

UNIVERSIDADE METODISTA DE SÃO PAULO
ESCOLA DE ENGENHARIAS, TECNOLOGIA E INFORMAÇÃO
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

ANTONIO CARLOS SILVA FILHO
DENIS LUIZ MARTINONI JUNIOR
RODRIGO MASCARINI GALINDO

TCCTEX
SISTEMA PARA FORMATAÇÃO DE TRABALHO
DE CONCLUSÃO DE CURSO

SÃO BERNARDO DO CAMPO

2015

ANTONIO CARLOS SILVA FILHO
DENIS LUIZ MARTINONI JUNIOR
RODRIGO MASCARINI GALINDO

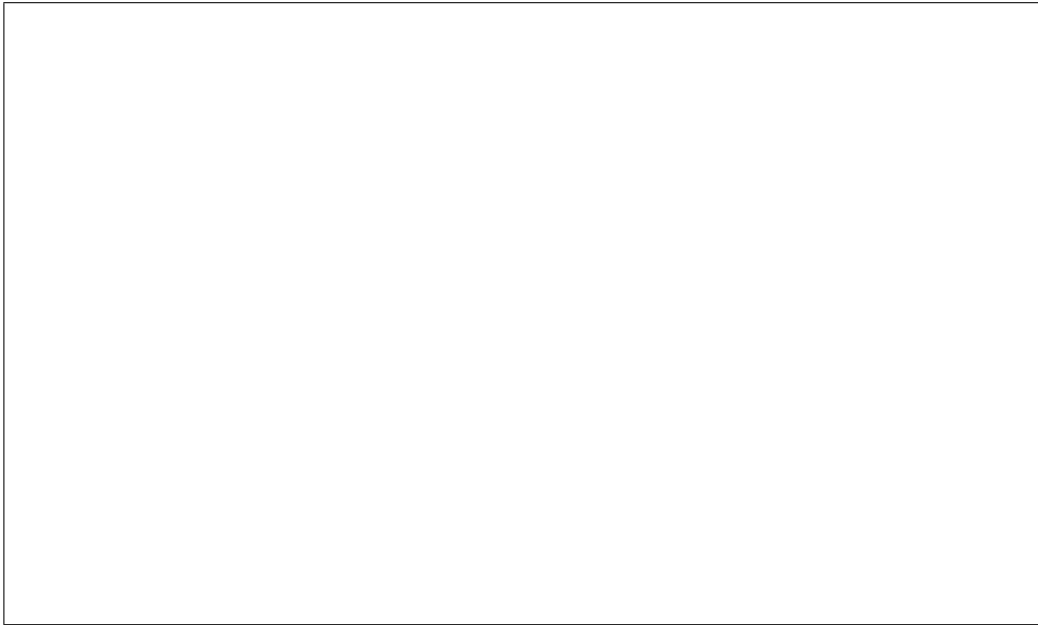
TCCTEX
SISTEMA PARA FORMATAÇÃO DE TRABALHO
DE CONCLUSÃO DE CURSO

Trabalho apresentado no curso de graduação
à Universidade Metodista de São Paulo, Es-
cola de Engenharias, Tecnologia e Informa-
ção como requisito parcial para obtenção do
título de Bacharel em Sistemas de Informa-
ção

Orientador: Profa. Dra. Silvia A. Brunini

SÃO BERNARDO DO CAMPO

2015



Dedicamos este trabalho a todos os nossos familiares, amigos e professores, que
caminharam juntos conosco até aqui.

“Diante da vastidão do tempo e da imensidão do universo, é um imenso prazer para mim
dividir um planeta e uma época com você”

Carl Sagan

RESUMO

A ideia por trás de computadores digitais pode ser explicada dizendo que estas máquinas têm a intenção de realizar qualquer operação que pode ser realizada por um computador humano (Alan Turing, 1950, tradução nossa). Na era da informação e automação, podemos mais do que nunca, moldar o mundo a nossa volta, de cadeiras de rodas com software de sintetização de voz, a celulares que encontram a loja que deseja. Softwares estão em todos os lugares, em sistemas embarcados nos aviões, no forno de micro-ondas, nos carros, talvez esta seja uma das invenções mais úteis já criadas. Nesse contexto, pensamos em cada coisa que pode ser automatizada, otimizada, padronizada, diante dessa realidade constatamos que poderíamos auxiliar alunos que estão passando por um importante processo acadêmico, o conhecido e temido Trabalho de Conclusão do Curso (TCC). A cada ano professores e alunos usam grande parte do seu precioso tempo para escrever, revisar, refazer seu texto para avaliação. Nossa proposta ambiciona a automatização de parte deste processo, nos propomos a desenvolver um software, que recebe arquivos de texto como entrada, processa, e padroniza o documento final nos padrões ABNT Metodista. Para tal, utilizamos técnicas e ferramentas de desenvolvimento de software como, Extreme Programming, uma metodologia de desenvolvimento ágil de software; NetBeans IDE, um ambiente integrado de desenvolvimento; \LaTeX , um conjunto de macros para o programa de diagramação de textos TeX, utilizado amplamente na produção de textos matemáticos e científicos, devido a sua alta qualidade tipográfica.

Palavras-chave: TCCTeX, TCC, desenvolvimento de software, XP, Extreme Programming, processador de texto, \LaTeX , TeX, odt, pdf.

ABSTRACT

The idea behind digital computers may be explained by saying that these machines are intended to carry out any operations that could be done by a human computer (Alan Turing, 1950). In the age of information and automation, more than ever, we can shape the world around us, wheelchairs with speech synthesis software to mobiles that find the store you want. Software is everywhere, embedded systems: in planes, microwaves oven, cars, maybe it is the most useful invention ever created. In this context, we think about several things that can be automated, standardized, given this reality, we found that we could automate an important academic process, known and feared, course completion assignment (CCA). As well know, every year teachers and students use much of your precious time to write, correct and redo. Our proposal aims to automate part of this process, we propose to develop a software that receives text files as input, processes, and standardizes the final document in the Metodista ABNT standards. To make it happen, we used techniques and tools for software development such as Extreme Programming, a methodology for agile software development, NetBeans IDE, an integrated development environment, \LaTeX , a set of macros for text typesetting program: Tex, widely used in the production of mathematical and scientific documents, due to their high typographical quality.

Key-words: TCCTeX, TCC, software development, XP, Extreme Programming, word processor, \LaTeX , TeX, odt, pdf.

LISTA DE ILUSTRAÇÕES

Figura 1 – Template \LaTeX com Linguagem Apache Velocity	28
Figura 2 – Exemplo de Configuração Writer2Latex	29
Figura 3 – Modelo de User Stories	32
Figura 4 – Caixa de Seleção dos Testes Funcionais	33
Figura 5 – Modelo de Cartão de Tarefa	34
Figura 6 – Parte da arquitetura do software TCCTeX	36
Figura 7 – Projeto TCCTeX	37
Figura 8 – Pacotes do TCCTeX	38
Figura 9 – Pacote br.metodista,TCCTeX	39
Figura 10 – Importações da Classe Conversor	39
Figura 11 – JLR para Geração de PDF	40
Figura 12 – Método odtToTex	41
Figura 13 – Parte do Método alterarTex()	41
Figura 14 – JLR para Preenchimento de Template Tex	42
Figura 15 – Método converterPre()	43
Figura 16 – Classe textuais	43
Figura 17 – Método converterTextuais()	44
Figura 18 – Parte do método abrir() Principal	45
Figura 19 – Pacote Contendo as Telas	46
Figura 20 – Pacote Contendo Imagens	46
Figura 21 – Interface da Tela Principal	48
Figura 22 – Etapa Concluída	48
Figura 23 – Interface da Tela de Informações Essenciais	49
Figura 24 – Interface da tela de Pré-Textuais	50
Figura 25 – Interface da Tela de Textuais	51
Figura 26 – Interface da Tela Pós-Textuais	52
Figura 27 – Interface da Tela Entrega Final	53
Figura 28 – Capa em abnTeX1	57
Figura 29 – Folha de Rosto em abnTeX1	58
Figura 30 – Preenchimento de Macros em principal.tex abnTeX2	59
Figura 31 – Pequeno Texto feito no Microsoft Word	60
Figura 32 – Pequeno texto em \LaTeX com mesmo conteúdo da figura anterior	60
Figura 33 – Pequena parte do código XML do docx	61

LISTA DE TABELAS

Tabela 1	– Regra de Negócio: Tela Informações Essenciais	54
Tabela 2	– Regra de Negócio: Tela Informações Essenciais	54
Tabela 3	– Regra de Negócio: Botão Terminei Esta Parte	54
Tabela 4	– Regra de Negócio: Arquivos de Entrada	54
Tabela 5	– Regra de Negócio: Arquivos Opcionais	54
Tabela 6	– Regra de Negócio: Botão Gerar Tex	55
Tabela 7	– Regra de Negócio: Botão Gerar PDF	55
Tabela 8	– Regra de Negócio: Adicionar Capítulos da Tela Textuais	55

SUMÁRIO

1	INTRODUÇÃO	15
2	REVISÃO BIBLIOGRÁFICA	17
3	REFERENCIAL TEÓRICO	19
3.1	XP – EXTREME PROGRAMMING	19
3.1.1	Por que utilizamos XP	21
3.2	LaTeX	21
3.2.1	Benefícios do Latex	22
3.2.2	Por que utilizamos LaTeX	22
3.3	AS CLASSES ABNTeX E ABNTeX2	23
3.3.1	Novidades do abnTeX2	23
3.3.2	Por que utilizamos abnTeX	25
3.4	OPEN DOCUMENT TEXT – ODT	25
3.5	OFFICE OPEN XML DOCUMENT – DOCX	25
3.5.1	Vantagens e desvantagens do docx	27
3.5.2	Por que utilizamos a extensão .docx	27
3.6	APIs UTILIZADAS	27
3.6.1	Java LaTeX Report	28
3.6.2	Writer2Latex	28
4	DESENVOLVIMENTO DO PROJETO	31
4.1	APLICAÇÃO DA METODOLOGIA XP	32
4.2	ARQUITETURA DO SOFTWARE UTILIZANDO ODT	35
4.3	FUNCIONAMENTO DO TCCTEX	37
4.3.1	Pacote br.metodista.TCCtex	38
4.3.1.1	Classe Conversor	39
4.3.1.2	Classe InfoEssenciais	42
4.3.1.3	Classe PreTextuais	42
4.3.1.4	Classe Textuais	43
4.3.1.5	Classe Principal	45
4.3.2	Pacote br.metodista.TCCtex.telas	46
4.3.2.1	Pacote br.metodista.TCCTex.telas.imagens	46
4.4	INTERFACES	47
4.4.1	INTERFACE DA TELA PRINCIPAL	47
4.4.2	INTERFACE DA TELA DE INFORMAÇÕES ESSENCIAIS	49

4.4.3	INTERFACE DA TELA DE PRÉ-TEXTUAIS	49
4.4.4	INTEFACE DA TELA DE TEXTUAIS	50
4.4.5	INTERFACE DA TELA PÓS-TEXTUAIS	51
4.4.6	INTERFACE DA TELA ENTREGA FINAL	52
4.5	REGRAS DE NEGÓCIO	53
4.6	ALTERAÇÃO DE abnTeX PARA abnTeX2	55
4.7	EXTENÇÃO DO PRESENTE TRABALHO	59
4.8	DISCUSSÃO	61
5	CONCLUSÃO	65
	APÊNDICE A – ATAS DE REUNIÕES COM ORIENTADOR	69
	APÊNDICE B – CRONOGRAMA	87
	ANEXO A – RELATÓRIO DE RECOMENDAÇÕES DA BANCA DE QUALIFICAÇÃO	91

1 INTRODUÇÃO

Nos dias atuais software faz parte da nossa vida, eles estão em todos os lugares, suas funções são as mais diversas, sua utilidade incalculável. Ainda neste tema de software, notamos a dificuldade dos alunos na hora de adequar a redação de seus trabalhos ao padrão ABNT; por outro lado professores dispendem horas corrigindo a formatação do trabalho, nesse cenário, o projeto, TCCTeX, se propõe a auxiliar o estudante a desenvolver seu TCC em etapas pré-definidas, padronizando-o segundo a norma ABNT Metodista e gerando automaticamente o documento no formato PDF, após ser processado em \LaTeX .

O principal objetivo é desenvolver um software (TCCTeX) que recebe dados e arquivos em formato comum ao usuário, faz um processamento destes e como saída, produz um documento completo no padrão \LaTeX seguindo as normas ABNT Metodista, assim agilizando o processo de padronização do trabalho feito pelo aluno e tornando desnecessária a correção da padronização feita pelo professor.

Durante o desenvolvimento do nosso projeto, utilizaremos diversas ferramentas de apoio ao desenvolvimento adequado ao nosso tipo software, entre elas Latex, NetBeans IDE. O desenvolvimento deste projeto utiliza o consagrado \LaTeX , um produtor de texto acadêmicos e matemáticos; Extreme Programming (XP), uma metodologia ágil para desenvolvimento de software que encoraja a colaboração entre as partes interessadas de um projeto de software; e NetBeans IDE, um ambiente de desenvolvimento muito utilizado pelo mercado.

A motivação do grupo é a possibilidade de aprender a lidar com metodologias, técnicas e tecnologias com grande potencial de mercado as quais esperamos poder ser um diferencial no desenvolvimento da nossa carreira. E mais a oportunidade de desenvolver um software simples que será útil a nós e a qualquer outro estudante que necessite seguir as normas da ABNT Metodista para o desenvolvimento de um documento de TCC.

Espera-se que este software possa contribuir para ajudar aos alunos da Escola de Engenharias, Tecnologia e Informação e futuramente qualquer outra faculdade da Metodista, no seu processo de construção do TCC, ganhando tempo, qualidade e para alunos e professores.

2 REVISÃO BIBLIOGRÁFICA

Durante o processo de desenvolvimento do TCC pesquisamos diversos softwares que são utilizados para o desenvolvimento da documentação de TCC a fim de ter o conhecimento sobre essas ferramentas, suas vantagens e desvantagens em relação ao desenvolvimento do TCC.

As principais ferramentas utilizadas para desenvolvimento da documentação do TCC são editores de texto WYSIWYG (What you see is what you get, em português “O que você vê é o que você obtém”) como o Microsoft Word que diferente do \LaTeX possibilita que o documento que está sendo criado possa ser editado e visualizado ao mesmo tempo com suas alterações, por meio de atalhos que são selecionados pelo mouse ou atalhos, enquanto no \LaTeX todo desenvolvimento do documento é feito por meio de códigos que são interpretados e apresentados visualmente ao usuário apenas após a conversão do documento para PDF ou algum outro formato pré estabelecido. A grande desvantagem de editores WYSIWYG é a dificuldade de formatação de documentos muito extensos e de abordagem científica, nos quais é exigida uma formatação padronizada e cheia de detalhes. Outra desvantagem é na criação e formatação de fórmulas matemáticas complexas que são muito comuns em documentos científicos.

Por esses motivos, muitas universidades levaram a exigir que os alunos utilizassem o \LaTeX para o desenvolvimento do TCC, pois ele possui fácil formatação de qualquer tipo de texto e é um programa padrão que existe há anos e não varia muito a cada nova versão além de ser gratuito.

