

IFBATEX - AMBIENTE WEB PARA EDIÇÃO DE DOCUMENTOS LATEX

Ruivaldo Azevedo Lobão Neto

Instituto Federal da Bahia – IFBA/Brazil

rneto@rneto.com.br

Manoel C. M. Neto

Instituto Federal da Bahia - IFBA/Brasil

manoelnetom@ifba.edu.br

Abstract: This work presents the development of IFBATEX, an open source web tool, whose aim is to facilitate the development of academic texts. The application provides an integrated environment for online editing of LaTeX documents. Its interface is one of its differentials. It allows the user to edit, compile and visualize documents without the need to install the numerous components that represent the LaTeX environment.

Keywords: IFBATEX; LATEX; Ferramentas de Edição; Trabalho de Conclusão de Curso.

I. INTRODUÇÃO

O trabalho de conclusão de curso (TCC) é um tipo de trabalho acadêmico comumente utilizado no ensino superior e que contempla a diversidade dos aspectos da formação universitária de um aluno. Ele é usado como forma de efetuar uma avaliação final dos graduandos. Em muitas instituições, o TCC é encarado como critério final de avaliação do aluno: em caso de reprovação, o aluno estará impedido de obter o diploma e conseqüentemente exercer a respectiva profissão até que seja aprovado.

Para os alunos dos cursos de graduação de Análise e Desenvolvimento de Sistemas (ADS) do Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA), o TCC deve ser elaborado e apresentado ao final do quinto semestre. Esta tarefa demanda uma quantidade considerável de tempo uma vez que os alunos precisam elaborar o TCC em paralelo ao cumprimento de outras atividades (disciplinas, estágio, etc.). Uma boa parte do tempo consumido é destinada à formatação do documento para que ele se enquadre às normas adotadas pelo curso. Os processadores de texto utilizados pela maioria dos usuários ajudam a tratar este problema, mas não são projetados para dar atenção especial a formatação de um texto. Para preencher esta lacuna, membros da comunidade científica passaram a usar o TeX e o LaTeX [13].

O TeX é um sistema de tipografia popular no meio acadêmico. A sua popularidade se deve a capacidade que este sistema tem de produzir textos, fórmulas e símbolos matemáticos de forma elegante e com alta qualidade tipográfica [17]. O LaTeX pode ser entendido como um conjunto de macros para um

processador de textos TeX. Este conjunto fornece ao usuário comandos de alto nível que permitem lidar com bibliografias, citações, formatos de páginas, referências, entre outras características da formatação de documentos, tornando a sua utilização mais fácil por iniciantes (quando comparado ao TeX puro).

A utilização do LaTeX, apesar de apresentar similaridades com outras linguagens de marcação como HTML, apresenta uma curva de aprendizado acentuada [12]. Além disso, a instalação do ambiente sofre influências da infraestrutura disponível (tipo de sistema operacional, quantidade de memória e espaço em disco, tipo de codificação de texto suportada, etc.). Assim, da mesma forma que outros sistemas, mudanças nessa infra podem influenciar no funcionamento do LaTeX. Uma solução para evitar a influência dessas mudanças no funcionamento de sistemas foi portá-los para o ambiente web.

O curso de ADS do IFBA incentiva o uso do LaTeX para elaboração do TCC. Este incentivo serviu como motivação inicial para criação deste projeto. Além disso, o IFBATEX também tem como objetivo solucionar os problemas citados, que ocorrem tanto no âmbito do curso de ADS quanto nos ambientes acadêmicos em geral, assim contribuindo com a elaboração de textos científicos (especialmente nos TCCs dos cursos de graduação).

Este artigo apresenta um sistema web, denominado IFBATEX, que foi criado para permitir que usuários editem de forma on-line documentos escritos em LaTeX. O IFBATEX permite ao usuário utilizar um editor LaTeX sem a necessidade de instalar este ambiente completo no seu computador. Além disso, o sistema oferece ferramentas para a edição simplificada de documentos em computadores que possuam um navegador compatível (com suporte a HTML5 e CSS3). O IFBATEX está disponível em <http://aws.rneto.com.br/ifbatex/public> e foi empregado na elaboração deste trabalho.

Este trabalho está estruturado da seguinte forma:

- A seção 2 aborda os processadores de texto TeX, LaTeX e a ferramenta auxiliar BibTeX.

- A seção 3 apresenta os trabalhos relacionados.
- A seção 4 trata do levantamento de requisitos do projeto.
- A seção 5 expõe o projeto arquitetural da aplicação desenvolvida.
- A seção 6 apresenta os modelos que compõem o projeto detalhado do IFBATEX.
- A seção 7 apresenta as principais funcionalidades da aplicação desenvolvida.
- As seções 8 e 9 abordam as metodologias de teste utilizadas.
- Nas seções seguintes são apresentadas as conclusões, os trabalhos futuros e as referências utilizadas.

II. TEX, LATEX E BIBTEX

O termo “Tipografia Digital” refere-se à preparação de documentos para publicação utilizando apenas meios digitais (softwares, computadores, impressoras, etc.). Nesta seção serão detalhados os sistemas de composição tipográfica digital TeX e LaTeX, além da ferramenta auxiliar BibTeX.

O TeX é um sistema de tipografia concebido por Donald Knuth [16], professor de ciências da computação da Universidade de Stanford, nos EUA, e autor de diversos livros na área. A idéia do TeX surgiu na segunda metade da década de 70, enquanto D. Knuth revisava seu livro *The Art of Computer Programming* e atestava a baixa qualidade tipográfica obtida. Ele trabalhou no desenvolvimento da ferramenta durante o restante da década de 70 e disponibilizou a primeira versão em 1979 [16].

Paralelo ao desenvolvimento do TeX, D. Knuth também criou o sistema METAFONT. Este sistema é composto por um interpretador e uma linguagem especializados para a criação de fontes [15]. Um dos principais diferenciais deste sistema é a utilização de fórmulas geométricas para definição dos caracteres. Este diferencial permite que o METAFONT tenha alta precisão e qualidade na renderização das fontes. A família de fontes “Computer Modern”, utilizada nos documentos produzidos pelo TeX, é gerada por este sistema.

O TeX é composto por um compilador e uma linguagem de programação. A linguagem de programação TeX dispõe de uma série de comandos que permitem ao usuário definir todos os aspectos visuais do documento a ser gerado. Uma característica importante da linguagem é a possibilidade de estendê-la através da criação de macros [21]. As coleções de macros TeX são chamadas de “formatos”. O primeiro formato, o Plain TeX, foi criado pelo próprio Knuth.

Conforme pode ser observado no código fonte 1, ao utilizar o formato Plain TeX muitos detalhes do

aspecto visual ainda devem ser definidos pelo usuário. Com o intuito de mitigar este problema, Leslie Lamport desenvolveu o formato LaTeX em 1985 [17].

```

\magnification=\magstep1
\baselineskip=12pt
\hsize=6.3truein
\vsiz=8.7truein
\font\footsc=cmcscl0 at 8truept
\font\footbf=cmbx10 at 8truept
\font\footrm=cmr10 at 10truept
\footline={\footsc the electronic journal of
combinatorics
{\footbf 16} (2009), \#R00\hfil\footrm\folio
}
\font\bigrm=cmr12 at 14pt
\centerline{\bigrm An elementary proof of
the reconstruction conjecture}

\bigskip\bigskip

\centerline{D. Remifa\footnote*{Thanks to
the editors of this wonderful journal!}}
\smallskip
\centerline{Department of Inconsequential
Studies}
\centerline{Solatido College, North Kentucky
, USA}
\centerline{\tt remifa@dis.solatido.edu}
\bigskip

\centerline{\bf Abstract}
\smallskip
{\narrower\noindent
Paragraph\par}
\bigskip

\beginsection 1. Introduction.
This is the start of the introduction.
\bye

