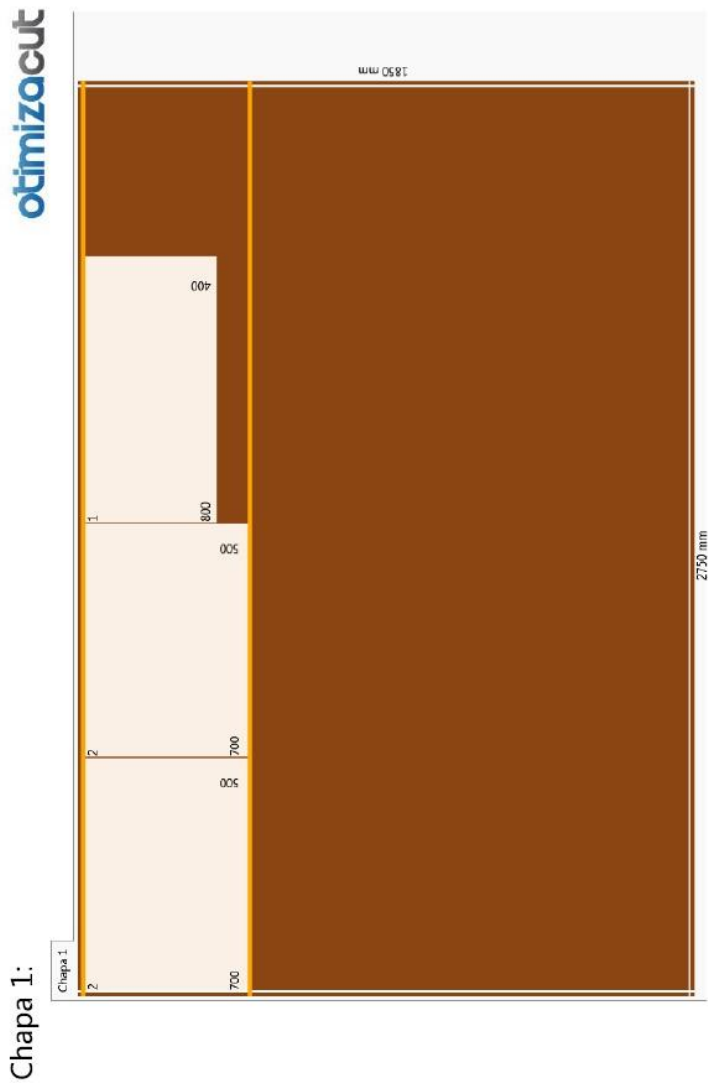


Figura 18 – Página contendo plano de corte



Chapa 1 - Estatísticas:

Material: 2750,0mm x 1850,0mm x 10,0mm					
Quantidade de peças: 3					
Área total da chapa: 5,0875m <sup>2</sup>					
Área total de peças na chapa: 1,02m <sup>2</sup>					
Aproveitamento da chapa: 20,05%					
CÓDIGO	DESCRIÇÃO	COMPRIMENTO	LARGURA	QUANTIDADE	ÁREA (M <sup>2</sup> )
2	Lateral	700	500	2	0,70
1	Tampa	800	400	1	0,32

Figura 19 – Página contendo dados estatísticos do plano de corte

## PARÂMETROS DE CONFIGURAÇÃO

A configuração dos parâmetros de otimização é realizada na lista de parâmetros existentes na aba “Configurações” conforme exibido na figura 20. Os parâmetros são:

OtimizaCut - Aislan Systems

Materiais Lotes Configurações Sobre

Parâmetros de configuração

Espessura da serra: 4,2 (mm)

Refilos laterais: 15,0 (mm)

☒ Exibir comprimento e largura no plano de corte

☒ Exibir linhas de refilos laterais no plano de corte

☒ Exibir apenas os cortes do primeiro ciclo (cortes verticais/cortes horizontais)

Orientação dos cortes iniciais da chapa

☒ Horizontais

☐ Verticais

Configurações para geração do relatório de cortes

☒ Exibir desenho do plano de corte no relatório

☒ Exibir estatísticas do plano de corte

Salvar

**Figura 20 – Parâmetros de configuração**

- **Espessura da serra:** É um campo numérico expresso em milímetros, onde informa-se a espessura da serra utilizada para corte na máquina seccionadora;
- **Refilos laterais:** É um campo numérico expresso em milímetros, onde informa-se as laterais que são refugos (pois foram estragadas durante transporte da matéria-prima ou a qualidade da chapa é baixa e deve-se refilar as bordas quebradas);

- **Exibir comprimento e largura no plano de corte:** Com esse parâmetro habilitado são exibidas as informações referentes às dimensões das peças inseridas no plano de corte (comprimento e largura);
- **Exibir linhas de refilos laterais no plano de corte:** Quando este parâmetro está habilitado, são exibidas as linhas que indicam onde estão os refilos laterais (superior, inferior, esquerdo e direito) da chapa;
- **Exibir apenas os cortes do primeiro ciclo (cortes verticais/cortes horizontais):** Quando este parâmetro está habilitado são exibidos os primeiros cortes da chapa, sendo eles horizontais ou verticais;
- **Orientação dos cortes iniciais da chapa:** Este parâmetro possibilita ao usuário do *software* otimizar planos de corte que iniciem com cortes na horizontal da chapa ou na vertical da mesma;
- **Exibir desenho do plano de corte no relatório:** Se este parâmetro estiver habilitado são exibidas as páginas contendo os planos de corte no relatório gerado pelo usuário;
- **Exibir estatísticas do plano de corte:** Se este parâmetro estiver habilitado são exibidas as páginas contendo as estatísticas de cada chapa (plano de corte) no relatório gerado pelo usuário.