Notamos que hoje, o melhor programa que permite o desenvolvimento de um TCC dentro de todas as normas ABNT e todas as funcionalidades exigidas para isso é o \LaTeX , porém ele não é um programa intuitivo e de fácil aprendizagem, principalmente para pessoas que não tem grande familiaridade com computação.

Em cada universidade a utilização das normas ABNT para formatação de trabalhos acadêmicos é feita de forma diferente e seguindo como base o documento “Manual de Apresentação para Trabalhos Acadêmicos” (TIMBÓ, 2013) desenvolvido pela Universidade Metodista de São Paulo, notamos a complexidade das normas e a grande quantidade de cuidados que devem ser tomados pelos alunos para fazer uma documentação corretamente. Usaremos os requisitos deste documento como base para desenvolver o nosso software que intermediará a interação do aluno que desenvolve o TCC com o \LaTeX e auxiliará na formatação deste documento segundo as normas ABNT adotadas pela UMESP.

3 REFERENCIAL TEÓRICO

Durante o desenvolvimento do nosso projeto, utilizamos de algumas ferramentas de apoio ao desenvolvimento adequado ao nosso tipo software, entre elas \LaTeX , NetBeans IDE, APIs Java como Writer2Latex e JLR (Java Latex Report), Também utilizamos a metodologia Extreme Programming (XP). Neste capítulo falaremos sobre cada uma dessas ferramentas e metodologias.

3.1 XP – EXTREME PROGRAMMING

O Extreme Programming (XP) foi criado em 1997 por Kent Beck. “A Extreme Programming (XP) é uma metodologia ágil para equipes pequenas e médias que desenvolvem software baseado em requisitos vagos e que se modificam rapidamente” (BECK, 2000).

O XP visa garantir a satisfação do cliente, enfatizando o desenvolvimento ágil do projeto para o cumprimento das estimativas e vem ocupando espaço considerável no mercado, que antes era dominado por metodologias tradicionais, como RUP – Rational Unified Process.

Este subcapítulo tem como objetivo fazer uma breve apresentação dessa metodologia, seus valores e algumas de suas práticas para justificar a utilização da mesma nesse projeto, evitando uma abordagem detalhada e desnecessária ao objetivo do projeto.

VALORES XP

“Metas individuais de curto prazo frequentemente entram em conflito com metas sócias de longo prazo. As sociedades aprenderam a lidar com esse problema desenvolvendo um conjunto de valores a serem compartilhados” (BECK, 2000). Extreme Programming possui cinco valores a partir dos quais são desenvolvidas todas as suas práticas.

Esses são os cinco valores de XP:

- **Comunicação:** deve ser priorizado o diálogo presencial que melhora o entendimento tanto entre cliente e desenvolvedor quanto dentro da própria equipe de desenvolvimento;

- **Simplicidade:** desenvolver apenas o que é essencial ao software evita que se produza algo que não utilizado. Vale mais a pena adicionar modificações a desenvolver códigos inúteis;
- **Feedback:** quanto mais cedo for identificada a necessidade de mudanças mais barato será para o desenvolvedor e para o cliente aplica-las. Feedback constante permite a rápida identificação de possíveis erros e da necessidade de mudanças ou melhorias no software;
- **Respeito:** todos devem ser respeitados igualmente dentro e fora da equipe, pois todos contribuem com o desenvolvimento. Todo o tipo de experiência e conhecimento deve ser respeitado;
- **Coragem:** é necessária para estar preparado para as constantes mudanças que podem ocorrer em um projeto de software, bem como para aplicar todos os valores de XP.

Práticas XP

Cada uma das práticas de Extreme Programming visa atender um ou mais dos valores citados acima e não possuem nenhum valor se não forem usadas buscando atender esses valores. Cada prática aplicada separadamente gera benefícios ao projeto, mas a utilização de várias práticas em conjunto pode-se notar um ganho de benefícios muito maior.

Listaremos a seguir algumas práticas de XP que foram adotadas por essa equipe com uma breve descrição das mesmas:

- **Cliente Presente:** o cliente faz parte da equipe do desenvolvimento de software e é presença constante nesse processo, fornecendo feedback com a maior frequência possível;
- **Jogo do Planejamento:** planejamento é uma atividade contínua desempenhada durante todo o projeto, esta postura admite que ao longo do projeto as prioridades e necessidades do cliente sofrem constantes mudanças, portanto o planejamento deve acompanhar essas mudanças;
- **Código Coletivo:** todo o desenvolvedor tem acesso a todas as partes do código (inclusive aquelas que ele não programou) e podem fazer qualquer alteração sem ter que pedir permissão, desde que haja necessidade de tal alteração;

- **Integração Contínua:** é feita constantemente a integração do código produzido ao restante do sistema assegurando, ao final de cada integração, a consistência da base de código.
- **Ritmo Sustentável:** recomenda que os membros da equipe trabalhem oito horas por dia e evitem ao máximo a utilização de horas-extras de trabalho.
- **Projeto Simples:** “o sistema deve ser projetado da maneira mais simples o possível em qualquer momento. A complexidade desnecessária é removida assim que for descoberta” (BECK,2000).

O usuário de XP não tem a obrigação de aplicar todas as suas práticas, mas apenas aquelas que se adequarem as necessidades do projeto.

3.1.1 Por que utilizamos XP

Decidimos aplicar XP devido a sua flexibilidade para com possíveis mudanças no decorrer do projeto e a possibilidade de escolher e adaptar as práticas de XP as necessidades de nosso projeto e as restrições de nossa equipe.

Possuíamos também um grande interesse, pessoal e profissional, em conhecer, aprender e aplicar essa metodologia que vem ganhando espaço no mercado de desenvolvimento de software.

3.2 \LaTeX

“LATEX é um sistema para formatação de textos de documentos. Sua primeira versão amplamente disponível, misteriosamente numerada 2.09, apareceu em 1985” (LAMPORT, 1995)

O \LaTeX é um pacote de macros para desenvolver e gerar documentos para impressão com alta qualidade tipográfica, criado por Leslie Lamport e mantido por Frank Mittelbach atualmente, podendo ser utilizado um layout profissional predefinido. O seu desenvolvimento usou o TeX como motor de formatação.

TeX é um software da onde se originou o LaTeX, foi desenvolvido por Donald E. Knuth que começou a escrevê-lo em 1977, época em que editoras gráficas começaram a utilizar equipamentos digitais para impressão.

Segundo Knuth (1984) ao “preparar um manuscrito em formato TEX, você estará dizendo a um computador exatamente como o manuscrito deve ser transformado em páginas cuja qualidade aparenta ser tipográfica”.

Os caracteres T, E, X no nome derivam respectivamente das letras gregas tau, épsilon, e chi, como o nome do TeX deriva do grego $\tau'εχνη$ (habilidade, arte, técnica) a sua pronúncia correta seria “téc”. Já a primeira sílaba do nome se pronuncia exatamente igual à palavra inglesa lay sendo a pronuncia correta para L^AT_EX é “lei-téc”.

3.2.1 Benefícios do Latex

Para o usuário comum, que não conhece ou apenas ouviu falar do Latex, imediatamente deve se perguntar, “esse é mais um processador de texto, mas porque devo usar?”. Alguns pontos para essa pergunta.

- Devido aos seus layouts padronizados, os textos nele baseado possuem uma aparência de texto de gráfica;
- O Latex é totalmente gratuito e pode funcionar em quase todas as plataformas;
- Produção de textos estruturados;
- Não é necessário montar um layout para uso personalizado, o usuário precisa apenas conhecer alguns comandos que estruturam o documento;
- A construção de fórmulas matemáticas, tabelas, referências, rodapés, índices no L^AT_EX é totalmente facilitado;
- O padrão Latex é amplamente usado pela comunidade acadêmica e científica, sendo em alguns casos obrigatório o seu uso em trabalhos acadêmicos.

3.2.2 Por que utilizamos L^AT_EX

Devido aos benefícios já citados, o L^AT_EX é muito utilizado em Universidades e Empresas de médio e grande porte para o desenvolvimento de relatórios, monografias, artigos e demais documentos que requerem um alto nível de padronização.

Observamos que o desenvolvimento do TCC demanda muito tempo dos alunos e a necessidade de aprender uma tecnologia desconhecida para a maioria dos alunos apenas para desenvolver a documentação do TCC limitava o tempo de trabalho dos mesmos.

Neste trabalho temos como objetivo principal desenvolver um software que intermediasse a interação entre um aluno que esteja desenvolvendo um TCC e o L^AT_EX, tornando o processo de desenvolvimento do TCC mais intuitivo e isentando o aluno da

necessidade de aprender \LaTeX para desenvolver um TCC, com nosso programa sendo o instrumento para a sua documentação final.

3.3 AS CLASSES ABNTEX E ABNTEX2

O \LaTeX , usado amplamente pela comunidade acadêmica para produção de textos matemáticos e científicos, não possuía uma classe que se adaptasse a ABNT (Associação Brasileira de Normas Técnicas), em 2001 nasceu o projeto para adequação do \LaTeX às normas ABNT, originando o pacote de macros chamado `abnTeX`, que auxilia na adequação da documentação dos trabalhos acadêmicos feitos com \LaTeX às normas da ABNT. Entretanto a última versão publicada do `abnTeX` por seus desenvolvedores originais, em 03/11/2004, foi a versão 0.8.2. “Em 2006 uma versão não estável foi publicada para testes, mas nunca foi evoluída” (ARAUJO, 2015).

Decorridos quase 10 anos do início do projeto original e praticamente 5 anos sem nenhuma atualização do `abnTeX`, notou-se diversas falhas em relação as normas ABNT vigentes e muita dificuldade em relação a instalação do mesmo, conseqüentemente surgiu a necessidade de um novo projeto o `abnTeX2` que, coordenado por Lauro César Araujo, teve início em maio de 2012 e teve sua primeira versão concluída em dezembro de 2012.

“O software é mantido desde então pela comunidade de indivíduos e de organizações que adotam e/ou investem em software livre.” (ARAUJO, 2015).

3.3.1 Novidades do `abnTeX2`

A classe `abnTeX2` é “um conjunto de customizações da classe `memoir` para elaboração de documentos técnicos e científicos condizentes com as normas da Associação Brasileira de Normas Técnicas” (ARAUJO, 2015).

A `abnTeX2` foi construído com base na classe `memoir` do `Latex`, sendo essa uma das mais completas, muitos pontos foram melhorados e outros criados, ambiente para errata e ficha catalográfica, posição para rótulos e legendas.

A evolução do `abnTeX`, o pacote `abnTeX2` possui classes, formatação de estilos, pacotes de citação e ainda modelos de documentos, artigos científicos, relatórios técnicos, dissertações e teses, assim a suíte facilita o aprendizado e adaptação ao novo pacote, tendo ainda a documentação disponível.

O pacote `abnTeX2` é compatível com as seguintes normas ABNT:

- **ABNT NBR 6022:2003:** Informação e documentação – artigo em publicação periódica científica impressa – Apresentação.
- **ABNT NBR 6023:2002:** Informação e documentação – Referencia – Elaboração.
- **ABNT NBR 6024:2012:** Informação e documentação – Numeração progressiva das seções de um documento – Apresentação.
- **ABNT NBR 6027:2012:** Informação e documentação – Sumário – Apresentação.
- **ABNT NBR 6028:2003:** Informação e documentação – Resumo – Apresentação.
- **ABNT NBR 6029:2006:** Informação e documentação – Livros e folhetos – Apresentação.
- **ABNT NBR 6034:2004:** Informação e documentação – Índice – Apresentação.
- **ABNT NBR 10520:2002:** Informação e documentação – Citações.
- **ABNT NBR 10719:2011:** Informação e documentação – Relatório técnico e/ou científico – Apresentação.
- **ABNT NBR 14724:2011:** Informação e documentação – Trabalhos acadêmicos – apresentação.
- **ABNT NBR 15287:2011:** Informação e documentação – Projeto de pesquisa – Apresentação.

Todos esses modelos podem ser compilados com programas que “rodam” \LaTeX através do pacote `abnTeX2`, provendo uma versão mais completa e atualizada da suíte.

O pacote `abnTeX2` possui 4 elementos principais são eles:

- A classe de formação de trabalhos acadêmicos `abnTeX2`;
- O pacote de citações bibliográficas `abnTeX2cite`;
- Modelos canônicos de uso do `abnTeX2`.
- Especificações de formatação de referências bibliográficas `abnTeX2`

Os modelos canônicos são exemplos de uso do `abnTeX2` que acompanham a instalação do software.

3.3.2 Por que utilizamos abnTeX

Por força normativa direcionada pela ABNT, a qualidade dos trabalhos acadêmicos depende não só do conteúdo nele incluso como também e edição e formatação dos trabalhos, sendo assim, com a necessidade de um modelo que atenda a padronização das normas ABNT, adotamos o pacote de macros abnTeX, e conseqüentemente a sua versão mais atual abnTeX2 para o nosso trabalho.

Devido ao seu caráter de software livre, ele pode ser melhorado, aperfeiçoado e customizado de acordo com as necessidades específicas, podendo ser construído um pacote ou classe que se adapte ao padrão específico de cada Universidade, sendo necessário apenas que o nome seja alterado e que o devido crédito seja dado aos seus autores nos termos do “The Latex project public License”.

O abnTeX2 traz também o suporte para a produção de documentos em diferentes idiomas, e ainda não há problemas de compatibilidade com a versão original do abnTeX, não causando nenhum problema para computadores que possuam as duas versões instaladas.

3.4 OPEN DOCUMENT TEXT – ODT

O ODT nada mais é do que um padrão de documento de texto, como o DOC ou DOCX, cujo proprietário é a Microsoft, sendo que o ODT faz parte do padrão Open Document Format (ODF), que foi desenvolvido de forma totalmente descentralizada, por diversas empresas, sendo reconhecido pela ISO e ABNT, tendo seu formato totalmente aberto, possibilitando assim o acesso a suas especificações de forma livre.

O TCCTeX utiliza o formato ODT como opção de entrada de arquivos de texto, por ele ser um formato aberto estável (última versão lançada há quatro anos) e ter total compatibilidade com os principais processadores de texto gratuitos como o LibreOffice Writer e o OpenOffice Writer, assim permitindo que o usuário utilize nosso software sem necessitar pagar por um processador de texto comercial como o Microsoft Word. O ODT também é compatível com o Microsoft Word, que é o líder do mercado de processadores de texto, porém caso o usuário utilize o Microsoft Word indicamos a ele que use o formato de arquivos DOCX, pois o TCCTeX também é compatível com este formato.

3.5 OFFICE OPEN XML DOCUMENT – DOCX

A extensão docx (Office Open XML Document) faz parte de um grupo de três

formatos Office Open XML que foram desenvolvidos pela gigante de tecnologia Microsoft para serem usadas pela sua suíte de aplicativos para escritório Microsoft Office 2007, sendo usada em todas as versões posteriores do Microsoft Office como o principal formato de arquivos. Os formatos Office Open XML são:

- Office Open XML Document cuja extensão é o .docx e foi desenvolvida para o Microsoft Word;
- Office Open XML Workbook cuja extensão é o .xlsx e foi desenvolvida para o Microsoft Excel;
- Office Open XML Presentation cuja extensão é o .pptx e foi desenvolvida para o Microsoft PowerPoint.