```

Código-fonte 1

O LaTeX é uma coleção de macros que implementa um sistema de preparação de documentos [17]. Este formato disponibiliza para o usuário uma linguagem de marcação de alto nível que permite descrever o documento em termos de sua estrutura lógica, sem preocupações com o aspecto visual. O formato também permite a utilização de classes de documentos e packages. Classes de documentos são especificações do LaTeX para produção de diferentes estilos de documento (cartas, artigos, livros, etc.) e packages são bibliotecas de código que fornecem funcionalidades adicionais. O código-fonte 2 demonstra um exemplo simples de código LaTeX.

```

\documentclass[a4paper,11pt]{article}
\usepackage[utf8]{inputenc}

\title{Sample title}
\author{Joe Doe}
\date{\today}

\begin{document}
\maketitle

\begin{abstract}
This is a sample abstract.
\end{abstract}

\section{First section}
This is the text that \textbf{you} want to
put here.

\section{Second section}
This is the text that you want to put here.

\end{document}

```

Código-fonte 2

Os documentos LaTeX são divididos em duas partes. A primeira parte é denominada “préambulo”, e o LaTeX exige que nesta parte seja definida a classe do documento através do comando `\documentclass{}`. Logo após esta definição, são elencados os packages utilizados. O comando `\usepackage[utf8]{inputenc}`, por exemplo, possibilita a inserção de caracteres acentuados no texto. No préambulo também são definidas as informações sobre o título (comando `\title`), sobre o autor (comando `\author`) e sobre a data (comando `\date`) do documento. Estes dados são utilizados na segunda parte para gerar o título (comando `\maketitle`).

O texto visível do documento é escrito na segunda parte, entre os comandos `\begin{document}` e `\end{document}`. Tipograficamente, o texto visível é dividido em seções e parágrafos. Os títulos das seções são feitos através da tag `\section{}`. De forma similar, títulos de subseções e subsubseções podem ser feitos, respectivamente, pelos comandos `\subsection{}` e `\subsubsection{}`. A separação de parágrafos é feita por linhas em branco. O LaTeX cuida da distribuição das palavras nas linhas (justificação), contudo, este arranjo pode ser modificado manualmente através do comando `\newline`, que força a quebra de linha. É possível aplicar formatações específicas para uma ou mais palavras do texto através de comandos LaTeX. O comando `\textbf{palavras}`, por exemplo, coloca as palavras passadas entre as chaves em negrito.

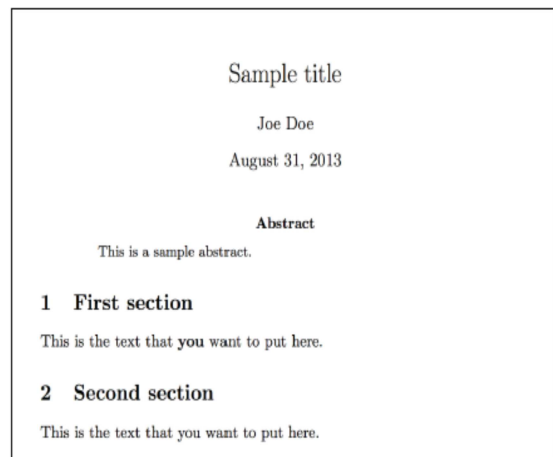


Figura 1 - Documento produzido a partir do exemplo LaTeX

O resultado da compilação do código-fonte 2 é exibido na Figura 1. Nesta figura é possível observar que o texto entre os comandos `\begin{abstract}` e `\end{abstract}` é formatado de forma distinta. O abstract é um exemplo de ambiente LaTeX. Os ambientes LaTeX permitem formatações diferenciadas dentro de um documento e a inclusão de estruturas como: figuras, tabelas, enumerações, etc. Todos estes ambientes apresentam uma estrutura de codificação similar, demonstrada no código-fonte 3.

```

\begin[opcoes]{nome-do-ambiente}
\comandol
\comando2
\end{nome-do-ambiente}

```

Código-fonte 3

A preparação de um documento LaTeX é realizada em três etapas. A primeira etapa consiste em escrever o código fonte `.tex`, o que pode ser feito com um editor de texto simples como Vim [18], Emacs [22], Bloco de Notas, etc. Na segunda etapa, o compilador converte o código fonte `.tex` para o formato `.dvi` (Device Independent), que contém instruções para visualizar ou imprimir o arquivo. Por fim, na terceira etapa, o arquivo `.dvi` é direcionado para o driver, que o converte em um formato mais adequado para publicação e distribuição como `.pdf` (Portable Document Format) ou `.ps` (PostScript). A Figura 2 exhibe graficamente este processo.



Figura 2 - Ciclo de preparação de documentos LaTeX

Em princípio, é possível construir referências bibliográficas diretamente no LaTeX, porém, a popularidade do sistema BibTeX vêm substituindo o

modo nativo. O BibTeX é um conjunto de pacotes e ferramentas auxiliares ao LaTeX, introduzido por Oren Patashnik [10], que facilita identificar as padronizações de referências bibliográficas e a organização delas no texto.

Para utilizar o BibTeX é necessário que todas as referências sejam armazenadas em arquivos do tipo BibTeX. O arquivo BibTeX é um arquivo texto que deve ser salvo no mesmo diretório que o arquivo .tex, porém com a extensão .bib. No arquivo, as referências são organizadas a partir de uma chave de referência. Um trecho de um arquivo .bib é apresentado no código-fonte 4.

```
@article{conallen,
  author = {Conallen, Jim},
  title = {Modeling Web application
    architectures with UML},
  journal = {Commun. ACM},
  issue_date = {Oct. 1999},
  volume = {42},
  number = {10},
  month = oct,
  year = {1999},
  issn = {0001-0782},
  pages = {63--70},
  numpages = {8},
  url = {http://doi.acm.org
    /10.1145/317665.317677},
  doi = {10.1145/317665.317677},
  acmid = {317677},
  publisher = {ACM},
  address = {New York, NY, USA},
}
```

Código-fonte 4

Durante a compilação do arquivo .tex, o arquivo .bib é lido pelo BibTeX e cruzado com as citações existentes no documento. Após a realização do cruzamento, o BibTeX produz uma lista das referências utilizadas e insere no documento final. O estilo da lista de referências (ordenação, dados a serem apresentados, etc.) e a posição no documento são definidos através de comandos LaTeX.

Apesar de sua curva de aprendizado inicial, usuários do LaTeX apresentam maior produtividade durante a elaboração de documentos complexos, quando comparados aos usuários de editores de texto convencionais. Segundo [24], ao se analisar o esforço dispendido pelos usuários de dois sistemas (LaTeX e Microsoft Word) durante a elaboração de um documento complexo, constatou-se que usuários do LaTeX apresentam menor produtividade nas etapas iniciais. Porém, a medida que o documento aumenta de complexidade, os usuários do LaTeX passam a apresentar maior eficiência do que os usuários do Microsoft Word.

Jessen [23] fez uma importante comparação entre o LaTeX e os sistemas WYSIWYG (What You See Is What You Get), listando as vantagens e desvantagens de sua utilização. A seguir são apresentadas as principais vantagens e desvantagens identificadas por este estudo:

- Qualidade Tipográfica - O processador TeX produz resultados com qualidade

tipográfica diferenciada, por possuir excelente suporte a hifenação, ao espaçamento de parágrafos e palavras, e para representar fórmulas matemáticas.

- Portabilidade - O ambiente LaTeX está disponível para os principais sistemas operacionais. É possível preparar um documento em uma máquina e obter uma saída idêntica compilando-o em outra máquina completamente distinta.
- Estabilidade - O processador TeX e suas ferramentas auxiliares são softwares exaustivamente testados e apresentam baixa incidência de falhas [11].
- Disponibilidade - TeX e LaTeX são softwares livres.
- Formato Universal - Como os documentos são montados a partir de códigos fonte, é menos provável que atualizações do sistema dificultem o reconhecimento de um documento antigo;
- Layout Lógico - Um dos principais objetivos do LaTeX é permitir ao autor focar no conteúdo, ao invés da apresentação visual.