Esses formatos de arquivos do pacote Office, construídos a partir da linguagem de marcação XML (eXtensible Markup Language), são open format, ou seja, não é necessário pagar uma licença comercial para utilizá-los. O XML, como também HTML (HyperText Markup Language), uma linguagem de marcação que compõe a base das páginas Web, são originadas do SGML (Standard Generalized Markup Language), uma metalinguagem que tem a função de definir outras linguagens de marcação para documentos.

A SGML inovou ao permitir o compartilhamento de documentos por máquinas que faziam parte de um mesmo projeto, estes arquivos se encontravam disponíveis e legíveis por longos anos, ainda a facilidade de impressão fez com que essa tecnologia logo conquistasse Governos e Industrias.

A tecnologia XML surgiu com o propósito de simplificar o SGML, criando uma estrutura única para outras linguagens, além de estendê-la a outras plataformas como RSS, XML-RPC e SOAP. O seu propósito foi alcançado e a partir dela outras linguagens surgiram como XHTML, XSIL, SVG, SMIL, MathML, RDF, XBRL, SDMX e NCL. Os novos formatos de documentos facilitaram o gerenciamento de arquivos e dados.

Os aplicativos que possuem suporte ao XML possibilitam a essa tecnologia a conversão dos seus binários no novo formato. Os requisitos de segurança também foram atendidos com essa tecnologia, o padrão XML é basicamente texto sem formatação, sendo assim, não possuem uma estrutura que possa ser identificada como maliciosa por firewalls. Essas características fizeram com que os aplicativos derivados dessa linguagem ganhassem o público em geral.

Quando se utiliza o pacote Office, que como dito acima, se baseia na linguagem XML, os arquivos criados a partir desses aplicativos como Microsoft Excel, por exemplo, adiciona-se a letra “x” ou “m” aos nomes dos documentos, quando os arquivos possuem a

letra “x” significa um arquivo XML sem macros, já os arquivos que utilizam a letra “m” significa que possui macros.

3.5.1 Vantagens e desvantagens do docx

As vantagens do docx para sua utilização neste projeto são:

- É o principal formato do Microsoft Word, processador de texto líder de mercado;
- É baseado na linguagem de marcação XML, formato amplamente aceito por desenvolvedores em todo o mundo;
- É um formato estável (ultima versão lançada há três anos) mantido pela gigante Microsoft.

As desvantagens do docx para sua utilização neste projeto são:

- O único processador de texto que possui total compatibilidade com ele é o Microsoft Word;
- O código XML gerado para seu funcionamento é muito burocrático.

3.5.2 Por que utilizamos a extensão .docx

Decidimos utilizar o docx como opção de entrada de arquivos de texto por ele ser utilizado pelo Microsoft Word, o processador de texto mais utilizado pelo usuário comum, e conseqüentemente ser o formato mais comum utilizado em todo o tipo de documentos, profissionais ou pessoais, acadêmicos ou informais.

3.6 APIS UTILIZADAS

API (*Application Programming Interface*) ou interface de programação de Aplicações, são conjuntos de funções pré-escritas que podem ser acessadas por aplicativos para que os mesmos ganhem novas funcionalidades sem que o programador tenha que reescrever código já existente, assim podendo agilizar o processo de desenvolvimento.

3.6.1 Java L^AT_EX Report

A API open source JLR (Java L^AT_EX Report) foi desenvolvida pela empresa Nixo Soft, e faz parte das bibliotecas utilizadas no TCC_{TeX}, sua função é a de realizar o preenchimento de *templates* L^AT_EX já desenvolvidos, estes *templates* contêm código L^AT_EX misturados com a linguagem Apache Velocity, por meio desta linguagem é possível que o JLR encontre uma lógica de preenchimento no arquivo LaTeX e o preencha, além disso a biblioteca JLR contém todo código necessário para que o documento L^AT_EX possa ser convertido para PDF.

Figura 1 – Template L^AT_EX com Linguagem Apache Velocity

```
\begin{titlepage}
  \vfill
  \begin{center}
    {\Large\MakeUppercase{$Instituicao}\par
      \MakeUppercase{$Faculdade}\par
      \MakeUppercase{$Curso}} \\\[3.5cm]
    {\foreach( $name in $Autores )
      \Large\MakeUppercase {$name}\par
      #end} \\\[3.5cm]
    {\Huge {$Titulo}}\\\[1cm]
    \vfill
    \MakeUppercase{$Cidade}\\\{$Ano}
  \end{center}
\end{titlepage}
```

Fonte: Autoria própria

3.6.2 Writer2Latex

A API open source Writer2L^AT_EX está sendo utilizada no desenvolvimento do TCC_{TeX}, pois reconhece todos textos, tabelas e figuras que existem em um documento ODT (*open document text*) e a partir desse reconhecimento converte o arquivo para L^AT_EX.

Essa conversão só é possível por meio de um arquivo de configuração, onde deverá estar escrito como o documento L^AT_EX deverá se comportar, como por exemplo: não deverá existir parágrafos sem texto ou muitos pulos de linha, capítulos e sub capítulos deverão ser substituídos pelo código declarado, entre outras coisas.

Figura 2 – Exemplo de Configuração Writer2Latex

```
<?xml version="1.0" encoding="utf-8" ?>
<config>

<option name="no_preamble" value="true" />

<option name="ignore_hard_page_breaks" value="true" />
<option name="ignore_hard_line_breaks" value="true" />
<option name="ignore_empty_paragraphs" value="true" />

<option name="page_formatting" value="ignore_all" />
<option name="save_images_in_subdir" value="true" />
<heading-map max-level="4">
<heading-level-map writer-level="1" name="chapter" level="0" />
<heading-level-map writer-level="2" name="section" level="1" />
<heading-level-map writer-level="3" name="subsection" level="2" />
<heading-level-map writer-level="4" name="subsubsection" level="3" /
</heading-map>
<custom-preamble />
```

Fonte: Autoria própria

O Writer2Latex consegue fazer o seu papel de converter um arquivo *Opendocument* para \LaTeX muito bem, mas adiciona ao documento gerado várias linhas de código desnecessárias para nosso projeto, além disso não consegue reconhecer padrões de legendas de figuras e tabelas, nem deixar o documento gerado no padrão ABNT, mas o TCCTeX conseguirá suprir todos estes problemas. null

4 DESENVOLVIMENTO DO PROJETO

A metodologia de *Extreme Programming* (XP) foi escolhida, devido ao fato de estar muito relacionada a agilidade e flexibilidade, características essas que condiziam com a forma como desejávamos trabalhar.

Logo em seguida notamos que, antes de iniciar o desenvolvimento do software, deveríamos dividir nossa equipe em três frentes de trabalho, para que pudéssemos desenvolver o projeto com mais agilidade e eficiência. Elas se comunicariam com frequência e trocariam suas experiências pesquisas e conclusões a cada etapa que fosse superada. As três frentes de trabalho que eram divididas em A, B e C tinham as seguintes funções:

- Frente de trabalho A: Verificar quais as exigências da Metodista em relação à padronização ABNT e planejar como nosso software ira auxiliar o usuário a atender essas exigências.
- Frente de trabalho B: Pesquisar como funciona a metodologia escolhida para desenvolver o software e preparar a equipe do projeto para aplicar essa metodologia.
- Frente de trabalho C: Pesquisar e aprender como utilizar as tecnologias que iriam permitir o desenvolvimento do TCCTeX e como seria feita a interação dessas tecnologias.

Essas três frentes de trabalho foram, como previsto, naturalmente se complementando e conforme as atividades de uma frente avançavam elas alcançavam um nível o qual dependeriam da interação com outra frente de trabalho para se desenvolver.

A primeira frente de trabalho que alcançou esse nível foi a frente de trabalho A que conseguiu captar as necessidades que o software deveria suprir e planejar como ele iria interagir com o usuário desenvolvendo protótipos visuais do TCCTeX. A partir dessa conclusão a frente de trabalho A passou a trabalhar em conjunto com a frente C para encontrar as tecnologias e ferramentas que viabilizariam o desenvolvimento do software dentro desses parâmetros e discutir possíveis alterações dos protótipos em prol de melhorias nas funcionalidades do TCCTeX.

Em seguida a frente de trabalho B conseguiu estabelecer como iriamos aplicar a metodologia XP em nosso projeto e passou a toda a equipe do projeto como seriam implantada a metodologia e como ela iria auxiliar no desenvolvimento do software.

A frente de trabalho C, por sua vez conseguiu encontrar diversas tecnologias que poderiam possibilitar no desenvolvimento do software e passou a testa-las para verificar a eficiência de cada uma delas e escolher quais melhor se adequavam com as necessidades do software. A frente de trabalho C só concluiria esse processo de pesquisa e escolha durante o desenvolvimento do software, pois seria durante o desenvolvimento do software que essas tecnologias e a capacidade de interação delas seriam realmente posta a prova.

4.1 APLICAÇÃO DA METODOLOGIA XP

Com base em nossos estudos sobre Extreme Programming (XP) notamos que essa metodologia requer contato direto e recorrente do cliente do projeto em desenvolvimento e neste quesito notamos uma diferença entre o nosso projeto e outros projetos desenvolvidos utilizando XP, pois como alunos da Universidade Metodista de São Paulo, nós podemos assumir o papel tanto de clientes desse projeto como de desenvolvedores do mesmo.

A partir do momento em que começamos a aplicar XP na prática tivemos que nos conscientizar dos dois papéis que exercemos e como lidar com essa dualidade sem prejudicar o andamento do projeto nem no produto final.

Para suprir a impossibilidade da equipe de nos reunir e compartilhar um espaço de trabalho único, portanto, não termos um local físico para colocarmos um quadro de tarefas ou de cartões de história, utilizamos a ferramenta de anotações Microsoft Office OneNote combinado com o serviço de armazenamento e partilha de arquivos Dropbox. Essa união nos possibilitou a flexibilidade do planejamento do projeto tal qual é previsto na metodologia XP.

Para iniciar o projeto aplicando XP deveríamos assumir o papel de clientes e escrever cartões de história (User Stories) contando o que um aluno da UMESP que está desenvolvendo a documentação do seu TCC deseja que o software TCCTeX faça por ele. Para escrever as histórias de usuário tínhamos o seguinte modelo:

Figura 3 – Modelo de User Stories



Num:
Custo:
História:

Fonte: Autoria própria

O qual deveria ser usado seguindo a seguinte referencia:

- **Num:** número de identificação do cartão.
- **Custo:** estimativa de custo de tempo de 1 à 3 semanas que deve ser feita em grupo.
- **História:** nós, no papel de clientes, deveríamos escrever histórias de o que gostaríamos que o software fizesse evitando termos técnicos.

Cada uma dessas histórias representa uma parte de software que pode ser entregue a parte. Após escrever as histórias nós em grupo, assumindo o papel de desenvolvedores deveríamos fazer uma Reunião de Planejamento de Releases (entregas, liberações do software) onde estimávamos o tempo que levaríamos para transformar cada uma dessas histórias em código e em seguida escolhíamos quais histórias desenvolveríamos primeiro.

Estas histórias eram organizadas, já na ordem de desenvolvimento em um documento do One Note chamado Plano de Liberação que continha também os Testes Funcionais que eram desenvolvidos durante as Reuniões de Planejamento de Iterações.

Os Testes Funcionais representam o resultado esperado do sistema. Cada história possui um ou mais Testes Funcionais que eram ser aplicados para verificar se essa história foi completamente implementada. Cada um dos Testes Funcionais possuía uma caixa de seleção que tinham a seguinte função:

Figura 4 – Caixa de Seleção dos Testes Funcionais

☐ Indica que o Teste não foi aceito.

☒ Indica que o Teste foi aceito.


Fonte: Autoria própria

Fizemos Testes Funcionais simples e de fácil verificação visual.

Decidimos que faríamos iterações de uma semana com Reuniões de Planejamento de Iteração no inicio de cada semana. As reuniões tinham como objetivo dividir histórias em tarefas, decidir quais tarefas seriam desenvolvidas durante a semana e escrever Testes Funcionais. Fazíamos também reuniões ao final de cada semana que eram utilizadas para verificar os resultados da semana e propor possíveis novas tarefas a partir das experiências da semana que se passou.

As tarefas que são planejadas são escritas seguindo o seguinte modelo:

Figura 5 – Modelo de Cartão de Tarefa

A imagem mostra um modelo de cartão de tarefa dentro de uma janela de aplicativo. O cartão é um retângulo branco com uma borda cinza. No topo da janela, há uma barra de título com o texto "*****" e ícones de minimizar, maximizar e fechar. O cartão contém três linhas de texto em negrito: "Num:", "Funcionalidade:" e "Para".

Fonte: Autoria própria

O qual deveria ser usado seguindo a seguinte referencia:

- **Num:** neste campo será colocado um numero será utilizado para fazer a referencia entre o cartão de história de usuário e cada tarefa.
- **Funcionalidade:** aquilo que deve ser desenvolvido.
- **Para:** motivação pela qual se deseja que essa tarefa seja implementada. É utilizado para evitar que o uma funcionalidade desnecessária seja desenvolvida.

Todas as tarefas eram organizadas em um documento do One Note chamado Tarefas do TCCTeX que era dividido em três colunas:

- **Não Implantadas:** tarefas que devem ser desenvolvidas, mas que ainda não foram designadas para nenhuma iteração;
- **Em Iteração:** tarefas que foram designadas para a próxima iteração;
- **Finalizadas:** tarefas que já passaram por iterações e estão finalizadas.

Durante as Reuniões de Planejamento de Iteração, quando decidíamos que iríamos desenvolver certa tarefa durante a próxima iteração nós movíamos o cartão desta tarefa para a coluna “Em Iteração” e em seguida copiávamos essa tarefa para o documento do One Note chamado Plano de Iteração.