No estudo também foram expostas as principais desvantagens do LaTeX, sendo elas:

- Necessidade de um visualizador externo para os arquivos finais;
- Utilização do LaTeX requer a instalação de diversos componentes e programas;
- Não adequação para elaboração de cartazes e material publicitário;

III. TRABALHOS RELACIONADOS

Nesta seção serão apresentados processadores de texto online convencionais e editores online para LaTeX. Serão analisadas as suas funcionalidades e características.

O Google Drive é uma suíte de escritório web fornecida pelo Google. A suíte dispõe de diversas aplicações, incluindo um processador de texto que apresenta funcionalidades similares às aplicações desktop.

Um dos diferenciais do Google Drive é a edição colaborativa em tempo real. Durante a edição a posição atual de cada autor é destacada no corpo do documento, a fim de evitar a sobreposição de modificações. Dentro da tela do documento também é possível comunicar-se com os outros autores através do bate-papo disponibilizado pela aplicação. Outra forma de comunicação é a inserção de comentários e notas no documento.

O Google Drive importa e exporta diversos formatos de arquivo, incluindo os formatos nativos dos principais editores desktop como, por exemplo, o .doc do Microsoft Word e o .odt do LibreOffice. Outra característica importante do sistema é o fato de ser gratuito, bastando apenas ter uma conta do Google para acessá-lo.

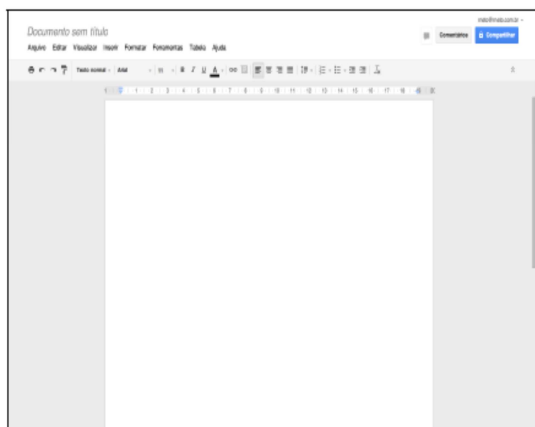


Figura 3 - Tela de edição do Google Drive

Visualmente o Google Drive segue o mesmo padrão limpo e objetivo dos outros sistemas do Google, conforme pode ser visto na Figura 3. São apresentados menus de opções (menu bar), atalhos visuais (barra de ferramentas) e atalhos de teclado.

O ThinkFree é uma suíte de escritório web comercial que também oferece um processador de texto convencional. Um dos seus destaques é a experiência de uso similar às aplicações desktop. Para atingir este objetivo, o sistema foi desenvolvido utilizando a linguagem Java. O ThinkFree é disponibilizado através de um applet, o que torna sua utilização dependente de plugins e o seu primeiro acesso mais demorado em comparação às ferramentas puramente web. Entretanto, a utilização da tecnologia de applets permite ao ThinkFree funcionar de forma offline, em casos de queda da conexão.

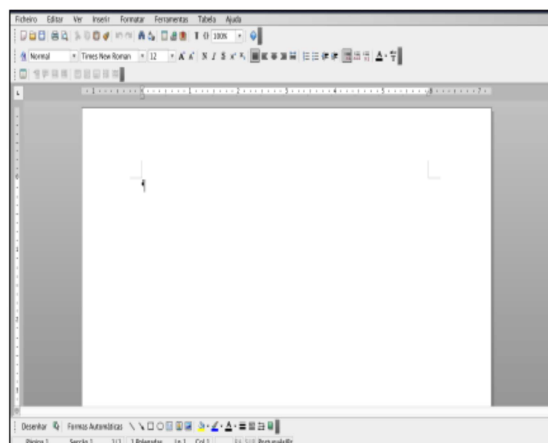


Figura 4 - Tela de edição do ThinkFree

A interface gráfica do editor ThinkFree, exibida na Figura 4, é semelhante ao popular processador de texto desktop Microsoft Word. Essa similaridade proporciona ao usuário uma melhor experiência de uso e aproveitamento do sistema, pois utiliza o conhecimento prévio adquirido. A aplicação, assim como o Google Drive, é compatível com os principais formatos de arquivos.

O Zoho Writer é mais uma alternativa comercial de suíte de escritório para a web que dispõe de um processador de texto convencional. A aplicação apresenta todas as funcionalidades essenciais para edição de textos através de uma interface gráfica simples e intuitiva. A Figura 5 mostra a apresentação visual da tela de edição da ferramenta.

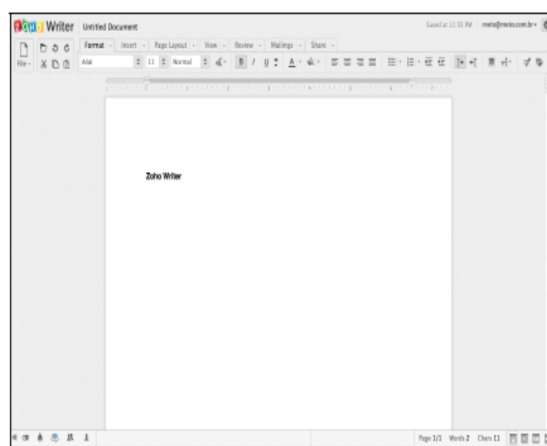


Figura 5 - Tela de edição do Zoho Writer

Dentre as características do Zoho Writer, destacam-se a opção de criação de templates, o histórico de revisões (que permite comparar diferenças entre as revisões) e a possibilidade de assinar digitalmente arquivos antes de compartilhá-los por e-mail. Assim como os outros processadores, o

Zoho Writer também é compatível com a maior parte dos formatos de arquivo comumente utilizados.

A utilização dos editores convencionais relacionados mitiga os problemas de infraestrutura necessária (instalação do software e recursos de hardware) e pervasividade computacional dos editores para desktop. Apesar desta evolução, a edição de textos longos que exigem formatação rígida (artigos, monografias, etc.) continua complexa e trabalhosa. Conforme dito anteriormente, a comunidade científica tem utilizado o processador de texto LaTeX para a elaboração destes tipos de documento. Assim como os editores de texto convencionais para desktop, o LaTeX também apresenta problemas de infraestrutura necessária e pervasividade.

Estes problemas levaram ao surgimento de sistemas online especializados na edição de arquivos LaTeX. No restante desta seção serão analisados os editores online especializados em LaTeX de maior relevância. Ao final da análise, será apresentado um quadro de posicionamento contendo as características funcionais das ferramentas descritas e as do IFBATEX.

O WriteLaTeX é um editor LaTeX para web. Ele é comercial, apesar de dispôr um plano de uso gratuito. O sistema apresenta uma interface amigável e graficamente rica. Dentre suas funcionalidades principais destacam-se o suporte a templates, a integração com a nuvem (Google Drive e Dropbox), a pré-visualização instantânea de modificações e a identificação de erros simplificada. A Figura 6 exibe a tela de edição do WriteLaTeX.