O Plano de Iteração era dividido horizontalmente em semanas/iterações e verticalmente em três colunas:

- **Não Implantadas:** tarefas que devem ser desenvolvidas;

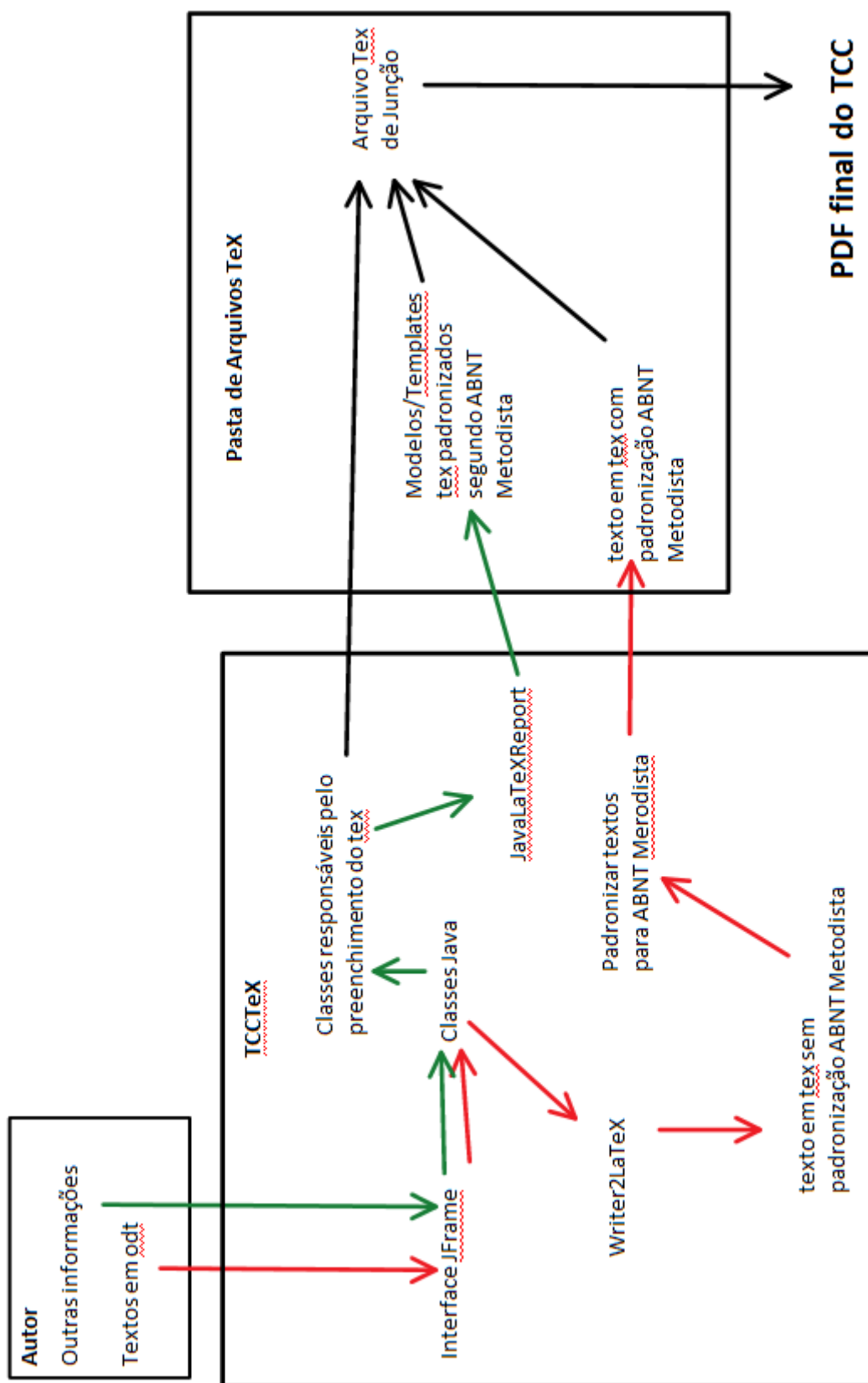
- **Em Desenvolvimento:** tarefas que estão sendo implementadas por um dos desenvolvedores;
- **Finalizadas:** tarefas que já tiveram suas funcionalidades desenvolvidas.

4.2 ARQUITETURA DO SOFTWARE UTILIZANDO ODT

O software tem uma arquitetura complexa em decorrência da necessidade do TCC-CTeX tem suprir a de grande quantidade de normas exigidas pela para a elaboração da documentação do TCC pela Universidade Metodista de São Paulo.

Abaixo disponibilizamos uma figura que demonstra a arquitetura criada para gerar os arquivos .tex a partir dos arquivos .odt e algumas informações específicas.

Figura 6 – Parte da arquitetura do software TCCTeX



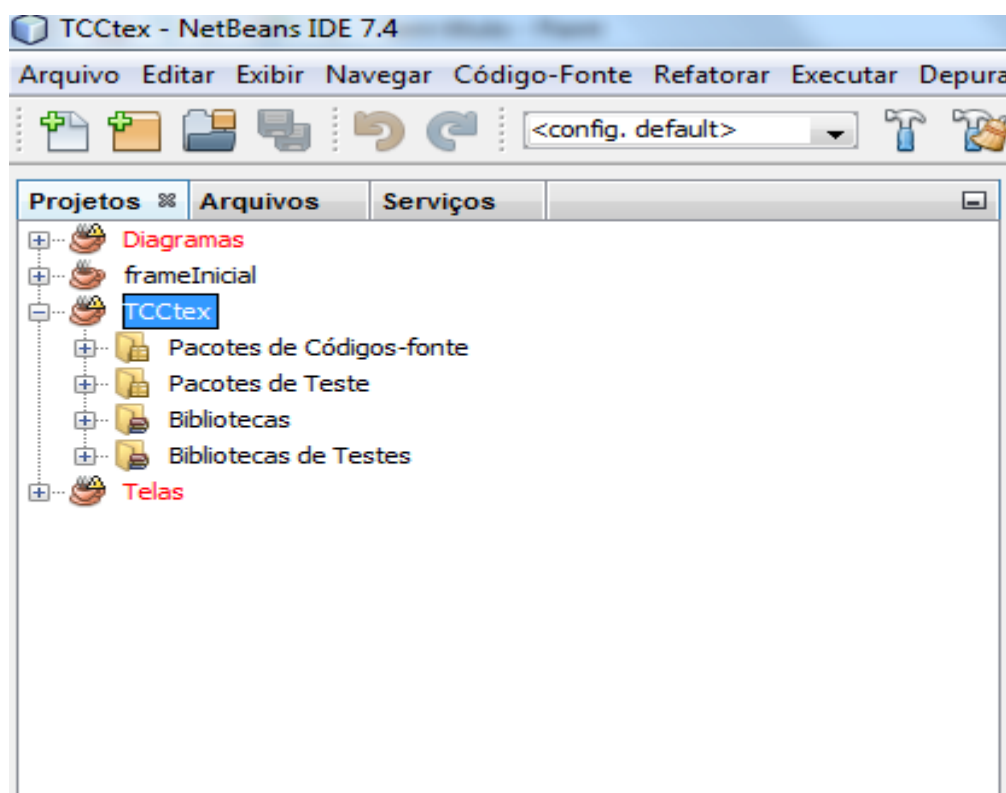
Fonte: Autoria própria

4.3 FUNCIONAMENTO DO TCCTEX

Neste subcapítulo explicaremos o funcionamento do software TCCTeX.

O programa se estrutura em 4 pacotes, pacotes de códigos-fonte (padrão do java), pacotes de teste, bibliotecas (padrão do java), bibliotecas de testes.

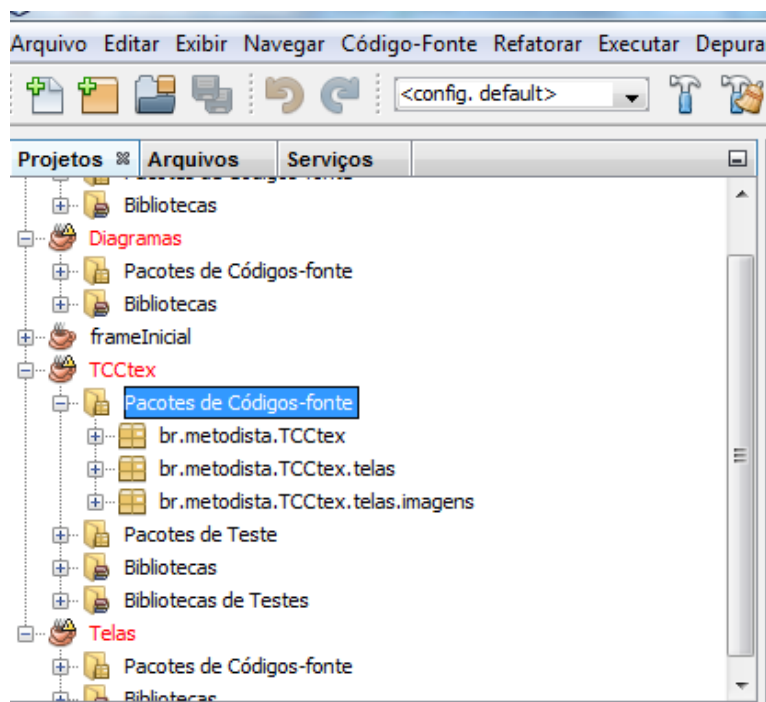
Figura 7 – Projeto TCCTeX



Fonte: Autoria própria

Pacotes de códigos-fonte são utilizados para abrigar os pacotes divididos em responsabilidades, no software TCCTeX temos 3 pacotes principais, br.metodista.TCCTex, br.metodista.TCCTex.telas, br.metodista.TCCTex.telas.imagens.

Figura 8 – Pacotes do TCCTeX

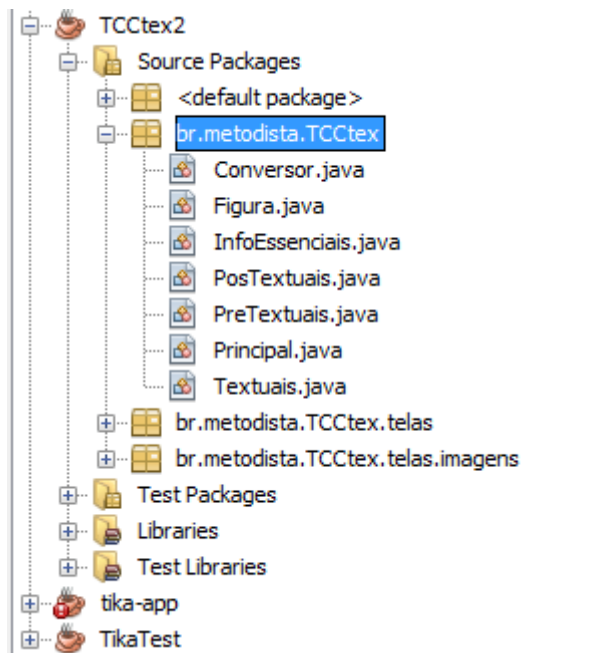


Fonte: Autoria própria

4.3.1 Pacote br.metodista.TCCTex

Este pacote contém as classes java Conversor, Figura, infoEssenciais, PreTextuais, Principal e Textuais.

Figura 9 – Pacote br.metodista,TCCTeX



Fonte: Autoria própria

4.3.1.1 Classe Conversor

Esta classe é utilizada para conversão, padronização e geração de arquivos no formato tex e pdf. Na classe Conversor, como em todas as outras, se faz necessário a importações de pacotes.

Figura 10 – Importações da Classe Conversor

```
import java.io.File;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.UnsupportedEncodingException;

import de.nixosoft.jlr.JLRGenerator;
import de.nixosoft.jlr.JLR opener;

import writer2latex.api.Config;
import writer2latex.api.Converter;
import writer2latex.api.ConverterFactory;
import writer2latex.api.ConverterResult;
```

Fonte: Autoria própria

Utilizamos pacotes para organizar as classes, que atuam com um mesmo grupo de características, o pacote `de.nixosoft.jlr.JLRGenerator` por exemplo é utilizado no preenchimento de templates feitos em \LaTeX , como citado anteriormente, já o pacote `nixosoft.jlr.JLR opener` é utilizado para conversão de documentos \LaTeX para PDF e possui um método `open(File)` para abrir mais rapidamente o PDF gerado.

Figura 11 – JLR para Geração de PDF

```

////////// GERADOR DE ARQUIVO PDF APARTIR DE ARQUIVO TEX ////
public void gerarPDF(File caminho) throws UnsupportedEncodingException,
] FileNotFoundException, IOException{
    File tex, pdf, workingDirectory, salvarEm;
    String current = new java.io.File( "." ).getCanonicalPath();
    tex = new File(current+"\\LaTex\\tcc.tex");
    workingDirectory = caminho;
    salvarEm = caminho;

    JLRGenerator pdfGen = new JLRGenerator();

    pdfGen.generate(tex, salvarEm, workingDirectory);|
    pdfGen.generate(tex, salvarEm, workingDirectory);

    pdf = pdfGen.getPDF();
    JLR opener.open(pdf);
- }
}

```

Fonte: Autoria própria

O pacote `writer2latex.api` possui a classe *Config* que armazena o local do arquivo xml de configuração do `writer2 \LaTeX` e algumas de suas configurações podem ser escritas diretamente na classe java como no exemplo abaixo onde `config.setOption("inputencoding","UTF-8")` significa que os arquivos ODT devem ser abertos utilizando o padrão de codificação UTF-8.

Figura 12 – Método odtToTex

```
//Converte arquivo odt para Tex
public File odtToTex(File arquivo) throws IOException{

//Pega nome da File (Ex: desktop/pasta/documento para documento
String nomeArquivo = arquivo.getName().replace( ".odt", "" );
// Cria um conversor LaTeX
Converter converter =
    ConverterFactory.createConverter("application/x-latex");

// Configura o conversor por meio do arquivo (myoconfig.xml)
Config config = converter.getConfig();
config.read(new File("myconfig.xml"));
config.setOption("inputencoding","UTF-8");

// Converte o documento odt para tex)
ConverterResult result =
    converter.convert(arquivo,
        nomeArquivo+".tex");
// Escreve o arquivo convertido
result.write(new File("LaTex"));
//Apaga todo código ruim gerado pelo Writer2LaTeXConversorvo.apaga
return new File("LaTex\\"+nomeArquivo+".tex");
```

Fonte: Autoria própria

A classe *Converter* é utilizada para conversão do arquivo ODT para um texto no padrão \LaTeX , já a classe *ConverterResult* recebe esse texto e o escreve em um arquivo com a extensão tex.

Figura 13 – Parte do Método alterarTex()

```
public String alterarTex(File arquivo, String tipo)
    throws FileNotFoundException, IOException{
    String comeco, meio = "", fim, texto;

    /* Abre arquivo tex utilizando o formato de arquivo utf-8
     * para que seja preservada toda acentuação e caracteres do arquivo*/
    Scanner scanner = new Scanner(new FileInputStream(
        arquivo.getAbsolutePath()), "utf-8")
        .useDelimiter("\\|\\|\\|\\n");

    //Se tipo de arquivo for igual resumo ou abstract
    if(tipo.equals("resumo") || tipo.equals("abstract")){
        //Adiciona Título (Tag) na variável: comeco (Ex: /begin {resumo})
        comeco="\\begin{" + tipo + "}\\n";

        //Adiciona Título Final (Tag) na variável: fim (Ex: /end {resumo})
        fim="\\n\\end{" + tipo + "}";

        // Lê primeira linha do tex e armazena na variável: primeiraLinha
        String primeiraLinha = scanner.nextLine();

        /* Caso primeira linha não for \\begin{resumo} por exemplo, este método
         * continua a rodar, para adicionar as linhas de começo e fim */
        if(!primeiraLinha.equals("\\begin{"+tipo+"}")){
            texto = juntarComecoMeioFim(arquivo, comeco, meio, fim);
        }
    }
```

Fonte: Autoria própria

Seguindo na classe `Conversor`, temos um outro método, chamado `alterarTex()`, que possui algumas funções como ler o arquivo convertido pelo `Writer2LATEX`, verificar se o arquivo é o resumo, abstract, dedicatória, etc, adicionar o título se necessário conforme o tipo de arquivo e utilizar outros métodos para remover código desnecessário gerado pelo `Writer2LATEX` e reescrever o arquivo `tex`.

4.3.1.2 Classe `InfoEssenciais`

A classe `InfoEssenciais` recebe várias informações necessárias para geração da capa, folha de rosto e folha de aprovação, além de gerar uma ficha catalográfica não preenchida para que prévias do TCC possam ser entregues, no próximo exemplo é demonstrado um método utilizado para geração desses arquivos.

Figura 14 – JLR para Preenchimento de Template Tex

```
1 private void preencherLayout(File t) throws IOException{
    File workingDirectory = new File("Templates\\");
    JLRConverter converter = new JLRConverter(workingDirectory);
    File template = new File("LaTex\\" + t.getName());
    File gerado = new File("LaTex\\" + t.getName());

    converter.replace("Faculdade", faculdade);

    ArrayList<ArrayList<String>> services = new ArrayList<ArrayList<String>>();
    services.add(autor);
    converter.replace("Instituicao", instituicao);
    converter.replace("Autores", autor);
    converter.replace("Curso", curso);
    converter.replace("Titulo", titulo);
    converter.replace("Ano", ano);
    converter.replace("Cidade", cidade);
    converter.replace("TipoCurso", tipoCurso);
    converter.replace("Orientador", orientador);
    converter.replace("Banca", banca);
    converter.parse(template, gerado);
- }
```

Fonte: Autoria própria

4.3.1.3 Classe `PreTextuais`

A classe `PreTextuais` recebe os arquivos `Odt` contendo o resumo, abstract, dedicatória, agradecimentos e epigrafe e faz a conversão dos mesmos para `tex` se os arquivos obrigatórios resumo e abstract existirem.

Figura 15 – Método converterPre()

```

public void converterPre() throws IOException{
    try{
        if( resumoPtBr.exists()
            && resumoEnUs.exists() ){
            this.gerarPre(this.resumoPtBr, "resumo");
            this.gerarPre(this.resumoEnUs, "abstract");
            if(dedicatoria.exists())
                this.gerarPre(this.getDedicatoria(), "dedicatoria");
            if(agradoecimento.exists())
                this.gerarPre(this.agradoecimento, "agradoecimento");
            if(epigrafe.exists())
                this.gerarPre(this.epigrafe, "epigrafe");
        }
    }
}

```

Fonte: Autoria própria

4.3.1.4 Classe Textuais

A classe Textuais possui algumas variáveis que guardam o caminho de cada arquivo utilizado na etapa de Textuais, são estas as variáveis introdução e conclusão do tipo *File* e a lista de capítulos também do tipo *File*.

Figura 16 – Classe textuais

```

public class Textuais {

    private File introducao;
    private File conclusao;
    private LinkedList<File> capitulos = new LinkedList();
    Conversor a = new Conversor();
    int i;
}

```

Fonte: Autoria própria

O método listarImagens() da classe Textuais reconhece todas legendas de figuras criadas no software Microsoft Word e coloca no padrão \LaTeX corretamente para que possa ser gerada a numeração de cada figura e a lista contendo todas. No pequeno trecho retirado do método listarImagens(), mostrado na próxima figura, é demonstrado como as

legendas são reconhecidas pelo software para que posteriormente sejam tratadas e gerado o código correto das mesmas.

Figura 17 – Método converterTextuais()

```
public boolean converterTextuais() throws IOException{
    try{
        if (this.verificarTextuais()){
            //Converte todos capitulos
            for( i= this.getCapitulos().size()-1; i>=0 ; i--){
                a.odtToTex(this.getCapitulos().get(i));
                this.listarTabelas(this.getCapitulos().get(i));
                this.listarImagens(this.getCapitulos().get(i));
            }
            //Converte introdução
            a.odtToTex(this.getIntroducao());
            //Converte conclusão
            a.odtToTex(this.getConclusao());
            System.out.println("Textuais foram convertidos");
            return true;
        }
        return false;
    }
}
```

Fonte: Autoria própria

4.3.1.5 Classe Principal

Figura 18 – Parte do método abrir() Principal

```
public void abrir(File arquivo) throws FileNotFoundException{
    //Verifica se arquivo existe
    if(arquivo.exists()){
        //Verifica se arquivo termina com .tcc
        if(arquivo.getAbsolutePath().toString().endsWith(".tcc")){
            //Reseta lista de autores e capitulos
            TelaPrincipal.t1.resetListas();
            TelaPrincipal.t3.resetCapitulos();
            //Le Arquivo
            Scanner s = new Scanner(new FileInputStream(
                arquivo.getAbsolutePath()), "utf-8");
            /---Preenche campos da Tela Informações Essenciais pegando cada linha do arquivo---
            TelaPrincipal.t1.setjTInstituicao(s.nextLine());
            TelaPrincipal.t1.setjTFaculdade(s.nextLine());
            TelaPrincipal.t1.setjTCurso(s.nextLine());
            TelaPrincipal.t1.setjTCidade(s.nextLine());
            TelaPrincipal.t1.setjCTipoCurso().setSelectedIndex(Integer.parseInt(s.nextL

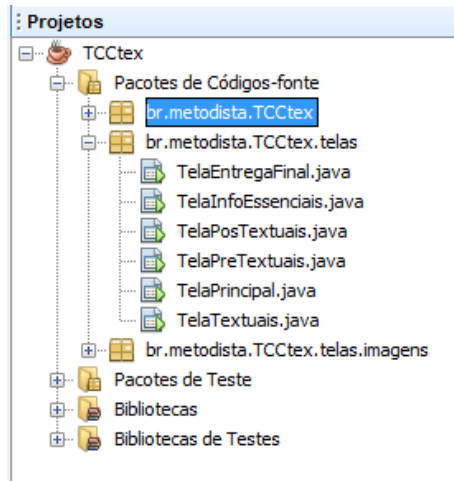
            TelaPrincipal.t1.setjTTitulo(s.nextLine());
            TelaPrincipal.t1.setjTSubTitulo(s.nextLine());
            TelaPrincipal.t1.setjTAutores(s.nextLine());
            TelaPrincipal.t1.setjTOrientador(s.nextLine());
            TelaPrincipal.t1.setjTCoorientador(s.nextLine());
```

Fonte: Autoria própria

A Classe principal tem a função de gerar o arquivo que salva o projeto desenvolvido pelo usuário e de carregar esse mesmo arquivo, assim o usuário pode salvar o um projeto em andamento e continuar mais tarde.

4.3.2 Pacote br.metodista.TCCtex.telas

Figura 19 – Pacote Contendo as Telas

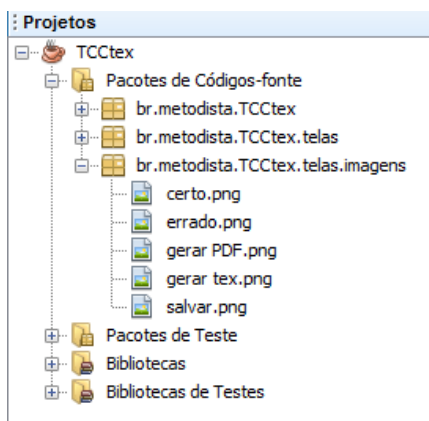


Fonte: Autoria própria

O pacote `br.metodista.TCCtex.telas`, possui os formulários da aplicação, esses formulários são JFrames, que nada mais é que uma classe, especializada em componentes visuais, como botões, menus, caixas de texto, e tudo que existem em janelas de aplicativos.

4.3.2.1 Pacote br.metodista.TCCTex.telas.imagens

Figura 20 – Pacote Contendo Imagens



Fonte: Autoria própria

As boas práticas da linguagem java, tanto em aplicações desktop, quanto WEB, sugerem que os arquivos de imagens fiquem em um pacote dedicado, na nossa aplicação

esse pacote é o `br.metodista.TCCtex.telas.imagens`, dentro dele temos as imagens usadas no aplicativo.

4.4 INTERFACES

O design e funcionamento das interfaces foram feitos de forma a garantir o processo de aprendizagem e uso o mais simples possível, para o usuário final.

4.4.1 INTERFACE DA TELA PRINCIPAL

A tela principal é composta por: botões de cada etapa a ser realizada para que o trabalho possa ser finalizado, sendo indicado se determinada etapa foi concluída ou não ao seu lado por uma imagem; botão Salvar, para guardar todas as informações preenchidas pelo usuário em todas as interfaces gráficas; botão Gerar PDF para converter o trabalho gerado de \LaTeX para PDF, salvando-o no local onde o usuário desejar.

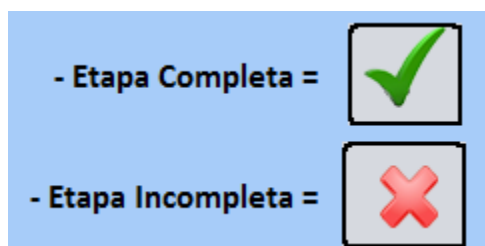
Figura 21 – Interface da Tela Principal



Fonte: Autoria própria

Quando o usuário conclui determinada etapa a imagem ao lado da opção da etapa é alterada para indicar que a mesma foi concluída.

Figura 22 – Etapa Concluída



Fonte: Autoria própria

4.4.2 INTERFACE DA TELA DE INFORMAÇÕES ESSENCIAIS

A tela de informações essenciais possui os campos de texto: nome da instituição, faculdade, curso, cidade da instituição, título do trabalho, subtítulo, autores, orientador, coorientador(es) e ano de entrega. Também possui o botão voltar e as caixas de seleção: tipo de curso, natureza e terminei esta parte que ao ser clicada verifica se o usuário preencheu todas informações necessárias, caso ele tenha preenchido, são gerados arquivos L^AT_EX para a capa, ficha catalográfica, folha de aprovação e folha de rosto, retornando o usuário para a tela inicial.

Figura 23 – Interface da Tela de Informações Essenciais

Fonte: Autoria própria

4.4.3 INTERFACE DA TELA DE PRÉ-TEXTUAIS

Na tela de pré-textuais o usuário deverá inserir arquivos com o resumo do trabalho em português e inglês (*abstract*), poderá também inserir arquivos de texto como a errata,

dedicatória, agradecimentos e epígrafe, sendo que não deverá escrever o título de cada parte do trabalho nos documentos. O usuário poderá abrir os documentos após selecioná-los para efetuar modificações clicando no botão: abrir. A caixa de seleção: Terminei esta parte irá verificar se o usuário inseriu todos arquivos obrigatórios, assim os convertendo para L^AT_EX no padrão ABNT utilizado pela metodista.

Figura 24 – Interface da tela de Pré-Textuais

Voltar

Pré Textuais

Obrigatório

* **Resumo em Português:** TestesArquivos\Resumo.odt ... **Abrir**

* **Resumo em Inglês:** TestesArquivos\Abstract.odt ... **Abrir**

Opcionais

Errata: ... **Abrir**

Dedicatória: TestesArquivos\Dedicatoria.odt ... **Abrir**

Agradecimentos: TestesArquivos\Agradecimentos.odt ... **Abrir**

Epígrafe: TestesArquivos\Epigrafe.odt ... **Abrir**

☐ **Terminei esta parte**

Fonte: Autoria própria

4.4.4 INTEFACE DA TELA DE TEXTUAIS

Nesta tela o usuário deverá inserir arquivos ODT contendo: a introdução, todos capítulos separados e conclusão, o arquivo onde estará escrito cada um, podendo conter tabelas e imagens, também poderá adicionar novos capítulos, remove-los e abrir o arquivo para edição.

Ao clicar na opção: terminei esta parte os arquivos inseridos serão convertidos para \LaTeX , no padrão ABNT da Metodista.

Figura 25 – Interface da Tela de Textuais

The interface is a web-based form with a light blue background. It features a menu bar with 'File' and 'Edit'. The main content area is divided into three sections, each with a title and a form element:

- Introdução:** Contains a text input field labeled 'Selecione o Arquivo de Introdução' and a button labeled 'Abrir'.
- Capítulos:** Contains a text input field labeled 'Selecione o Arquivo e clique em Adicionar' and a button labeled 'Adicionar'. Below this is a table with three columns: 'Numero', 'Arquivo', and 'Caminho'. At the bottom of this section are three buttons: 'Abrir', 'Alterar', and 'Remover'.
- Conclusão:** Contains a text input field labeled 'Selecione o Arquivo de Conclusão' and a button labeled '...'.

At the bottom of the interface is a checkbox labeled 'Terminei esta parte'.

Fonte: Autoria própria

4.4.5 INTERFACE DA TELA PÓS-TEXTUAIS

Por meio desta tela será possível inserir um arquivo com todas referências citadas no texto, para isso o usuário poderá utilizar o sistema Online MORE para criar as citações e referências e colar as mesmas no documento a ser inserido. Alguns anexos são obrigatórios, sendo o Relatório de recomendações da banca de qualificação (RRBQ) um

deles. O anexo contendo todas Atas e Cronogramas desenvolvidos durante o curso devem ser entregues no desenvolvimento da entrega final do TCC, sendo assim podem ser adicionados ao documento, caso o usuário não esteja fazendo uma prévia do trabalho.

Figura 26 – Interface da Tela Pós-Textuais

Arquivo Sobre

Voltar

Pós Textuais

Obrigatório

* Referências: ... **Abrir**

- Para fazer as referências utilize o sistema MORE [Clicando Aqui](#)
- Cole todas referências em um arquivo ODT como [neste exemplo](#)
Atenção: Não coloque título no documento de Referencias

* RRBQ: Seleccione o Arquivo em: DOC, DOCX, ou pdf ... **Adicionar**

Atas e Cronograma: Seleccione o Arquivo compactado em ZIP com as atas e cron... ... **Adicionar**

Atenção: As Atas e Cronograma não são necessárias nas prévias do trabalho

Opcionais: **Obras Consultadas** **Anexos** **Apêndices** **Glossário**

☐ **Terminei esta parte**

Fonte: Autoria própria

4.4.6 INTERFACE DA TELA ENTREGA FINAL

A tela de entrega final é utilizada apenas como o próprio nome indica, no desenvolvimento do TCC para entrega final, para isso o aluno ou seu grupo deve procurar auxilio na biblioteca da Universidade Metodista para produção da ficha catalográfica, ao enviar algumas informações para a biblioteca como: título do trabalho, autores, orientador, assunto, etc, o aluno receberá um documento no qual existirá informações como a classificação de assunto e notação do autor para que o mesmo insira nesta tela e seja gerado o trabalho para impressão.

Figura 27 – Interface da Tela Entrega Final

Voltar **Entrega Final**

Banca: Separe o nome dos professores com uma vírgula e um espaço (Ex: Prof01, Prof02)

Classificação de Assunto: ? **Notação do Autor:** ?

Atenção: Para obter a classificação de Assunto e Notação do Autor contate a biblioteca

Nº de Páginas do PDF Gerado: ?

Assuntos:

Terminei Esta Parte

Fonte: Autoria própria

4.5 REGRAS DE NEGÓCIO

Aqui estarão listadas algumas regras para que o aluno faça o preenchimento da interface:

Tabela 1 – Regra de Negócio: Tela Informações Essenciais

Regra de negócios	1
Descrição	O aluno deve escrever algumas informações essenciais. Como, nome instituição, faculdade, curso, autores, entres outras.
Justificativa	Ao preencher as informações essenciais, é gerado um arquivo L ^A T _E X para capa, ficha catalográfica, folha de aprovação e folha de rosto. Caso fosse para inserir arquivos, seria necessário entradas e telas para cada uma dessas páginas, o que aumentaria o trabalho do aluno

Tabela 2 – Regra de Negócio: Tela Informações Essenciais

Regra de Negócio	2
Descrição	Campos em asterisco são obrigatórios.
Justificativa	Informações necessárias para os arquivos.

Tabela 3 – Regra de Negócio: Botão Terminei Esta Parte

– Regra de Negócio	3
Descrição	Clicar em terminei esta parte.
Justificativa	Ao clicar nesta marcação os arquivos Tex serão gerados, e o programa volta para a tela inicial.

Tabela 4 – Regra de Negócio: Arquivos de Entrada

Regra de Negócio	4
Descrição	Os arquivos de entradas para transformação no padrão ABNT precisam estar no formato ODT (Open Document Text) Documento de Texto Aberto.
Justificativa	Por ser Open Source, é possível utilizar junto a várias API's, entre elas o Writer2Latex a qual estamos usando.