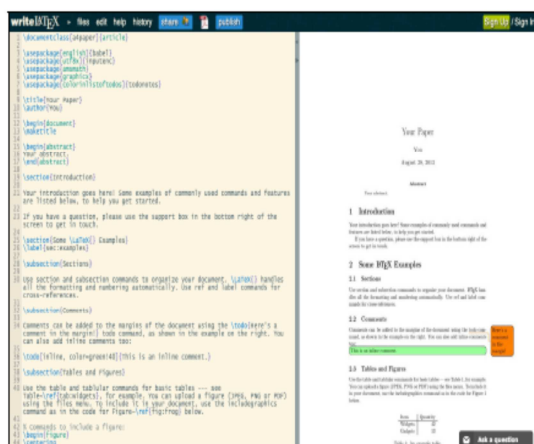


Figura 6 - Tela de edição WriteLaTeX

O ShareLaTeX também é um editor LaTeX para web. Assim como o WriteLaTeX, o ShareLaTeX é comercial, porém dispõe de um plano de uso gratuito. Sua interface gráfica é bastante simples, o que facilita a sua utilização por novos usuários. É a ferramenta comercial com maior número de funcionalidades,

dentre as quais evidenciam-se: colaboração em tempo real, suporte a templates, integração com a nuvem (Dropbox), code-completion e corretor ortográfico integrado. Na Figura 7 é exibido o ambiente de autoria do sistema.



Figura 7 - Ambiente de edição ShareLaTeX

O VerboSUS é um editor online para LaTeX e Octave (uma linguagem para computação numérica com capacidade de plotar gráficos). Ele também é comercial. Dentre as opções comerciais é a que apresenta a interface gráfica com menor usabilidade. Apesar disso, também dispõe de funcionalidades avançadas como: templates, visualizador PDF integrado e complementação de código. A Figura 8 exibe o ambiente de edição desta ferramenta.



Figura 8 - Ambiente de edição VerboSUS

A ferramenta MonkeyTeX é uma ferramenta gratuita, apesar de não ser de código aberto. Ela apresenta funcionalidades básicas para editar documentos LaTeX e um módulo de colaboração primário. A aplicação apresenta uma interface gráfica de baixa qualidade e de difícil uso. O ambiente de edição é demonstrado na Figura 9.



Figura 9-Ambiente de edição MonkeyTeX

O LaTeX Lab é um editor LaTeX para web de código aberto. Ele é fortemente acoplado a plataforma do Google, inclusive no aspecto visual. A aplicação conta com funcionalidades similares às das aplicações comerciais, sendo as principais: pré-visualização integrada, histórico de revisões, atalhos visuais e wizards para inserção de marcações LaTeX (tabelas, imagens, etc.). Todavia, a aplicação aparenta ter sido descontinuada, posto que a última atualização no repositório de código data do ano de dois mil e dez. A Figura 10 exhibe a interface de edição do LaTeX Lab.

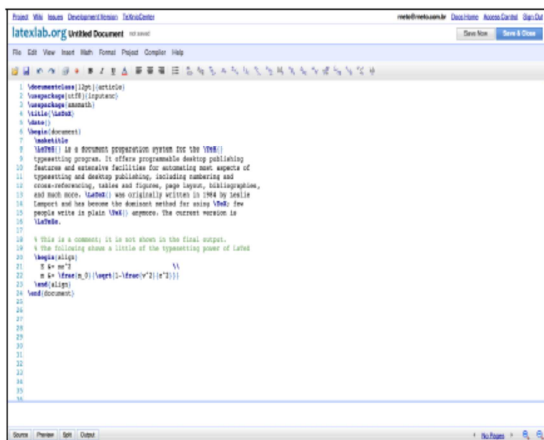


Figura 10 - Ambiente de edição LaTeX Lab

O JaxEdit é um editor online de código aberto para documentos LaTeX. A aplicação é bastante simples e dispõe de poucas funcionalidades. Sua principal característica é a renderização dos documentos para HTML, o que permite a pré-visualização instantânea. O projeto apresenta uma interface gráfica pouco atrativa e de baixa usabilidade. A tela de edição da aplicação é demonstrada na Figura 11.

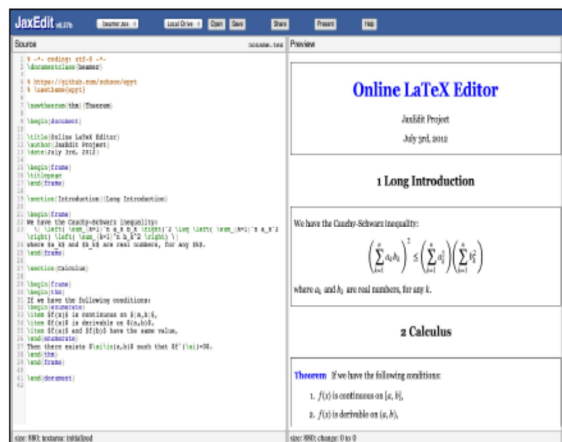


Figura 11 - Tela de edição JaxEdit

A Tabela 1 apresenta as características funcionais do IFBATEX e das ferramentas apresentadas. Conforme pode ser observado nesta tabela e na análise dos trabalhos relacionados, não existia uma ferramenta de código aberto, intuitiva, sem restrições de plataforma, que possibilitasse a edição online de documentos LaTeX com o mesmo grau de funcionalidades das ferramentas comerciais. O desenvolvimento do IFBATEX buscou contribuir com a comunidade científica e acadêmica ao desenvolver uma solução que preenchesse esta lacuna. O IFBATEX, apesar de prover funcionalidades similares às ferramentas comerciais, apresenta vantagens por dispôr de uma interface intuitiva e em português, ser de código aberto e arquitetado para tornar simples a sua extensão e customização.

	IFBATEX	WriteLaTeX	ShareLaTeX	Verbosus	MonkeyTeX	LaTeXLab	JaxEdit
Código Aberto	✓	X	X	X	X	✓	✓
Templates	✓	✓	✓	X	X	X	X
Atalhos de edição	✓	X	X	✓	X	✓	X
Visualizador Integrado	✓	✓	✓	✓	X	✓	✓
Gerador de Tabelas	✓	X	X	X	X	X	X
Citação Simplificada	✓	X	X	X	X	X	X

Tabela 1

IV. LEVANTAMENTO DE REQUISITOS

Um levantamento de requisitos bem feito é fundamental para o sucesso do projeto e para a produção de um software de qualidade. Conforme [7], o levantamento de requisitos de um projeto é a formalização do conhecimento coletado sobre o que a aplicação deve fazer e quais propriedades globais o sistema deve possuir. Nesta seção será apresentado o

levantamento de requisitos realizado para o desenvolvimento do IFBATEX.

Conforme [6], requisitos funcionais podem ser definidos como especificações de funcionalidades do sistema. Isto é, como o sistema deve reagir a entradas específicas e como deve se comportar em determinadas situações. Para que não existam ambiguidades no projeto, é importante que a especificação destes requisitos seja completa e consistente. Os requisitos funcionais identificados para o IFBATEX são apresentados na Tabela 2.

ID	Requisito Funcional	Ator
RF1	O sistema deve permitir o cadastro de novos usuários.	Usuário
RF2	O sistema deve permitir a criação, edição e exclusão de projetos.	Usuário
RF3	O sistema deve permitir, durante a criação do projeto, a escolha do template a ser utilizado.	Usuário
RF4	No ambiente de edição, o sistema deve prover atalhos para as formatações mais comuns (negrito, itálico e sublinhado).	Usuário
RF5	No ambiente de edição, o sistema deve permitir o upload de novos arquivos (imagens, .bib e .tex).	Usuário
RF6	No ambiente de edição, o sistema deve prover um wizard para inclusão facilitada de citações.	Usuário
RF7	No ambiente de edição, a aplicação deve apresentar os erros de compilação de forma simplificada, indicando claramente a posição do erro no documento.	Usuário
RF8	No ambiente de edição, a aplicação deve disponibilizar ao usuário uma interface simplificada e completamente visual para edição de tabelas.	Usuário
RF9	A aplicação deve permitir o download completo do projeto em formato zip.	Usuário

Tabela 2

A definição dos requisitos não-funcionais está relacionada às propriedades globais (qualidades) que o sistema deve apresentar. Os requisitos não-funcionais tratam de características como escalabilidade, interoperabilidade, facilidade de manutenção, desempenho, portabilidade e segurança

[6]. A análise dos requisitos não-funcionais do IFBATEX é demonstrada na Tabela 3.

ID	Requisito Não-Funcional	Categoria
RNF1	O sistema deverá ser acessado por um navegador de Internet moderno.	Usabilidade
RNF2	A interface deve ser de fácil utilização para o usuário.	Usabilidade
RNF3	O sistema deverá utilizar folhas de estilo CSS.	Padronização
RNF4	A conexão com banco de dados deve ser feita através de um ORM, abstraindo as consultas SQL.	Portabilidade
RNF5	Usuários não devem ter acesso a projetos de outros usuários.	Segurança
RNF6	O sistema será desenvolvido utilizando a linguagem PHP e o banco de dados PostgreSQL.	Software

Tabela 3

O levantamento de requisitos exposto nesta seção serviu como base para elaboração do projeto arquitetural e para modelagem do sistema através da UML. O projeto arquitetural será apresentado na próxima seção e os modelos construídos serão detalhados na seção 6.

V. ARQUITETURA

A arquitetura do IFBATEX foi dividida em 3 camadas, que seguem o padrão Model-View-Controller (MVC). O padrão MVC separa a representação da informação da interação do usuário com a mesma. A camada de modelo deve consistir de dados da aplicação, regras de negócio, lógica e funções. Uma visão pode ser qualquer saída que represente dados, como um gráfico, um diagrama ou uma página da web. O padrão possibilita a existência de múltiplas visões para um mesmo leque de dados. A camada de controle media a entrada, convertendo-a em comandos para camada de modelos ou de visão ou outros subsistemas. Este padrão tem como objetivos centrais a reutilização de código e separação de responsabilidades [20].

As camadas interagem através de conectores do tipo Procedure Call. Os conectores do tipo Procedure Call modelam o fluxo de controle através de técnicas de invocação e realizam a transferência de dados entre os componentes envolvidos através do uso de parâmetros. Exemplos de conectores Procedure Call

incluem funções, procedimentos, métodos da orientação a objetos, chamadas de callback e system calls [14]. Na arquitetura do IFBATEX os conectores são representados pelos métodos da orientação a objetos.

A aplicação conta ainda com um subsistema de compilação de documentos (compilador baseado em uma distribuição LaTeX). Toda a arquitetura do sistema pode ser visualizada na Figura 12.

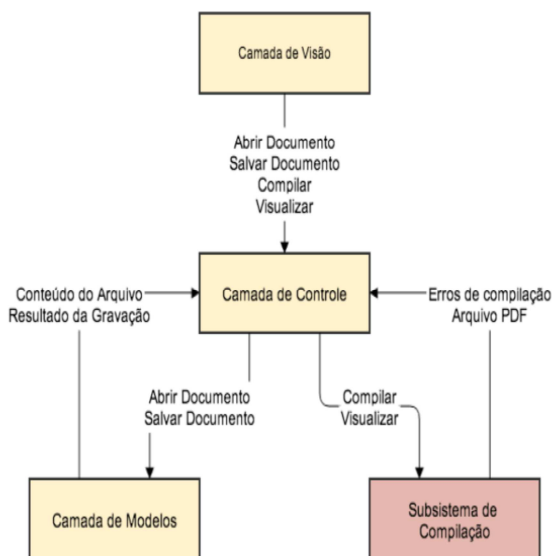


Figura 12 - Arquitetura do IFBATEX

A camada de visão é representada pela interface gráfica e é responsável por fornecer as funções de edição e visualização de textos. As ações executadas na camada de visão geram requisições que são encaminhadas para a camada de controle. Esta camada é responsável por orquestrar os pedidos enviados a partir da camada de visão, que servem para acionar as funcionalidades da camada de modelo e do subsistema de compilação de documentos. A camada de modelo e o subsistema de compilação executam suas ações e retornam o resultado para a camada de controle. As informações retornadas são tratadas e repassadas à camada de visão, onde são apresentadas para o usuário.

A camada de modelo tem como função abstrair o acesso aos dados de um projeto em edição. O subsistema de compilação é responsável por realizar a compilação dos arquivos fontes para o formato comumente aceito para publicações: Portable Document Format - PDF. Este módulo será tratado com mais detalhes na seção 7.

O desenvolvimento do IFBATEX priorizou o uso de tecnologias não proprietárias e comumente utilizadas na internet. Foram utilizadas:

- PHP como linguagem de programação;

- Laravel como framework MVC;
- PostgreSQL para banco de dados;
- Servidor web Apache em sistema operacional Linux.
- Frameworks PHPUnit [4] para testes unitários e Selenium HQ [8] para realização de testes.

Estas tecnologias foram escolhidas por serem de código aberto e contarem com uma grande comunidade de usuários e desenvolvedores. Além disso, são tecnologias de fácil implantação e disponíveis para os principais sistemas operacionais.

VI. MODELAGEM DO SISTEMA

Nesta seção será apresentada a modelagem do sistema IFBATEX. Os modelos do sistema foram elaborados utilizando a UML.

O diagrama de classes do IFBATEX é apresentado na Figura 13. Conforme apresentado no diagrama, o IFBATEX é composto por 6 classes principais, três classes do tipo controller e outras três do tipo modelo. As classes de visão são criadas dinamicamente pelo framework MVC adotado na aplicação.

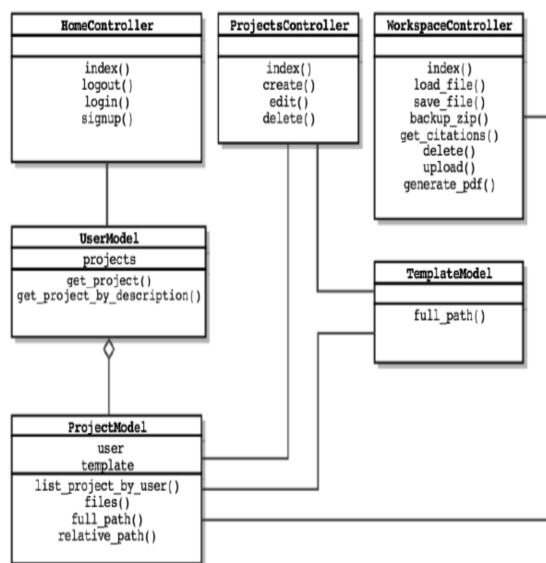


Figura 13 - Diagrama de classes

O principal fluxo do IFBATEX é o processo de editar, compilar e visualizar um documento. Este processo é demonstrado no diagrama de sequência exibido na Figura 14.

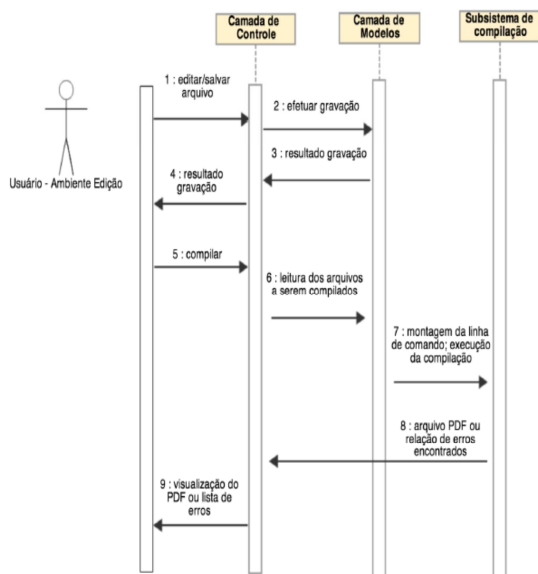


Figura 14 - Diagrama de sequência

O IFBATEX apresenta apenas um ator (usuário) que dispõe de sete casos de uso. Esta relação é apresentada graficamente no diagrama de caso de uso exposto na Figura 15.