Tabela 5 – Regra de Negócio: Arquivos Opcionais

Regra de Negócio	5
Descrição	Arquivos opcionais.
Justificativa	Nem todas as partes de um TCC são obrigatórias, algumas são opcionais como, Errata, Dedicatória, Epígrafe.

Tabela 6 – Regra de Negócio: Botão Gerar Tex

Regra de Negócio	6
Descrição	Botão Gerar Tex funciona apenas se usuário terminou as 4 etapas: pré-textuais, textuais, pós-textuais.
Justificativa	Gera os arquivos no formato Tex que poderá gerar um arquivo no formato PDF

Tabela 7 – Regra de Negócio: Botão Gerar PDF

– Regra de Negócio	7
Descrição	Botão Gerar PDF funciona se tex foi gerado.
Justificativa	Gera o arquivo no formato pdf. Todas as partes fundamentais do TCC precisam estar prontas para ser gerado o arquivo.

Tabela 8 – Regra de Negócio: Adicionar Capítulos da Tela Textuais

Regra de Negócio	8
Descrição	Adicionar capítulos
Justificativa	Para que o usuário possa escrever seu arquivo de maneira que achar melhor, não necessitando de uma ordem para utilizar o programa.

4.6 ALTERAÇÃO DE ABNTEX PARA ABNTEX2

Para diferenciar a versão original *abnTeX* do *abnTeX2* vamos chamar a versão original de *abnTeX1*.

Como o *abnTeX1* não estava catalogada na principal base de dados do *L^AT_EX* para utiliza-lo era necessário fazer a instalação manualmente, ainda sim a ultima versão disponível estava defasada devido a muitos anos sem atualizações. Devido a essa defasagem tivemos que adicionar diversos pacotes no arquivo principal do *L^AT_EX*, sendo que alguns destes também tiveram de ser instalados manualmente para que o pdf final gerado estivesse dentro do padrão ABNT atual. Todo este processo de instalação manual de pacotes *L^AT_EX* traria grande dificuldade aos usuários do *TCC_{TeX}* que nunca haviam utilizado o *L^AT_EX* e até mesmo os mais habituados ao *L^AT_EX* teriam dificuldades.

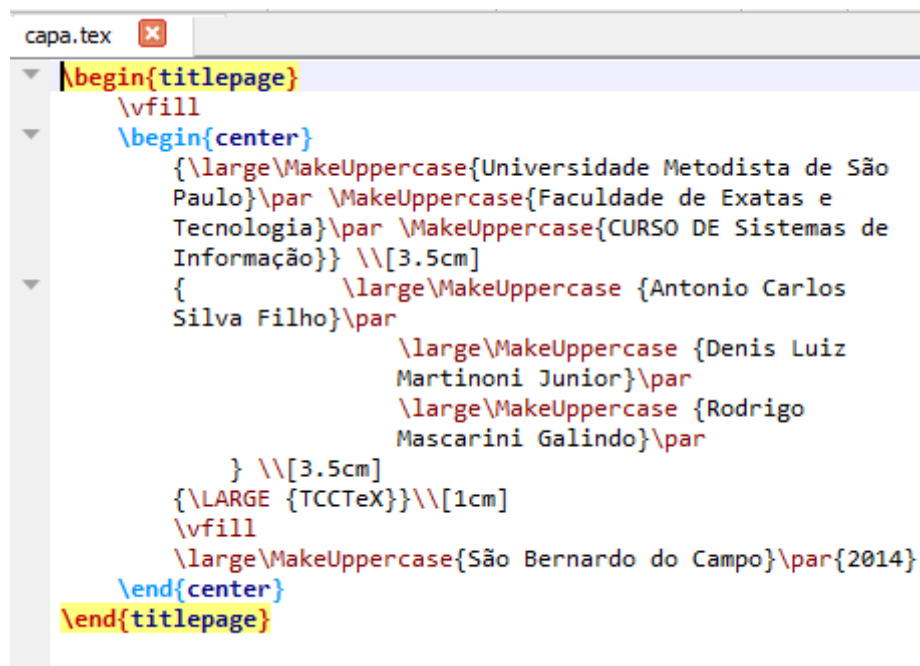
Com a alteração para o abnTeX2 estes pacotes não são mais necessários pois todos os pacotes necessários para a formatação do documento, bem como o abnTeX2, estão disponíveis do CTAN (Comprehensive TEX Archive Network) que é uma base de dados que possui todo tipo de materiais referentes a TeX. “As principais distribuições LATEX são construídas à partir de pacotes e classes do CTAN” (ARAUJO,2015).

Com a utilização do ABNTeX1 houve uma dificuldade maior de aprendizado, pois no site do mesmo não haviam mais manuais atualizados de como utiliza-lo e modifica-lo, sendo que tivemos que corrigir diversos problemas do mesmo reescrevendo diversos comandos.

Com a alteração, o código fonte do TCCTeX foi alterado em diversos aspectos, removendo coisas desnecessárias e atualizando alguns métodos, como por exemplo métodos relacionados a identificação de figuras, tabelas e o que deve gerar o arquivo principal com a extensão .tex que conecta todos os outros arquivos tex que formam o documento PDF do trabalho no padrão ABNT.

Um bom exemplo das modificações que foram feitas no código do programa para que ele funcione com abnTeX2 são as modificações feitas para a capa e folha de rosto. Como o abnTeX1 estava defasado em relação as ultimas versões das normas ABNT era necessário criar templates específicos para a capa e folha de rosto colocando-as na versão mais recente das normas ABNT.

Abaixo você pode verificar uma figura qual era o resultado do preenchimento do template da capa, que gerava o arquivo “capa.tex”:

Figura 28 – Capa em *abnTeX1*


```

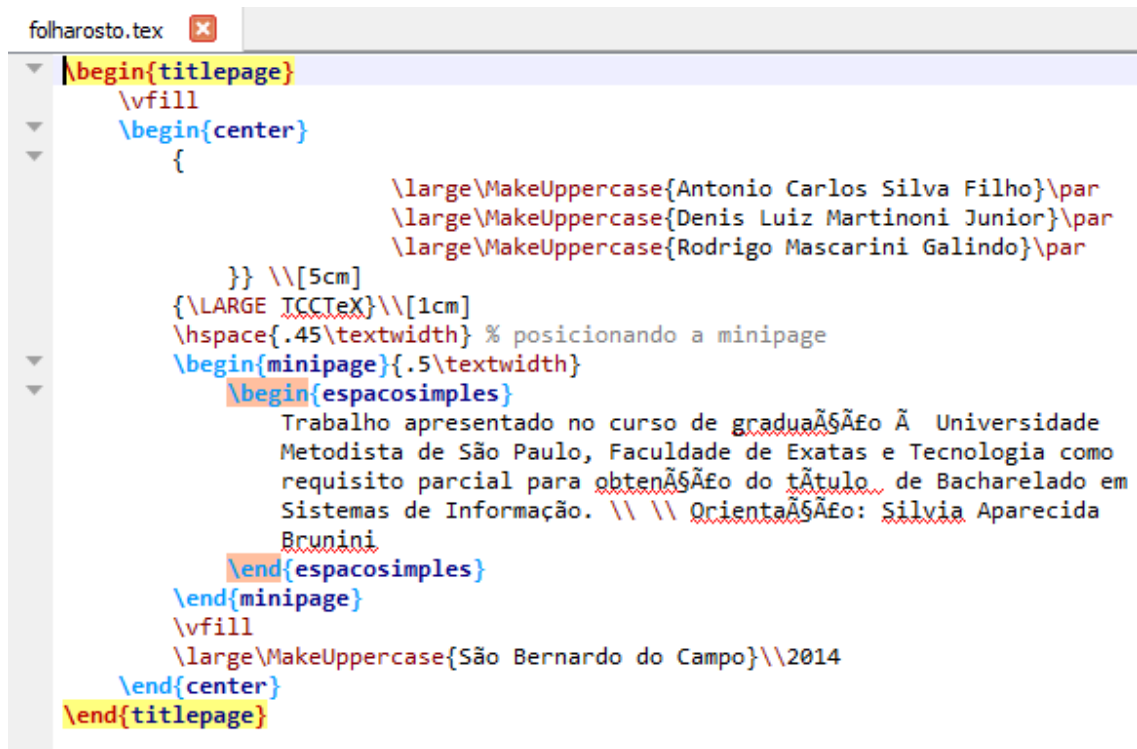
\begin{titlepage}
\vfill
\begin{center}
{\large\MakeUppercase{Universidade Metodista de São
Paulo}\par \MakeUppercase{Faculdade de Exatas e
Tecnologia}\par \MakeUppercase{CURSO DE Sistemas de
Informação}} \\\[3.5cm]
{
\large\MakeUppercase {Antonio Carlos
Silva Filho}\par
\large\MakeUppercase {Denis Luiz
Martinoni Junior}\par
\large\MakeUppercase {Rodrigo
Mascarini Galindo}\par
} \\\[3.5cm]
{\LARGE {TCCTeX}}\|[1cm]
\vfill
\large\MakeUppercase{São Bernardo do Campo}\par{2014}
\end{center}
\end{titlepage}

```

Fonte: Autoria própria

Abaixo você pode verificar uma figura qual era o resultado do preenchimento do template da folha de rosto, que gerava o arquivo “folharosto.tex”:

Figura 29 – Folha de Rosto em abnTeX1

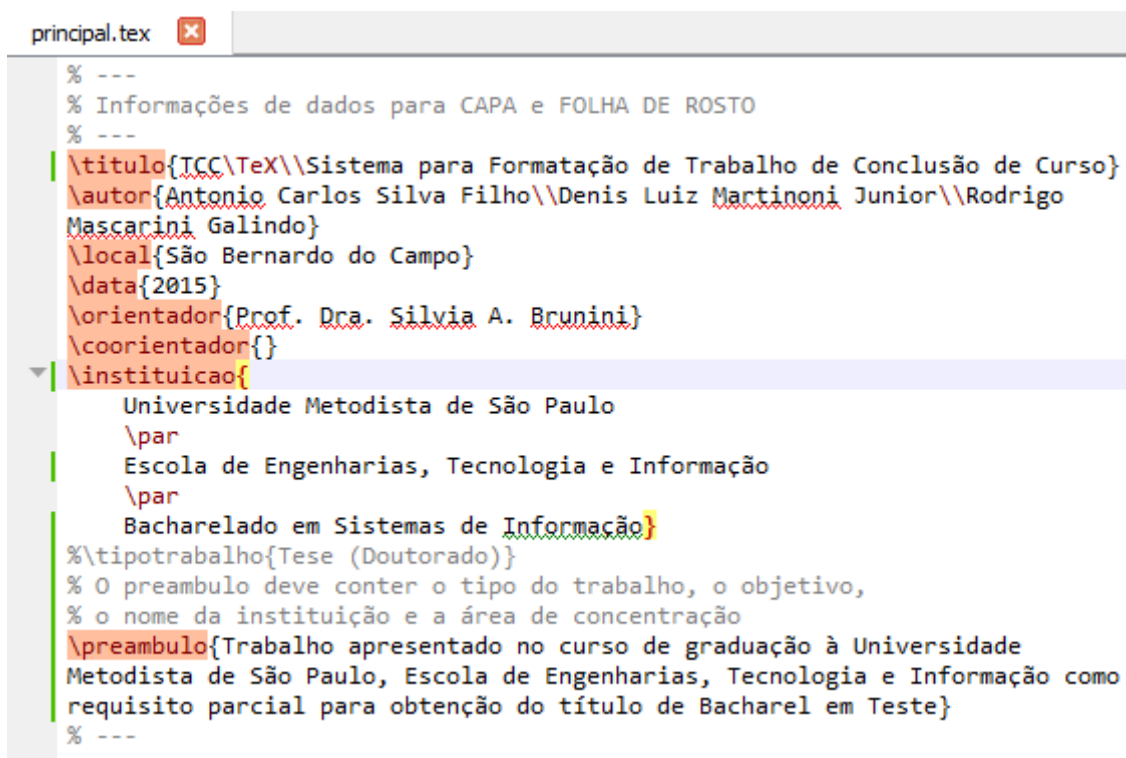


```
folharosto.tex
\begin{titlepage}
  \vfill
  \begin{center}
    {
      \large\MakeUppercase{Antonio Carlos Silva Filho}\par
      \large\MakeUppercase{Denis Luiz Martinoni Junior}\par
      \large\MakeUppercase{Rodrigo Mascarini Galindo}\par
    } \\\[5cm]
    {\LARGE TCCTeX}\\\[1cm]
    \hspace{.45\textwidth} % posicionando a minipage
    \begin{minipage}{.5\textwidth}
      \begin{espacosimples}
        Trabalho apresentado no curso de graduaÃ§Ã£o Ã Universidade
        Metodista de SÃo Paulo, Faculdade de Exatas e Tecnologia como
        requisito parcial para obtenÃ§Ã£o do tÃtulo de Bacharelado em
        Sistemas de InformaÃ§Ã£o. \\\ \ OrientaÃ§Ã£o: SÃlvia Aparecida
        Brunini
      \end{espacosimples}
    \end{minipage}
    \vfill
    \large\MakeUppercase{SÃo Bernardo do Campo}\2014
  \end{center}
\end{titlepage}
```

Fonte: Autoria pr3pria

Quando utilizamos o abnTeX2 s3 precisamos fazer o programa preencher uma s3rie de macros de dados do documento no arquivo principal.tex que 3 utilizado para unir todos os arquivos do documento. A figura abaixo mostra como s3o preenchidos esse macros no arquivo principal.tex:

Figura 30 – Preenchimento de Macros em principal.tex abnTeX2



```

principal.tex
% ---
% Informações de dados para CAPA e FOLHA DE ROSTO
% ---
\titulo{ICC\TeX\Sistema para Formatação de Trabalho de Conclusão de Curso}
\autor{Antonio Carlos Silva Filho\Deniz Luiz Martinoni Junior\Rodrigo
Mascarin Galindo}
\local{São Bernardo do Campo}
\data{2015}
\orientador{Prof. Dra. Silvia A. Brunini}
\coorientador{}
\instituicao{
  Universidade Metodista de São Paulo
  \par
  Escola de Engenharias, Tecnologia e Informação
  \par
  Bacharelado em Sistemas de Informação}
\tipotrabalho{Tese (Doutorado)}
% O preambulo deve conter o tipo do trabalho, o objetivo,
% o nome da instituição e a área de concentração
\preambulo{Trabalho apresentado no curso de graduação à Universidade
Metodista de São Paulo, Escola de Engenharias, Tecnologia e Informação como
requisito parcial para obtenção do título de Bacharel em Teste}
% ---

```

Fonte: Autoria própria

Após preencher todos os macros basta utilizar os comandos `\imprimircapa` para gerar a capa e `\imprimirfolhaderosto` para gerar a folha de rosto. Esses comandos são inseridos no próprio arquivo `principal.tex`.

4.7 EXTENÇÃO DO PRESENTE TRABALHO

Após terminar essa fase de desenvolver o software para que ele transforme arquivos em odt para \LaTeX colocando-os dentro dos padrões ABNT, pretendemos desenvolver a mesma funcionalidade para transformar arquivos docx em \LaTeX como uma extensão deste projeto.

Para fazer a transformação de textos .docx para um arquivo .tex precisamos reconhecer os padrões do código xml utilizado pelo docx e comparar com a linguagem \LaTeX . Para isso escrevemos no Microsoft Word arquivos em docx com possuem diferentes formatações de texto, tabela, imagem e divisão de capítulos, tudo fora da formatação da norma ABNT.

Figura 31 – Pequeno Texto feito no Microsoft Word

Este parágrafo está todo em negrito com um trecho sublinhado e outro *itálico*.

Só um pedaço do paragrafo terá **negrito**, outro terá *itálico*, outro terá sublinhado, outro terá **negrito e itálico**, outro terá **negrito e sublinhado**, outro terá itálico e sublinhado e por fim **negrito itálico e sublinhado**. Fim do paragrafo.

Abaixo alguns símbolos e sinais:

\ | / ' " ! @ * () + - = [] : ; < > , . ç ' £ § ? " ' ¢ ¬ ¹ ² ³ # \$ % & _ { } ^ ~

Fonte: Autoria própria

Em seguida escrevemos, com linguagem L^AT_EX e utilizando o pacote abnTeX2, arquivos .tex com o mesmo conteúdo dos arquivos .docx, porém obedecendo as normas abnt.

Figura 32 – Pequeno texto em L^AT_EX com mesmo conteúdo da figura anterior

```
Só um pedaço do paragrafo terá \textbf{negrito}, outro terá \textit{itálico}, outro terá
\underline{sublinhado}, outro terá \textbf{\textit{negrito e itálico}}, outro terá
\textbf{\underline{negrito e sublinhado}}, outro terá \textit{\underline{itálico e sublinhado}} e
por fim \textbf{\textit{\underline{negrito itálico e sublinhado}}}. Fim do paragrafo.
```

Abaixo alguns símbolos e sinais:

```
\backslash$
| / ' " ! @ * ( ) + - = [ ] : ; < > , . ç ' £ § ? " ' ¢ ¬ ¹ ² ³
\#
\$
\%
\&
\_
\{
\}
\textasciicircum
\textasciitilde
```

Fonte: Autoria própria

Em seguida fazemos uma cópia do arquivo docx e renomeamos a extensão dessa cópia para .zip para assim podermos descompactar os arquivos xml do docx.

Após descompactar os arquivos .docx, abrimos, utilizando o NetBeans IDE, o arquivo document.xml que possui o parte principal do código xml

Figura 33 – Pequena parte do código XML do docx

```

279 <w:p w:rsidR="00A77C8A" w:rsidRDefault="00A77C8A" w:rsidP="00A77C8A">
280   <w:r>
281     <w:t xml:space="preserve">Só um pedaço do paragrafo terá </w:t>
282   </w:r>
283   <w:r w:rsidRPr="00952C71">
284     <w:rPr>
285       <w:b/>
286     </w:rPr>
287     <w:t>negrito</w:t>
288   </w:r>
289   <w:r>
290     <w:t xml:space="preserve">, outro terá </w:t>
291   </w:r>
292   <w:r w:rsidRPr="00952C71">
293     <w:rPr>
294       <w:i/>
295     </w:rPr>
296     <w:t>itálico</w:t>
297   </w:r>
298   <w:r>
299     <w:t xml:space="preserve">, outro terá </w:t>
300   </w:r>
301   <w:r w:rsidRPr="00952C71">
302     <w:rPr>
303       <w:u w:val="single"/>
304     </w:rPr>
305     <w:t>sublinhado</w:t>
306   </w:r>
307   <w:r>
308     <w:t xml:space="preserve">, outro terá </w:t>
309   </w:r>

```

Fonte: Autoria própria

Este arquivo document.xml é bastante extenso e possui todo o conteúdo inserido no arquivo docx. Cada arquivo docx possui seu próprio arquivo document.xml.

A partir disso teremos que, utilizando o NetBeans IDE, reconhecer os padrões da linguagem xml utilizada no arquivo .docx, compará-los com os padrões da linguagem \LaTeX utilizada no arquivo .tex e criar histórias de como o código Java irá converter os arquivos de uma linguagem para outra. A partir dessas histórias será criado o conversor de .docx para \LaTeX em Java.

4.8 DISCUSSÃO

Uma questão relevante que pode surgir a quem está desenvolvendo um TCC e toma conhecimento da existência do TCCTeX é: “Por que eu deveria utilizar o TCCTeX?” Essa é uma questão pertinente tendo em vista que todo estudante universitário já teve algum contato com os softwares editores de texto do tipo WYSIWYG (What You See Is What You Get, ou O que você vê é o que você obtém) e sem duvida já fez diversos trabalhos, atividades e diversos tipos de textos nesse tipo de editor de texto.

A resposta para essa pergunta foi a nossa grande preocupação ao desenvolver nosso projeto. O que faria um usuário de programas como Microsoft Word ou LibreOffice Writer deixar de trabalhar exclusivamente com esses programas e começar a utilizar o TCC_{TeX}?

O que levaria uma pessoa que esta empenhada em desenvolver seu TCC desviar o seu foco do desenvolvimento do mesmo para dedicar um pouco de seu tempo a aprender a utilizar o TCC_{TeX}? Se o período para o desenvolvimento de um TCC é, na grande maioria dos casos, apertado, o fato de ter que se familiarizar com um novo software não diminuiria o tempo do aluno dedicado ao TCC?

Na nossa visão a questão correta a ser levantada é: “O TCC_{TeX} é realmente capaz de me fazer economizar tempo durante a documentação do meu projeto de TCC?” A resposta para essa pergunta é: Sim. Na realidade o tempo gasto para aprender a utilizar e se familiarizar com o TCC_{TeX} é sem duvida muito menor que o tempo gasto para colocar um TCC perfeitamente dentro de todos os padrões estabelecidos pela ABNT (Associação Brasileira de Normas Técnicas) e exigidos pela Universidade Metodista de São Paulo.

Cada página de um trabalho acadêmico deve estar dentro de padrões muito bem detalhados e cada uma das etapas da documentação possuem diferenças de formatação entre si, isso sem contar com as inúmeras regras para se colocar tabelas e figuras e suas listas.

Mas o problema de colocar um trabalho nos padrões estabelecidos utilizando os processadores de texto altamente difundidos pelo mercado não está só na grande quantidade de pequenos detalhes que devem ser atentados nessa tarefa, mas também a grande irregularidade e inconsistência desses programas, especialmente no que diz respeito à formatação de grandes documentos.

No desenvolvimento de um grande documento de texto diversas vezes o autor sente a necessidade de modificar a organização do mesmo, seja mudando a ordem dos capítulos, inserindo ou até mesmo removendo parágrafos, mas quando se trabalha com processadores de texto tarefas simples como essas podem se tornar dor de cabeça. O simples ato de alterar o texto de uma página pode refletir negativamente na formatação de todas as páginas subsequentes. E quando o autor deixa para formatar o documento após estar com todos os textos completos pode levar muitas horas e deixar muitas falhas no caminho, principalmente para documentos muito grandes.

Vale lembrar que até mesmo manter um padrão único da primeira folha do primeiro capítulo até a última folha do último capítulo de um grande documento pode se tornar uma tarefa árdua quando se trabalha exclusivamente com processadores de texto WYSIWYG.

A grande vantagem que o aluno pode obter ao documentar seu TCC com o auxílio do TCC_{TeX} é justamente não ter que se preocupar com a formatação do documento. Após aprender a utilizar o TCC_{TeX} o seu foco poderá estar voltado apenas para o conteúdo

do documento, sem ter que se preocupar com margens, índices, listas ou legendas.

Acreditamos que o usuário conseguira se familiarizar com o TCC_{TeX} rapidamente devido a simplicidade da interface e a grande quantidade de dicas de como utilizar o software que serão disponibilizadas no manual. O aluno facilmente aprenderá a utilizar o TCC_{TeX} no dia a dia.

5 CONCLUSÃO

Acreditamos que conseguimos desenvolver o que foi proposto pelo grupo e pela nossa orientadora, utilizando a metodologia de XP, que se demonstrou adequada para acompanhar e desenvolver projetos de software principalmente quando o projeto envolve incertezas do cliente e requer flexibilidade no desenvolvimento.

O Software TCCTeX foi desenvolvido para ser simples e usual, a metodologia envolvida permitiu diminuir a sua complexidade, durante o desenvolvimento podemos mudar os rumos da nossa pesquisa e idealização do software em várias oportunidades, finalizado, o programa se mostrou adequado ao que foi proposto, ajudar a comunidade acadêmica no processo de elaboração do seu TCC.

O próximo desenvolvedor que deseje melhorar o programa terá ampla documentação disponível, código comentado, base teórica da metodologia, e conhecimento de quais ferramentas e em que momentos usar. O programa está pronto para padronizar automaticamente, no padrão ABNT Metodista, o mais importante empreendimento acadêmico do aluno em sua jornada pela graduação.

REFERÊNCIAS

ARAUJO, Lauro César. **A classe abntex2**: Documentos técnicos e científicos brasileiros compatíveis com as normas ABNT. [S.l]: [S.n.], 2015. Disponível em: <<http://linorg.usp.br/CTAN/macros/latex/contrib/abntex2/doc/abntex2.pdf>>. Acesso em: 06 nov. 2015.

BECK, Kent. **Extreme Programming Explained**: Embrace Change. S.l: Addison-wesley, 2000.

KNUTH, Donald E.. **The TeXbook**. Massachusetts: Addison-wesley, 1984.

LAMPORT, Leslie. **L^AT_EX**: A Document Preparation System. 2. ed. S.l: Addison-wesley Publishing Company, 1994.

TIMBÓ, Noeme Viana. **Manual de Apresentação para Trabalhos Acadêmicos**: Segundo ABNT NBR 14724 DEZ. 2011; NBR 6027 MAIO 2003; NBR 6028 NOV. 2003; NBR 12225 JULHO 2004. S.l: S.n., 2013. Disponível em: <https://portal.metodista.br/biblioteca/servicos/manual_apresentacao_3.pdf>. Acesso em: 08 nov. 2015.

TURING, Alan Mathison. **Computing Machinery and Intelligence**. 1950. Disponível em: <<http://www.loebner.net/Prizet/TuringArticle.html> >. Acesso em: 26 maio 2014.

APÊNDICE A – ATAS DE REUNIÕES COM ORIENTADOR

ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
19/02/2014, das 19h00 às 19h30	Pres.	Aus.	Aluno	Versão: 0.1
Local	X		Antonio Carlos Silva Filho	Preparação: Antonio em 09/03/2014
	X		Denis Luiz Martinoni	
	X		Rodrigo G. Mascarini	
Edifício Lambda				Revisão: Denis em 10/03/2014

Orientador: Silvia Brunini

Pauta

- 1.0-Discussão sobre o nome do Software.
- 1.1-Discussão sobre estrutura do Software.
- 1.2-Discussão sobre interface do Software.
- 2.0-Discussão sobre tecnologias a serem usadas.

Pontos de Discussão / Anotações

- 1.1-Linguagens de programação que serão usadas no projetos.
- 1.2-O grupo decidiu que o nome do projeto será TCCTEX.

Encaminhamentos / Pendências

O que	Quem	Para Quando
Pesquisar tecnologias a serem usadas no projeto	O grupo	10/03/2014

Pontos sem Discussão (para a próxima reunião)

- 1.1-Apresentação de tecnologias para desenvolvimento de software.

ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
24/03/2014, das 19h00 às 19h30	Pres.	Aus.	Aluno	Versão: 0.2
Local	X		Antonio Carlos Silva Filho	Preparação: Antonio em 23/03/2014
		X	Denis Luiz Martinoni	
		X	Rodrigo G. Mascarini	
Edifício Lambda				Revisão: Denis em 24/03/2014

Orientador:

Pauta

1.0-Análise da apresentação para banca.

Pontos de Discussão / Anotações

1.0- Definição da utilização de apenas o padrão ABNT seguido pela metodista.
2.0-Adoção do pacote ABNTEX 2.
3.0-Orientações gerais aos alunos.

Encaminhamentos / Pendências

O que	Quem	Para Quando

Pontos sem Discussão (para a próxima reunião)



ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
25/03/2014, das 19h00 às 19h30	Pres.	Aus.	Aluno	Versão: 0.2
Local	X		Antonio Carlos Silva Filho	Preparação: Denis em 24/03/2014
	X		Denis Luiz Martinoni	
	X		Rodrigo G. Mascarini	
Edifício Lambda				Revisão: Antonio em 05/04/2014

Orientador:

Pauta

- 1.0-Discussão em análise sobre apresentação de banca
- 2.0-Discussão sobre a
- 3.0-Discussão sobre a interface do software.
- 4.0-Discussão sobre utilização do ABNTEX 2
- 5.0-Esclarecimento sobre padrões ABNT adotados pela metodista.

Pontos de Discussão / Anotações

- 1.0- Definição da utilização de apenas o padrão ABNT seguido pela metodista.
- 2.0-Adoção do pacote ABNTEX 2.
- 3.0-Definição do Layout.
- 4.0-Indicação do estudo do código fonte do software docx2tex.
- 5.0-Definição da utilização do formato docx como documento de entrada no software.

Encaminhamentos / Pendências

O que	Quem	Para Quando
Estudar o código fonte do software doc2tex	Denis	03/04/2014
Estudar ABNTEX 2 e LaTeX	Antonio	06/04/2014
Estudar padrões ABNT utilizados na metodista	Rodrigo	02/04/2014

Pontos sem Discussão (para a próxima reunião)

ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
15/04/2014, das 19h00 às 19h30	Pres.	Aus.	Aluno	Versão: 0.3
Local	X		Antonio Carlos Silva Filho	Preparação: Denis em 14/04/2014
	X		Denis Luiz Martinoni Junior	
		X	Rodrigo Galindo Mascarini	Revisão: Antonio em 15/04/2014
Edifício Lambda				

Orientador:

Pauta

Análise do design da interface do software

Pontos de Discussão / Anotações

- 1.0-Design de interface do software
- 2.0-Métodos para entrada de arquivo
- 3.0-Métodos de programação

Encaminhamentos / Pendências

O que	Quem	Para Quando
Desenvolvimento de diagramas de classe	Todos	07/05/2014
Criação de classes e testes básicos	Todos	09/05/2014

Pontos sem Discussão (para a próxima reunião)



ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
7/05/2014, das 19h00 às 19h30	Pres.	Aus.	Aluno	Versão: 0.4
Local	X		Antonio Carlos Silva Filho	Preparação: Antonio em 07/05/2014
		X	Denis Luiz Martinoni Junior	
		X	Rodrigo Mascarini Galindo	Revisão: Rodrigo em 01/06/2014
Edifício Lambda				

Orientador: Silvia Aparecida Brunini

Pauta DIAGRAMAS UML		
Pontos de Discussão / Anotações JFRAME PARA DESKTOP PLUGINS PARA UML CHAMADAS DE PROGRAMAS EXTERNOS EM BACKGROUND		
Encaminhamentos / Pendências		
O que	Quem	Para Quando
Modificar e melhorar diagrama de classes UML	Grupo	12/05/2014
Pontos sem Discussão (para a próxima reunião) UML E CODIFICAÇÃO.		

ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
21/05/2014, das 19h00 às 19h30	Pres.	Aus.	Aluno	Versão: 0.4
Local	X		Antonio Carlos Silva Filho	Preparação: Denis em 21/05/2014
	X		Denis Luiz Martinoni Junior	
		X	Rodrigo Mascarini Galindo	Revisão: Antonio em 02/06/2014
Edifício Lambda				

Orientador: Silvia Aparecida Brunini

Pauta Protótipo do software Metodologias ágeis Documentação do TCC		
Pontos de Discussão / Anotações		
Encaminhamentos / Pendências		
O que	Quem	Para Quando
Melhorar protótipo do software	Denis	24/05/2014
Desenvolver classes Java com base no diagrama de classes	Antonio	25/05/2014
Finalizar prévia do trabalho	Grupo	26/05/2014
Pontos sem Discussão (para a próxima reunião)		



ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
02/06/2014, das 19h00 às 19h30	Pres.	Aus.	Aluno	Versão: 0.4
Local	X		Antonio Carlos Silva Filho	Preparação: Rodrigo em 02/06/2014
	X		Denis Luiz Martinoni Junior	
	X		Rodrigo Mascarini Galindo	
Edifício Lambda				Revisão: Denis em 02/06/2014

Orientador: Silvia Aparecida Brunini

Pauta

Extreme Programming (XP)
Utilização do RTF ou DOCX como arquivo de entrada do software
Funcionamento do software TCCTeX em relação a citações, imagens e tabelas

Pontos de Discussão / Anotações

Encaminhamentos / Pendências

O que	Quem	Para Quando
Pesquisar sobre metodologia de <i>Extreme Programming</i> (XP)	Grupo	11/06/2014
Definir arquivo de entrada	Grupo	19/06/2014

Pontos sem Discussão (para a próxima reunião)

ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
29/08/2014, das 19h00 às 19h30	Pres.	Aus.	Aluno	Versão:
	x		Antonio Carlos Silva Filho	
Local		x	Denis Luiz Martinoni JR	Preparação: Rodrigo em 30/08/2014
	x		Rodrigo M. Gallindo	
Lambda				Revisão: Antonio em 30/08/2014

Orientador:



ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
12/08/2014, das 19h00 às 19h30	Pres.	Aus.	Aluno	Versão:
	x		Antonio Carlos Silva Filho	
Local	x		Denis Luiz Martinoni JR	Preparação: Denis em 13/08/2014
	x		Rodrigo M. Gallindo	
Lambda				Revisão: Antonio em 14/08/2014

Orientador:

Pauta

Congresso, Definir a participação da utilização do BDD, Início da Codificação

Pontos de Discussão / Anotações

Definição da não utilização do BDD

Tiramos duvidas sobre a formulação do documento a ser entregue para o congresso

Encaminhamentos / Pendências

O que	Quem	Para Quando
Definir a uma Metodologia Ágil de processo	Grupo	17/08/2014
Definir funcionalidades do Software	Grupo	20/08/2014

Pontos sem Discussão (para a próxima reunião)

- Codificação utilizando TDD

ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
08/09/2014 das 21h10 às 22h30	Pres.	Aus.	Aluno	Versão:
	X		Antonio Carlos Silva Filho	
Local	X		Denis Luiz Martinoni Junior	Preparação: Denis em 08/09/2014
	X		Rodrigo Mascarini Galindo	
Edifício Lambda				Revisão: Rodrigo em 10/09/2014

Orientador:

</



ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
08/09/2014 das 19h00 às 19h30	Pres.	Aus.	Aluno	Versão:
	X		Antonio Carlos Silva Filho	
Local	X		Denis Luiz Martinoni Junior	Preparação: Denis em 10/09/2014
	X		Rodrigo Mascarini Galindo	
Edifício Lambda				Revisão: Rodrigo em 12/09/2014

Orientador: Silvia Aparecida Brunini

Pauta

- Divisão capítulos em arquivos separados, tanto na entrada quanto saída.

Pontos de Discussão / Anotações

- Divisão de documento em vários arquivos “.odt” na entrada..
- Divisão de documento em vários arquivos “.tex” na saída.
- Divisão de capítulos de um TCC em arquivos separados.
- Subcapítulos dentro dos arquivos de cada capítulo.
- Criação de manual de usuário para correta utilização do software.

Encaminhamentos / Pendências

O que	Quem	Para Quando
Terminar interface gráfica da tela de informações essenciais.	Antonio	12/09/2014
Estudar modelo tex de TCC que utiliza ABNTE _X	Grupo	17/09/2014
Integrar interface gráfica e modelos tex da divisão informações essenciais para preencher templates LaTeX.	Denis	15/09/2014

Pontos sem Discussão (para a próxima reunião)

- Reconhecimento de subcapítulos de um arquivo odt pelo Writer2Latex

ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
17/09/2014 das 18h50 às 19h30	Pres.	Aus.	Aluno	Versão:
	X		Antonio Carlos Silva Filho	
Local	X		Denis Luiz Martinoni Junior	Preparação: Denis em 17/09/2014
	X		Rodrigo Mascarini Galindo	
Edifício Lambda				Revisão: Antonio em 28/09/2014

Orientador: Silvia Aparecida Brunini

Pauta

- Reconhecimento de subcapítulos de um arquivo odt pelo Writer2Latex

Pontos de Discussão / Anotações

- Configuração de arquivo XML do writer2LaTeX para conversão de capítulos de um arquivo *Opendocument* (odt) para LaTeX (tex).
- Retirada de preâmbulo dos arquivos tex por meio da configuração do arquivo XML.
- Realização de testes de conversão de imagens e tabelas de odt para tex.
- Resolução de problemas na conversão de odt para tex..

Encaminhamentos / Pendências

O que	Quem	Para Quando
Criar método para gerar arquivo tex principal que interligará todas partes do TCC.	Denis	23/09/2014
Criar método para adicionar comandos iniciais e finais em arquivos tex gerados (ex: <code>begin {abstract}</code> , <code>end {abstract}</code>).	Denis e Rodrigo	25/09/2014
Começar a adicionar mais informações ao documento do trabalho.	Antonio	27/09/2014

Pontos sem Discussão (para a próxima reunião)

- Conversão do modelo de TCC padrão metodista de ABNTEX para ABNTEX 2



ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
01/09/2014, das 21h10 às 21h20	Pres.	Aus.	Aluno	Versão:
	x		Antonio Carlos Silva Filho	
Local	x		Denis Luiz Martinoni JR	Preparação: Rodrigo em 03/09/2015
	x		Rodrigo M. Gallindo	
Delta				Revisão: Antonio em 08/09/2015

Orientador:

Pauta

Congresso, Tipo de Arquivo de texto para TCCTeX, ABNTEX2

Pontos de Discussão / Anotações

Falamos sobre quais pontos ressaltar na apresentação do Congresso.
Discutimos sobre a possibilidade de mudar o tipo de texto de entrada para o TCCTeX

Encaminhamentos / Pendências

O que	Quem	Para Quando
Avaliação e pesquisa sobre viabilidade de usar docx	Grupo	11/09/2015
Conclusão do trabalho em ABNTEX2	Grupo	11/09/2015

Pontos sem Discussão (para a próxima reunião)

- Funcionalidades inacabadas do software
- Elaboração do trabalho escrito



ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
28/09/2015, das 19h30 às 21h10	Pres.	Aus.	Aluno	Versão:
	x		Antonio Carlos Silva Filho	
Local		X	Denis Luiz Martinoni JR	Preparação: Antonio em 03/09/2015
	x		Rodrigo M. Gallindo	
Biblioteca				Revisão: Rodrigo em 08/09/2015

Orientador:

Pauta

Como utilizar .docx

Pontos de Discussão / Anotações

Orientação sobre como reconhecer os padrões xml do docx.

Encaminhamentos / Pendências

O que	Quem	Para Quando
Criar Arquivos docx para reconhecermos os padrões xml	Grupo	05/10/2015
Reconhecer padrões xml dos capítulos	Grupo	05/10/2015

Pontos sem Discussão (para a próxima reunião)

- Partes importantes do código xml

ATA DE REUNIÃO COM ORIENTADOR

Data e Horário	Participantes			Documento
05/10/2015, das 19h30 às 21h10	Pres.	Aus.	Aluno	Versão:
	x		Antonio Carlos Silva Filho	
Local		X	Denis Luiz Martinoni JR	Preparação: Antonio em 03/09/2015
	x		Rodrigo M. Gallindo	
Biblioteca				Revisão: Rodrigo em 08/09/2015

Orientador:

Pauta

Duvidas sobre o xml do .docx

Pontos de Discussão / Anotações

Tirar duvidas sobre como reconhecer os padrões xml do docx.

Encaminhamentos / Pendências

O que	Quem	Para Quando
Criar mais arquivos docx para reconhecermos os padrões xml	Grupo	19/10/2015
Reconhecer padrões xml das tabelas	Grupo	19/10/2015

Pontos sem Discussão (para a próxima reunião)

- Como o xml cria as tabelas do docx

APÊNDICE B – CRONOGRAMA



Faculdade de Exatas e Tecnologia

CRONOGRAMA DO TCC

PROJETO	TCCTeX
DATA FINAL	
VERSÃO	4.0
ALUNO 1	Antonio Carlos Silva Filho
ALUNO 2	Denis Luiz Martinoni Junior
ALUNO 3	Rodrigo Mascarini Galindo
ALUNO 4	

STATUS
EAP = Em Andamento no Prazo
EAA = Em Andamento no Atraso
NIP = Não Iniciado no Prazo
NIA = Não Iniciado em Atraso
CON = Concluído

Nº. Atividade	Atividade Principal	Sub Atividade	Responsável (is)	Data Início	Data Entrega	Observações	Status	Precede
1	Reunião	Pesquisa de Tecnologias	Grupo	19/02/2014	14/03/2014	Definições de melhores tecnologias a serem utilizadas.	CON	
2	Reunião	Definição de Tecnologias	Grupo	14/03/2014		Segunda Reunião do Ano	CON	
3	Definição de tipo de arquivo de entrada	Estudo do código fonte do Docx2Tex	Denis	25/03/2014	03/04/2014		CON	
4	Definição de pacotes LaTeX	Estudo do ABNTEX 2 e LaTeX	Antonio	25/03/2014	03/04/2014		CON	
5	Estudo de Padrões ABNT	Verificação do padrão ABNT Metodista	Rodrigo	25/03/2014	06/04/2014		CON	
6	Prototipação de layout	Consulta ao padrão ABNT Metodista	Grupo	31/03/2014	02/04/2014	Idealização inicial de Layout	CON	
7	Criação do Layout		Grupo	07/03/2014	14/04/2014		CON	
8	Desenvolvimento de Diagramas de		Grupo	26/04/2014	07/05/2014		CON	

	Classe			14	14			
9	Criação das Classes java		Grupo	29/04/2014	09/05/2014		CON	
10	Criação de Casos de Uso		Grupo	05/08/2014	10/08/2014		CON	
11	Modificação das Classes		Antonio	12/08/2014	15/08/2014		CON	
12	Criação de Classe de Teste Pre Textuais		Denis e Rodrigo	19/08/2014	22/08/2014		EAP	
13	Desenvolvimento da Classe Java Pre Textuais		Denis e Rodrigo	23/08/2014	04/09/2014		EAP	
14	Criação de Outras Classes		Grupo	27/08/2014	20/09/2014		EAP	
15	Identificar Pacotes Java para trabalhar com RTF		Denis	01/09/2014	05/09/2014		NIP	

Nº. Atividade	Atividade Principal	Sub Atividade	Responsável (is)	Data Início	Data Entrega	Observações	Status	Precede
17	Documentação	Padronização ABNTEX2	Rodrigo e Antonio	19/07/2015	11/09/2015	Elaboração da documentação atual no padrão ABNTEX2	EAP	
18	Adptação ao Padrão UMESP de TCC	Programação	Grupo	14/07/2015		Falta alguns pontos para o programa elaborar o arquivo totalmente no padrão UMESP.	EAP	
19	Reunião	Congresso e andamento do desenvolvimento	Grupo	01/09/2015	01/09/2015	Discussão de pontos em aberto	CON	
20	Reunião	Como utilizar .docx	Grupo	28/09/2015	05/09/2015	Orientação sobre como reconhecer os padrões xml do docx.	CON	
21	Reunião	Duvidas sobre xml	Grupo	05/10/2015	28/09/2015	Como o xml cria as tabelas do docx	CON	

Link para uma apresentação do professor Douglas Murilo Siqueira que explica como preencher esse cronograma: <http://breeze.metodista.br/cronograma/>

ANEXO A – RELATÓRIO DE
RECOMENDAÇÕES DA BANCA DE
QUALIFICAÇÃO



Faculdade de Exatas e Tecnologia

RELATÓRIO DE RECOMENDAÇÕES DA BANCA DE QUALIFICAÇÃO

IDENTIFICAÇÃO

Curso: Sistemas de Informação

Data: 24/03/2014

Componentes do Grupo

Antonio Carlos Silva Filho

Denis Luiz Martinoni

Rodrigo Galindo Mascarini

ORIENTADOR

Silvia Brunini

TÍTULO DO TRABALHO

TCCTeX

Reproduza cada recomendação feita pelos professores da banca de qualificação em um dos itens RECOMENDAÇÃO a seguir. Escreva o nome do professor que fez a recomendação. Analise-a junto com seu orientador e informe se vocês estão de acordo com a recomendação. Caso julguem a recomendação NÃO VIÁVEL, devem escrever uma justificativa. Se necessário podem aumentar o número de recomendações.

RECOMENDAÇÃO 1

Falta do LaTeX no índice, sendo que a parte mais complicada do trabalho será aprender o funcionamento do mesmo, e utiliza-lo para fazer o software funcionar.

Feita pelo professor: Marcelo Módulo

Parecer do grupo/orientador

(X) de acordo

() não viável - JUSTIFICAR

RECOMENDAÇÃO 2
Planejar todo futuro do projeto e refazer o cronograma, que no momento está focando mais no passado.
Feita pelo professor: Marcelo Módulo
Parecer do grupo/orientador <input checked="" type="checkbox"/> de acordo <input type="checkbox"/> não viável - JUSTIFICAR
RECOMENDAÇÃO 3
Não faz sentido descrever 2 capítulos falando de TDD e BDD, sendo que só estarão utilizando ele, sem estar comparando, nem discutindo ou apresentando as dificuldades e problemas ao utiliza-los.
Feita pelo professor: Marcelo Módulo
Parecer do grupo/orientador <input type="checkbox"/> de acordo <input checked="" type="checkbox"/> não viável - JUSTIFICAR
O orientador não está de acordo, pois, apesar de ser um software simples para processamento de texto, o foco principal do trabalho é entregar um software gráfico com qualidade.
RECOMENDAÇÃO 4
Esquecer o BDD e TDD no momento e focar no LaTeX e como irão fazer o trabalho.
Feita pelo professor: MARCELO MÓDULO
Parecer do grupo/orientador <input type="checkbox"/> de acordo <input checked="" type="checkbox"/> não viável - JUSTIFICAR
O orientador não está de acordo, pois, apesar de ser um software simples para processamento de texto, o foco principal do trabalho é entregar um software gráfico com qualidade.
RECOMENDAÇÃO 5
Utilizar o pacote ABNTeX e seguir os padrões de formatação de TCC da Metodista, pois a utilização dos padrões do ABNT variam conforme a instituição de ensino.
Feita pelo professor: Leonardo Angiolim

**Faculdade de Exatas e Tecnologia****Parecer do grupo/orientador****(X) de acordo****() não viável - JUSTIFICAR**