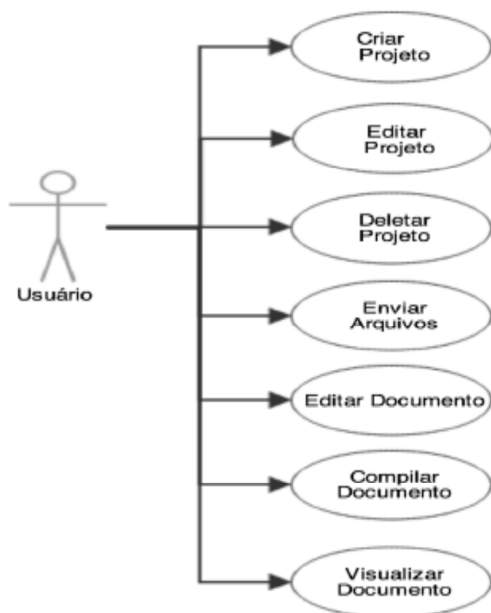


Figura 15 - Diagrama de caso de uso

VII. PRINCIPAIS FUNCIONALIDADES

O acesso ao IFBATEX é feito mediante um login tradicional com solicitação de usuário e senha. Assim, para um usuário acessar o sistema, ele deve ser previamente cadastrado. O cadastro requer apenas informações de login e senha e é público. Após a autenticação, o usuário é apresentado a tela de criação ou seleção de um projeto existente (Figura 16).

Para criar um novo projeto, o usuário deve inicialmente nomeá-lo e depois escolher um dos três templates disponíveis no sistema: Association for Computing Machinery - ACM, Springer e Elsevier. Para selecionar um projeto já criado, o usuário deve escolher um dos itens listados na tabela localizada no centro da tela. Os templates oferecidos por padrão no sistema são aqueles utilizados no TCC do curso de ADS. Porém, é importante ressaltar que outros templates podem ser adicionados ao IFBATEX.

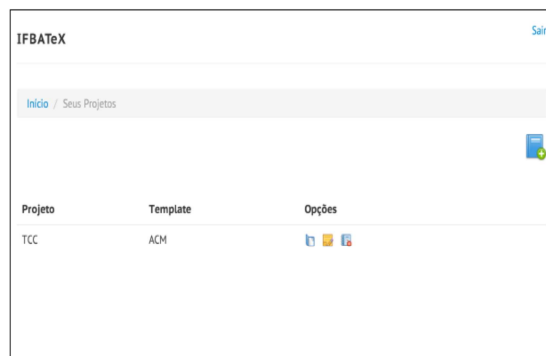


Figura 16 - Tela para criação ou escolha de um projeto

Feita a escolha ou criação do projeto, o usuário é apresentado ao ambiente de edição conforme pode ser visto na Figura 17. Nela, pode-se observar os principais componentes voltados para edição e compilação de documentos do sistema. São eles:



Figura 17 - Tela principal do ambiente de edição

A lista de arquivos;

1. As abas que contêm o editor de código, o visualizador PDF e o log de erros de compilação.
2. A barra de ferramentas de projeto; e
3. A barra de ferramentas de edição.

A lista de arquivos apresenta os documentos divididos em 3 categorias: textos, bibliografias e imagens. Além disso, ela permite que o usuário escolha um arquivo para edição. Uma vez escolhido, a lista destaca o arquivo mudando a cor do seu nome para vermelho (ver Figura 17). É possível pré-visualizar as imagens, através de uma janela modal, clicando sobre seus respectivos nomes. A exclusão de arquivos é feita posicionando o cursor sobre o nome do arquivo e acionando o ícone de exclusão que aparecerá à direita.

O editor de código LaTeX do IFBATEX é baseado no componente Ace Editor. Este componente foi especializado para apresentar funcionalidades semelhantes a editores LaTeX para desktop [1]. As suas principais funcionalidades básicas são: realce de sintaxe, indentação automática, busca e substituição de texto, quebra e numeração de linhas com erro. Além destas funcionalidades, foram adicionadas: acesso a funções via atalhos de teclado e a barra de ferramentas de edição.



Figura 18 - Barra de ferramentas de edição

A barra de ferramentas de edição (demonstrada na Figura 18) permite marcação do texto em negrito, itálico e sublinhado além de oferecer a opção de “Citação”. Esta função permite incluir referências armazenadas em arquivos BibTeX. O conteúdo deste tipo de arquivo é visualizado no sistema através de uma tabela conforme observado na Figura 19. As linhas desta tabela apresentam cada uma das referências do arquivo BibTeX. As colunas mostram o título do trabalho, o nome do autor e um botão que permite usar a função citar. Ao utilizar esta função, o comando de citação `\cite{}` com a referência da obra selecionada será incluído no ponto exato do texto onde o cursor foi posicionado. O processamento (parse) do conteúdo dos arquivos BibTeX foi feito com o auxílio da biblioteca BibTeXParser [3].

Escolha a obra que deseja citar:

Titulo	Autor	Opções
Usability Engineering	Nielsen, Jakob	
EMACS the extensible, customizable self-documenting display editor	Stallman, Richard M.	
Vi improved---vim and happy hacking keyboard lite 2	Marti, Don	
Software Engineering (7th Edition)	Sommerville, Ian	
Designing UML diagrams for technical documentation: continuing the collaborative approach to publishing class diagrams	MacKinnon, Neil, Murphy, Steve	
Modeling Web application architectures with UML	Conallen, Jim	
Research Directions in Requirements Engineering	Cheng, Betty H. C., Atlee, Joanne M.	

Figura 19 - Inclusão de referência através de interface amigável

Ao lado do botão citar, encontram-se funções para: abrir o gerador de tabelas LaTeX, aumentar o tamanho do editor (ocultando a lista de arquivos) e ocultar a numeração das linhas.

```
1 \begin{table}[h]
2 \begin{tabular}{lllll}
3
4 & & & & \\
5 & & & & \\
6 & & & & \\
7 & & & & \\
8 \end{tabular}
9 \end{table}
```

Figura 20 - Editor visual de tabelas LaTeX

Conforme dito anteriormente, umas das principais dificuldades ao utilizar o LaTeX é a construção de tabelas. Com o objetivo de simplificar este processo, foi desenvolvido para o IFBATEX um editor visual de tabelas. A interface deste editor pode ser visualizada na Figura 20.

O editor de tabelas é composto por um menu, uma barra de ferramentas, uma tabela interativa e uma caixa de texto que exhibe o código LaTeX gerado. O menu apresenta as opções “Tabela”, “Coluna”,

“Linha” e “Célula”. Na opção “Tabela”, é possível definir o tamanho da tabela, a disposição das bordas e retornar a tabela para a configuração inicial. Na opção “Coluna”, é possível definir o alinhamento do texto, adicionar uma nova coluna à esquerda ou à direita da tabela e remover a coluna selecionada. Em “Linha”, é possível adicionar uma nova linha abaixo ou acima da tabela e remover uma linha selecionada. A opção “Célula” permite agrupar e desagrupar células selecionadas.



Figura 21 - Barra de ferramentas do gerador de tabelas

A barra de ferramentas do editor visual de tabelas, apresentada na Figura 21, é dividida em blocos de funcionalidades. O primeiro bloco permite ao usuário definir o alinhamento do texto nas células (esquerda, centro ou direita). No segundo bloco é possível formatar o texto das células selecionadas como negrito ou itálico. O terceiro bloco permite ocultar ou exibir todas as bordas da tabela, ou ativar o modo de edição das bordas célula a célula. O quarto bloco permite agrupar ou desagrupar as células selecionadas.

A barra de ferramentas de projeto (exibida na Figura 22) oferece as opções:

1. Salvar;
2. Gerar PDF (compilar);
3. Download completo do projeto empacotado em formato ZIP; e
4. Upload de novos arquivos.

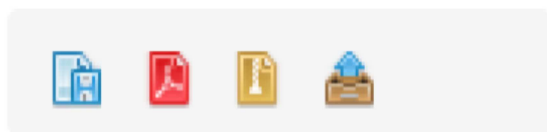


Figura 22 - Barra de ferramentas de projeto

A primeira opção permite que alterações feitas através do editor sejam gravadas em disco no servidor. A segunda opção permite que um texto seja compilado para gerar um documento em formato PDF. A terceira permite que um projeto seja compactado em formato ZIP e baixado para a máquina de um usuário. Por fim, a quarta opção permite o envio de arquivos existentes (documentos tex, bibliografias .bib e imagens) da máquina do usuário para o projeto.

O processo de compilação do IFBATEX é realizado através de um subsistema de compilação baseado na distribuição TeX Live. Ele é transparente ao usuário, que precisa apenas acionar o botão “Gerar

PDF” ou usar o atalho de teclado correspondente (CTRL + P ou Command + P). Quando um usuário utiliza a função compilar, o IFBATEX faz uma chamada ao subsistema de compilação usando a função `shell_exec(string $cmd)` do PHP, onde o parâmetro `cmd` é uma String que contém o comando de compilação `pdflatex`, os nomes dos arquivos de texto, da bibliografia utilizada e do template escolhido. Algumas características específicas do processo de compilação do LaTeX também são tratadas como, por exemplo, a necessidade de múltiplas execuções para resolução das referências e a utilização de caracteres acentuados.

A visualização do documento PDF gerado é realizada através da aba “PDF” do editor. Para implementar esta funcionalidade o IFBATEX utiliza a biblioteca PDF.js. A PDF.js implementa um visualizador de arquivos PDF utilizando apenas as tecnologias HTML e Javascript [2]. A Figura 23 demonstra a visualização de um documento compilado através do IFBATEX que, além de exibir arquivos PDF, também permite:

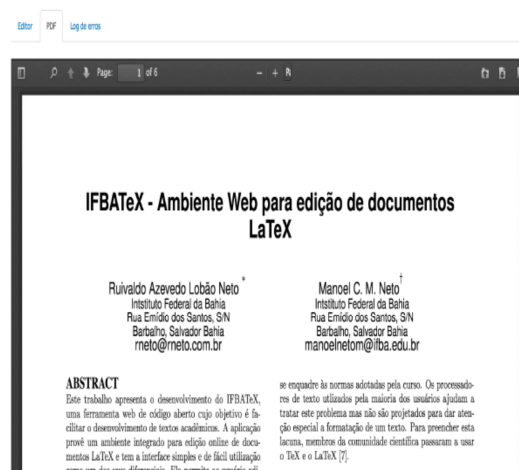


Figura 23 - Amostra do visualizador PDF integrado.

Outra funcionalidade oferecida pelo IFBATEX é permitir que erros encontrados durante o processo de compilação sejam visualizados a partir da aba “Log de Erros”. A Figura 24 exibe um exemplo deste processo. Quando um usuário digita na linha 95 o comando `\error`, que não é um comando LaTeX válido, o erro de sintaxe `undefined control sequence` é detectado. Ele é capturado, identificado (linha onde ocorreu) e apresentado na aba de log de erros. Ao clicar em um dos erros listados nesta aba, o cursor é posicionado na linha de ocorrência, que fica destacada no editor, conforme pode ser observado na Figura 25. A captura dos erros é feita através do arquivo texto log de compilação (salvo pelo subsistema de compilação). A extração das linhas com erro deste arquivo é realizada por meio de expressões regulares.

```
94
95 \error
./ifbatex.tex:95: Undefined control sequence.
98 \includegraphics[width=0.5\textwidth]{latexlab}
99 \caption{Ambiente de edição LaTeX Lab}
```

Figura 24 - Demonstração da aba de log de

VIII. TESTES AUTOMATIZADOS

O processo de validação do IFBATEX foi realizado através de testes automatizados e testes com usuários. Nesta seção será apresentada a metodologia empregada na realização dos testes automatizados e os resultados obtidos.

Testes unitários, também chamados de testes de unidade, testes de módulos ou testes de componentes, avaliam a menor unidade lógica ou bloco de um programa. Na programação estruturada, por exemplo, esta unidade pode ser definida como um procedimento ou função [9].

Os testes unitários fazem parte de um conjunto de testes chamados de testes de caixa branca [25]. Os testes de caixa branca são testes que conhecem e utilizam a estrutura interna do sistema (funções, classes, etc.). Já os testes de caixa preta, enviam uma entrada e aguardam uma saída, de forma similar a como um usuário faria.

Testes automatizados têm como objetivo verificar se o software faz corretamente e adequadamente todos os detalhes especificados no projeto. Em sistemas grandes e complexos, compostos por camadas de lógica, dados e interface com o usuário, estes testes devem abranger todos os possíveis caminhos de código, avaliar telas, menus, mensagens, validação de campos e o tratamento de exceções.

Uma boa estratégia para aumentar a abrangência dos testes é aliar testes de caixa branca com testes de caixa preta [5]. Segundo [9], testes de caixa branca como os testes unitários geralmente utilizam cerca de 60% do esforço e custo dos testes em um sistema, por outro lado, podem garantir um aumento de cerca de 40% na qualidade do software desenvolvido.

Para a criação dos testes automatizados do IFBATEX foram utilizados dois frameworks: PHPUnit e SeleniumHQ. O PHPUnit é um framework para codificação de testes unitários na linguagem PHP. Através do PHPUnit é possível criar classes de teste que podem ser organizadas hierarquicamente, permitindo a realização de testes em partes separadas

ou um teste completo do código [4]. O Selenium HQ pode ser definido como um conjunto de ferramentas de código aberto para o desenvolvimento rápido de testes funcionais do tipo caixa preta. Este conjunto de ferramentas oferece diversas opções para a realização de testes em aplicações baseadas na web [8]. O IFBATEX utiliza uma extensão do PHPUnit que permite controlar as ferramentas do Selenium HQ através de comandos PHP. Os códigos-fonte 5 e 6 exibem, respectivamente, exemplos de um teste unitário (utilizando PHPUnit) e de um teste funcional (utilizando PHPUnit e SeleniumHQ), do sistema IFBATEX.

```
class CompileTest extends TestCase {
    public function setUp() {
        parent::setUp();
        $this->bootstrapTests();
    }

    public function testCompilationACM(){
        $response = $this->action(
            'POST',
            'ProjectsController@select',
            array('project_id' => 1)
        );
        $response->assertResponseOk();

        $response = $this->action(
            'POST',
            'WorkspaceController@compile',
            array('project_id' => 1)
        );
        $response->assertResponseOk();
    }

    public function testCompilationElsevier(){
        $response = $this->action(
            'POST',
            'ProjectsController@select',
            array('project_id' => 2)
        );
        $response->assertResponseOk();

        $response = $this->action(
            'POST',
            'WorkspaceController@compile',
            array('project_id' => 2)
        );
        $response->assertResponseOk();
    }
}
```

Código-fonte 5

Os testes utilizaram como amostra os exemplos de documentos LaTeX distribuídos juntamente com os templates utilizados no IFBATEX (Elsevier, ACM e Springer). Observou-se que em todas as situações o sistema apresentou um comportamento correto, ou seja, houve a compilação e a exibição do arquivo PDF obtido sem falhas. Considerando os cenários de teste identificados e também a amostra utilizada para execução dos testes unitários, existem evidências do funcionamento adequado do sistema no que diz respeito ao ambiente de edição, compilação e visualização de documentos LaTeX.

IX. CONCLUSÃO

A elaboração de textos científicos é uma tarefa que demanda uma quantidade considerável de tempo. Boa parte deste tempo é destinado à formatação do documento, tendo em vista que os editores de texto convencionais não foram projetados para elaboração de textos que requeiram formatação rígida. A comunidade científica, buscando simplificar este processo, passou a utilizar os processadores de texto TeX e LaTeX. Entretanto, estes processadores apresentam problemas de infraestrutura necessária e pervasividade computacional.

Os problemas de infraestrutura necessária do LaTeX incentivaram o surgimento de ambientes online para edição destes documentos. Contudo, não existia uma ferramenta de código aberto, intuitiva, sem restrições de plataforma, que possibilitasse a edição online de documentos LaTeX com o mesmo grau de funcionalidades das ferramentas comerciais. A existência desta lacuna e o incentivo dado pelo curso de ADS do IFBA para a utilização do LaTeX na elaboração dos TCC's serviram como motivação inicial para a elaboração deste trabalho.

```
class WebTest extends
    PHPUnit_Extensions_Selenium2TestCase {

    public function setUp() {
        $this->setBrowser('firefox');
        $this->setBrowserUrl('http://ifbatex.
            rneto.com.br');
    }

    public function testCompilation() {
        $link = $this->byId('.projects')
            .first();
        $link->click();

        $compile = $this->byId(
            '.compile_button'
        );
        $compile->click();

        $pdf = ".mozilla_pdf_viewer";
        $this->assertElementPresent(
            $pdf
        );
    }

    public function testSave() {
        $link = $this->byId('.projects')
            .first();
        $link->click();

        $compile = $this->byId(
            '.save_button'
        );
        $compile->click();

        $okmessage = ".okmessage";
        $this->assertElementPresent(
            $okmessage
        );
    }
}
```

Código-fonte 6

O objetivo deste projeto, conforme apresentado no início, é oferecer um ambiente online de código aberto para edição de documentos LaTeX com funcionalidades similares às soluções comerciais. Através da ferramenta desenvolvida o usuário pode editar, compilar e visualizar documentos LaTeX sem precisar instalar este ambiente em seu computador.

A validação da ferramenta foi realizada através de testes unitários e testes com usuários. Os resultados obtidos permitem constatar que a ferramenta apresenta corretude em seu funcionamento e boa usabilidade. Sendo capaz de auxiliar na edição de documentos LaTeX em geral, inclusive textos longos. Este trabalho foi completamente redigido utilizando o IFBATEX.

Um dos resultados obtidos com o desenvolvimento deste projeto foi a publicação de um artigo sobre a ferramenta no XII Workshop on Tools and Applications do evento Webmedia 2013, realizado na cidade de Salvador, Bahia, entre 5 e 8 de novembro.

Pode-se concluir que o desenvolvimento do IFBATEX pode ser caracterizado como uma relevante contribuição para a área acadêmica. A sua relevância se dá, pois apesar de existirem outros editores online para LaTeX, o IFBATEX se diferencia em três aspectos principais: i) é uma iniciativa criada em uma universidade pública sem fins lucrativos (é gratuito e de código aberto), ii) possui uma interface de utilização simples e intuitiva e iii) é desenvolvido de forma modular, tornando-o de fácil customização e extensão.

Apesar dos principais objetivos deste trabalho terem sido alcançados, a análise dos trabalhos relacionados e os resultados obtidos através de testes com usuários, demonstram que ainda existem trabalhos futuros a serem realizados. Sugestões para trabalhos futuros serão apresentadas na próxima seção.

REFERÊNCIAS

- [1] Ace editor. <http://ace.ajax.org/>. Acessado em: 28/06/2013.
- [2] Pdf.js. <https://github.com/mozilla/pdf.js/>. Acessado em: 26/06/2013.
- [3] Structures bibtex parser php class. http://pear.php.net/package/Structures_BibTex. Acessado em: 22/06/2013.
- [4] S. Bergmann. PHPUnit Pocket Guide. O'Reilly Media, Inc., 2005.
- [5] S. Beydeda, V. Gruhn, and M. Stachorski. A graphical class representation for integrated black- and white-box testing. In Proceedings of the IEEE International Conference on Software Maintenance (ICSM'01), ICSM '01, pages 706–, Washington, DC, USA, 2001. IEEE Computer Society.
- [6] B. H. C. Cheng and J. M. Atlee. Research directions in requirements engineering. In 2007 Future of Software Engineering, FOSE '07, pages 285–303, Washington, DC, USA, 2007. IEEE Computer Society.
- [7] J. Conallen. Modeling web application architectures with uml. *Commun. ACM*, 42(10):63–70, Oct. 1999.
- [8] B. David. Selenium 2 Testing Tools: Beginner's Guide. Packt Publishing, 2012.
- [9] E. Dustin, J. Rashka, and J. Paul. Automated software testing: introduction, management, and performance. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999.
- [10] J. Fenn. Managing citations and your bibliography with BibTeX. *the PracTeX journal*, 4, 2006.
- [11] G. Grätzer. More Math Into LaTeX. Springer, 4 edition, Aug. 2007.
- [12] G. Grätzer. A gentle learning curve for LATEX. *The PracTEX Journal*, 2008.
- [13] P. W. D. Helmut Kopka. Guide to LaTeX. Addison-Wesley Publishing Company, 2003.
- [14] A. W. Kiwelekar. Architectural connectors. <http://www.cse.iitb.ac.in/~awk/connector.pdf>, 2004. Acessado em: 27/07/2013.
- [15] D. E. Knuth. Metafont: The program, volume d of computers and typesetting, 1986.
- [16] D. E. Knuth. TeX: The Program, volume B of Computers & Typesetting. Addison-Wesley, Reading, Massachusetts, 1986.
- [17] L. Lamport. LaTeX User's Guide and Document Reference Manual. Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.
- [18] D. Marti. Vi improved—vim and happy hacking keyboard lite 2. *Linux J.*, 2001(91):24–, Nov. 2001.
- [19] J. Nielsen. Usability Engineering. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.
- [20] E. Rozanski, N. e Woods. Software Systems Architecture: Working with Stakeholders Using Viewpoints and Perspectives. Pearson Education, 2011.
- [21] B. Salzberg and T. Murphy. Latex: when word fails you. In Proceedings of the 33rd annual ACM SIGUCCS fall conference, SIGUCCS '05, pages 241–243, New York, NY, USA, 2005. ACM.
- [22] R. M. Stallman. Emacs the extensible, customizable self-documenting display editor. *SIGPLAN Not.*, 16(6):147–156, Apr. 1981.
- [23] K. Steding-Jessen. Latex: Uma alternativa mais eficiente comparada aos sistemas wysiwyg. http://www.dicas-l.com.br/arquivo/latex_uma_alternativa_mais_eficiente_comparada_aos_sistemas_wysiwyg.php, 1998. Acessado em: 28/07/2013.
- [24] A. Tomé. Latex: Uma introdução, um incentivo. <http://biquinho.furg.br/tex-br>, 1999. Acessado em: 27/07/2013.
- [25] H. Zhu, P. A. V. Hall, and J. H. R. May. Software unit test coverage and adequacy. *ACM Comput. Surv.*, 29(4):366–427, Dec. 1997.

X. TRABALHOS FUTUROS

Como sugestão para trabalhos futuros, pode-se elencar:

1. Possibilitar a edição colaborativa dos documentos em tempo real;
2. Implementar um sistema de ajuda integrado a ferramenta;
3. Implementar um histórico de revisões que permita a análise comparativa entre as revisões;
4. Implementar uma relação de orientação para os projetos, possibilitando que orientadores acompanhem os projetos dos alunos através do sistema. Podendo acompanhar modificações, estabelecer prazos, metas e sugerir modificações.